

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**ÜÇ BOYUTLU TANIMLANAN KONTEYNER
YÜKLEME PROBLEMİNİN BOŞLUK TANIMLAMA
YAKLAŞIMI İLE ÇÖZÜMÜ VE BENZETİMİ**

Yüksek Lisans Tezi

Tezi Hazırlayan:
Selim SELÇUK

İstanbul, 2016

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**ÜÇ BOYUTLU TANIMLANAN KONTEYNER
YÜKLEME PROBLEMİNİN BOŞLUK TANIMLAMA
YAKLAŞIMI İLE ÇÖZÜMÜ VE BENZETİMİ**

Yüksek Lisans Tezi

Tezi Hazırlayan:
Selim SELÇUK

Öğrenci No:
110820035

Danışman:
Doç. Dr. Gökhan SİLAHTAROĞLU

İstanbul, 2016

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum "**Üç Boyutlu Tanımlanan Konteyner Yükleme Probleminin Boşluk Tanımlama Yaklaşımı İle Çözümü Ve Benzetimi**" başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiği ve çalışmamın içinde kullandıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 14.03.2016

Aday: Selim SELÇUK



T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi..11082005...no'lu.....'in 07.01.2016 tarihinde yapılan tez savunma sınavı¹ sonucunda 50...dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle, kabul (...kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

Tez Başlığı³ : İLİ BÜYÜTLÜ TANIMLANAN KONTEYNER YÜKLEME
PROBLEMİNİN BÖLÜK TANIMLAMA YAKLAŞIMI İLE ÇÖZÜMÜ VE BENZETİMİ

Tez Sınav Jürisi

Öğretim Üyesi

İmza

Danışman : Doç. Dr. Gökhan SİLAHTAROĞLU
Üye : Yrd. Doç. Dr. R. Haluk KUL
Üye : Yrd. Doç. Dr. Ediz ŞAYKOL

¹ Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayın tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

² Tez sınavının tamamlanmasından sonra jüri, tez hakkında "kabul", "düzeltme" veya "red" kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişkisi kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişkisi kesilir. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

³ İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

TEŐEKKÜR

Bu tezin yapılıőında emeđi geen, sűre boyunca yanımda olan danıőmanım Do. Dr. Gökhan SİLAHTAROĐLU ve Yrd. Do. Dr. Ediz ŐAYKOL hocalarıma saygılarımı sunar, teőekkűrű bir bor bilirim.

Adı ve Soyadı : Selim SELÇUK
Danışmanı : Doç. Dr. Gökhan SİLAHTAROĞLU
Türü ve Tarihi : Yüksek Lisans, 2016
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Lojistik, Konteyner Yükleme Problemi, Paketleme, 3D Bin Packing, Kutulama.

ÖZ

ÜÇ BOYUTLU TANIMLANAN KONTEYNER YÜKLEME PROBLEMİNİN BOŞLUK TANIMLAMA YAKLAŞIMI İLE ÇÖZÜMÜ VE BENZETİMİ

Lojistik sektöründeki işleyiş hiç bir zaman durmamaktadır. Her geçen saniye çok sayıda yük bir yerden başka bir yere taşınmaktadır. Lojistik sektöründe taşıma maliyeti taşınan yüklerin temel maliyetleri kadar önemli hale gelmiştir. Bu durumda konteynerlerin iç hacminin olabildiğince dolu, olabildiğince verimli kullanılması zorunlu hale gelmiştir. Konteyner içinde boş kalan en ufak bir alan o konteyner içinde taşınan ürünlerin maliyetini artıracaktır. Bu çalışmada yazılım teknolojileri kullanılarak üç boyutlu olarak tanımlanan konteyner yükleme probleminin çözümü boşluk tanımlama yaklaşımı ile çözülecektir. Ayrıca bu çözüm web uygulaması olarak kodlanıp internet ortamında yayına açılacaktır.

Name and Surname : Selim SELÇUK
Supervisor : Assoc. Dr. Gökhan SİLAHTAROĞLU
Degree and Date : Master, 2016
Major : Computer Engineering
Key Words : Logistics, 3D Bin Packing Problem, Container, Box,

ABSTRACT

A SUGGESTION FOR THE 3D BIN PACKING PROBLEM BY USING WEB APPLICATION

As is known, it never stops operation in the logistics sector . Many loads are transported from one place to another place with each passing second. Transport costs in the logistics sector has become as important as the basic costs of goods transported. In this case, the internal volume of the container is full as possible , as efficiently has become imperative to use . The remaining empty space in a small container that will increase the cost of products transported in containers . In this study, a solution of three-dimensional software technology as defined container loading problems will be solved by using the space identification approach. In addition, this solution will be offered to broadcast the coded internet as a web application

İÇİNDEKİLER

	Sayfa No.
ÖZ	iii
ABSTRACT	iv
ŞEKİLLER LİSTESİ	ix
KISALTMALAR	xv
1. GİRİŞ	1
2. KODLANAN WEB UYGULAMASININ İÇERDİĞİ TEKNOLOJİLERİN İNCELENMESİ	3
2.1. Yazılım Nedir?	3
2.1.1. Yazılım Dilleri.....	3
2.2. ASP.NET Framework, ASP ve ASP.NET	4
2.3. ASPX Belge Yapısı.....	5
3. ÖRNEK KONTEYNER YÜKLEME PROBLEMİNİN TANIMI	6
3.1. Konteyner Tanımı	6
3.2 Konteyner Sayısı	6
3.3. Konteynere Yüklenecek Yüklerin Tanımı	6
3.4. Konteyner Ve Yüklenecek Yüklerin X, Y, Z Koordinat Düzlemlerine Göre Konumu	7
4. TEZ ÇALIŞMASI DAHİLİNDE KODLANAN WEB UYGULAMASININ KODLARININ İNCELENMESİ	9
4. 1. Sınıflar (classes)	9
4.1.1. Manager Sınıfı (Manager class, Manager.cs).....	9
4.1.1.1. Manager Sınıfının İçerdiği Metotlar	9
4.1.1.1.1. Sort Boxes Metodu.....	9
4.1.1.1.2. AnyBoxNotProcessed Metodu	9
4.1.1.1.3. AnyBoxNotChecked Metodu	9
4.1.1.1.4. CreateNewContainer Metodu.....	9
4.1.1.1.5. SkipBoxesWithGreaterVolumeAndWeight Metodu	9
4.1.1.1.6. SkipBoxesWithGreaterDimension Metodu	9
4.1.1.1.7. AddBoxIntoContainerReally Metodu	10
4.1.1.1.8. PerfromAllTasks Metodu.....	10

4.1.1.1.9. GetBestContainer Metodu.....	10
4.1.1.1.10. GetBiggestBox Metodu.....	10
4.1.1.1.11. AddBoxIntoContainerVirtually Metodu	10
4.1.1.1.12. MakeCombinations Metodu.....	10
4.1.1.2. Manager Sınıfının İçerdiği Üyeler (Members)	11
4.1.1.2.1.Container, PossibleContainers (Konteyners, Makul, İhtimal Dahilinde Olan Konteynerler).....	11
4.1.1.2.2.List, Container, Containers (Konteyner Listesi)	11
4.1.1.2.3.Box, BoxList (Kutu, Kutu Listesi)	11
4.1.1.2.4.Container, CurrentContainer (Konteyner, An İtibarı İle İşlemde Olan Konteyner).....	11
4.1.1.2.5.Box, CurrentBox (Kutu, An itibarı İle İşlemde Olan Kutu)	11
4.1.1.2.6.Int Stepnumber (Tamsayı Tanımında Adım Sayısı)	11
4.1.2. Space Sınıfı (Space class, Space.cs).....	11
4.1.2.1. Space Sınıfı Altında Tanımlı Metotlar.....	11
4.1.2.1.1. Get Volume Metodu.....	11
4.1.2.2. Space Sınıfının İçerdiği Üyeler (Members).....	12
4.1.2.2.1. Length:	12
4.1.2.2.2. Height:.....	12
4.1.2.2.3. Width:.....	12
4.1.2.2.4. Int Volume:	12
4.1.2.2.5. x1:.....	12
4.1.2.2.6. y1:.....	12
4.1.2.2.7. z1:.....	12
4.1.2.2.8. x2:.....	12
4.1.2.2.9. y2:.....	12
4.1.2.2.10. z2:.....	12
4.1.3. Box Sınıfı (Box class, Box.cs, Kutu sınıfı)	13
4.1.3.1. Box Sınıfı Altında Tanımlı Metotlar	13
4.1.3.1.1. Get Volume Metodu.....	13
4.1.3.2. Box Sınıfının İçerdiği Üyeler (Members).....	13
4.1.3.2.1. Length:	13

4.1.3.2.2. Height:	13
4.1.3.2.3. Width:	13
4.1.3.2.4. Int Volume:	13
4.1.3.2.5. x1:	13
4.1.3.2.6. y1:	13
4.1.3.2.7. z1:	13
4.1.3.2.8. x2:	13
4.1.3.2.9. y2:	14
4.1.3.2.10. z2:	14
4.1.3.2.11. OriginalLength:	14
4.1.3.2.12. OriginalHeight:	14
4.1.3.2.13. OriginalWidth:	14
4.1.3.2.14. Int ID:	14
4.1.3.2.15. isRotated:	14
4.1.3.2.16. ContainerID:	14
4.1.3.2.17. ProcessStepNumber:	14
4.1.3.2.18. IsProcessed:	14
4.1.3.2.19. EverFit:	15
4.1.3.2.20. IsChecked:	15
4.1.3.2.21. Volume:	15
4.1.3.2.22. OriginalWeight:	15
4.1.4. Konteyner Sınıfı (Container class, Container.cs, Kutu sınıfı)	15
4.1.4.1. Konteyner Sınıfı Altında Tanımlı Metotlar	15
4.1.4.1.1. GetBoxInsertionNumber Metodu	15
4.1.4.1.2. AnySpaceAndWeightCapacityLeft Metodu	15
4.1.4.1.3. SetSpaceList Metodu	15
4.1.4.1.4. SetMyOriginalSpace Metodu	16
4.1.4.1.5. GetSpaceConsumed Metodu	16
4.1.4.1.6. GetWeightConsumed Metodu	16
4.1.4.1.7. OccupyThisWeight Metodu	16
4.1.4.1.8. IsAccommodatable Metodu	16
4.1.4.1.9. SortSpaces Metodu	16

4.1.4.2. Konteyner Sınıfının İçerdiği Üyeler (Members)	16
4.1.4.2.1. Space List (Boşluk tanım listesi):	16
4.1.4.2.2. BoxToBeInserted:	17
4.1.4.2.3. Original Space:.....	17
4.1.4.2.4. Original Weight:.....	17
4.1.4.2.5. RemainingWeight:	17
5. ÜÇ BOYUTLU OLARAK TANIMLANAN ÖRNEK KONTEYNER	
YÜKLEME PROBLEMİNİN ADIM ADIM ÇÖZÜMÜ.....	18
5.1. Örnek Problemin Tanımı.....	18
5.2. Örnek Problemin, Kodlanan Web Uygulamasının Arayüzüne Tanımlanması	25
5.3. Web Uygulamasının Örnek Problemin Tüm Değerlerini Paketle İşleme	
Başlamadan Önce Kontrol Etmesi	26
5.4. Konteyner İçine Yerleştirilecek Kutu Listesi Oluşturulur	27
5.5. Ağırlık Kontrolü.....	27
5.6. Konteyner İçine Yerleştirilmesi Mümkün Olmayan Kutu Listesi Oluşturulur:	
.....	28
5.7.Hacim Kontrolü.....	28
5.8. Genişlik, Yükseklik, Derinlik Değerlerinin Kontrolü.....	30
5.9. Konteyner İçine Yerleştirilecek Olan Kutu Listesinin Kutu Hacim Değerlerine	
Göre Yeniden Sıralanması.....	31
5.10. Konteyner İçine Yerleştirilecek Her Kutu İçin Aynı Olacak İşlemler Dizisine	
Başlanır.....	32
5.10.1. Boşluk Tanımı (Space Defination).....	35
SONUÇ	133
EKLER	146
Ek 1. BR1 Testlerinin, Tez Çalışması Dâhilinde Kodlanan Web Uygulaması İle	
Çözümünün Çıktısı.....	146

ŞEKİLLER LİSTESİ

Sayfa No.

Şekil 1. X, Y, Z Koordinat Düzlemi	7
Şekil 2. Ortogonal Yerleşimin Gösterimi	8
Şekil 3. Ortogonal Olmayan Yerleşimin Gösterimi	8
Şekil 4. Örnek Problem İçin Tanımlanan Konteyner Ve Kutuların Tamamının Gösterimi.....	18
Şekil 5. Tanımlı Konteynerin Gösterimi. (2x2x2, Taşıyabileceği Maksimum Ağırlık 80 KG).....	19
Şekil 6. Tipinde Tanımlanan Kutunun Gösterimi (2x2x1, 40 KG, 1 Adet).....	19
Şekil 7. B Tipinde Tanımlanan Kutunun Gösterimi (1x2x1, 20 KG, 1 Adet)	20
Şekil 8. C Tipinde Tanımlanan Kutunun Gösterimi (1x1x2, 30 KG, 1 Adet)	21
Şekil 9. D Tipinde Tanımlanan Kutunun Gösterimi (1x1x1, 10 KG, 4 adet).....	22
Şekil 10. D Tipinde Tanımlanan Kutunun Gösterimi (1x1x1, 10 KG, 4 adet).....	23
Şekil 11. F Tipinde Tanımlanan Kutunun Gösterimi (2x2x3, 40 KG, 1 adet).....	24
Şekil 12. G Tipinde Tanımlanan Kutunun Gösterimi (2x2x2, 81 KG, 1 adet).....	25
Şekil 13. Örnek Problemin Web Arayüzüne Tanımlanmasının Gösterimi.....	26
Şekil 14. A Tipinde Tanımlı Kutunun İlk Pozisyonunun Gösterimi	33
Şekil 15. A Tipinde Tanımlı Kutunun, Yükseklik Ve Derinlik Tanımı Oluşturan Yüzey Üzerinde Bulunduğu Düzleme Temas Halindeyken Görünümü	34
Şekil 16. A Tipinde Tanımlı Kutunun, Genişlik Ve Yükseklik Tanımı Oluşturan Yüzey Üzerinde Bulunduğu Düzleme Temas Halindeyken Görünümü	35
Şekil 17. Ölçüleri 3x3x3 Olan Konteynere Ait Boşluk Tanımının Gösterilmesi	36
Şekil 18. Ölçüleri 3x3x3 Olan Konteyner İçine Ölçüsü 1x1x1 Olan Kutu Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor	36
Şekil 19. Oluşan 3 Boşluk Tanımı Tüm Hacmi İle Gösteriliyor	37
Şekil 20. Ölçüleri 3x3x3 Olan Konteyner İçinde 2 Boşluk Tanımının Oluşması.....	37
Şekil 21. Ölçüleri 3x3x3 Olan Konteyner İçinde 2 Boşluk Tanımının Oluşması.....	38
Şekil 22. Boşluk Tanımlarının Birleştirilmesine Bir Örnek	38
Şekil 23. Boşluk Tanımlarının Birleştirilmeden Önce Tam Hacmi İle Gösterilmesi	39
Şekil 24. Boşluk Tanımlarının Birleştirilmesinin Gösterimi	39
Şekil 25. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon.....	44
Şekil 26. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon.....	44

Şekil 27. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon.....	45
Şekil 28. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon.....	45
Şekil 29. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon.....	46
Şekil 30. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon.....	46
Şekil 31. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren Kombinasyon Seçiliyor	47
Şekil 32. Problemden Tanımlanan Konteynerin Gösterimi	48
Şekil 33. Problemden Tanımlı Konteynerin İçinde Oluşan Boşluk Tanımı (İçeride Herhangi Bir Kutu Yok İken)	49
Şekil 34. A Tipinde Tanımlı Kutunun Çevrilmeden Önceki Görüntüsü	49
Şekil 35. A Tipinde Tanımlı Kutunun Konteyner İçine Hangi Pozisyonda Yerleştirileceğini Gösteren Şekil	50
Şekil 36. A Tipinde Tanımlanan Kutunun Konteyner İçine Yerleştirilmesi	51
Şekil 37. A Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinden Sonra Konteyner İçinde Oluşan Yeni Boşluk Tanımının Gösterimi.....	53
Şekil 38. A Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinden Sonra Konteyner İçinde Oluşan Yeni Boşluk Tanımının Konteyner Gölgelendirilemesi İle Gösterimi.....	54
Şekil 39. B Tipinde Tanımlı Kutunun Gösterimi.....	55
Şekil 40. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğinin Karar Aşaması.....	56
Şekil 41. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon.....	57
Şekil 42. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon.....	58
Şekil 43. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon.....	59
Şekil 44. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon.....	60
Şekil 45. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon.....	61
Şekil 46. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon.....	62

Şekil 47. B Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinin Mümkün Olmadığı Kombinasyonun Gösterimi	63
Şekil 48. B Tipinde Tanımlı Kutunun Konteyner İçine Ne Şekilde Yerleştirileceğine Karar Veriliyor	64
Şekil 49. B Tipinde Tanımlı Kutu, A Tipinde Kutunun Üzerine Yerleştiriliyor	66
Şekil 50. B Tipinde Tanımlı Kutunun Yerleştirilmesi Gösteriliyor.....	66
Şekil 51. B Tipinde Tanımlı Kutunun Konteyner İçine Yerleşimi Konteyner Gölgeleştirilmesi İle Gösteriliyor	67
Şekil 52. B Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinden Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor	68
Şekil 53. C Tipinde Tanımlı Kutu Konteyner İçine Yerleştirilmek İçin Seçiliyor	69
Şekil 54. D Tipinde Tanımlı 4 Numaralı Kutu Konteyner İçine Yerleştirilmek İçin Listeden Seçiliyor	70
Şekil 55. D Tipinde Tanımlı 4 Numaralı Kutu 1 Numaralı Kutunun Üzerine, 2 Numaralı Kutun İse Önüne Yerleştiriliyor.....	72
Şekil 56. D Tipinde Tanımlı 4 Numaralı Kutunun Konteynere Yerleştirilmesi Gösteriliyor	72
Şekil 57. D Tipinde Tanımlı 4 Numaralı Kutunun Konteynere Yerleştirilmesi Konteyner Gölgeleştirilmesi İle Gösteriliyor	73
Şekil 58. D Tipinde Tanımlı 4 Numaralı Kutunun Konteyner İçine Yerleşiminden Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor.....	74
Şekil 59. D Tipinde Tanımlı 5 Numaralı Kutu Konteynere Yerleştirilmek İçin Listeden Seçiliyor	76
Şekil 60. D Tipinde Tanımlı 5 Numaralı Kutu Konteynere Yerleştiriliyor	77
Şekil 61. Problemden Tanımlı Konteynerin İçinde Oluşan Boşluk Tanımı (İçeride Herhangi Bir Kutu Yok İken)	78
Şekil 62. C Tipinde Tanımlanmış Kutu 2.Konteynere Yerleştirilmek Üzere Listeden Seçiliyor	81
Şekil 63. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon.....	83
Şekil 64. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor).....	83
Şekil 65. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon.....	84
Şekil 66. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor).....	84

Şekil 67. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon.....	85
Şekil 68. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor).....	85
Şekil 69. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon.....	86
Şekil 70. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor).....	86
Şekil 71. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon.....	87
Şekil 72. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor).....	87
Şekil 73. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon.....	88
Şekil 74. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor).....	88
Şekil 75. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğine Karar Veriliyor.....	90
Şekil 76. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı).....	91
Şekil 77. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 2 numaralı boşluk tanımı).....	92
Şekil 78. C Tipinde Tanımlı Kutunun Konteynere Yerleşmiş Hali Gösteriliyor.....	92
Şekil 79. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor (2 numaralı olasılık tanımı dahilinde oluşan 1 ve 2 numaralı boşluk tanımları).....	94
Şekil 80. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı).....	96
Şekil 81. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı).....	97
Şekil 82. İkinci Konteynere iki kutu yerleştikten sonra oluşan boşluk tanımlarını gösteriliyor	100

Şekil 83. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı).....	101
Şekil 84. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 ve 2 numaralı boşluk tanımları).....	102
Şekil 85. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor	104
Şekil 86. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor	105
Şekil 87. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor	106
Şekil 88. Konteyner İçinde Oluşan Boşluk Tanımlarının Birleştirilmesi Gösteriliyor	107
Şekil 89. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor	108
Şekil 90. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor	110
Şekil 91. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor	111
Şekil 92. Konteyner İçinde Oluşan Boşluk Tanımlarının Birleştirilmesi Gösteriliyor	112
Şekil 93. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor	113
Şekil 94. Konteyner İçinde Oluşan 2 Boşluk Tanımı Konteyner Gölgeleştirilmesi İle Gösteriliyor	114
Şekil 95. Konteyner İçinde Oluşan 2 Boşluk Tanımı Konteyner Gölgeleştirilmesi İle Gösteriliyor	116
Şekil 96. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor	118
Şekil 97. Konteyner İçinde Oluşan 2 Boşluk Tanımı Konteyner Gölgeleştirilmesi İle Gösteriliyor	118
Şekil 98. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor	121
Şekil 99. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor	123
Şekil 100. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor	123
Şekil 101. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor	126
Şekil 102. 1 Numaralı Konteynerin Son Görüntüsü	131
Şekil 103. 2 Numaralı Konteynerin Son Görüntüsü	132
Şekil 104. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(1)	135
Şekil 105. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(2)	136
Şekil 106. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(3)	137

Şekil 107. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(4)	138
Şekil 108. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(5)	139
Şekil 109. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(6)	140
Şekil 110. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dâhilinde Kodlanan Web Uygulamasının Karşılaştırılması(7)	140
Şekil 111. BR Serisi Test Verilerinin İncelenmesi	141
Şekil 112. BR1 Testinin, Tez Çalışması Dâhilinde Kodlanan Web Uygulamasının Ara Yüzüne Tanımlanması	143
Şekil 113. BR1 Testinin, Tez Çalışması Dâhilinde Kodlanan Web Uygulaması İle Çözümü	145

KISALTMALAR

İnt	: c sharp yazılım dili için tamsayı tanımlaması
space defination	: Koonteyner içindeki boş hacmi belirten tanım
x	: koordinat düzleminde x eksenini
y	: koordinat düzleminde y eksenini
z	: koordinat düzleminde z eksenini
length	: genişlik ölçüsünü yani koordinat düzleminde x eksenini ifade eder, kısaltması için L harfi kullanılmıştır
height	: yükseklik ölçüsünü yani koordinat düzleminde y eksenini ifade eder, kısaltması için H harfi kullanılmıştır
width	: derinlik ölçüsünü yani koordinat düzleminde z eksenini ifade eder, kısaltması için W harfi kullanılmıştır
new	: c sharp yazılım dili içinde kullanılan anahtar sözcük
class	: c sharp yazılım dili içinde kullanılan, bir nesnenin şeklini tanımlayan bir şablondur, kısaltması için .cs kullanılmıştır.
object	: c sharp yazılım dili içinde kullanılan nesne tanımıdır.
possibility	: olasılık
H.B.	: Bischoff vd (1995)'in sezgiseli (The heuristic of Bischoff et al.,1995).
H.BR.	: Bischoff ve Ratcliff (1995)'in sezgiseli (The heuristic of Bischoff and Ratcliff ,1995).
GA.GB.	: Gehring ve Bortfeldt (1997)'in genetik algoritması (The genetic algorithm of Gehring and Bortfeldt ,1997).
TS.BG.	: Bortfeldt ve Gehring (1998)'in yasaklı araması (The tabu search of Bortfeldt and Gehring ,1998).
H.T.	: Terno vd (2000) sezgiseli (The heuristic of Terno et al,2000).
HGA.GB.	: Bortfeldt ve Gehring (2001)'in genetik algoritması (The hybrid genetic algorithm of Bortfeldt and Gehring, 2001).
H.M.	: Eley (2002)'in sezgiseli (The heuristic of Eley, 2002).
PTS.BG	: Bortfeldt (2003)'in paralel yasaklı araması (The parallel tabu search of Bortfeldt et al, 2003).

1. GİRİŞ

Hangi sektörde, hangi alanda olursa olsun, her geçen gün artan rekabet ortamı ve taleplere hızlı cevap verme gereksinimi bazı zorunlulukların doğmasına neden olmaktadır. Bu anlamda en önem arz eden zorunluluk, maliyetleri minimize etmektir.

Lojistik sektöründe de maliyetlerin minimize edilmesi son derece önemlidir. Yükleme işleminin en verimli şekilde yapılması ile maliyetler minimize edilmiş olacaktır. "Yükleme Problemi" olarak tanımlanmış olan problem en çok bilinen problemlerdendir. Ve bu problemin üzerine çok sayıda çalışmalar yapılmıştır.

Yükleme probleminde, taşınacak yükü minimum maliyetle taşımak hedeflenmiştir. Bu çalışmada yükleme problemi bu yönü ile ele alınacaktır.

Lojistik hizmeti hiç bir zaman durmayan sektörlerden biridir. Her geçen saniye çok sayıda yük bir yerden başka bir yere taşınmaktadır. Lojistik sektöründe taşıma maliyeti birimlerin temel maliyetleri kadar önemli hale gelmiştir. Bu durumda taşımak için kullanılan yapıların (konteyner, koli..vb) iç hacminin olabildiğince dolu, olabildiğince verimli kullanılması gerekmektedir. Konteynır içinde boş kalan en ufak bir alan o konteynır içinde taşınan ürünlerin maliyetini artıracaktır.

Lojistik hizmetine olan talebinin sürekli artması taşımada kullanılan yapıların (konteyner, koli) yükleme algoritmalarına olan ilgiyi de artırmıştır. Bu konuda sürekli iyileştirmeler yapılmış, problemin çözümüne farklı tekniklerle yaklaşılmıştır. Öncelerde kullanılan pratik yöntemler yerini bilgisayar sistemlerine bırakmıştır. Lojistik sektöründe konteynır yükleme yazılımları yaygın olarak kullanılmaktadır.

Bu tez çalışmasında bu problemi çözmek için bazı bakış açıları temel prensip olarak kabul edilmiştir. Çalışmada konteyner yükleme problemi ve çözümü üç boyutlu olarak ele alınmıştır. Yük taşıyacak yapılar, (konteyner, koli.. vb) ve taşınacak yapılar (koli, kutu..vb) ancak ve ancak dikdörtgen prizması ya da küp şekillerinden oluşabilir. Piramit, silindir, koni, küre gibi cisimler bu çalışma için geçersizdir. Çözümde kullanılan algoritma büyük hesaplar içermektedir. Algoritma, ASP.NET teknolojisi kullanılarak C# yazılım dili ile Web Uygulaması haline getirilmiş <http://pakettle.webyuk.com/WebForm1.aspx> adresinde yayına açılmıştır.

Tez Çalışmasının 2. bölümünde, tez çalışması için kodlanan Web Uygulamasının yazılım tarafının içerdiği yazılım teknolojileri incelenmiştir.

3.Bölümde, kodlanan Web Uygulamasının kabul edeceği örnek konteyner problemlerinin tanımı üzerinde durulmuştur.

4.Bölümde Web Uygulamasına ait kodların class (sınıf) ve method (metot) bazında incelenmesi yapılmıştır.

5.Bölümde Web Uygulamasına ait işleyiş, yükleme algoritması bir örnek problem üzerinden şekillerle açıklanmıştır.

2. KODLANAN WEB UYGULAMASININ İÇERDİĞİ TEKNOLOJİLERİN İNCELENMESİ

Bu tez çalışmasında üç boyutlu olarak tanımlanan yükleme probleminin çözümü bir Web Uygulaması ile simule edilmiştir. Web Uygulamasında birçok yazılım teknolojisi kullanılmıştır.

2.1. Yazılım Nedir?

Teknoloji terimi olan yazılım, hem bilgisayar sistemini oluşturan donanım parçalarının hem de kullanıcıların işlerini yapmak için gerekli olan işlemleri tanımlayan komutlar topluluğudur.

2.1.1. Yazılım Dilleri

Yazılım için çeşitli diller mevcuttur. Bunlardan bazıları Pascal, C++, C#, Java'dır. Bu çalışmadaki programda kullanılan yazılım dili C (#)Sharp'dır. C # yazılım dili, Microsoft ASP.NET Web Uygulama teknolojisi ile geliştirilmiştir.

C# Nedir?, C (#) Sharp Yazılım Dili

C#, C/C++ ve Java dillerinden türeyen, güçlü, basit, esnek, tip güvenli, modern ve Microsoft .NET platformu için sıfırdan geliştirilmiş tek programlama dilidir.

Bilindiği üzere, bilgisayarları kullanılabilmesi için önce programlanması gerekir. Bilgisayarlar ilk çıktıkları günden bu yana değişik düzeyde birçok programlama dili kullanılarak programlanıp insanlara faydalı araçlar haline getirilmiştir. C# orta düzeyli diller grubuna girmektedir. C# ile hem üst düzey hem de alt düzeyde programlar yazılabilir. C# gerçek manada nesne yönelimli (object-oriented) bir dildir. C# XML desteğini içerir. Bu sayede XML verileri ile çalışmada C# dili büyük esneklikler sağlar. Ayrıca C# Form mantığını da destekler. C# ile Windows için program yazılabilir, konsol uygulamaları geliştirilebilir, web servisler ve dll (Dynamic Link Library) bileşenleri de yazılabilir. .NET ile tüm IP(internet protokolünü kullanan) cihazlara erişim mümkündür. Bunlar cep telefonları ya da PDA(Personal Data Assistant – Küçük cep bilgisayarları) olabilir. .NET Mobile Geliştirme Aracı ve C# ile mobil cihazlara yönelik yazılımlar geliştirilebilir. [4]

ASP.NET için en uygun dil C# kabul edilir çünkü .NET'in tamamı C# ile yazılmıştır. C#, ASP gibi script yorumlamalı değil aksine tam olarak nesne yönelimli, derlenen ve çalıştırılan bir model ifade etmektedir.

2.2. ASP.NET Framework, ASP ve ASP.NET

.NET Framework, Microsoft tarafından geliştirilmiş, uygulama geliştirme platformudur. Açık internet protokolleri ve standartları üzerine kurulmuştur. Sun Microsystems tarafından geliştirilmiş olan Java platformu ile oldukça benzerdir. .NET Framework uygulama kavramının kapsamı oldukça geniştir. Masaüstü ya da Web Uygulamalarına ait tüm içerikler bu çatı altında desteklenmiştir. Geliştirilen ortamdan bağımsız olarak uygulamalarla iletişim için web servisleri oluşturulmasına imkan verilmiştir. Ayrıca bu web servisleri uygulamaların birbirleriyle olan iletişimini de sağlamaktadır. .NET Framework mimarisi birçok içerikten oluşturulmuştur; Ortak bir yürütme ortamı, ortak bir değişken tür sistemi ve devingen bağlantılı kütüphaneler. Kısaca .NET kütüphanesi programcılar için bir çok fonksiyonun sınıflandırılmış halidir. .NET Framework sayesinde işletim sistemi ile program kolayca uyum içinde çalışabilir.

ASP ve ASP.NET Web uygulama teknolojileridir. Bu alanda Microsoft ilk olarak ASP betik motorunu geliştirmiştir. (Active Server Page). ASP ayrıca Classic ASP olarak da adlandırılmıştır. ASP Kodları server tarafından yorumlandığında HTML kodlarına dönüştürülür. Bu sayede yazılımcılar kodların kaynaklarını gizleyebilmektedir. ASP kodları HTML kodlarının içine gömülü olarak oluşturulabilir.

ASP.NET yine Microsoft tarafından geliştirilmiştir. Özdevinimli (dinamik) web sayfaları, web uygulamaları, XML tabanlı Web hizmetlerinin geliştirilmesine olanak sağlayan teknolojilerden biridir. ASP.NET teknolojisi her ne kadar isim olarak ASP teknolojisine benzer olsa da ASP'ye oranla ciddi değişiklikler içermektedir. ASP.NET, ASP'nin devamı niteliğindedir. ASP.NET, framework üzerinde çalışan C#, J#, VB.Net gibi yazılım dilleri ile düzenlenebilen bir .NET teknolojisidir.

2.3. ASPX Belge Yapısı

ASPX yapısı ASP.NET belgeleri tarafından kullanılan yapıdır. Bir ASPX yapısı statik (öz devinimsiz) HTML ya da XHTML tabanlı olabilir. ASP.NET'in kullandığı diğer belge yapılarına örnek verilecek olursak; ascx, asmx, ashx, master, sitemap, skin ve config sayılabilir.

3. ÖRNEK KONTEYNER YÜKLEME PROBLEMİNİN TANIMI

Yükleme problemi üç boyutlu olarak kabul edilmiştir.

3.1. Konteyner Tanımı

Kodlanan Web Uygulamasına göre konteyner, sadece dikdörtgenler prizması ya da küp şekillerinden oluşabilir. Bu durumda konteynere ait üç ölçü tanımlaması olabilir; genişlik, yükseklik, derinlik. Konteynerin pozisyonunu belirtecek 6 nokta olabilir; (x_1, y_1, z_1) , (x_2, y_2, z_2) .

3.2 Konteyner Sayısı

Kodlanan Web Uygulamasına göre uygulamada en az 1 adet konteyner tanımlamasının yapılması gerekiyor. Tanımlanan her konteyner için genişlik, yükseklik, derinlik ve konteyner içine yüklenebilecek maksimum ağırlık tanımlamalarının da yapılması gerekiyor. Sisteme tanımlanan konteyner, yine sisteme tanımlanan kutular için yetersiz ise, başka bir deyişle tanımlanan konteyner ya da konteynerlerin içleri kutu ile dolmuş ve içeride herhangi bir kutu için yer kalmamış ise sistem tanımlanmış konteyner ya da konteynerlerden uygun olanı oluşturur, kalan kutuları bu şekilde yeni oluşturduğu konteyner içine yerleştirir.

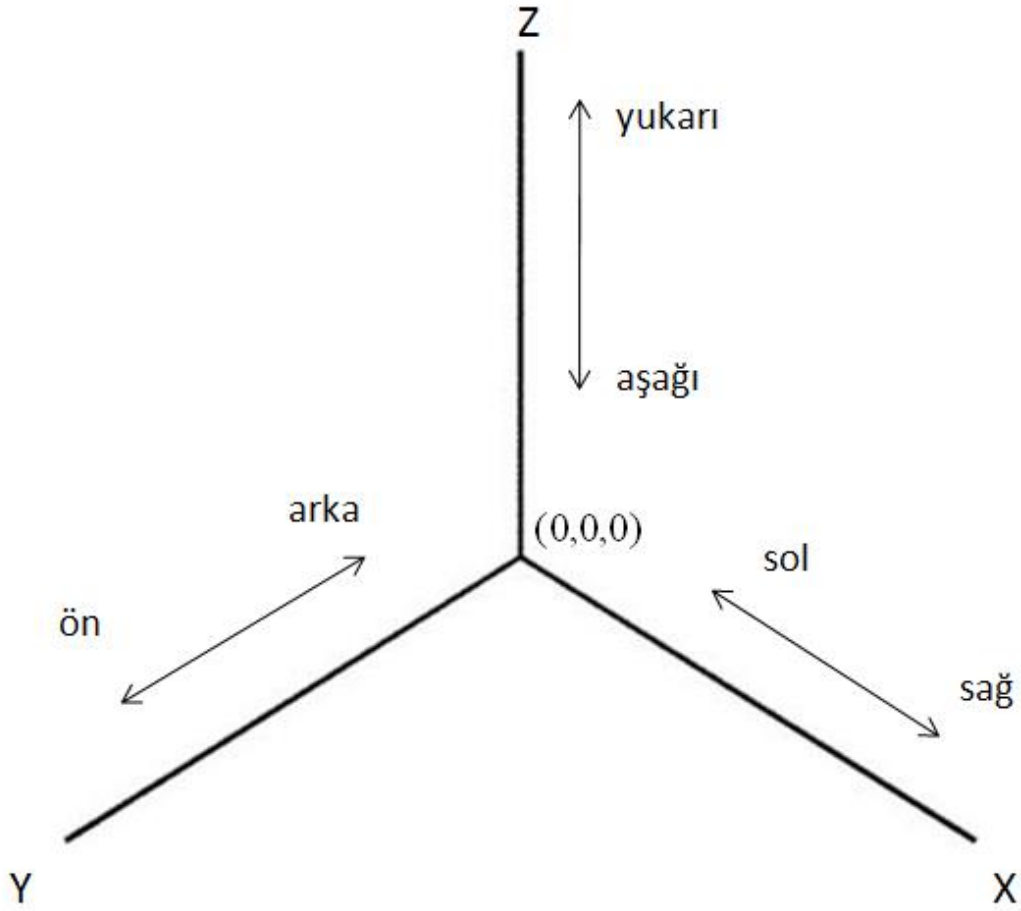
Sistem, tanımlanan konteynerlerden hesaplama sonucu gerekli görmediğini hiç kullanmayabilir. Uygulamanın çıktısında hesaplama katmadığı konteyner ya da konteynerleri liste şeklinde raporlayabilir.

3.3. Konteynere Yüklenecek Yüklerin Tanımı

Kodlanan Web Uygulamasına göre konteynere yüklenecek yükler de sadece dikdörtgenler prizması ya da küp şekillerden oluşabilir. Yükleme işleminin hesaplanabilmesi için en az 1 kutunun (yük) sisteme genişlik, yükseklik, derinlik ve ağırlık değerleri ile tanımlanmalıdır. Bu durumda her bir kutunun konteyner içindeki konumu belirleyecek 6 nokta tanımlanmış olur; (x_1, y_1, z_1) , (x_2, y_2, z_2) . Sisteme bir birinden farklı boyutlarda kutu tanımlaması da yapılabilir. Farklı boyutlarda girilen her yeni kutu yeni bir kutu tanımlamasını ifade eder. Ayrıca her kutu tipi için adet girilmesi gerekmektedir. Kutu adetleri için herhangi bir kısıtlama yoktur, istenilen sayıda kutu tanımlaması yapılabilir tanımlanan kutular için de istenilen sayıda kutu sisteme tanımlanabilir.

3.4. Konteyner Ve Yüklenecek Yüklerin X, Y, Z Koordinat Düzlemlerine Göre Konumu

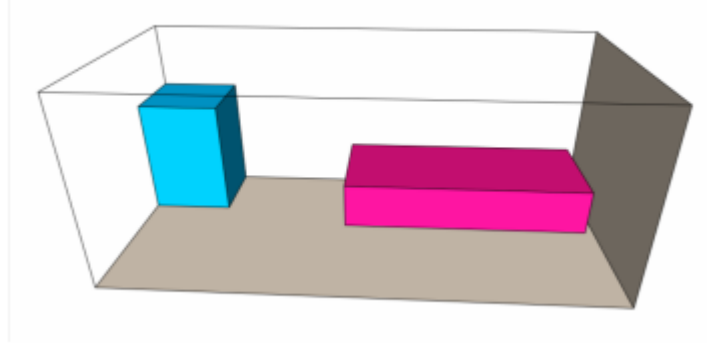
3-boyutlu konteyner yükleme problemi, X, Y, Z eksenlerinin pozitif bölümlerinde tanımlı üç boyutlu bir ortamda çözülür. (Şekil 1). Birbirini 90o açıyla kesen eksenlerden X eksenini genişliği, Y eksenini yüksekliği, Z eksenini ise derinliği ifade eder. Orijin noktası ise bu üç eksenin kesiştiği (0, 0, 0) noktasıdır.



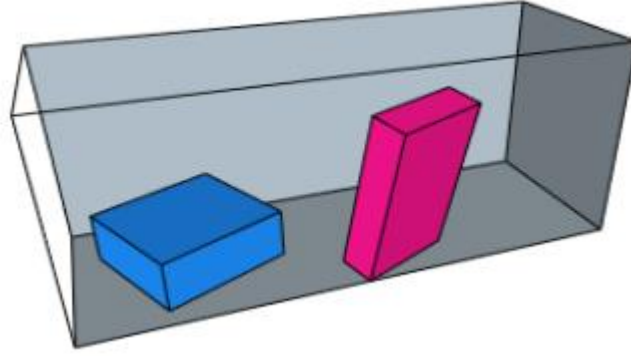
Şekil 1. X, Y, Z Koordinat Düzlemi

Bir konteynerin genişlik değerini ifade eden ölçü x_2 noktası ile x_1 noktasının farkı, yükseklik değerini ifade eden ölçü y_2 noktası ile y_1 noktasının farkı, derinlik değerini ifade eden ölçü ise z_2 noktası ile z_1 noktasının farkı kadardır.

Bu tez çalışmasında konteyner ve kutular ortogonal yerleşim şekli ile kabul edilmiştir. (Şekil 2). Ortogonal olmayan yerleşim şekli bu tez çalışması dışındadır. (Şekil 3)



Şekil 2. Ortogonal Yerleşimin Gösterimi



Şekil 3. Ortogonal Olmayan Yerleşimin Gösterimi

4. TEZ ÇALIŞMASI DAHİLİNDE KODLANAN WEB UYGULAMASININ KODLARININ İNCELENMESİ

Web uygulaması, 4 ana class'dan(sınıf), bir de web form'dan (uygulama arayüzü) oluşmaktadır.

4. 1. Sınıflar (classes)

4.1.1. Manager Sınıfı (Manager class, Manager.cs)

4.1.1.1. Manager Sınıfının İçerdiği Metotlar

4.1.1.1.1. Sort Boxes Metodu

Bu metot hacimlerine göre kutuları listeler.

4.1.1.1.2. AnyBoxNotProcessed Metodu

Bu metot işleme girmemiş herhangi bir kutu var ise true döner.

4.1.1.1.3. AnyBoxNotChecked Metodu

Bu metot kontrol edilmemiş herhangi bir kutu var ise true döner.

4.1.1.1.4. CreateNewContainer Metodu

Bu metot problem tanımını dışında ekstradan konteynere ihtiyaç varsa yeni konteyner oluşturur.

4.1.1.1.5. SkipBoxesWithGreaterVolumeAndWeight Metodu

Hacminden dolayı konteynere giremeyecek kutuları işaretleyen metod.

4.1.1.1.6. SkipBoxesWithGreaterDimension Metodu

Herhangi bir kutunun genişlik, yükseklik ya da derinlik ölçülerinden sebep konteynere girememesi durumunda bu metot bu kutuları işaretler.

4.1.1.1.7. AddBoxIntoContainerReally Metodu

Ana görevlere yardımcı olan metod AddBoxIntoContainerReally metodudur. Bu metod oldukça uzun kod bloğundan oluşur. Kutuları konteyner'a ekler, konteyner değerlerini atar. Kutuyu AddBoxIntoContainer metodu gibi sanal olarak değil gerçekten konteynere ekler.

4.1.1.1.8. PerfromAllTasks Metodu

Ana görevlere yardımcı olan bir diğer metot ise PerformAllTasks metodudur. Bu metod da paketleme performansı ile alakalı işlemler yapar.

4.1.1.1.9. GetBestContainer Metodu

Henüz paketlenmemiş kutular için en uygun konteyneri bulur.

4.1.1.1.10. GetBiggestBox Metodu

Henüz paketlenmemiş kutular içinde işlem halinde olan konteyner için en büyük kutuyu bulur

4.1.1.1.11. AddBoxIntoContainerVirtually Metodu

Bu metod seçili kutuyu konteyner içine almış farz ederek hesaplamalar yapar. Yapılan işlem için sanal diyebiliriz. Kutu konteyner içine yerleştirilmez, yerleştirilmiş gibi düşünülerek işlemler, hesaplamalar yapar.

4.1.1.1.12. MakeCombinations Metodu

Bu metod konteyner içine yerleştirilmek üzere olan kutunun, döndürülmesi gerekiyor mu? gerekmiyor mu? sorgusunu hesaplar. Tüm kombinasyonları değerlendirir, hesaplar kutuyu olması gerektiği pozisyonda konteyner içine yerleştirmek için hazır eder.

4.1.1.2. Manager Sınıfının İçerdiği Üyeler (Members)

4.1.1.2.1.Container, PossibleContainers (Konteyners, Makul, İhtimal Dahilinde Olan Konteynerler)

Kutuları paketlemek için makul, ihtimal dahilinde olan konteyner/ konteynerler.

4.1.1.2.2.List, Container, Containers (Konteyner Listesi)

İşlemlerde kullanılan konteynerlerin listesi.

4.1.1.2.3.Box, BoxList (Kutu, Kutu Listesi)

Paketlemeye dahil olan kutular

4.1.1.2.4.Container, CurrentContainer (Konteyner, An İtibarı İle İşlemde Olan Konteyner)

An itibarı ile işlem halinde olan konteyneri ifade eder.

4.1.1.2.5.Box, CurrentBox (Kutu, An itibarı İle İşlemde Olan Kutu)

An itibarı ile işlem halinde olan konteyneri ifade eder.

4.1.1.2.6.Int Stepnumber (Tamsayı Tanımında Adım Sayısı)

İşlem adımlarını ifade eder.

4.1.2. Space Sınıfı (Space class, Space.cs)

4.1.2.1. Space Sınıfı Altında Tanımlı Metotlar

4.1.2.1.1. Get Volume Metodu

Boşluk tanımlarının (Space Defination) hacimlerini değer olarak döndürür.

4.1.2.2. Space Sınıfının İÇerdiği Üyeler (Members)

4.1.2.2.1. Length:

Boşluk tanımı için genişlik değeri

4.1.2.2.2. Height:

Boşluk tanımı için yükseklik değeri

4.1.2.2.3. Width:

Boşluk tanımı için derinlik değeri

4.1.2.2.4. Int Volume:

Boşluk tanımının tamsayı değerinde hacim değeri

4.1.2.2.5. x1:

Boşluk tanımı için x düzleminde x1 noktası

4.1.2.2.6. y1:

Boşluk tanımı için y düzleminde y1 noktası

4.1.2.2.7. z1:

Boşluk tanımı için z düzleminde z1 noktası

4.1.2.2.8. x2:

Boşluk tanımı için x düzleminde x2 noktası

4.1.2.2.9. y2:

Boşluk tanımı için y düzleminde y2 noktası

4.1.2.2.10. z2:

Boşluk tanımı için z düzleminde z2 noktası

4.1.3. Box Sınıfı (Box class, Box.cs, Kutu sınıfı)

4.1.3.1. Box Sınıfı Altında Tanımlı Metotlar

4.1.3.1.1. Get Volume Metodu

Box (kutu) hacimlerini değer olarak döndürür.

4.1.3.2. Box Sınıfının İçerdiği Üyeler (Members)

4.1.3.2.1. Length:

Kutu için genişlik değeri

4.1.3.2.2. Height:

Kutu için yükseklik değeri

4.1.3.2.3. Width:

Kutu için derinlik değeri

4.1.3.2.4. Int Volume:

Kutu için tamsayı değerinde hacim değeri

4.1.3.2.5. x1:

Kutu tanımı için x düzleminde x1 noktası

4.1.3.2.6. y1:

Kutu tanımı için y düzleminde y1 noktası

4.1.3.2.7. z1:

Kutu tanımı için z düzleminde z1 noktası

4.1.3.2.8. x2:

Kutu tanımı için x düzleminde x2 noktası

4.1.3.2.9. y2:

Kutu tanımı için y düzleminde y2 noktası

4.1.3.2.10. z2:

Kutu tanımı için z düzleminde z2 noktası

4.1.3.2.11. OriginalLength:

Kutu için web arayüzüne tanımlanan ilk genişlik değeri

4.1.3.2.12. OriginalHeight:

Kutu için web arayüzüne tanımlanan ilk yükseklik değeri

4.1.3.2.13. OriginalWidth:

Kutu için web arayüzüne tanımlanan ilk derinlik değeri

4.1.3.2.14. Int ID:

Her kutu için tamsayı olarak tanımlanıp atananan benzersiz kutu numarası

4.1.3.2.15. isRotated:

Kutu çevrildi mi? Çevrilmedi Mi? Bu sorgunun sonucunu belirleyen değişken

4.1.3.2.16. ContainerID:

Kutunun hangi konteynere yerleştirildiğini ifade eden tamsayı olarak tanımlanmış benzersiz konteyner numarası

4.1.3.2.17. ProcessStepNumber:

Kutunun hangi adımda konteyner içine yerleştirildiğini ifade eden sıra sayısı

4.1.3.2.18. IsProcessed:

Kutu işleme alındı mı? alınmadı mı? Bu sorgunun cevabını tutan değişken

4.1.3.2.19. EverFit:

Kutu konteyner içine yerleşebilecek mi? yerleşemeyecek mi? Bu sorgunun cevabını tutan değişken. Kutu ağırlığından dolayı, hacminden dolayı ya da ölçütlerinden dolayı konteyner içine giremeyebilir.

4.1.3.2.20. IsChecked:

Kutu tüm değerleri ile kontrol edildi mi? Edilmedi mi? Bu sorgununun cevabını tutan değişken

4.1.3.2.21. Volume:

Kutunun hacmi

4.1.3.2.22. OriginalWeight:

Kutu için girilen ağırlık değeri

4.1.4. Konteyner Sınıfı (Container class, Container.cs, Kutu sınıfı)

4.1.4.1. Konteyner Sınıfı Altında Tanımlı Metotlar

Kutu tüm değerleri ile kontrol edildi mi? Edilmedi mi? Bu sorgununun cevabını tutan değişken

4.1.4.1.1. GetBoxInsertionNumber Metodu

Konteyner içine hangi kutunun girmesi gerektiğini belirtir.

4.1.4.1.2. AnySpaceAndWeightCapacityLeft Metodu

Konteyner içinde boşluk tanımı kaldı mı onu kontrol eder.

4.1.4.1.3. SetSpaceList Metodu

Boşluk tanımlarının listesini yönetir. (Boşluk tanımına ait değerleri, ve boşluk tanımının sayısı)

4.1.4.1.4. SetMyOriginalSpace Metodu

Kutu tüm deęerleri ile kontrol edildi mi? Edilmedi mi? Bu sorgununun cevabını tutan deęişken

4.1.4.1.5. GetSpaceConsumed Metodu

Kutu tüm deęerleri ile kontrol edildi mi? Edilmedi mi? Bu sorgununun cevabını tutan deęişken

4.1.4.1.6. GetWeightConsumed Metodu

Kutu tüm deęerleri ile kontrol edildi mi? Edilmedi mi? Bu sorgununun cevabını tutan deęişken

4.1.4.1.7. OccupyThisWeight Metodu

Kutu tüm deęerleri ile kontrol edildi mi? Edilmedi mi? Bu sorgununun cevabını tutan deęişken

4.1.4.1.8. IsAccommodatable Metodu

Kutu tüm deęerleri ile kontrol edildi mi? Edilmedi mi? Bu sorgununun cevabını tutan deęişken

4.1.4.1.9. SortSpaces Metodu

Boşluk tanımlarını y1 deęerine göre sıralar.

4.1.4.2. Konteyner Sınıfının İçerdiği Üyeler (Members)

4.1.4.2.1. Space List (Boşluk tanım listesi):

Konteyner içindeki boşluk tanım listesini ifade eder.

4.1.4.2.2. BoxToBeInserted:

Konteyner içinde kaç adet kutu var ve an itibarı ile hangi kutunun konteyner içine yerleşeceğini ifade eder.

4.1.4.2.3. Original Space:

Konteynerin tanımının yapıldığı anda konteyner içinde oluşan ilk boşluk tanımı ifade eder.

4.1.4.2.4. Original Weight:

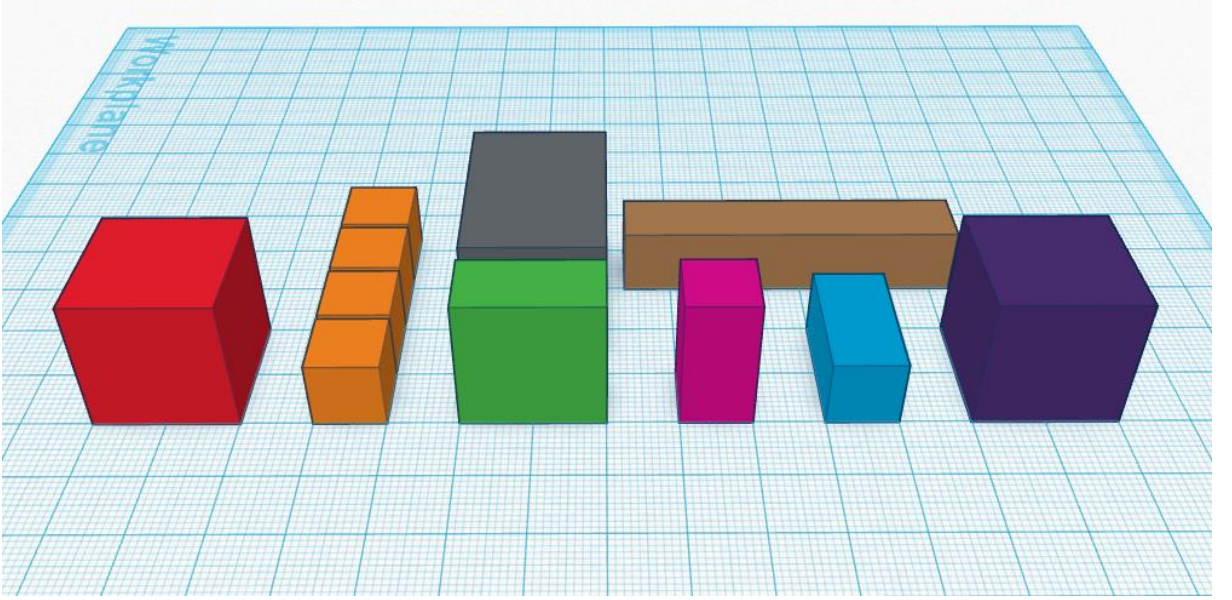
Konteyner için tanımlanan maksimum yük kapasitesini ifade eder.

4.1.4.2.5. RemainingWeight:

An itibarı ile konteyner içine yerleştirilebilecek maksimum ağırlığı ifade eder.

5. ÜÇ BOYUTLU OLARAK TANIMLANAN ÖRNEK KONTEYNER YÜKLEME PROBLEMİNİN ADIM ADIM ÇÖZÜMÜ

5.1. Örnek Problemin Tanımı



Şekil 4. Örnek Problem İçin Tanımlanan Konteyner Ve Kutuların Tamamının Gösterimi

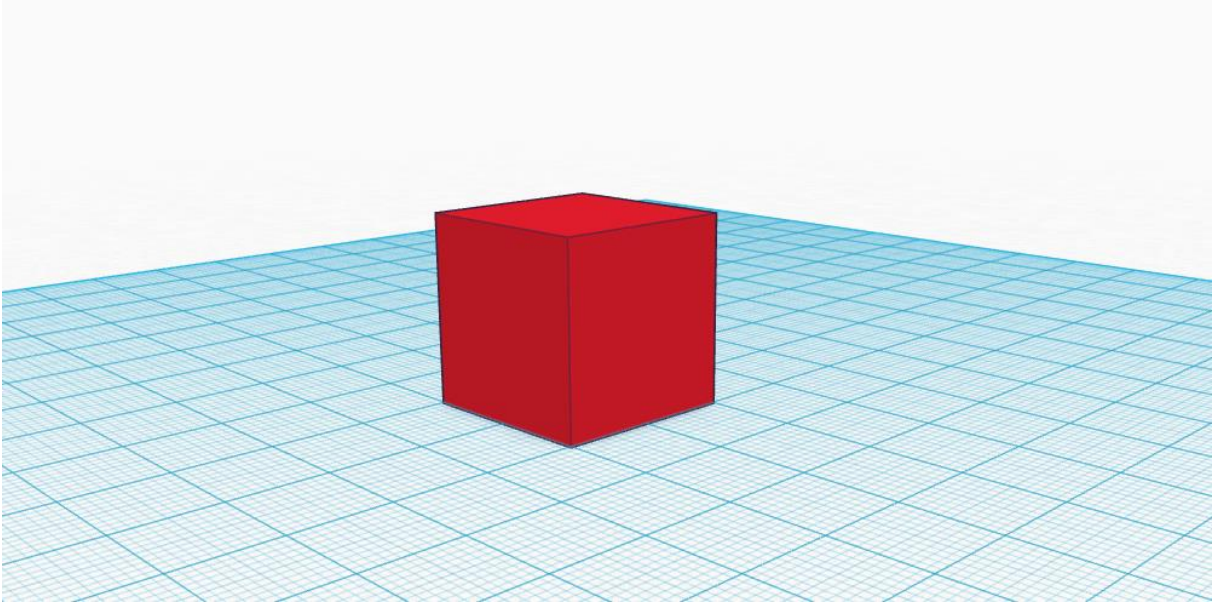
Örnek problem için tanımlanan konteyner tipi 1 tanedir. Bu konteynere ait tüm değerler aşağıdaki gibidir:

Konteyner genişlik değeri: 2 birim (metre)

Konteyner yükseklik değeri: 2 birim (metre)

Konteyner derinlik değeri: 2 birim (metre)

Konteynerin taşıyabileceği maksimum ağırlık: 80 birim (kilogram)

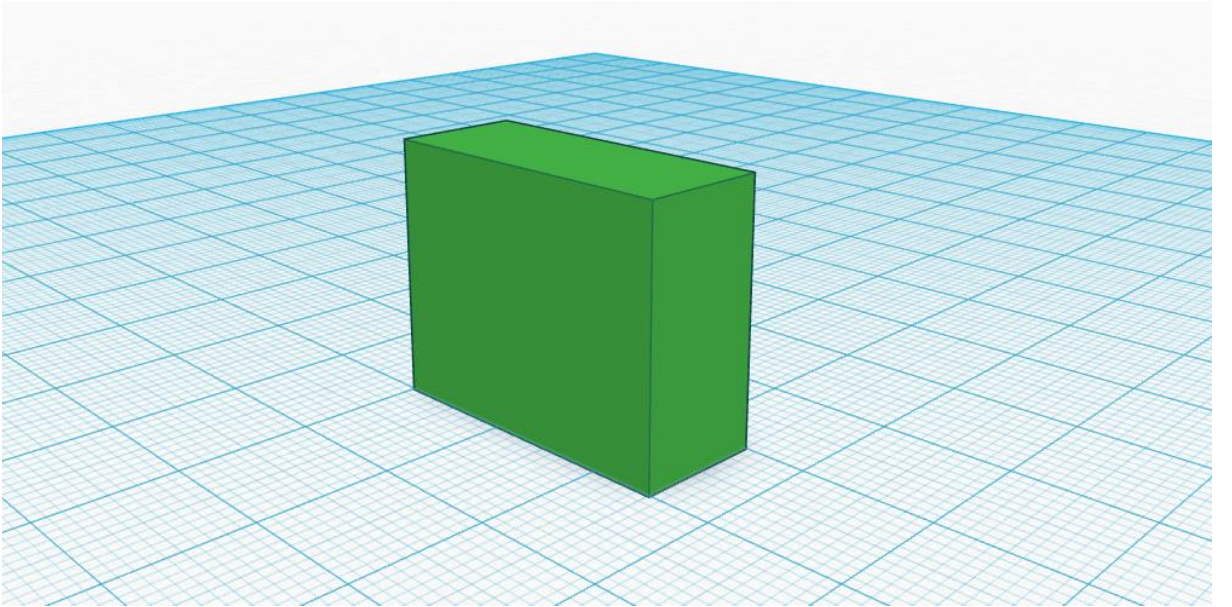


Şekil 5. Tanımlı Konteynerin Gösterimi. (2x2x2, Taşıyabileceği Maksimum Ağırlık 80 KG)

Bu örnekte, konteyner ve kutu boyutları için yapılan birim tanımı metre olarak, konteynerin taşıyabileceği maksimum ağırlık birimi ile kutuların ağırlık birimleri ise kilogram olarak kabul edilmiştir.

Konteyner içine yerleştirilmesi düşünülen kutu tipi 7 tanedir.

Kutu tipi A:



Şekil 6. Tipinde Tanımlanan Kutunun Gösterimi (2x2x1, 40 KG, 1 Adet)

Genişlik değeri: 2 birim (metre)

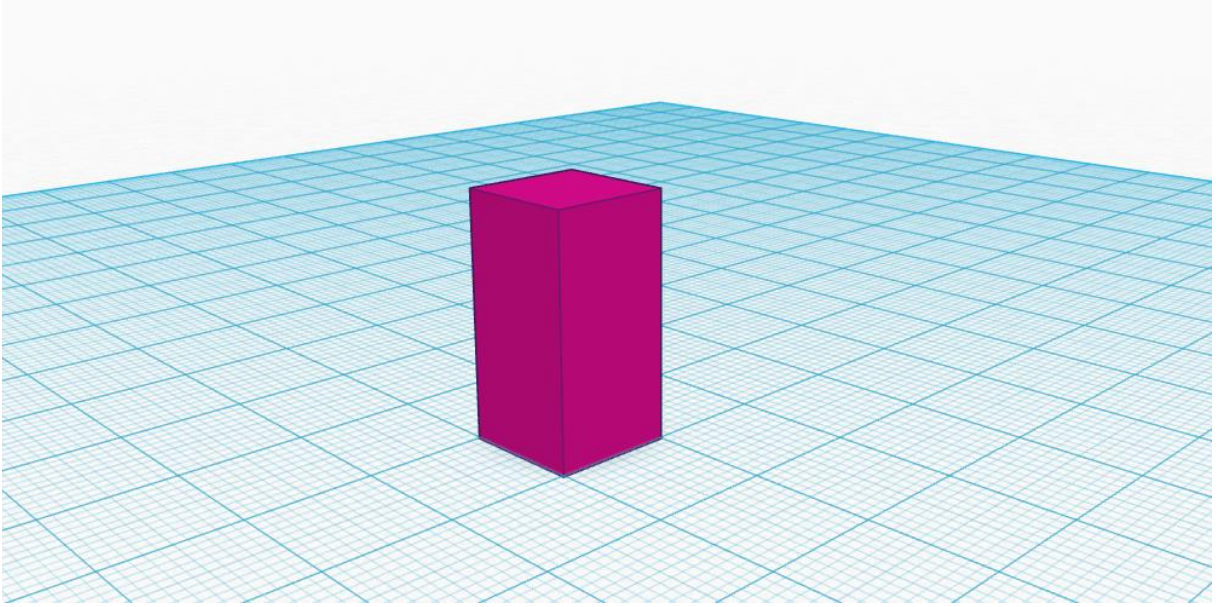
Yükseklik değeri: 2 birim (metre)

Derinlik değeri: 1 birim (metre)

Kutunun ağırlığı: 40 kg

Problem için tanımlanan A tipinde kutu sayısı: 1

Kutu tipi B:



Şekil 7. B Tipinde Tanımlanan Kutunun Gösterimi (1x2x1, 20 KG, 1 Adet)

Genişlik değeri: 1 birim (metre)

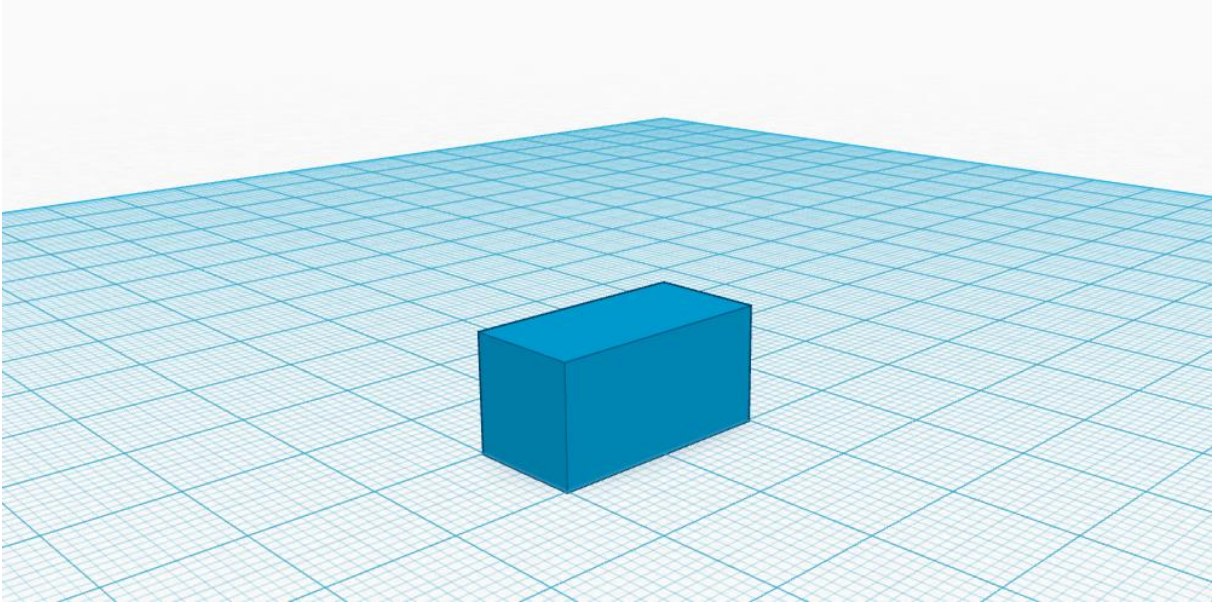
Yükseklik değeri: 2 birim (metre)

Derinlik değeri: 1 birim (metre)

Kutunun ağırlığı: 20 kg

Problem için tanımlanan b tipinde kutu sayısı: 1

Kutu tipi C:



Şekil 8. C Tipinde Tanımlanan Kutunun Gösterimi (1x1x2, 30 KG, 1 Adet)

Genişlik değeri: 1 birim (metre)

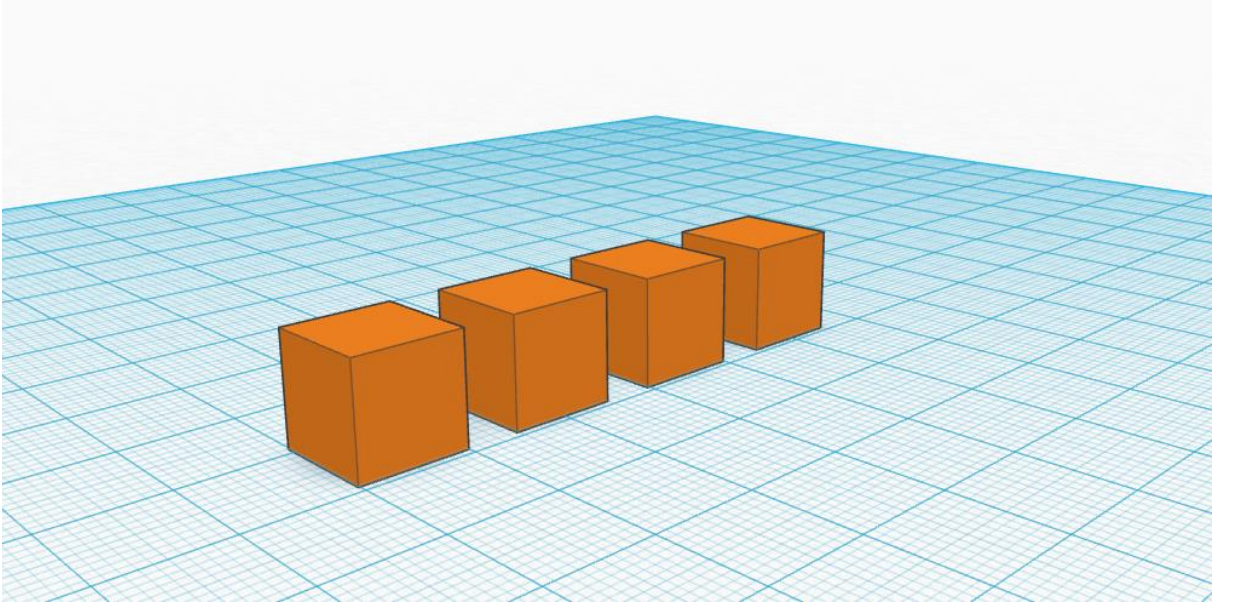
Yükseklik değeri: 1 birim (metre)

Derinlik değeri: 2 birim (metre)

Kutunun ağırlığı: 30 kg

Problem için tanımlanan C tipinde kutu sayısı: 1

Kutu tipi D:



Şekil 9. D Tipinde Tanımlanan Kutunun Gösterimi (1x1x1, 10 KG, 4 adet)

Genişlik değeri: 1 birim (metre)

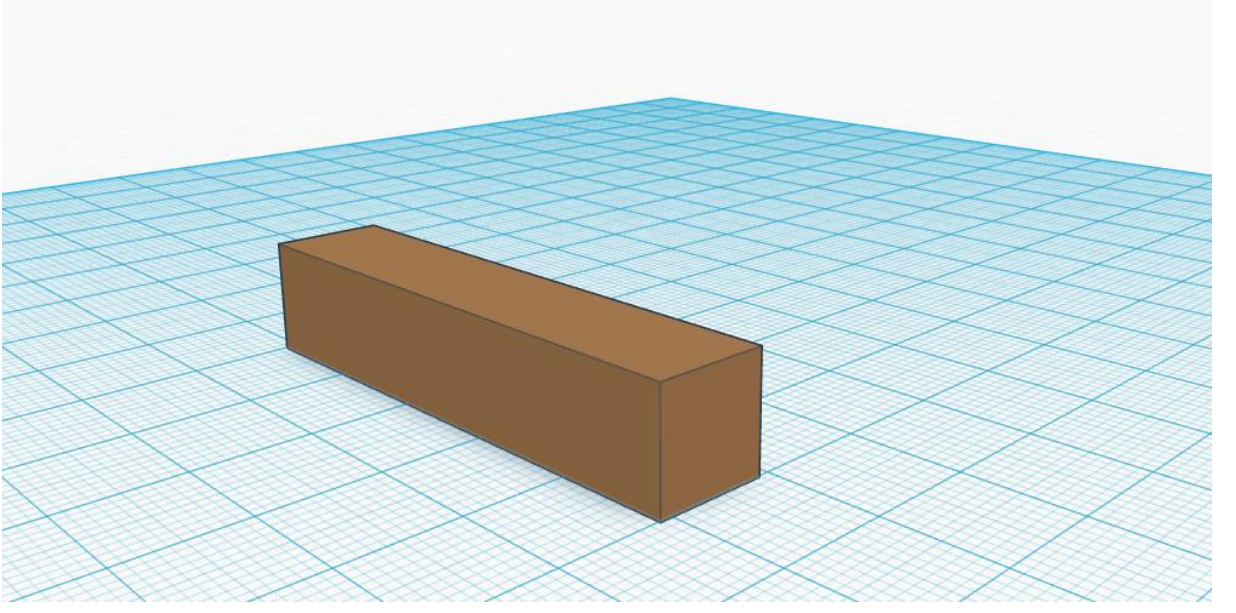
Yükseklik değeri: 1 birim (metre)

Derinlik değeri: 1 birim (metre)

Kutunun ağırlığı: 10 kg

Problem için tanımlanan A tipinde kutu sayısı: 4

Kutu tipi E:



Şekil 10. D Tipinde Tanımlanan Kutunun Gösterimi (1x1x1, 10 KG, 4 adet)

Genişlik değeri: 5 birim (metre)

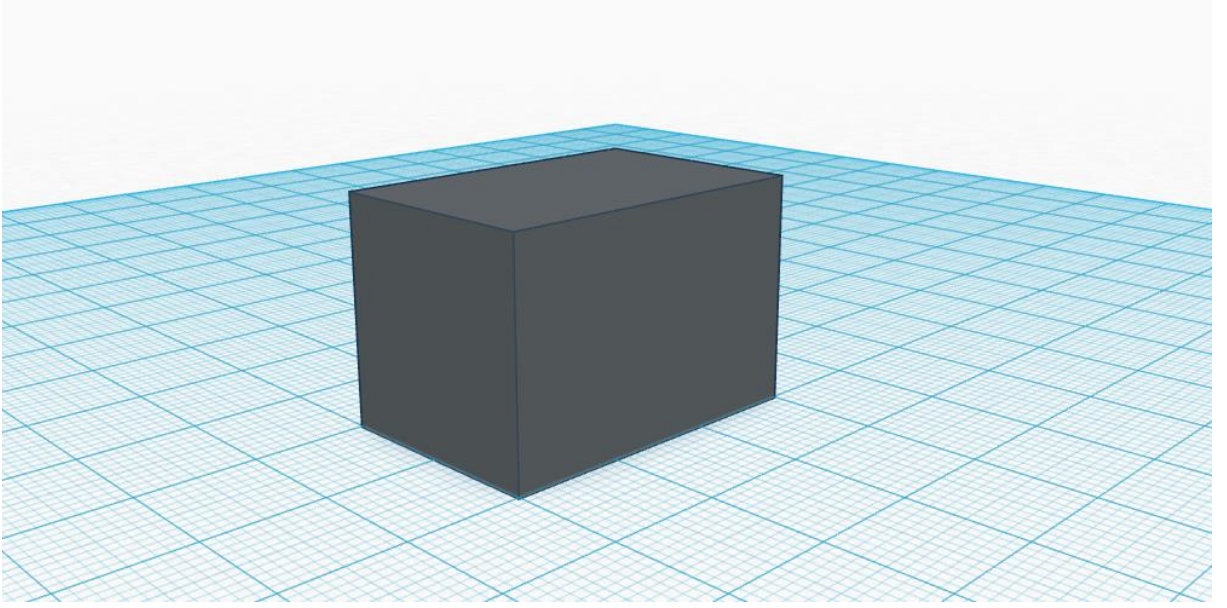
Yükseklik değeri: 1 birim (metre)

Derinlik değeri: 1 birim (metre)

Kutunun ağırlığı: 40 kg

Problem için tanımlanan E tipinde kutu sayısı: 1

Kutu tipi F:



Şekil 11. F Tipinde Tanımlanan Kutunun Gösterimi (2x2x3, 40 KG, 1 adet)

Genişlik değeri: 2 birim (metre)

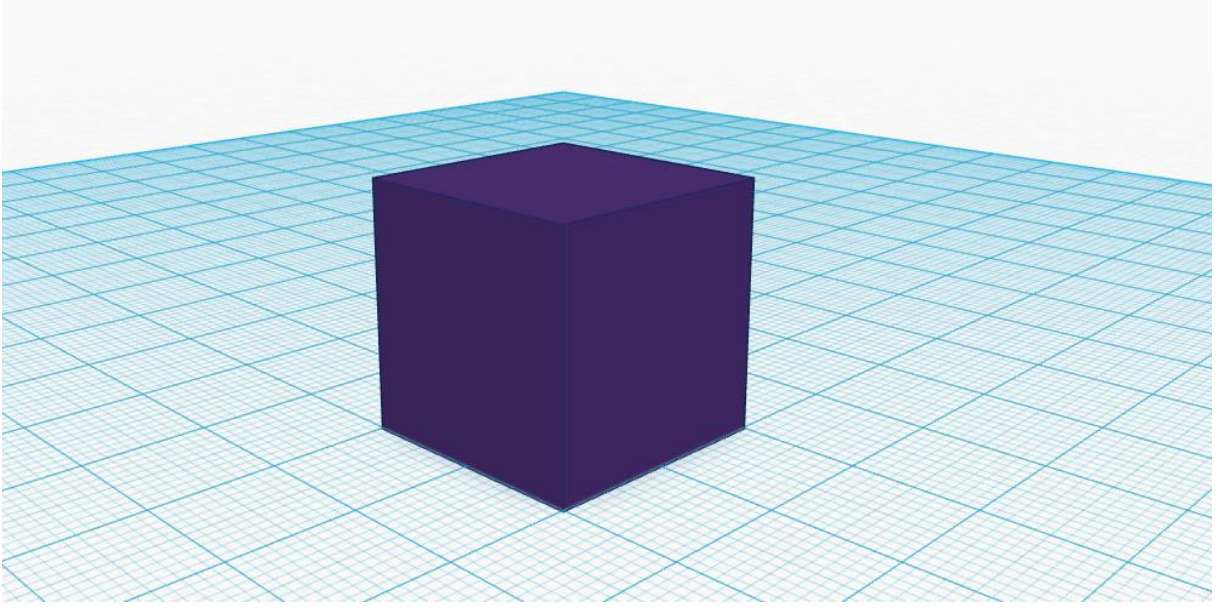
Yükseklik değeri: 2 birim (metre)

Derinlik değeri: 3 birim (metre)

Kutunun ağırlığı: 40 kg

Problem için tanımlanan A tipinde kutu sayısı: 1

Kutu tipi G:



Şekil 12. G Tipinde Tanımlanan Kutunun Gösterimi (2x2x2, 81 KG, 1 adet)

Genişlik değeri: 2 birim (metre)

Yükseklik değeri: 2 birim (metre)

Derinlik değeri: 2 birim (metre)

Kutunun ağırlığı: 81 kg

Problem için tanımlanan G tipinde kutu sayısı: 1

5.2. Örnek Problemin, Kodlanan Web Uygulamasının Arayüzüne Tanımlanması

Yukarıda tanımlanan problem, tez çalışması için kodlanmış olan web uygulamasına aşağıdaki şekilde gösterildiği gibi tanımlanır;

Container			
Length	Height	Width	Weight
2	2	2	80

Box				
Length	Height	Width	Quantity	Weight
2	2	1	1	40
1	2	1	1	20
1	1	2	1	30
1	1	1	4	10
5	1	1	1	40
2	2	3	1	40
2	2	2	1	81

Şekil 13. Örnek Problemin Web Arayüzüne Tanımlanmasının Gösterimi

Problemin web uygulamasına tanımlanmasından sonra paketle(Pack) bağlantısına tıklanır.

5.3. Web Uygulamasının Örnek Problemin Tüm Değerlerini Paketle İşleme Başlamadan Önce Kontrol Etmesi

Bu adımda örnek problem için web kullanıcısı tarafından sisteme tanımlanan tüm değerler kontrol edilir. Eksik ya da hatalı bir tanımlama var ise paketleme işlemi başlamaz. Eksik ya da hatalı bir tanımlamaya örnek verecek olursak; herhangi bir kutu için genişlik, yükseklik, derinlik, ağırlık ya da adet değerlerinden birinin eksik girilmesi gibi.

Kodlanan web uygulamasında bu değerleri kontrol eden metot `isInputsValidated()` metodudur. Integer (tamsayı) değerler yazılmış ise bu metot true değeri döndürür.

Probleme tanımlanan konteyner ve kutular `Manager.cs` (class) tarafından oluşturulur. `CreateNewContainer()` metodu. `CreateNewContainer()` metodu konteyner ve kutu listesini `GetBestContainer()` metoduna gönderir. `GetBestContainer()` metodunun geriye döndüreceği konteyner 2x2x2 olan konteynerdir. `CreateNewContainer()` metodu `Manager` class'ı tarafından kaç tane konteyner oluşturulmuş bunu kontrol

eder. An itibarı ile oluşturulan konteyner sayısı 1 adettir. CreateNewContainer metodu bu konteynere benzersiz bir numara olması için ID değerini 1 olarak tanımlar. Ayrıca bu metod bu konteyner için boşluk tanımını (space defination) da oluşturur. $2 \times 2 \times 2$, $(0, 0, 0)$, $(2, 2, 2)$ (x_1, y_1, z_1) , (x_2, y_2, z_2) . Boşluk tanımı tez çalışmasının bu bölümünde ilerleyen aşamalarda açıklanacaktır. Bu aşamaların en sonunda Manager sınıfının kutuları yerleştirmesi için kullanacağı 1 adet konteyner, boşluk tanımı ile oluşmuş olacaktır.

5.4. Konteyner İçine Yerleştirilecek Kutu Listesi Oluşturulur

- 1.Sıradaki kutu: A tipinde tanımlanan 1 numaralı kutu ($2 \times 2 \times 1$, 40 kg)
- 2.Sıradaki kutu: B tipinde tanımlanan 2 numaralı kutu ($1 \times 2 \times 1$, 20kg)
- 3.Sıradaki kutu: C tipinde tanımlanan 3 numaralı kutu ($1 \times 1 \times 2$, 30kg)
- 4.Sıradaki kutu: D tipinde tanımlanan 4 numaralı kutu ($1 \times 1 \times 1$, 10kg)
- 5.Sıradaki kutu: D tipinde tanımlanan 5 numaralı kutu ($1 \times 1 \times 1$, 10kg)
- 6.Sıradaki kutu: D tipinde tanımlanan 6 numaralı kutu ($1 \times 1 \times 1$, 10kg)
- 7.Sıradaki kutu: D tipinde tanımlanan 7 numaralı kutu ($1 \times 1 \times 1$, 10kg)
- 8.Sıradaki kutu: E tipinde tanımlanan 8 numaralı kutu ($5 \times 1 \times 1$, 40kg)
- 9.Sıradaki kutu: F tipinde tanımlanan 9 numaralı kutu ($2 \times 2 \times 3$, 40kg)
- 10.Sıradaki kutu: G tipinde tanımlanan 10 numaralı kutu ($2 \times 2 \times 2$, 81kg)

5.5. Ağırlık Kontrolü

Web uygulaması sisteme tanımlanmış olan tüm konteyner tiplerini ve bu konteynerlere ait maksimum ağırlık değerlerini okur. Örnek problemde tanımlı konteyner tipi 1 adettir. Bu konteyner için tanımlanmış maksimum ağırlık değeri 80 kg'dır. Bu değer ile tüm kutuların ağırlıkları karşılaştırılır.

A tipinde tanımlanan kutu ağırlığı 40 kg'dır. Bu değer konteynerin taşıyabileceği bir değerdir.

B tipinde tanımlanan kutu ağırlığı 40 kg'dır. Bu değer konteynerin taşıyabileceği bir değerdir.

C tipinde tanımlanan kutu ağırlığı 40 kg'dır. Bu değer konteynerin taşıyabileceği bir değerdir.

D tipinde tanımlanan kutu ağırlığı 40 kg'dır. Bu değer konteynerin taşıyabileceği bir değerdir.

E tipinde tanımlanan kutu ağırlığı 40 kg'dır. Bu değer konteynerin taşıyabileceği bir değerdir.

F tipinde tanımlanan kutu ağırlığı 40 kg'dır. Bu değer konteynerin taşıyabileceği bir değerdir.

G tipinde tanımlanan kutu ağırlığı 81 kg'dır. Bu değer konteynerin taşıyabileceği değer dışındadır. Bu nedenle bu adımda bu G tipinde tanımlı olan tek kutu paketlenen kutu listesinden çıkartılır. Bu kutu, " konteyner içine yerleştirilmesi mümkün olmayan kutu listesine" dâhil edilir.

Bu aşama sonunda 5.4. adımda oluşturulmuş olan " konteyner içine yerleştirilecek kutu listesi " tekrar güncellenir. Listenin son hali aşağıdaki gibidir:

Konteyner içine yerleştirilecek kutu listesi:

- 1.Sıradaki kutu: A tipinde tanımlanan 1 numaralı kutu (40 kg)
- 2.Sıradaki kutu: B tipinde tanımlanan 2 numaralı kutu (20kg)
- 3.Sıradaki kutu: C tipinde tanımlanan 3 numaralı kutu (30kg)
- 4.Sıradaki kutu: D tipinde tanımlanan 4 numaralı kutu (10kg)
- 5.Sıradaki kutu: D tipinde tanımlanan 5 numaralı kutu (10kg)
- 6.Sıradaki kutu: D tipinde tanımlanan 6 numaralı kutu (10kg)
- 7.Sıradaki kutu: D tipinde tanımlanan 7 numaralı kutu (10kg)
- 8.Sıradaki kutu: E tipinde tanımlanan 8 numaralı kutu (40kg)
- 9.Sıradaki kutu: F tipinde tanımlanan 9 numaralı kutu (40kg)

5.6. Konteyner İçine Yerleştirilmesi Mümkün Olmayan Kutu Listesi Oluşturulur:

- 10.Sıradaki kutu: G tipinde tanımlanan 10 numaralı kutu (81kg)

5.7.Hacim Kontrolü

Bu adımda problem için tanımlanan kutuların hacimleri hesaplanır. Bu hacim değerlerinden konteyner hacminden büyük olan varsa yukarıda ifade edilen listeler tekrar güncellenir.

Dikdörtgenler prizması ve küp şeklinde olan cisimlerin hacim hesabı:

Hacim=genişlik*yükseklik*derinlik

Sisteme tanımlanan konteyner hacmi= $2 \times 2 \times 2 = 8 \text{ m}^3$, (genişlik*yükseklik*derinlik)

Konteyner içine yerleştirilecek kutu listesindeki kutuların hacimlerinin hesaplanması:

- 1.Sıradaki kutu: A tipinde tanımlanan 1 numaralı kutu. Hacmi= $2*2*1=4 \text{ m}^3$,
- 2.Sıradaki kutu: B tipinde tanımlanan 2 numaralı kutu. Hacmi= $1*2*1=2 \text{ m}^3$,
- 3.Sıradaki kutu: C tipinde tanımlanan 3 numaralı kutu. Hacmi= $1*1*2=2 \text{ m}^3$,
- 4.Sıradaki kutu: D tipinde tanımlanan 4 numaralı kutu. Hacmi= $1*1*1=1 \text{ m}^3$,
- 5.Sıradaki kutu: D tipinde tanımlanan 5 numaralı kutu. Hacmi= $1*1*1=1 \text{ m}^3$,
- 6.Sıradaki kutu: D tipinde tanımlanan 6 numaralı kutu. Hacmi= $1*1*1=1 \text{ m}^3$,
- 7.Sıradaki kutu: D tipinde tanımlanan 7 numaralı kutu. Hacmi= $1*1*1=1 \text{ m}^3$,
- 8.Sıradaki kutu: E tipinde tanımlanan 8 numaralı kutu. Hacmi= $5*1*1=5 \text{ m}^3$,
- 9.Sıradaki kutu: F tipinde tanımlanan 9 numaralı kutu. Hacmi= $2*2*3=12 \text{ m}^3$,

Görüldüğü üzere 9.sırada bulunan F tipinde tanımlı olan 1 adet kutu hacminden dolayı konteyner içine yerleştirilemeyecektir. Çünkü konteynere ait hacim değeri 8 m^3 iken bu kutuya ait hacim değeri 12 m^3 'dür. Bu durumda 4.adım ve sonrasında oluşturulan listeler tekrar düzenlenecektir:

Konteyner içine yerleştirilecek kutu listesi:

- 1.Sıradaki kutu: A tipinde tanımlanan 1 numaralı kutu (40 kg)
- 2.Sıradaki kutu: B tipinde tanımlanan 2 numaralı kutu (20kg)
- 3.Sıradaki kutu: C tipinde tanımlanan 3 numaralı kutu (30kg)
- 4.Sıradaki kutu: D tipinde tanımlanan 4 numaralı kutu (10kg)
- 5.Sıradaki kutu: D tipinde tanımlanan 5 numaralı kutu (10kg)
- 6.Sıradaki kutu: D tipinde tanımlanan 6 numaralı kutu (10kg)
- 7.Sıradaki kutu: D tipinde tanımlanan 7 numaralı kutu (10kg)
- 8.Sıradaki kutu: E tipinde tanımlanan 8 numaralı kutu (40kg)
- 9.Sıradaki kutu: F tipinde tanımlanan 9 numaralı kutu (40kg)

Konteyner içine yerleştirilmesi mümkün olmayan kutu listesi:

- 10.Sıradaki kutu: G tipinde tanımlanan 10 numaralı kutu (81kg)
- 9.Sıradaki kutu: F tipinde tanımlanan 9 numaralı kutu (40kg, 12 m^3)

5.8. Geniřlik, Ykseklik, Derinlik Deęerlerinin Kontrol

Bu adımda kutulara ait geniřlik, ykseklik ve derinlik deęerlerinin her biri konteynere ait aynı deęerler ile kıyaslanır.

1.Sıradaki kutu: A tipinde tanımlanan 1 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 2,2,1. Bu kutunun konteyner iine yerleřtirilmesi iin herhangi bir engel yoktur.

2.Sıradaki kutu: B tipinde tanımlanan 2 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 1,2,1. Bu kutunun konteyner iine yerleřtirilmesi iin herhangi bir engel yoktur.

3.Sıradaki kutu: C tipinde tanımlanan 3 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 1,1,2. Bu kutunun konteyner iine yerleřtirilmesi iin herhangi bir engel yoktur.

4.Sıradaki kutu: D tipinde tanımlanan 4 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 1,1,1. Bu kutunun konteyner iine yerleřtirilmesi iin herhangi bir engel yoktur.

5.Sıradaki kutu: D tipinde tanımlanan 5 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 1,1,1. Bu kutunun konteyner iine yerleřtirilmesi iin herhangi bir engel yoktur.

6. Sıradaki kutu: D tipinde tanımlanan 6 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 1,1,1. Bu kutunun konteyner iine yerleřtirilmesi iin herhangi bir engel yoktur.

7.Sıradaki kutu: D tipinde tanımlanan 7 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 1,1,1. Bu kutunun konteyner iine yerleřtirilmesi iin herhangi bir engel yoktur.

8.Sıradaki kutu: E tipinde tanımlanan 8 numaralı kutu. Geniřlik, ykseklik ve derinlik deęerleri sırası ile 5,1,1. Bu kutuya ait geniřlik deęeri 5 birimdir. Bu yzden bu kutunun konteyner iine yerleřtirilmesi sz konusu deęildir. Bu nedenle bu ařamada bu kutu elenir.

Listeler tekrar gncellenir:

Konteyner içine yerleştirilecek kutu listesi:

- 1.Sıradaki kutu: A tipinde tanımlanan 1 numaralı kutu (40 kg, 4 m³)
- 2.Sıradaki kutu: B tipinde tanımlanan 2 numaralı kutu (20kg, 2 m³)
- 3.Sıradaki kutu: C tipinde tanımlanan 3 numaralı kutu (30kg, 2 m³)
- 4.Sıradaki kutu: D tipinde tanımlanan 4 numaralı kutu (10kg, 1 m³)
- 5.Sıradaki kutu: D tipinde tanımlanan 5 numaralı kutu (10kg, 1 m³)
- 6.Sıradaki kutu: D tipinde tanımlanan 6 numaralı kutu (10kg, 1 m³)
- 7.Sıradaki kutu: D tipinde tanımlanan 7 numaralı kutu (10kg, 1m³)

Konteyner içine yerleştirilmesi mümkün olmayan kutu listesi:

Manager Sınıfı konteyner içine yerleştirilmesi mümkün olmayan kutuları IsProcessed ile işaretler. (ağırlık, hacim ya da ölçütler yönünden konteyner içine giremeyen kutular). Ve yine

- 10.Sıradaki kutu: G tipinde tanımlanan 10 numaralı kutu (81kg)
- 9.Sıradaki kutu: F tipinde tanımlanan 9 numaralı kutu (40kg, 12 m³)
- 8.Sıradaki kutu: E tipinde tanımlanan 8 numaralı kutu (40kg, kutuya ait genişlik değeri 5 birim)

Bu aşama sonunda konteyner içine yerleştirilecek kutular ile konteyner içine yerleştirilmesi mümkün olmayan kutular belli olmuş olur. Konteyner için yerleştirilecek kutu sayısı 7 iken konteyner içine yerleştirilemeyecek kutu sayısı 3'dür.

5.9. Konteyner İçine Yerleştirilecek Olan Kutu Listesinin Kutu Hacim Değerlerine Göre Yeniden Sıralanması

Bu aşama bu tez çalışmasında problemin çözümünün temel prensiplerinden birini ifade eder. Paketlemek için ilk olarak hacmi en büyük olan kutu hesaplara dâhil edilecektir. Bu prensibe göre konteyner içine yerleştirilecek kutu listesinde kutu hacim değerlerine göre tekrar bir sıralama yapılır.

Bu işlemi yapan yine Manager Sınıfıdır.

Hacmi büyük olan kutular en başta olacak şekilde sıralanır:

- 1.Sıradaki kutu: A tipinde tanımlanan 1 numaralı kutu (40 kg, 4 m³)
- 2.Sıradaki kutu: B tipinde tanımlanan 2 numaralı kutu (20kg, 2m³)
- 3.Sıradaki kutu: C tipinde tanımlanan 3 numaralı kutu (30kg, 2 m³)

4.Sıradaki kutu: D tipinde tanımlanan 4 numaralı kutu (10kg, 1 m³)

5.Sıradaki kutu: D tipinde tanımlanan 5 numaralı kutu (10kg, 1 m³)

6.Sıradaki kutu: D tipinde tanımlanan 6 numaralı kutu (10kg, 1 m³)

7.Sıradaki kutu: D tipinde tanımlanan 7 numaralı kutu (10kg, 1 m³)

5.10. Konteyner İine Yerleřtirilecek Her Kutu İin Aynı Olacak İřlemler Dizisine Bařlanır

Bu adımı kontrol eden Manager Sınıfının PerformAllTasks() metodudur. Manager sınıfı konteyner iine yerleřtirilecek kutu listesini kontrol eder. İřleme alınmayan kutu var mı? Sorgusunu deęerlendirir. Sorgunun cevabı iřleme alınmayan kutuların olduęudur. Daha sonra aynı sınıf ařaęıdaki iřlemleri iřleme alınmamıř her kutu iin yineler.

İřlem 1: Konteyner İine Yerleřtirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seilir

Kutu No:1

Kutu Tipi: A

Kutu geniřlik deęeri: 2 birim

Kutu ykseklik deęeri: 2 birim

Kutu geniřlik deęeri:1 birim

Kutu aęırlıęı: 40 kg.

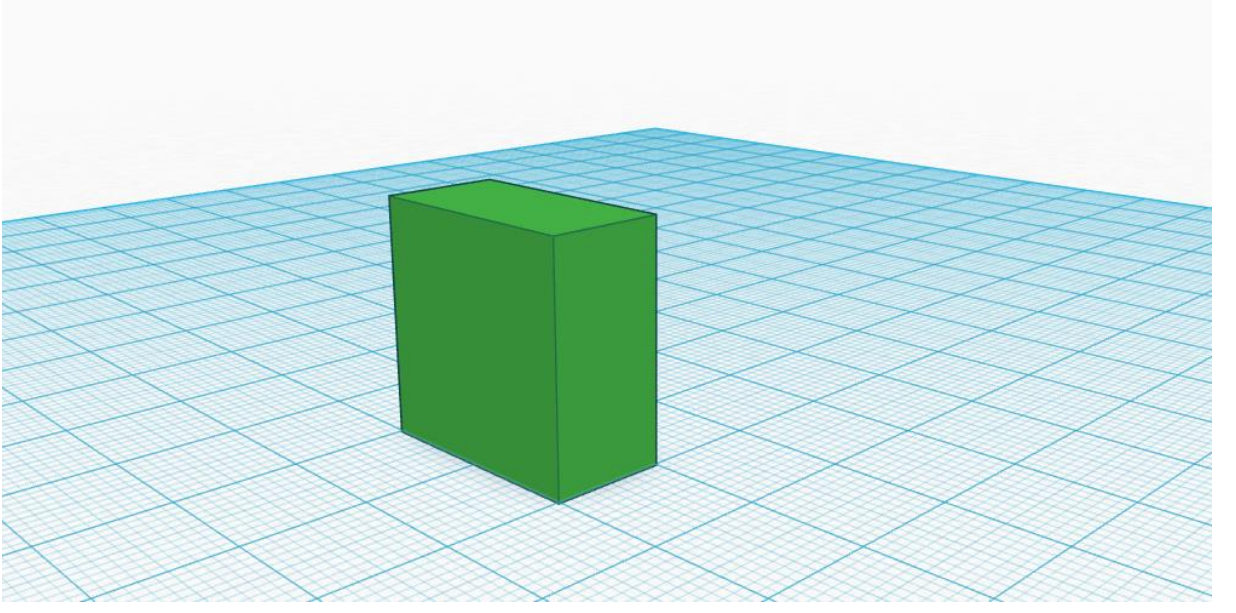
Kutu Hacmi: 4 m³

İřlem 2:Seilen Kutunun Dndrlmesi Gerekip Gerekmedięi Sorgulanır

Bu iřlemi yapan metotlar da Manager Sınıfı dahilinde yazılmıř metotlardır.

Kutu A iin tanımlı deęerlere bakıldıęında, kutunun geniřlik deęeri 2, ykseklik deęeri 2, derinlik deęeri ise 1 birimdir.(2, 2, 1)

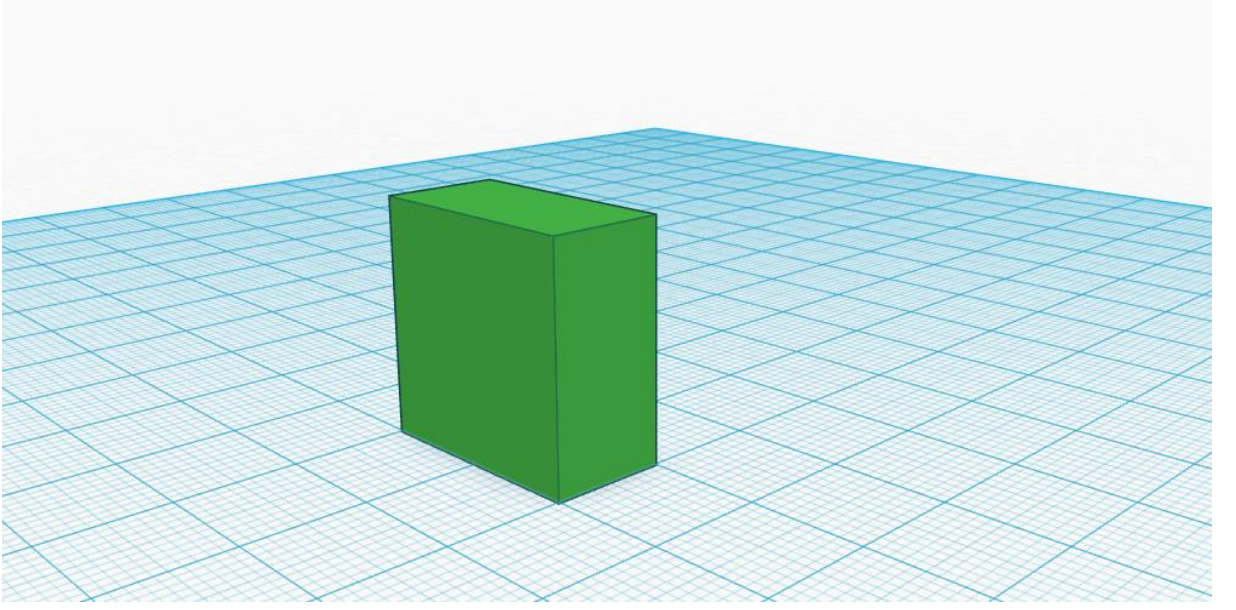
Kutu, bu tanımlamalara gre ařaęıdaki řekilde gsterildięi pozisyonda yzey ile temas halindedir.



Şekil 14. A Tipinde Tanımlı Kutunun İlk Pozisyonunun Gösterimi

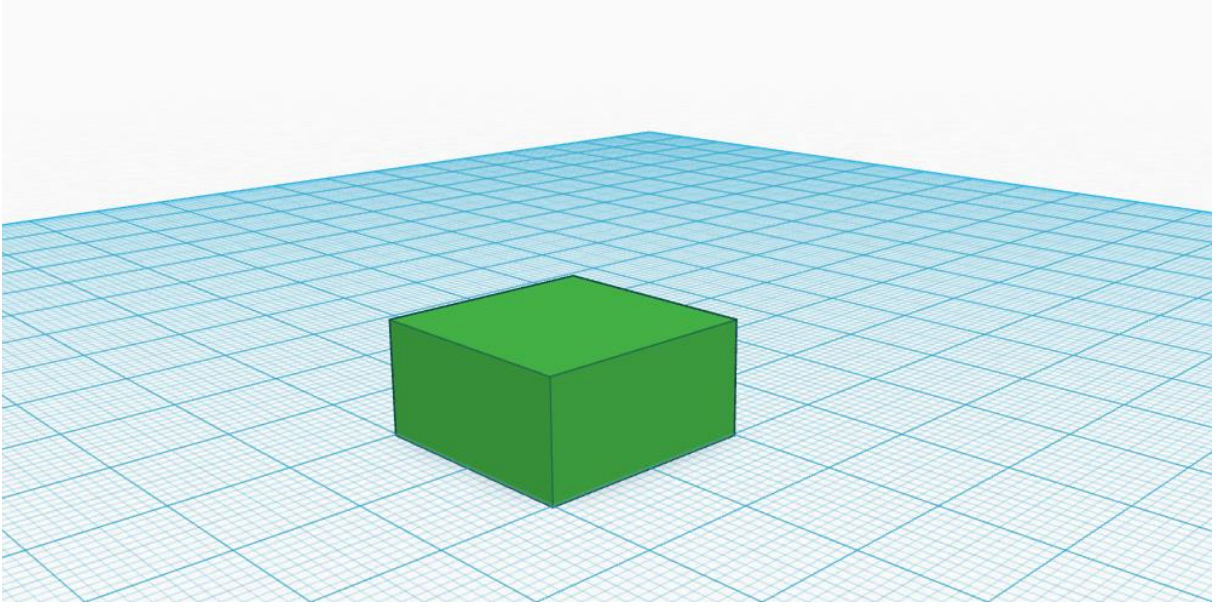
Bu durumda bu kutu, genişlik ve derinlik tanımını oluşturan yüzey üzerinde bulunduğu düzleme temas halindedir.

Kutu yükseklik ve derinlik tanımını oluşturan yüzey üzerinde bulunduğu düzleme temas halinde olursa kutu için yeni genişlik değeri 2, yükseklik değeri 2 ve derinlik değeri 1 olacaktır.(2x2x1) Kutu bu tanımlamalara göre aşağıdaki gösterildiği pozisyonda yüzey ile temas halindedir.



Şekil 15. A Tipinde Tanımlı Kutunun, Yükseklik Ve Derinlik Tanımı Oluşturan Yüzey Üzerinde Bulunduğu Düzleme Temas Halindeyken Görünümü

Bir diğer seçenek iste kutunun genişlik ve yükseklik tanımını oluşturan yüzey üzerinde bulunduğu yüzeye temas halinde olduğu durumdur. Bu durumda, genişlik değeri 2, yükseklik değeri 1, derinlik değeri ise 2 olacaktır. (2x1x2). Kutu bu tanımlamalara göre aşağıdaki gösterildiği pozisyonda yüzey ile temas halindedir.



Şekil 16. A Tipinde Tanımlı Kutunun, Genişlik Ve Yükseklik Tanımı Oluşturan Yüzey Üzerinde Bulunduğu Düzleme Temas Halindeyken Görünümü

5.10.1. Boşluk Tanımı (Space Defination)

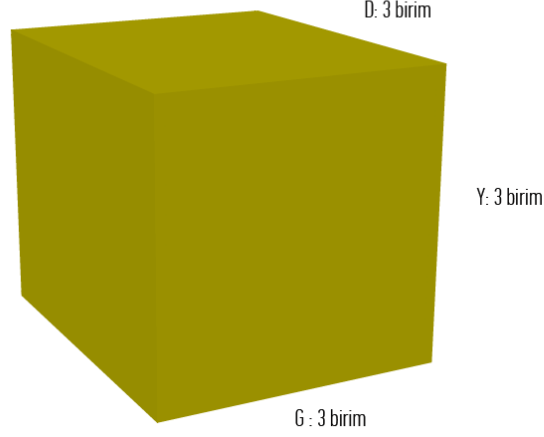
Boşluk tanımı (SpaceDefination):

Bu tanım, bu tez çalışmasında problemin çözümü için uygulanan yöntemi ifade eder. Boşluk tanımı konteyner içinde yer alan boş alanları ifade eder. Bu boşluk tanımları sadece düzgün prizma şekillerinden oluşabilir. Ayrıca bu boşluk tanımlarının bir araya gelerek oluşturabileceği şekiller ile de tanımlanabilir.

Konteyner içine henüz bir kutu yerleştirilmediğinde konteyner içindeki boşluk tanımının ölçüleri konteyner ölçüleri ile aynıdır. Dolayısı ile boşluk tanımının hacmi ile konteynerin hacmi de aynıdır.

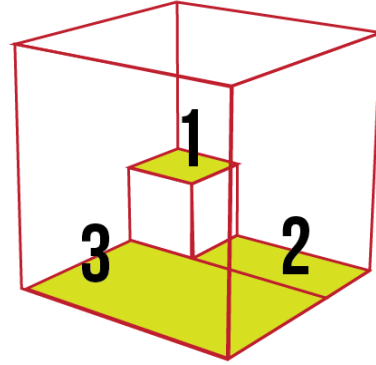
Aşağıdaki resimde boşluk tanımı başka bir örnek üzerinden açıklanmıştır. Ölçüleri 3x3x3 (genişlik, yükseklik, derinlik) olan bir konteynere ait boşluk tanımı gösterilmiştir.

Konteyner içinde kutu olmadığı zaman “boşluk” tanımının görüntüsü:



Şekil 17. Ölçüleri 3x3x3 Olan Konteynere Ait Boşluk Tanımının Gösterilmesi

Aynı konteyner içine 1x1x1 ölçülerinde kutu yerleştirildiği zaman, konteynerin görüntüsü ve konteyner içinde oluşan yeni boşluk tanımlarının oluşacağı alanlar 1, 2 ve 3 numaraları ile gösterilmiştir.



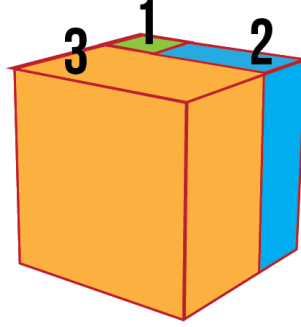
Şekil 18. Ölçüleri 3x3x3 Olan Konteyner İçine Ölçüsü 1x1x1 Olan Kutu Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor

Aşağıdaki resimde ise konteyner içine yerleştirilen kutudan sonra konteyner içinde oluşan bu 3 boşluk tanımının hacimsel olarak tamamı gösterilmiştir.

Boşluk tanımını 1'i ifade eden ölçüler: 1x2x1 (genişlik, yükseklik, derinlik)

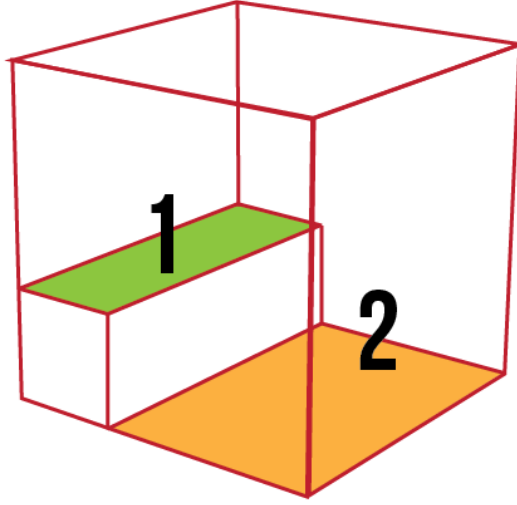
Boşluk tanımını 2'yi ifade eden ölçüler: $2 \times 3 \times 1$ (genişlik, yükseklik, derinlik)

Boşluk tanımını 3'ü ifade eden ölçüler: $3 \times 3 \times 2$ (genişlik, yükseklik, derinlik)



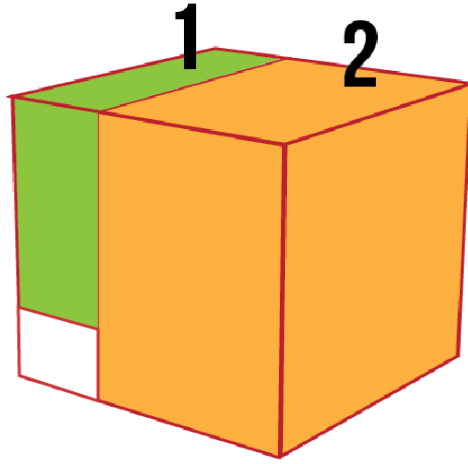
Şekil 19. Oluşan 3 Boşluk Tanımı Tüm Hacmi İle Gösteriliyor

Aynı konteynere yerleştirilen kutu $1 \times 1 \times 3$ ebatlarında ise konteyner içinde oluşan boşluk tanım sayısı 2 olacaktır. Aşağıdaki resimde bu senaryo gösterilmiştir:



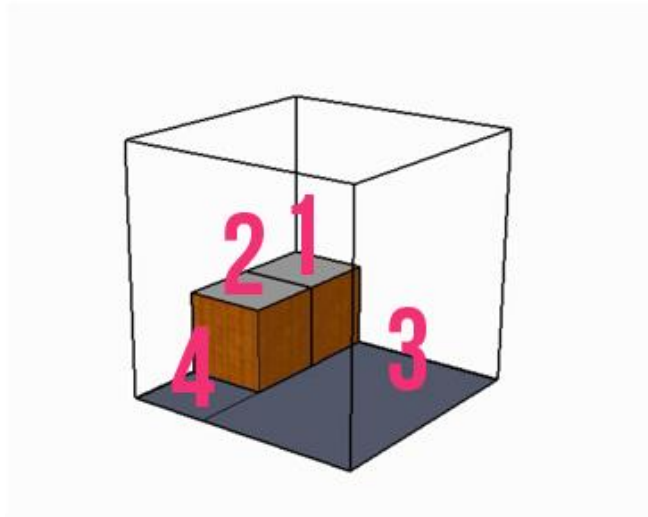
Şekil 20. Ölçüleri $3 \times 3 \times 3$ Olan Konteyner İçinde 2 Boşluk Tanımının Oluşması

Bu durumda 1 numaralı alanda üzerinde yükselen boşluk tanımını 1' e ait ölçüler: $1 \times 2 \times 3$ iken 2 numaralı alan üzerinde yükselen boşluk tanımını 2'ye ait ölçüler: $2 \times 3 \times 2$ 'dir.



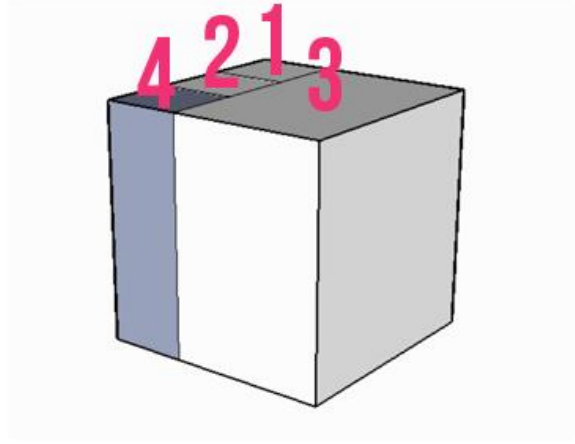
Şekil 21. Ölçüleri 3x3x3 Olan Konteyner İçinde 2 Boşluk Tanımının Oluşması

Web uygulaması bazı durumlarda boşluk tanımlarını birleştirebilir. Aşağıdaki şekilde boyutları 3x3x3 olan konteynere boyutları 1x1x1 olan aynı türde 2 tane kutu yerleştirilmiştir. Bu aşamadan sonra konteyner içinde oluşan boşluk tanım sayısı ilk aşamada 4 adettir. Fakat web uygulaması 1 ve 2 numara ile gösterilen boşluk tanımını birleştirmiştir. Aşağıdaki resimde boşluk tanımlarının birleşmeden önceki görüntüsü gösterilmektedir:

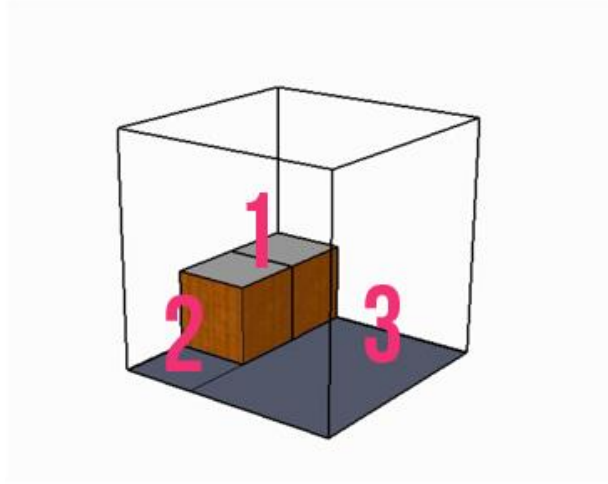


Şekil 22. Boşluk Tanımlarının Birleştirilmesine Bir Örnek

Boşluk tanımları birleşmeden önce tüm hacimleri ile gösteriliyor:



Şekil 23. Boşluk Tanımlarının Birleştirilmeden Önce Tam Hacmi İle Gösterilmesi



Şekil 24. Boşluk Tanımlarının Birleştirilmesinin Gösterimi

Yukarıdaki resimlerden ve anlatılanlardan da anlaşıldığı üzere web uygulaması 1 ve 2 numaralı boşluk tanımını ayrı olarak hesaba katmıyor, bu iki boşluk tanımını gerekli kıyaslamalar sonra da hesaplamalar yaparak birleştiriyor. Uygulamanın bu şekilde davranmasının sebebi, yaptığı hesaplamalara dayanıyor. Uygulama boşluk tanımlarına, oluşan tüm ihtimalleri hesaplayarak karar veriyor. Hangi boşluk tanımını en çok kutuyu konteyner içine koyabiliyor ise o boşluk tanımına karar veriliyor. Uygulamanın yapmış olduğu bu hesaplamalar, konteyner boyutu $2 \times 2 \times 2$ gibi olan problemlerden bir seviye büyük olan konteyner problemleri için oldukça büyüktür. Örnek olarak, Şekil.21'deki gibi $3 \times 3 \times 3$ boyutlarında olan konteynere aşağıdaki kutuların yerleştirildiği bir problem üzerinden konuşacak olursak,

Kutu tipi A: 1x1x1, kutu adet:24

Kutu tipi B: 1x1x3, kutu adet:1

Kutu tipi C: 1x1x4, kutu adet:1

Kutu tipi D: 3x3x4, kutu adet: 1

Problemin çözümünde ana metotların çalıştırılma sayıları aşağıdaki gibidir:

SortBoxes metodu 1 kez çağrılmıştır

AnyBoxNotProcessed metodu 54 kez çağrılmıştır

AnyBoxNotChecked metodu 1042442 kez çağrılmıştır

CreateNewContainer metodu 1 kez çağrılmıştır

SkipBoxesWithGreaterVolume metodu 1 kez çağrılmıştır

SkipBoxesWithGreaterDimension metodu 1 kez çağrılmıştır

AddBoxIntoContainerReally metodu 27 kez çağrılmıştır

GetBestContainer metodu 806316 kez çağrılmıştır

GetBiggestBox metodu 806327 kez çağrılmıştır

MakeCombinations metodu 806327 kez çağrılmıştır

AddBoxIntoContainerVirtually metodu 806300 kez çağrılmıştır.

Bu rakamlardan da anlaşılacağı üzere, yukarıda bahsedilen hesaplamalar içinde çok fazla olasılık olduğundan yapılan hesaplamaları içeren metotlar çok fazla sayıda özynameli fonksiyonlar dahilinde (recursive functions) tekrar tekrar çağrılır.

Bu tez çalışmasında yer alan sezgisel algoritmada, temel yaklaşım olarak ifade edilen boşluk tanımı aslında literatürde aynı türde tanımlanan problemin çözümü ile alakalı bir çok yaklaşım ile oldukça benzerdir.

3-boyutlu konteyner yükleme probleminin çözümü için literatürde en çok kullanılan yaklaşım [2]The Wall Building'dir. (Duvar İnşası). Bu algoritmanın geliştiricileri George ve Robinson'dur. (1980) Bu yaklaşımda ilk yapılan işlem, konteyneri derinlik doğrultusunda katlara(layer) ayırmaktır. Bir diğer ifade ile bu işlem sayesinde birimlerden katlar oluşturulur. Bu katman mimarisinden dolayı en çok kullanılan ve güncelleştirilen algoritmalardandır. Bu yaklaşım, kat derinliğinin belirlenmesi için sınıflandırma kuralını kabul etmiştir. Kuralın ilk adımı, yüklenmiş birimler içinden (x, y, z koordinatları üzerinde en kısa olanı) en büyük değere sahip

birim seçimidir. Seçilen birim döndürülür. Döndürme işlemindeki amaç, derinlik boyutunun en büyük olmasını sağlamaktır. Yani z koordinat düzlemi. Bu sayede kat derinliği tanımlanmış olur. Seçilip döndürülen birimin derinliği kat derinliğini ifade eder. Bu sayede zor olan aşama ilk olarak değerlendirilmiş olmaktadır. Kat derinliği tanımlandıktan sonra sınıflandırma kuralı yeniden devreye girerek katın yüklenmesi yatay olarak gerçekleşir. Bu kural uygulanırken en üst sınıf değerine sahip olan birimden en alt sınıf değerine sahip birime doğru seçim yapılır.

Bu aşamada önemli olan bir diğer ayrıntı, seçilen kat içindeki yüklemelerin mümkün olduğunca aynı tip birimlerce yapılmasıdır. Bu tipe ait birimin olmadığı durumda ise algoritma bir önceki kuralı devreye alabilir. Kısaca tüm birimlerin en kısa boyutları sıralanır, bu sıralamada en uzun boyutta olan birim tür olarak tanımlanır. Daha önce yüklenmiş tip birimi sayısı birden fazla ise algoritma iki farklı şekilde davranır. Birinci yol, daha önce yüklenmiş tiplerden yüklenmemiş olan birim sayısının çok olduğu tipi belirlemek. İkinci yol ise yukarıda tanımlanan sınıflandırma kuralı devreye girer. Sonrasında ise en yüksek sınıf değeri olan birim seçilir. Bu süreç işledikçe katlar yükselir. Boş kalan alanlar sistem tarafından belirlenir, birimler bu alanlara yüklemeye çalışılır.

Bu tez çalışması, George ve Robinson'un The Wall Building(Duvar İnşası) yaklaşımına oldukça yakın bir algoritma içerir. Bu tez çalışmasında da konteyner içindeki boşluk tanımları derinlemesine katlar şeklinde düşünülür. Algoritmanın temel prensiplerinden birisi, ilk olarak zemin katını doldurmak daha sonra ise zemin kat üzerinde katlar inşa etmektir. Bu durumu sağlamak için konteyner içinde oluşan boşluk tanımlarından ilk etapta y1 koordinatında en düşük değere sahip olan seçilmeye çalışılır. Yine The Wall Building algoritmasında olduğu gibi konteyner içine yerleştirilecek olan birimler (yükler, kutular) döndürülerek konteyner içine yerleştirildiği farz edilerek hesaplamalar yapılır, en verimli sonucu veren kombinasyon seçilir. Bir başka deyiş ile konteyner içine yerleşecek olan birimin konteyner içine hangi pozisyonda yerleşeceği bu hesaplamalar sonucunda bulunmuş olur. The Wall Building algoritması ile bir benzer yaklaşım ise her zaman hacim olarak en büyük birimden başlayarak zor olan aşama ilk olarak değerlendirilir. The Wall Building ile bu tez çalışmasının yaklaşım konusunda farklı olan tarafları ise tez çalışmasında herhangi bir birim sınıflandırma mantığı içermemesidir. Ayrıca seçilen

katlarda mümkün olduğunca aynı tip birimlerin yüklenmesi gibi bir yaklaşım bu tez çalışmasındaki algoritmada yoktur.

1982 yılında Dowsland ve Bischoff, yukarıda anlatılan Wall Building yaklaşımını inceleyerek bir başka algoritma üzerinde çalışmışlardır. Algoritma Temel olarak incelendiğinde Wall Building ile benzerdir. Fakat iki önemli değişiklik ile Wall Building algoritmasından ayrılır. [3,4,5,6]. Algoritmanın Wall Building yaklaşımından ayrıldığı ilk nokta, oluşturulacak katların sadece tek birimden oluşuyor olmasıdır. İkinci farklılık ise katların yükselmesi iki boyutlu yükleme olarak ele alınmıştır. Sonuç olarak her kat oluşumunda kullanılacak olan birimlerin derinlik ölçüleri aynı olacaktır. Daha sonra ise iki boyutlu yükleme problemi çözülecektir. Bu tez çalışmasındaki yaklaşım Dowsland ve Bischoff'un yukarıda bahsedilen iki farklı yöntemden sebep farklıdır. Benzer olan yaklaşım ise konteyner içine yerleştirilen her yükten sonra geriye kalan hacmin ilk etapta en küçük olması hedef alınmasıdır. Ayrıca Bischoff'un önerdiği yaklaşımda önemli olan her katın derinliğinin seçimidir. Bir birim seçilir, bu birimin her boyuttaki uzunluğu kat derinliği olarak düşünülür. Derinlik seçiminden sonra iki boyutlu yükleme probleminin çözüm aşamasına geçilir. Bu yönüyle de bu tez çalışması Bischoff ile Dowsland'ın yaklaşımından farklıdır.

Bu tez çalışmasında kabullerden olan konteyner hacminin kaybını en aza indirgeyecek şekilde boşluk tanımı oluşturmak Chen, Shen'in geliştirdiği yöntemle benzerdir.[7]. Bu tez çalışmasında konteynere herhangi bir kutu yerleşiminden sonra oluşan boşluk tanımları ve boşluk tanımlarından en fazla kutuyu içeriye alabilecek olanın seçilmesi yönüyle Gomory ve Gilmore[8]'ün yaklaşımına benzerdir. Gomory ve Gilmore, konteyner yükleme probleminin kule yaklaşımını ön görerek çalışmışlardır. Bu yaklaşımda parçalar kule olarak inşa edilir, inşa edilen bu kuleler de konteynerin zeminine iki boyutlu kesme problemini çözerek yerleştirilir. Bu yaklaşımda farklı kule inşaları denenerek daha verimli çözümler elde edilmeye çalışılmıştır. Kule yapıları bu tez çalışmasının temelini oluşturan boşluk tanımları ile benzerdir. Yine aynı şekilde Eley'in [9] çalışmasında da kutu kümesi adında bir yapı tanımlanır. Temel hedef, parçalar oluşturarak konteyner yüklemektir. Bu kutu kümesi tanımı bu tez çalışmasındaki boşluk tanımı ve boşluk tanımlarının inşası yaklaşımına oldukça benzerdir. Resende ve Gonçalves [10], konteyner yükleme

probleminin çözümüne populasyon temelli rassal anahtarlı genetik algoritma ile yaklaşmışlardır. İki boyutlu ve üç boyutlu olarak tanımlı problemler için algoritma geliştirmişlerdir. Konteyner içine yerleştirilen bir parçadan sonra konteyner içinde kalan boş alanı en yüksek hacimsel verimlilikte yönetmek üzerine algoritmalarını geliştirmişlerdir. Bu tez çalışmasında problem çözümüne ilişkin yaklaşım Resende ve Gonçalves'in yöntemine kabaca benzerlik göstermektedir.

İşlem 3:Konteyner İçindeki Boşluk Tanımlamaları Okunur

Seçilen kutunun döndürülmesi gerekiyor mu? Sorgusuna geri dönülecek olursak, ilk olarak konteyner içindeki boşluk tanımları incelenir. Web uygulaması konteyner içindeki boşluk tanımları okur. Konteyner içinde hiçbir kutu olmadığından boşluk tanımı sayısı 1 tanedir. Bu tez çalışmasında problemin çözümü için temel prensiplerden bir diğeri de, boşluk tanımlarını seçerken, ilk etapta yüksekliği en az olanı seçmektir. (y1) Bu sayede kutu ya da kutular ilk etapta tabanı hacimsel olarak doldurmuş olacaktır. Taban dolduktan daha sonra ise yükselen katlar doldurulacaktır.Boşluk tanımına ait tüm değerler aşağıdaki gibidir:

Boşluk tanımı 1 için genişlik değeri: 2,

Boşluk tanımı 1 için yükseklik değeri: 2,

Boşluk tanımı 1 için derinlik değeri: 2,

Konteyner içine yerleştirilebilecek yük ağırlığı an itibarı ile 80 kg'dır.

Listenden seçilen 1 numaralı kutu için genişlik değeri, 2 birim

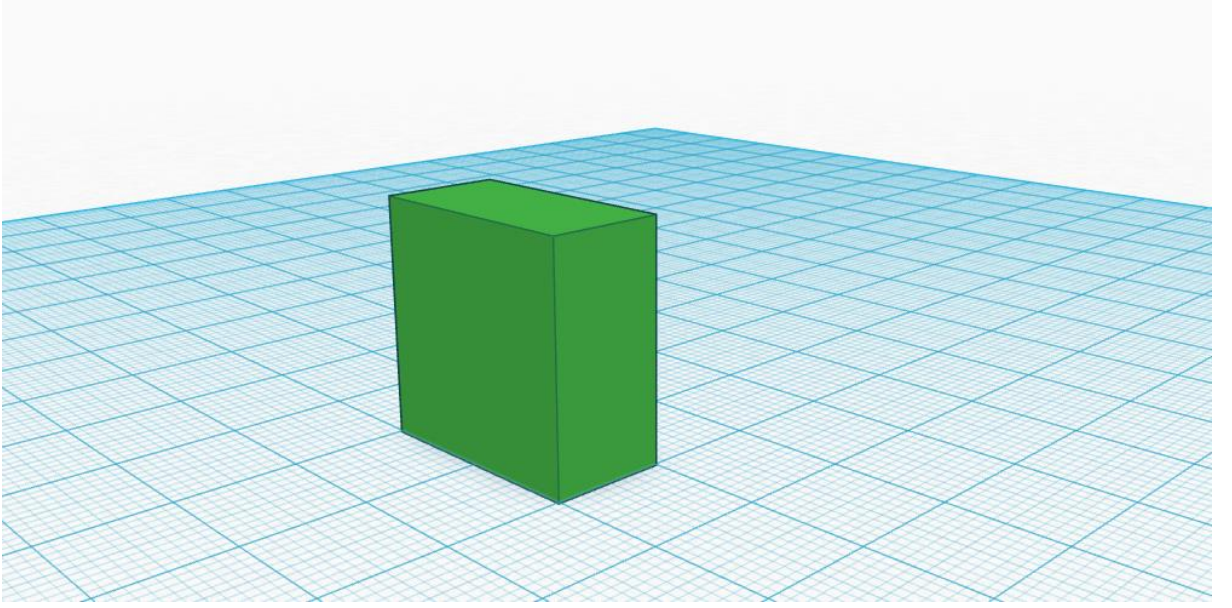
Listenden seçilen 1 numaralı kutu için yükseklik değeri, 2 birim

Listenden seçilen 1 numaralı kutu için derinlik değeri, 1 birim

Kutu ağırlığı: 40 kg.

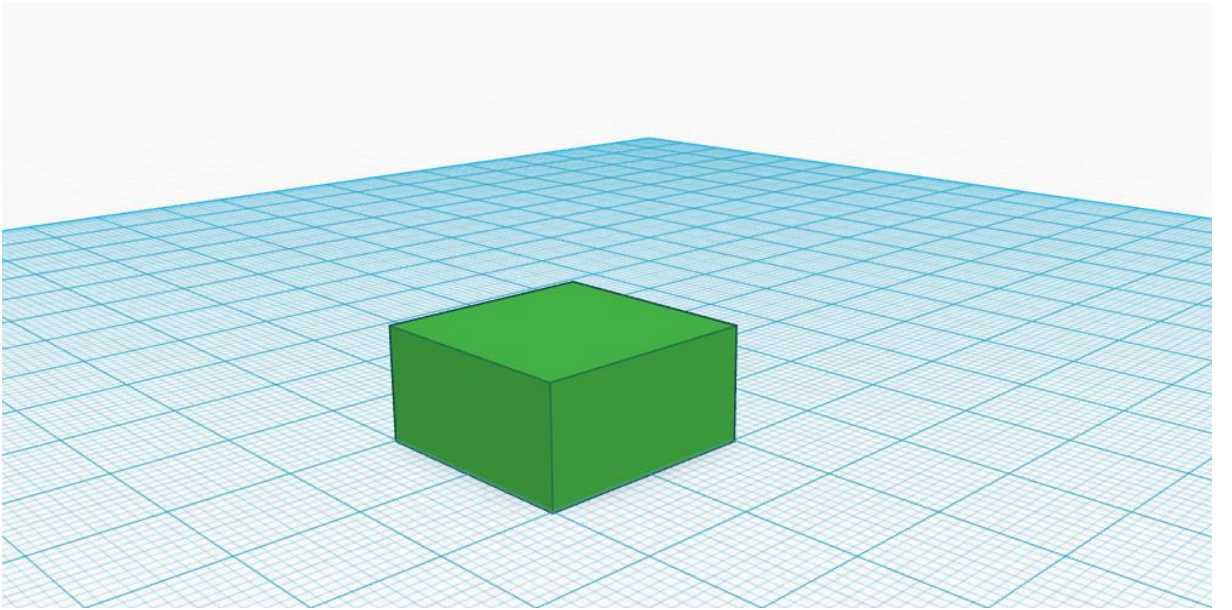
Kombinasyonlar (kutunun konteyner içinde alabileceği pozisyonlar)

Kombinasyon 1: 2x2x1 (genişlik, yükseklik, derinlik)



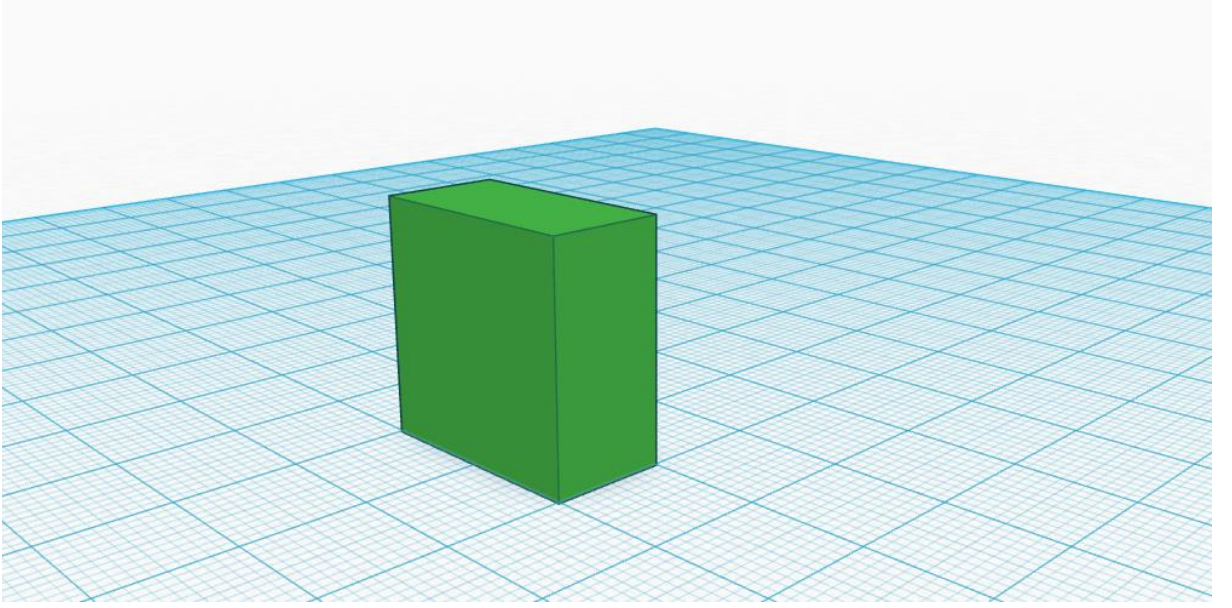
Şekil 25. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon

Kombinasyon 2: $2 \times 1 \times 2$ (genişlik, yükseklik, derinlik)



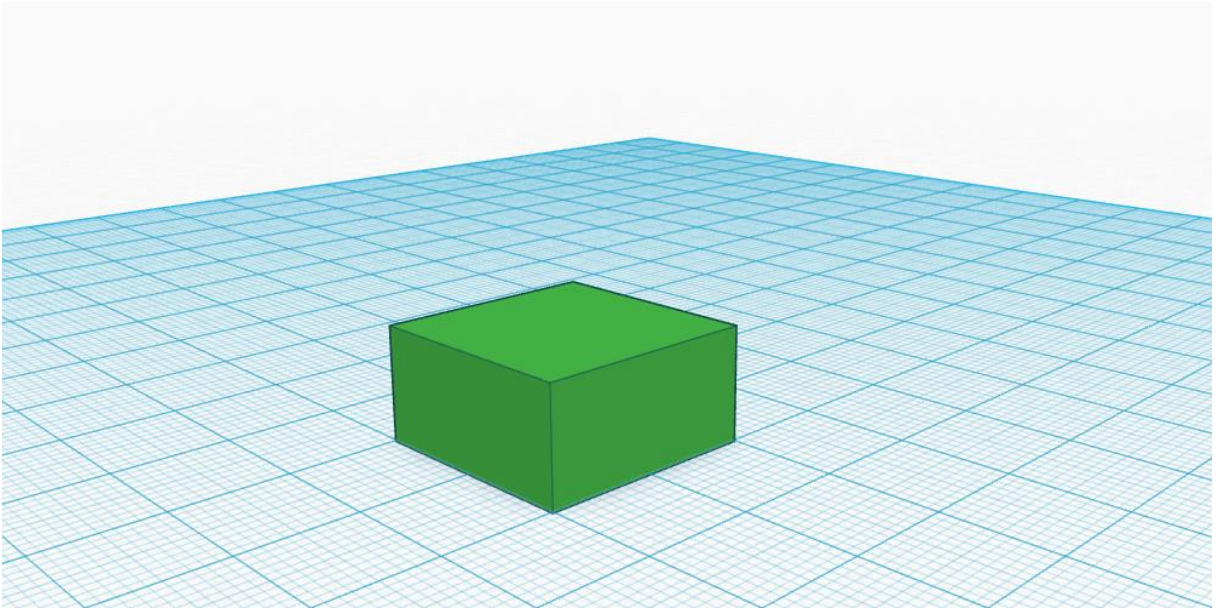
Şekil 26. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon

Kombinasyon 3: $2 \times 2 \times 1$ (genişlik, yükseklik, derinlik)



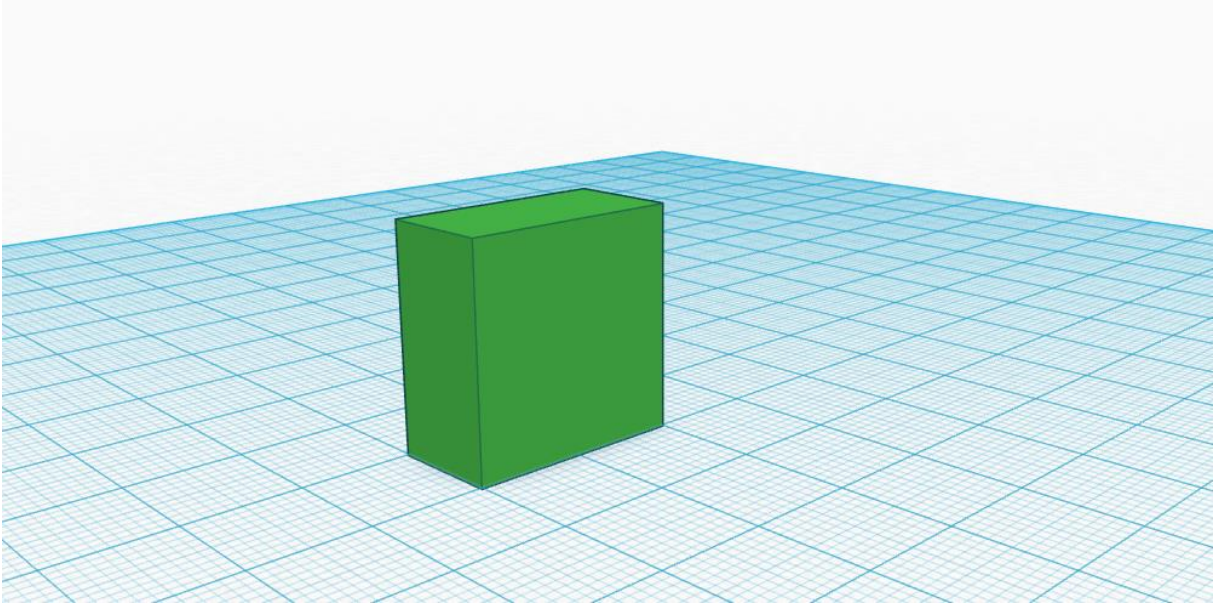
Şekil 27. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon

Kombinasyon 4: 2x1x2 (genişlik, yükseklik, derinlik)



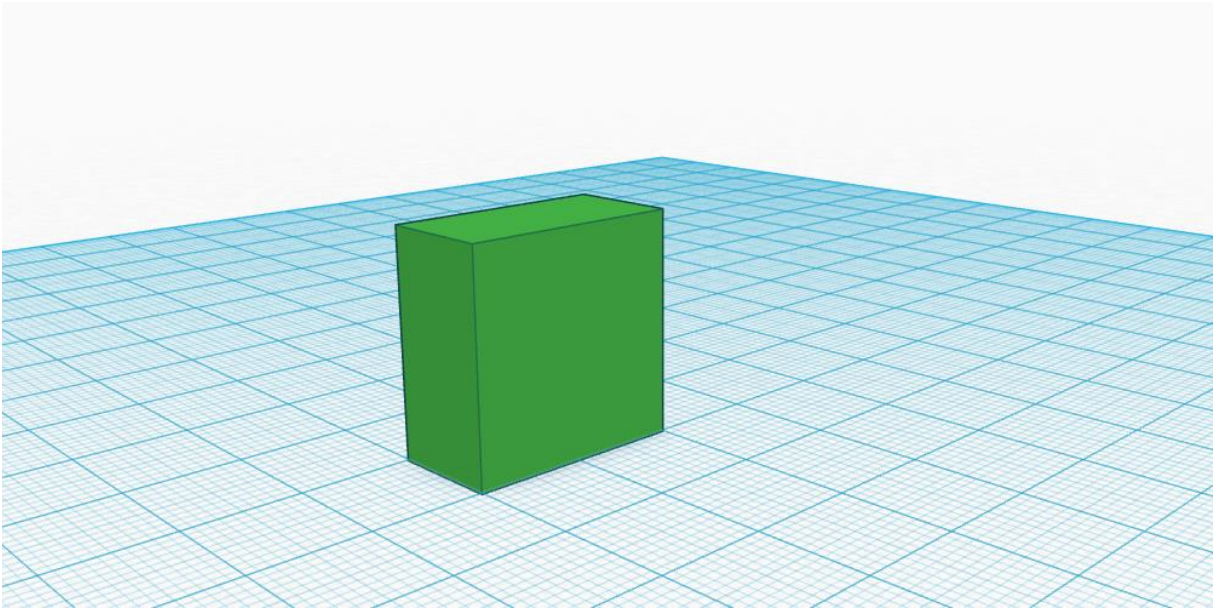
Şekil 28. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon

Kombinasyon 5: 1x2x2 (genişlik, yükseklik, derinlik)



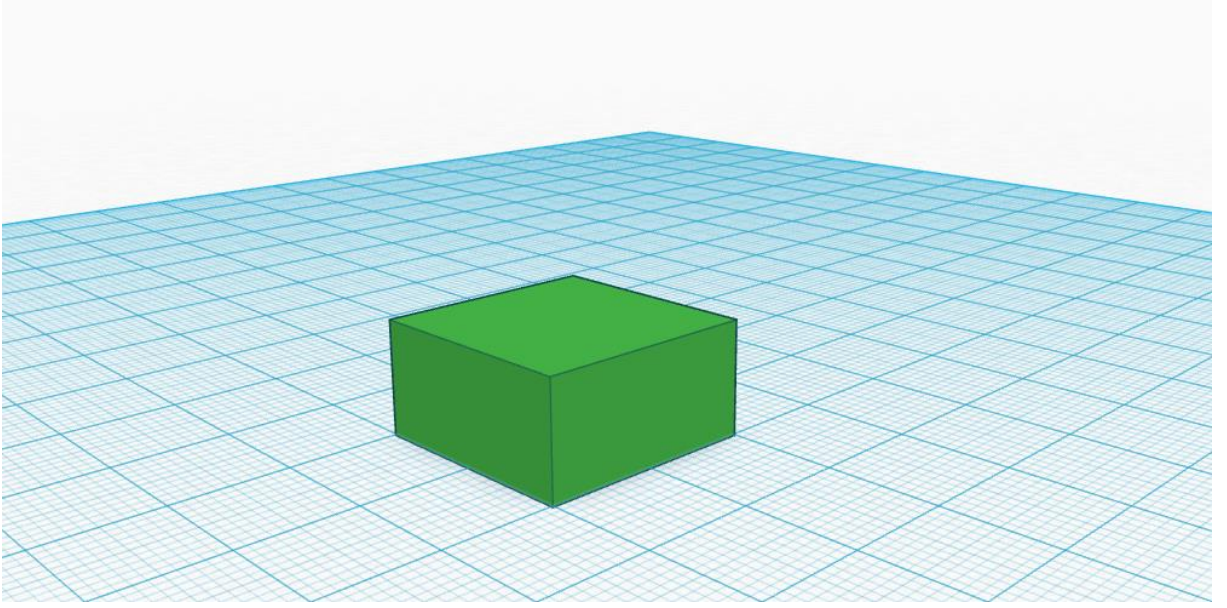
Şekil 29. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon

Kombinasyon 6: 1x2x2 (genişlik, yükseklik, derinlik)



Şekil 30. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon

İlk etapta seçmeye çalışacağımız kombinasyon, yükseklik değeri en düşük olan kombinasyondur. Dolayısı ile bu kombinasyon: 2x1x2'dir. (genişlik, yükseklik, derinlik)



Şekil 31. A Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren Kombinasyon Seçiliyor

Kombinasyon genişliği: 2 birim,

Kombinasyon yüksekliği: 1 birim,

Kombinasyon derinliği: 2 birim,

Kutu değerleri tekrar okunuyor;

Listenden seçilen 1 numaralı kutu için genişlik değeri, 2 birim

Listenden seçilen 1 numaralı kutu için yükseklik değeri, 2 birim

Listenden seçilen 1 numaralı kutu için derinlik değeri, 1 birim

Değerler karşılaştırıldığında kutunun çevrilmesi gerektiği anlaşılır. Kutuyu ifade eden yeni ölçü değerleri :2x1x2 (genişlik, yükseklik, derinlik)

İşlem 4:Seçilen Kutu Konteyner İçine Yerleştiriliyor

Konteyner içine yerleştirilecek kutu, A tipinde tanımlı 1 numaralı kutudur.

An itibarı ile konteyner içindeki boşluk tanımları listelenir.

Bir önceki adımda da belirtildiği gibi an itibarı ile konteyner içindeki boşluk tanım sayısı 1 adettir. Boşluk tanımına ait tüm değerler:

Genişlik değeri: 2 birim,

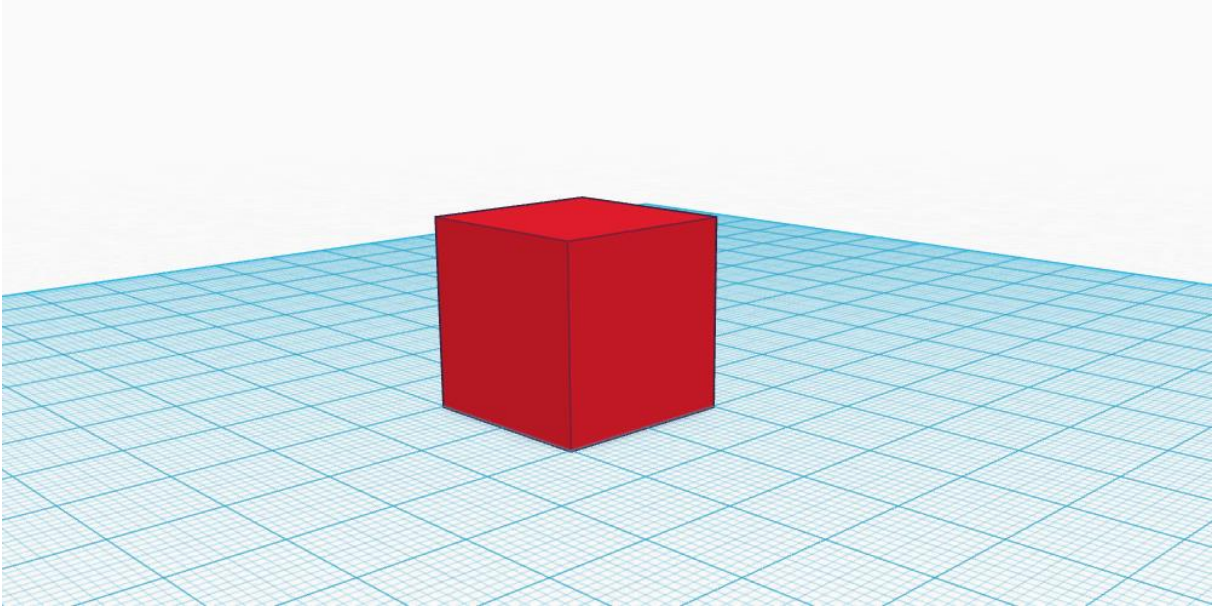
Yükseklik değeri: 2 birim,

Derinlik değeri: 2 birim,

Boşluk tanımını konteyner içinde belirleyen x, y, z koordinat düzlemine ait noktalar.

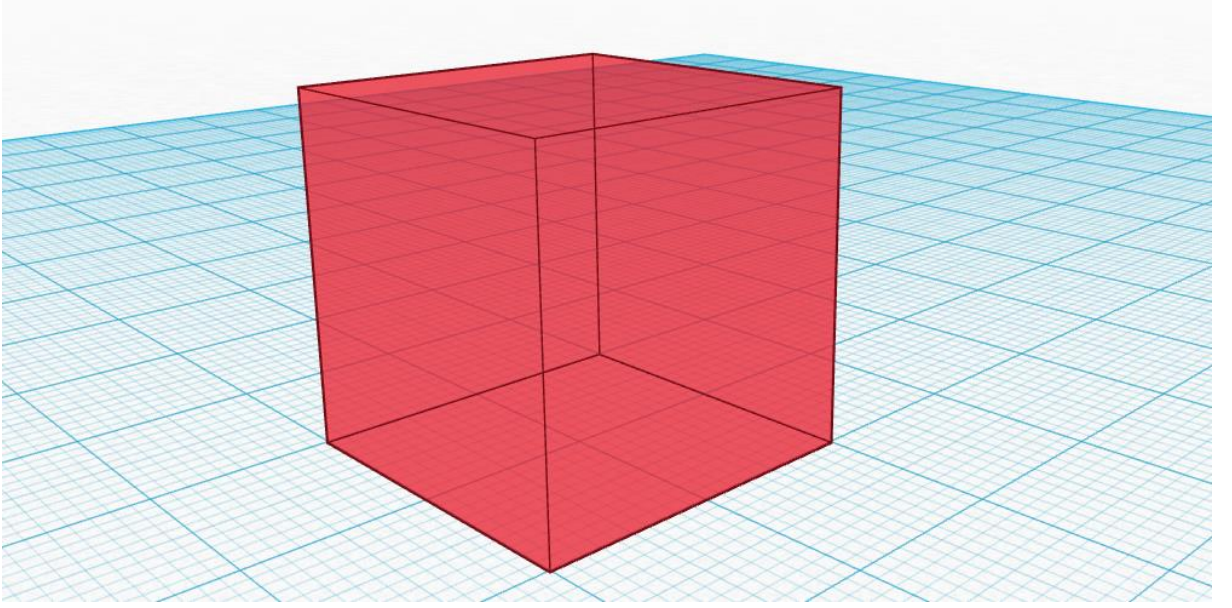
$(0, 0, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)$

Aşağıdaki şekilde, problemde tanımlanan, $2 \times 2 \times 2$ boyutlarındaki konteyner gösterilmektedir:



Şekil 32. Problemde Tanımlanan Konteynerin Gösterimi

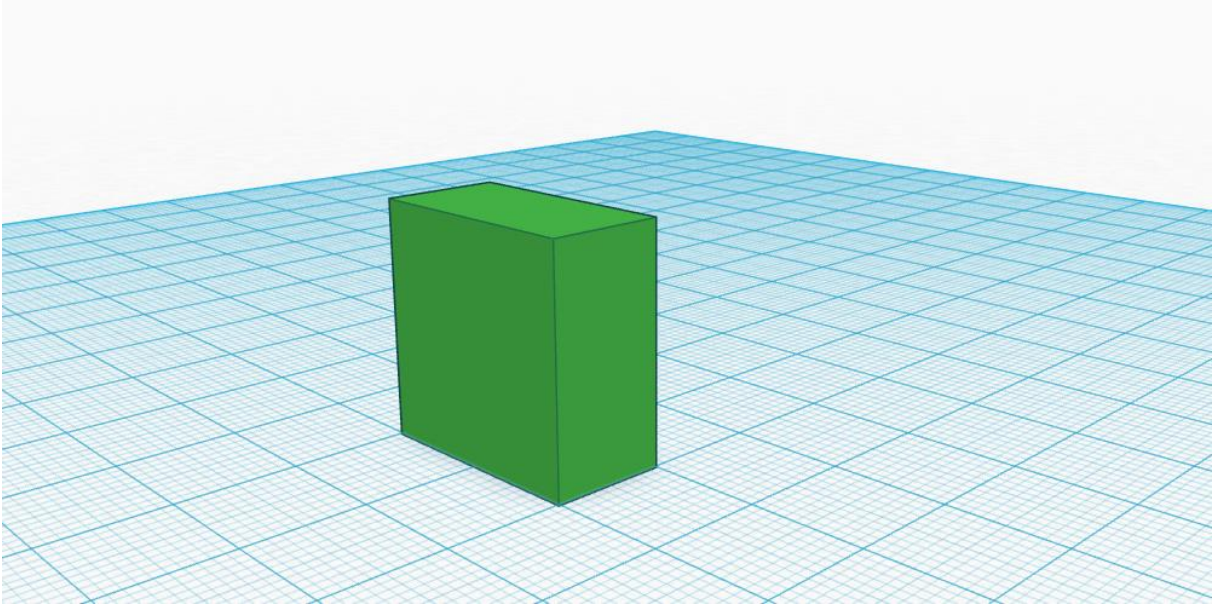
Aşağıdaki şekilde, $2 \times 2 \times 2$ ebatlarındaki konteyner içinde herhangi bir kutu yok iken konteyner içinde tanımlanan boşluk tanımını gösterilmiştir:



Şekil 33. Problemden Tanımlı Konteynerin İçinde Oluşan Boşluk Tanımı (İçeride Herhangi Bir Kutu Yok İken)

Listeden seçilen 1 numaralı kutunun çevrilmeden önceki boyutları:

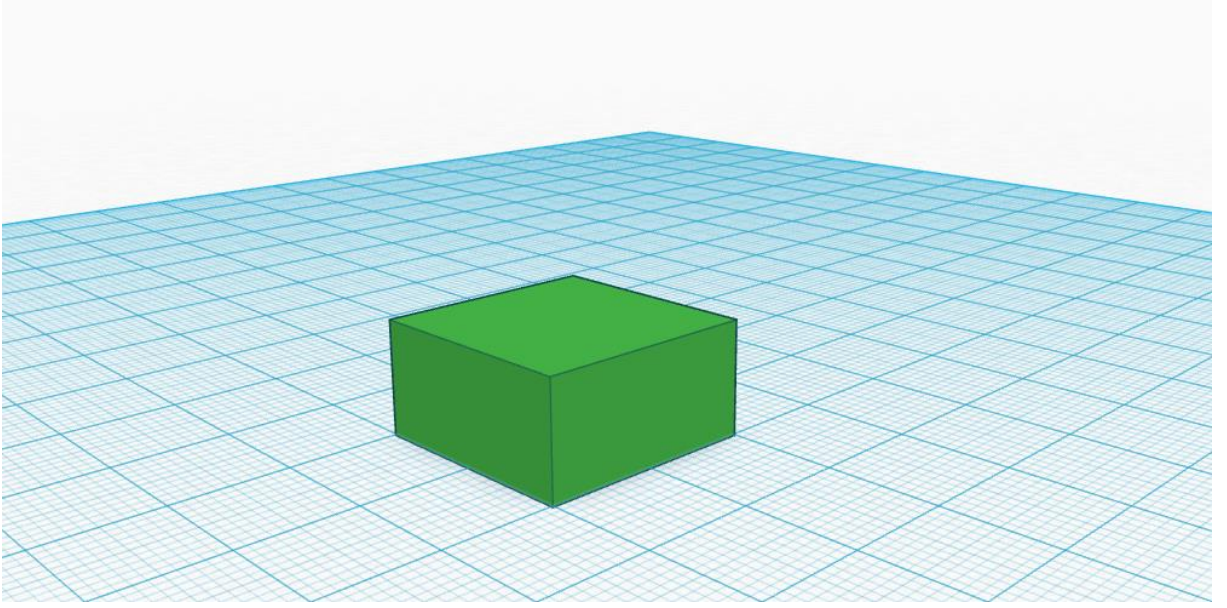
2x2x1 (genişlik, yükseklik, derinlik) . Kutu bu şekilde iken aşağıdaki gibi gözükmektedir:



Şekil 34. A Tipinde Tanımlı Kutunun Çevrilmeden Önceki Görüntüsü

Kutu çevrildikten sonra kutuya ait boyutlar ise 2x1x2 (genişlik, yükseklik, derinlik) olmuştur.

Kutu çevrildikten sonra aşağıdaki gibi gözükmektedir:



Şekil 35. A Tipinde Tanımlı Kutunun Konteyner İçine Hangi Pozisyonda Yerleştirileceğini Gösteren Şekil

Seçilen kutu ile boşluk tanımının ölçüleri kıyaslanır.

Boşluk tanımı 1'e ait genişlik değeri ile kutunun genişlik değeri bir birine eşittir. (2=2),

Boşluk tanımı 1'e ait genişlik değeri ile kutunun genişlik değeri bir birine eşit değildir. (2=1),

Boşluk tanımı 1'e ait derinlik değeri ile kutunun derinlik değeri bir birine eşittir. (2=2).

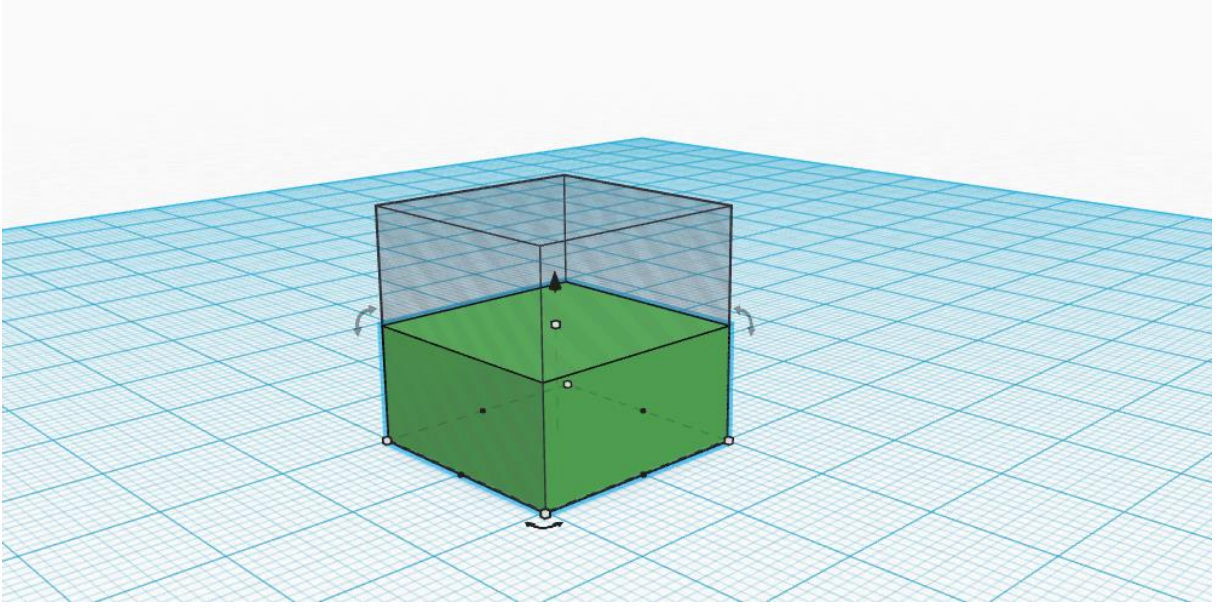
Kıyaslama sonucunda tek düzlemde fark olduğu ortaya çıkar. (yükseklik)

Kutu aşağıdaki koordinatlara yerleştirilmiş sayılır:

(0, 0, 0), (2, 1, 2) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

İşlem adımı kayıt edilir. (1)

Bu adımdan sonra konteyner içine yerleştirilebilecek maksimum ağırlık : 80kg - 40kg =40kg'dır.



Şekil 36. A Tipinde Tanımlanan Kutunun Konteyner İçine Yerleştirilmesi

Kutunun yerleşiminden sonra konteyner içinde oluşabilecek yeni boşluk tanım ya da tanımları için hesaplamalar yapılır. Yukarıdaki analize göre (sadece yükseklik ölçütünün farklı olmasından dolayı) bu olasılık ihtimali 1 adettir. Çünkü kutu x düzleminde boşluk tanımını hacim olarak tamamen kaplamıştır. Aynı şekile yerleştirilen kutu z düzleminde boşluk tanımını tamamen kaplamıştır. Fakat yerleştirilen kutu y düzleminde boşluk tanımını hacimsel olarak tamamen kaplamamıştır, boşluk tanımının y düzleminde hacimsel olarak sadece yarısını kaplayabilmiştir. Bu durumdan dolayı seçilen bu kutu (2x1x2) konteyner (2x2x2) içine yerleştikten sonra konteyner içinde oluşan yeni boşluk tanımının sayısı 1 tanedir. Bu yorumları rakamlar ile ifade edecek olursak:

Olasılık 1 (Possibility 1)

Olasılık 1 dahilinde tanımlı boşluk tanımı 1

Kutu yüksekliği: 1,

Boşluk tanımı yüksekliği: 2

Sadece y düzleminde farklılık olduğu için:

(Problemin çözümüne dair bu açıklamaların daha anlaşılır olması için bu aşamadan itibaren boşluk tanımlarına ait değerlerinin hesaplanması web uygulamasının kodları ile ifade edilecektir)

New spaceLength = oldspacelength = 2

$$\text{New spaceHeight} = \text{oldspaceheight} - \text{boxheight} = 1$$

$$\text{New spacewidth} = \text{oldspacewidth} = 2$$

$$\text{New space x1} = \text{oldspace x1} = 0$$

$$\text{New space y1} = \text{oldspace y1} + \text{boxheight} = 1$$

$$\text{New space z1} = \text{oldspace z1} = 0$$

$$\text{New space x2} = \text{New space x1} + \text{New spacelength} = 2$$

$$\text{New space y2} = \text{New space y1} + \text{New spaceheight} = 2$$

$$\text{New space z2} = \text{New space z1} + \text{New spacewidth} = 2$$

Bu hesaplamalar sonucunda konteyner içinde oluşan yeni boşluk tanımını ifade eden değerler aşağıdaki gibidir:

Yeni boşluk tanımına ait genişlik değeri, 2 birim

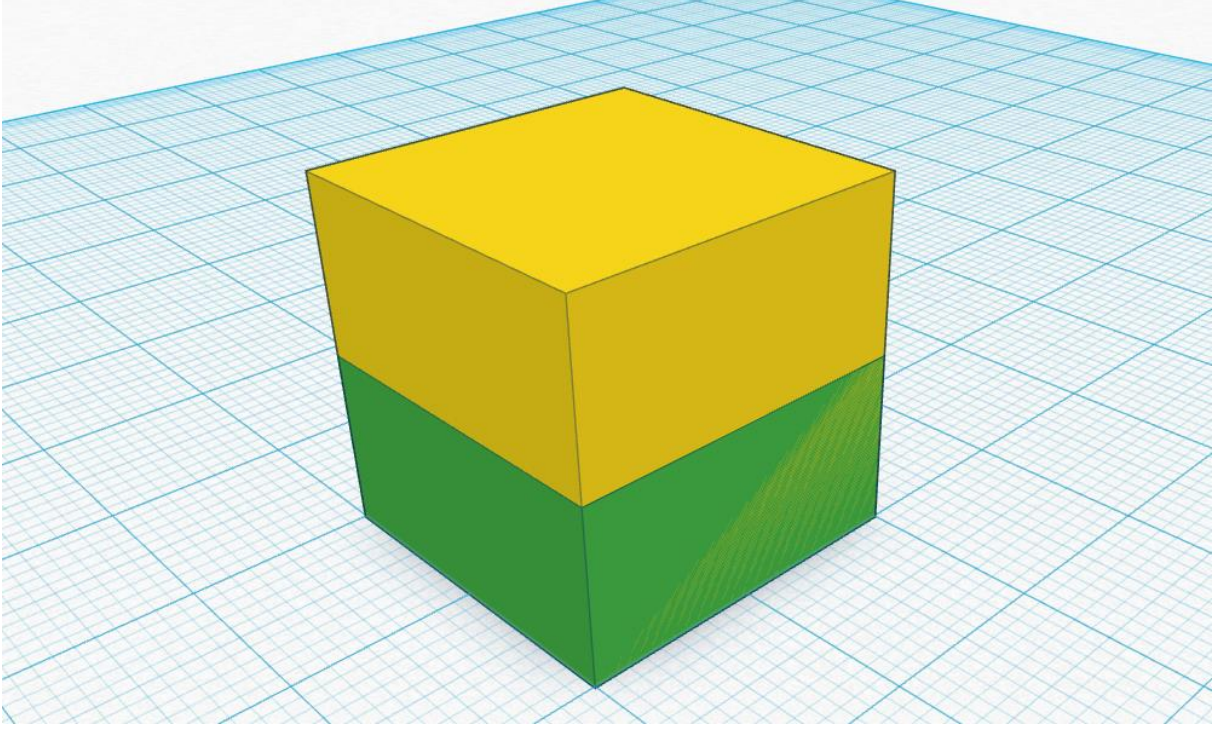
Yeni boşluk tanımına ait yükseklik değeri, 1 birim

Yeni boşluk tanımına ait derinlik değeri, 2 birim

Yeni boşluk tanımını x, y, z koordinat düzleminde ifade eden noktalar:

$$2 \times 1 \times 2 \text{ (0, 1, 0), (2, 2, 2) } (x_1, y_1, z_1), (x_2, y_2, z_2)$$

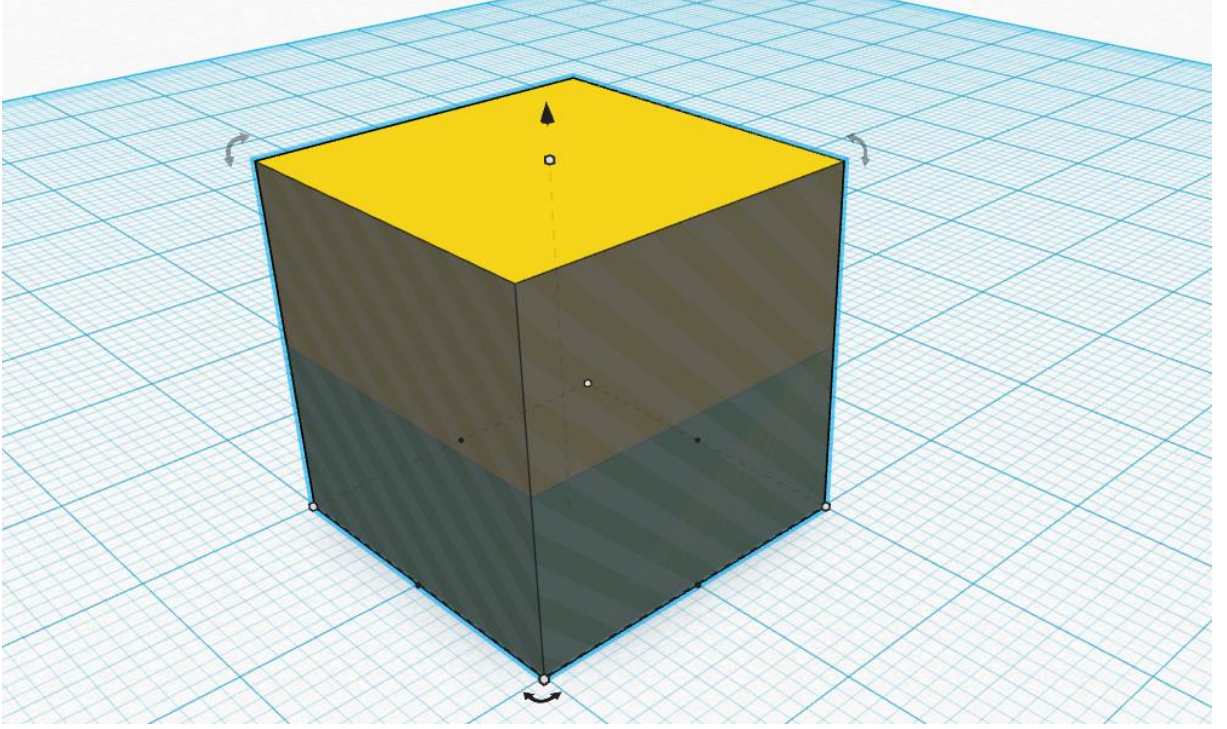
ilk kutunun konteyner içine yerleşiminden sonra oluşan yeni boşluk tanımını aşağıdaki şekilde gösterilmiştir. Yeni boşluk tanımını ifade eden alan sarı ile renklendirilmiştir.



**Şekil 37. A Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinden
Sonra Konteyner İçinde Oluşan Yeni Boşluk Tanımının Gösterimi**

Bu alanın konteyner içinde gösterimi de aşağıdaki şekilde gösterildiği gibidir:

$2 \times 1 \times 2, (0, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)$



**Şekil 38. A Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinden
Sonra Konteyner İçinde Oluşan Yeni Boşluk Tanımının Konteyner
Gölgelendirilemesi İle Gösterimi**

Boşluk tanımlarının birleştirilme işlemi başlar

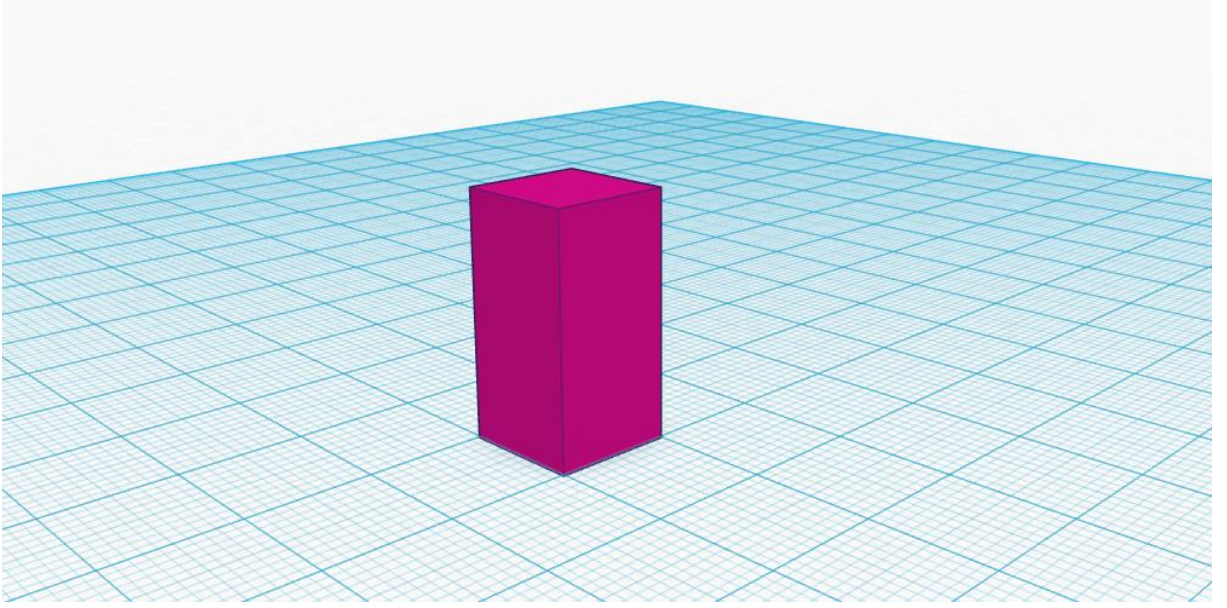
Web uygulaması konteyner içinde oluştan yeni boşluk tanımlarını sıradaki kutuyu yerleştirmeye geçmeden kontrol eder. Bu işlemin amacı boşluk tanımlarını birleştirmektir. Bu sayede konteyner içine yerleştirilecek kutu listesinden daha fazla kutuyu konteyner içine yerleştirebilir miyiz sorgusunu hesaplar. Daha fazla kutunun yerleşebileceği boşluk tanımı belirlenir. Geline noktada boşluk tanım sayısı 1 adet olduğu için bu işlem bu aşamada geçilir. Ve boşluk tanımı 1 adet olduğu için bu boşluk tanımı bir sonraki adım için konteyner içinde tanımlı boşluk tanımı ifade eder.

Boşluk tanımlarının birleştirilme işlemi sona erer

İşlem 5:Konteyner İçine Yerleştirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seçilir

Konteyner içine yerleştirilecek kutu listesine tekrar dönülür. Her kutu için aynı olan işlemler dizisine tekrar başlanacaktır. Listede sırada olan kutu seçilir. Bu kutu aşağıdaki şekilde gösterilen kutudur.

Kutu tipi B:



Şekil 39. B Tipinde Tanımlı Kutunun Gösterimi

Genişlik değeri: 1 birim (metre)

Yükseklik değeri: 2 birim (metre)

Derinlik değeri: 1 birim (metre)

Kutunun ağırlığı: 20 kg

Sırası: 2

İşlem 6:Seçilen Kutunun Döndürülmesi Gerekip Gerekmediği Sorgulanır

Öncelikle konteyner içindeki tüm boşluk tanımları listelenir. İlk aşamada boşluk tanımlarından yükseklik değeri en az olan seçilmeye çalışılacaktır. An itibarı ile konteyner içindeki boşluk tanım sayısı 1 tanedir. Bu boşluk tanımına ait tüm değerler;

Yeni boşluk tanımına ait genişlik değeri, 2 birim

Yeni boşluk tanımına ait yükseklik değeri, 1 birim

Yeni boşluk tanımına ait derinlik değeri, 2 birim

Yeni boşluk tanımını x, y, z koordinat düzleminde ifade eden noktalar:

(0, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Ayrıca an itibarı ile konteyner içine yüklenebilecek maksimum yük 40kg'dır.

Konteyner içine koyulabilecek maksimum ağırlık (80) - konteyner içindeki A tipinde tanımlı kutu(40) = 40 kg'dır.

Bu kutunun konteynere yüklenmesi ağırlık bakımından uygundur.

Boşluk tanımının değerleri ile yerleştirilecek kutu değerleri bir ileriki adımlarda karşılaştırılacaktır.

2 numaralı kutunun genişlik değeri: 1 birim,

2 numaralı kutunun yükseklik değeri: 2 birim,

2 numaralı kutunun derinlik değeri: 1 birim,

Kutu ağırlığı 20 kg.

Konteyner içindeki boşluk tanımı 1 adet olduğundan

Kutunun döndürülmesi ile alakalı tüm kombinasyonlar aşağıda resimlerle açıklanmıştır:

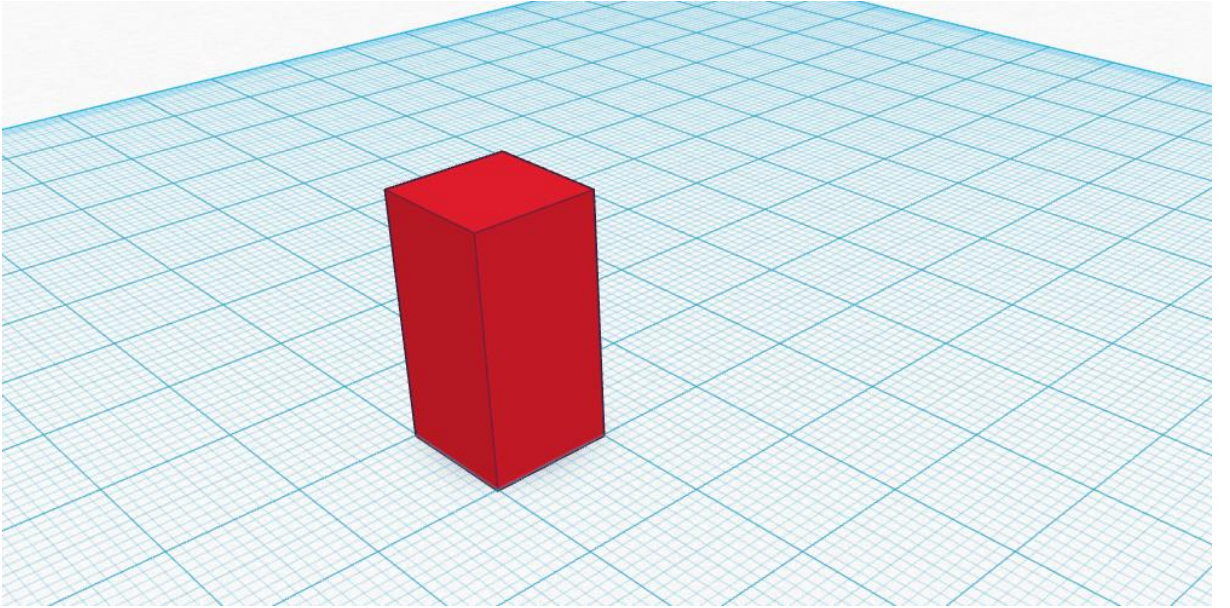
1 numaralı kombinasyon

Kutuya ait genişlik değeri \leq boşluk tanımına ait genişlik değeri

Kutuya ait yükseklik değeri \leq boşluk tanımına ait yükseklik değeri

Kutuya ait derinlik değeri \leq boşluk tanımına ait derinlik değeri

Bu durumda kutunun aldığı pozisyon aşağıdaki gibidir:



Şekil 40. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğinin Karar Aşaması

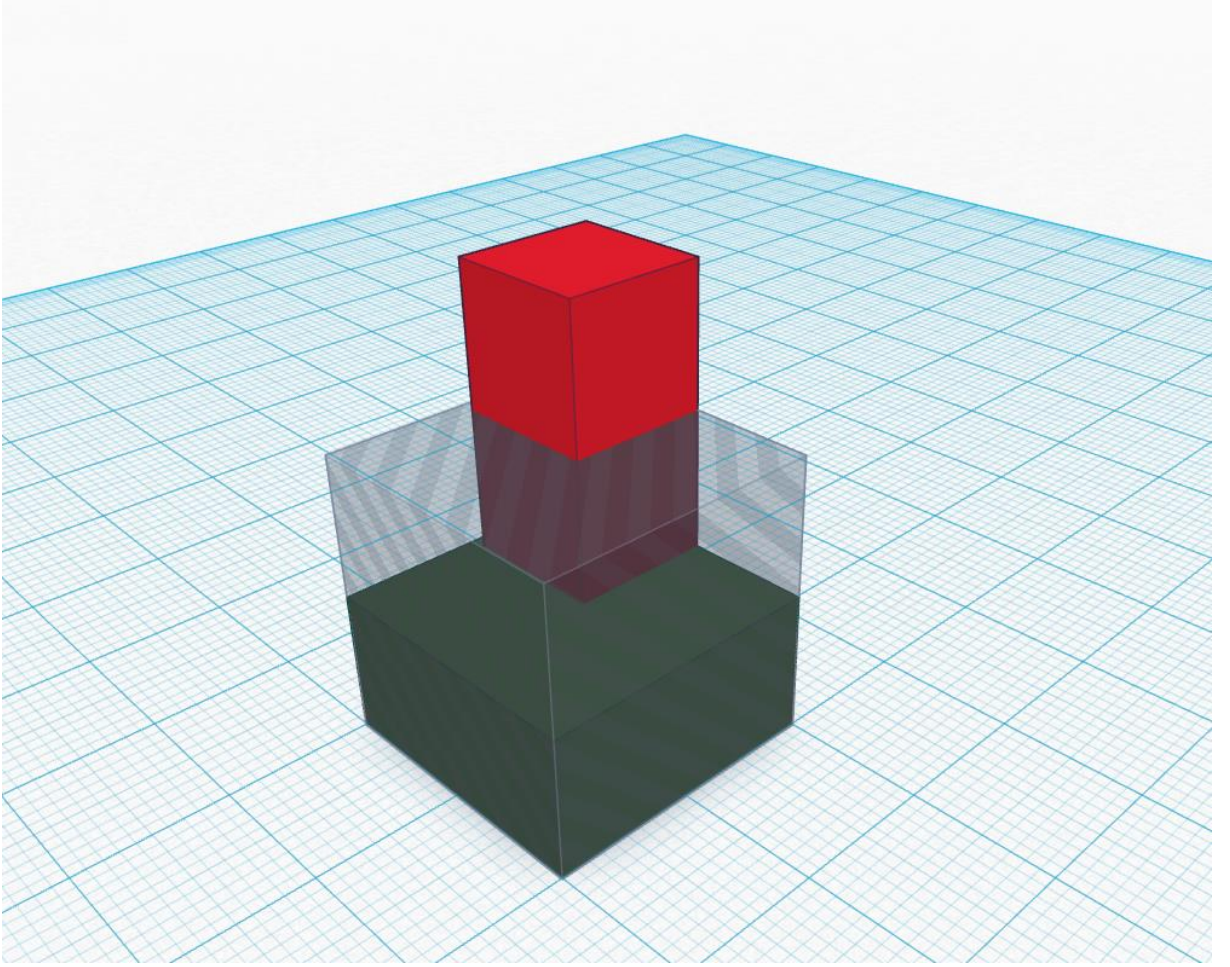
Kutu 1 numaralı kombinasyonda gösterildiği şekilde iken konteyner içinde olmasının ihtimali yoktur. Aşağıdaki hesaplar bu durumu ifade eder:

kutuya ait genişlik değeri (1) \leq boşluk tanımına ait genişlik değeri (2)

kutuya ait yükseklik değeri (2) $>$ boşluk tanımına ait yükseklik değeri (1)

kutuya ait derinlik değeri (1) \leq boşluk tanımına ait derinlik değeri (2)

Bu durum aşağıdaki resimde gösterilmiştir:



Şekil 41. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon

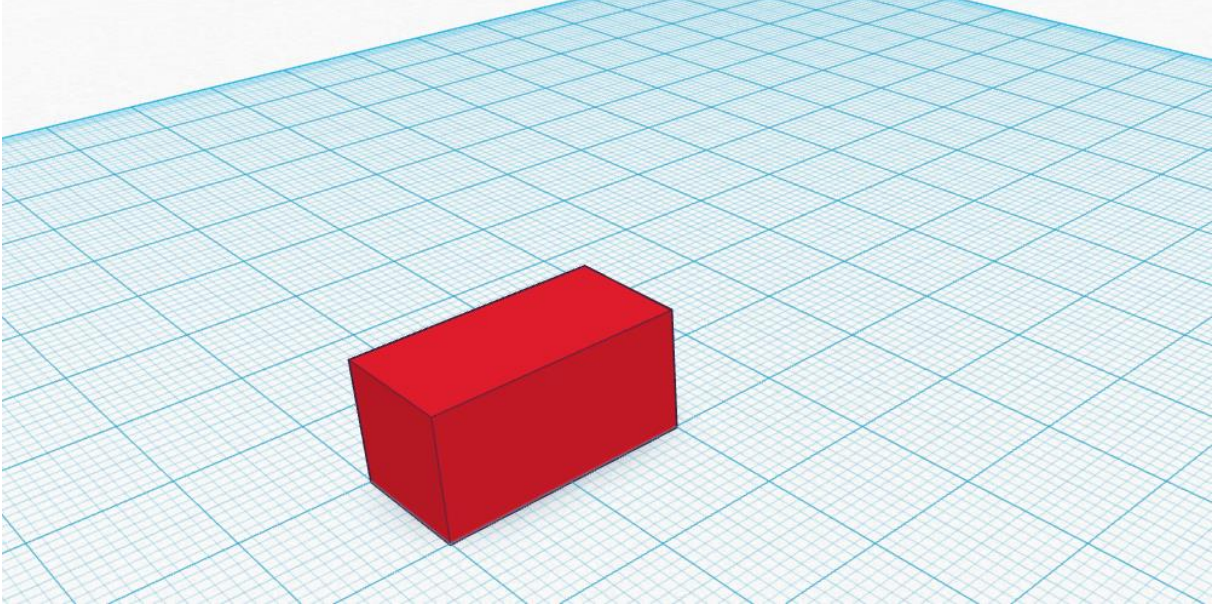
2 numaralı kombinasyon 1x1x2

kutuya ait genişlik değeri \leq boşluk tanımına ait genişlik değeri

kutuya ait derinlik değeri \leq boşluk tanımına ait yükseklik değeri

kutuya ait yükseklik değeri \leq boşluk tanımına ait derinlik değeri

bu durumda kutunun aldığı pozisyon aşağıdaki gibidir:



Şekil 42. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon

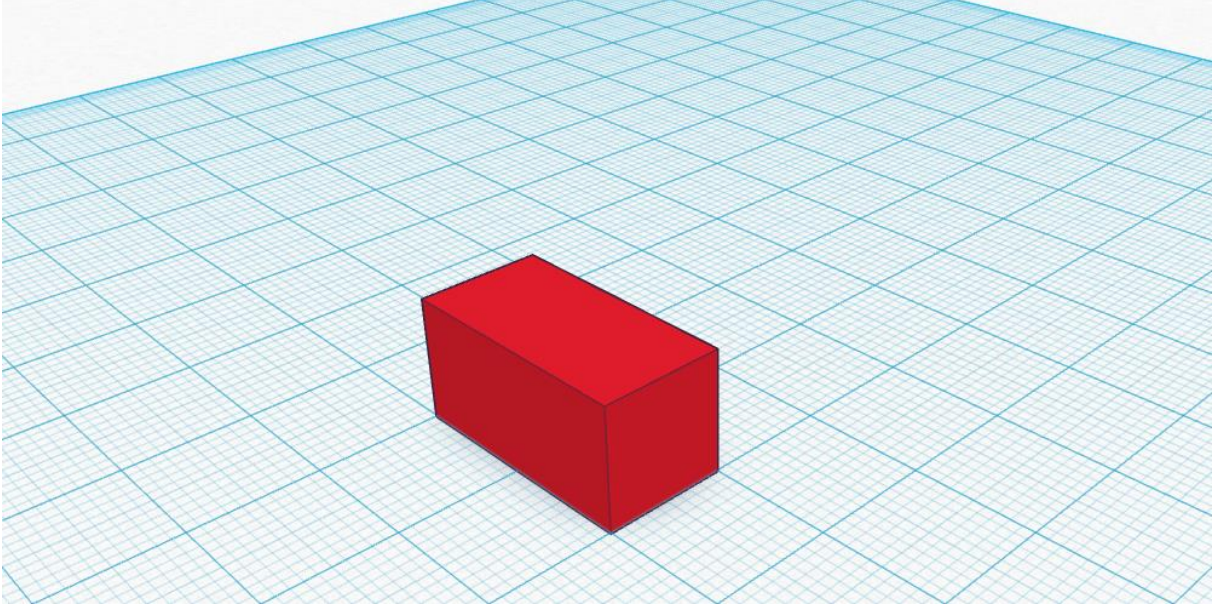
3 numaralı kombinasyon 2x1x1

kutuya ait yükseklik değeri \leq boşluk tanımına ait genişlik değeri

kutuya ait genişlik değeri \leq boşluk tanımına ait yükseklik değeri

kutuya ait derinlik değeri \leq boşluk tanımına ait derinlik değeri

bu durumda kutunun aldığı pozisyon aşağıdaki gibidir:



Şekil 43. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon

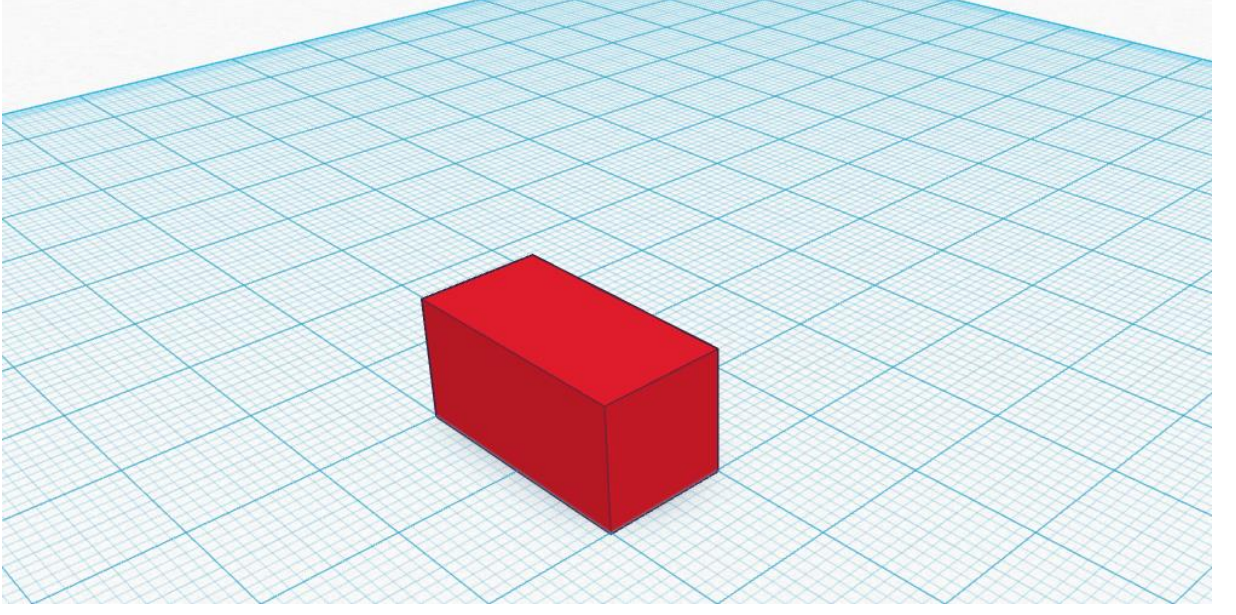
4 numaralı kombinasyon 2x1x1

kutuya ait yükseklik değeri \leq boşluk tanımına ait genişlik değeri

kutuya ait derinlik değeri \leq boşluk tanımına ait yükseklik değeri

kutuya ait genişlik değeri \leq boşluk tanımına ait derinlik değeri

bu durumda kutunun aldığı pozisyon aşağıdaki gibidir:



Şekil 44. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon

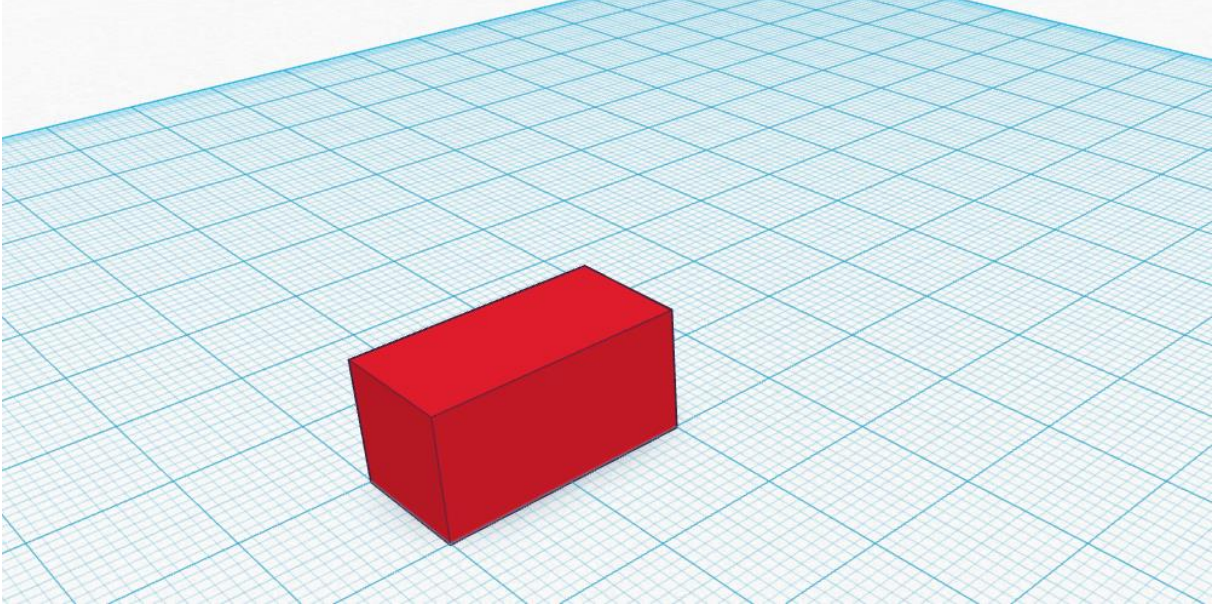
5 numaralı kombinasyon 1x1x2

kutuya ait derinlik değeri \leq boşluk tanımına ait genişlik değeri

kutuya ait genişlik değeri \leq boşluk tanımına ait yükseklik değeri

kutuya ait yükseklik değeri \leq boşluk tanımına ait derinlik değeri

bu durumda kutunun aldığı pozisyon aşağıdaki gibidir:



Şekil 45. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon

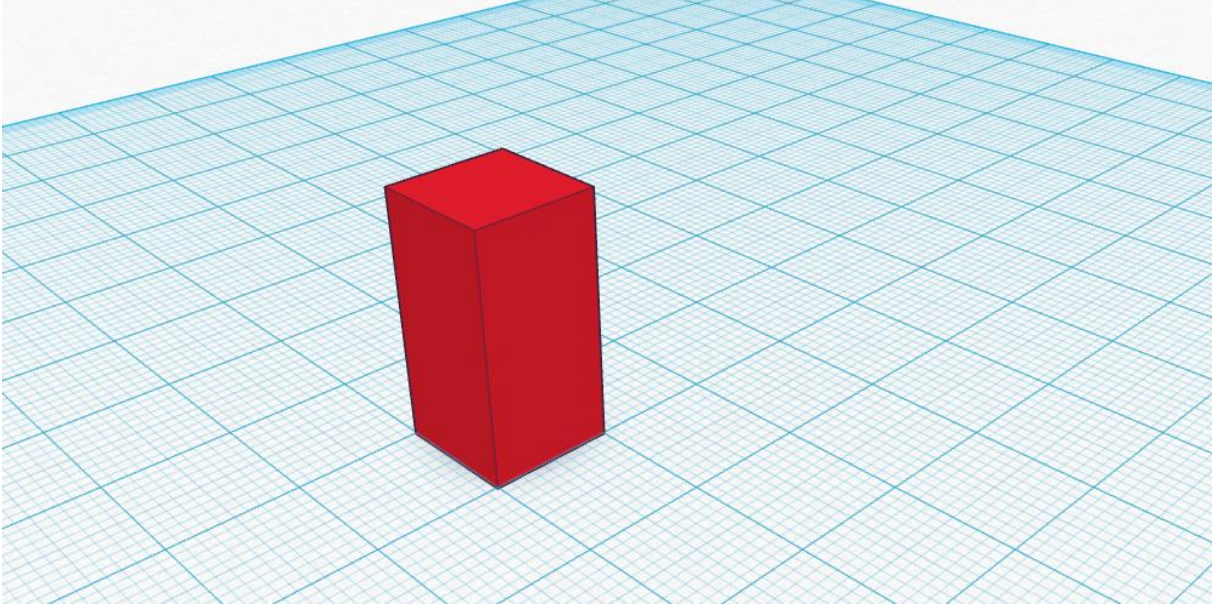
6 numaralı kombinasyon

kutuya ait derinlik değeri \leq boşluk tanımına ait genişlik değeri

kutuya ait yükseklik değeri \leq boşluk tanımına ait yükseklik değeri

kutuya ait genişlik değeri \leq boşluk tanımına ait derinlik değeri

bu durumda kutunun aldığı pozisyon aşağıdaki gibidir:



Şekil 46. B Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon

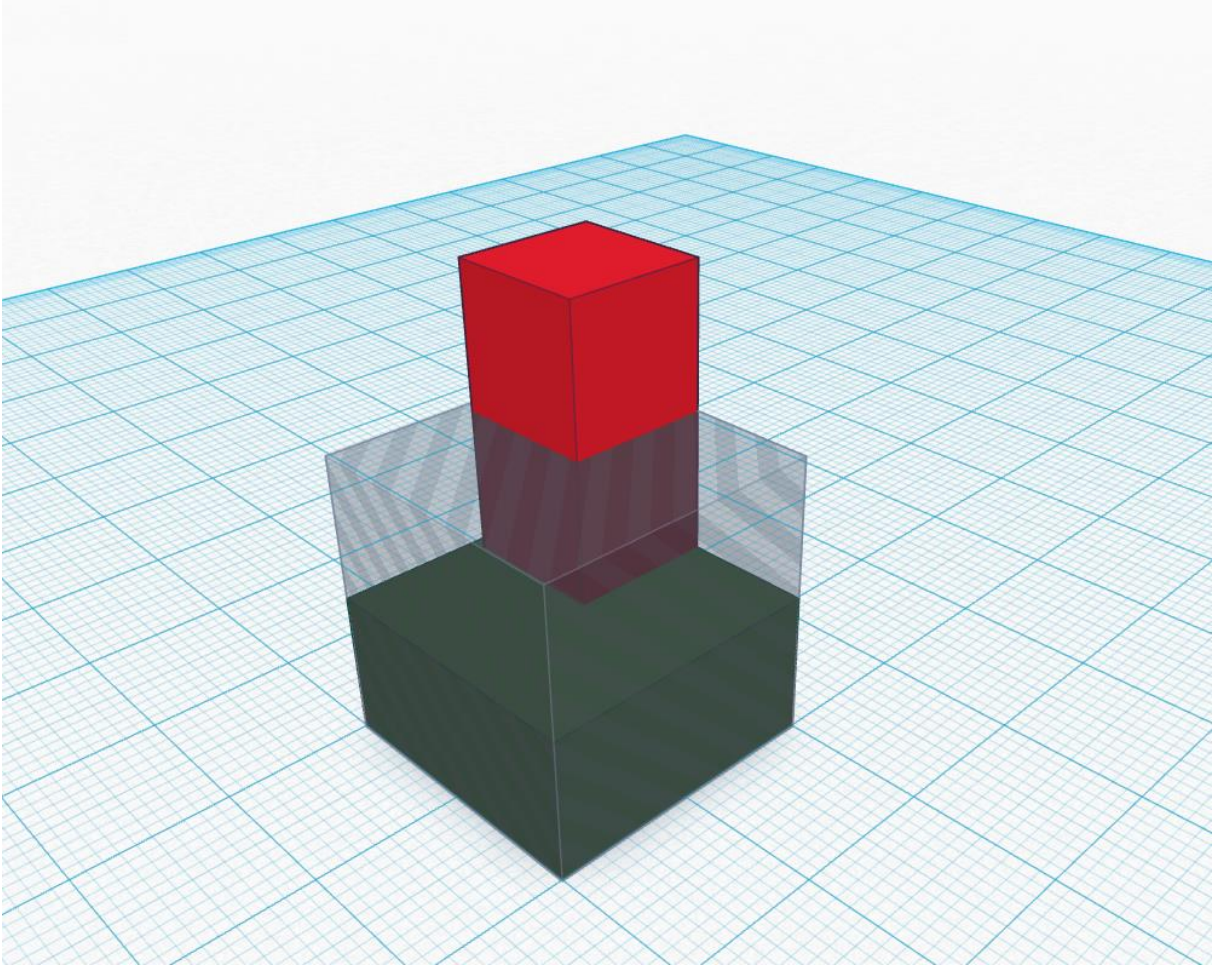
Kutu 6 numaralı kombinasyonda gösterildiği şekilde iken konteyner içinde olmasının ihtimali yoktur. Aşağıdaki hesaplar bu durumu ifade eder:

kutuya ait genişlik değeri (1) \leq boşluk tanımına ait genişlik değeri (2)

kutuya ait yükseklik değeri (2) $>$ boşluk tanımına ait yükseklik değeri (1)

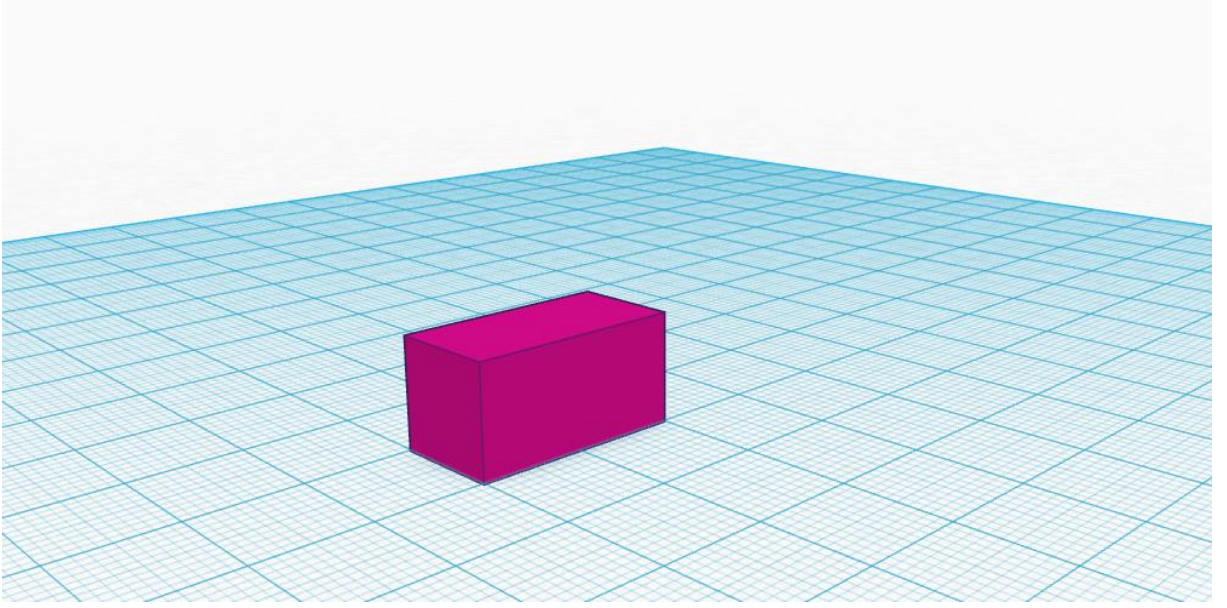
kutuya ait derinlik değeri (1) \leq boşluk tanımına ait derinlik değeri (2)

Bu durum aşağıdaki resimde gösterilmiştir:



**Şekil 47. B Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinin
Mümkün Olmadığı Kombinasyonun Gösterimi**

bu kombinasyonlardan yükseklik(y_1) değeri düşük olan kombinasyon seçilir. dolayısı kutu pozisyonu için seçilecek olan kombinasyon $1 \times 1 \times 2$ 'dir. Bu değer kutunun problem tanım aşamasında tanımlanan ölçüleri ile karşılaştırılırsa $(1 \times 2 \times 1)$ kutunun aşağıdaki şekilde çevrildiği gözlemlenebilir.



Şekil 48. B Tipinde Tanımlı Kutunun Konteyner İçine Ne Şekilde Yerleştirileceğine Karar Veriliyor

İşlem 7:Seçilen Kutu Konteyner İçine Yerleştiriliyor

Konteyner içindeki boşluk tanımları listelenir. Konteyner içindeki boşluk tanım sayısı 1 tanedir.

boşluk tanımı genişlik değeri: 2 birim

boşluk tanımı yükseklik değeri: 1 birim

boşluk tanımı derinlik değeri: 2 birim

bu boşluk tanımını x, y, z koordinat düzleminde ifade eden noktalar:

(0, 1, 0), (2, 2, 2) (x₁, y₁, z₁), (x₂, y₂, z₂)

kutu genişlik değeri ile boşluk tanımının genişlik değeri bir birine eşit değildir(1≠2)

kutu yükseklik değeri ile boşluk tanımının yükseklik değeri bir birine eşittir. (1=1)

kutu derinlik değeri ile boşluk tanımının derinlik değeri bir birine eşittir. (2=2).

Görüldüğü üzere farklılık tek düzlemedir. (genişlik ya da x eksen). Bu nedenle kutu bu noktaya yerleştirilir.

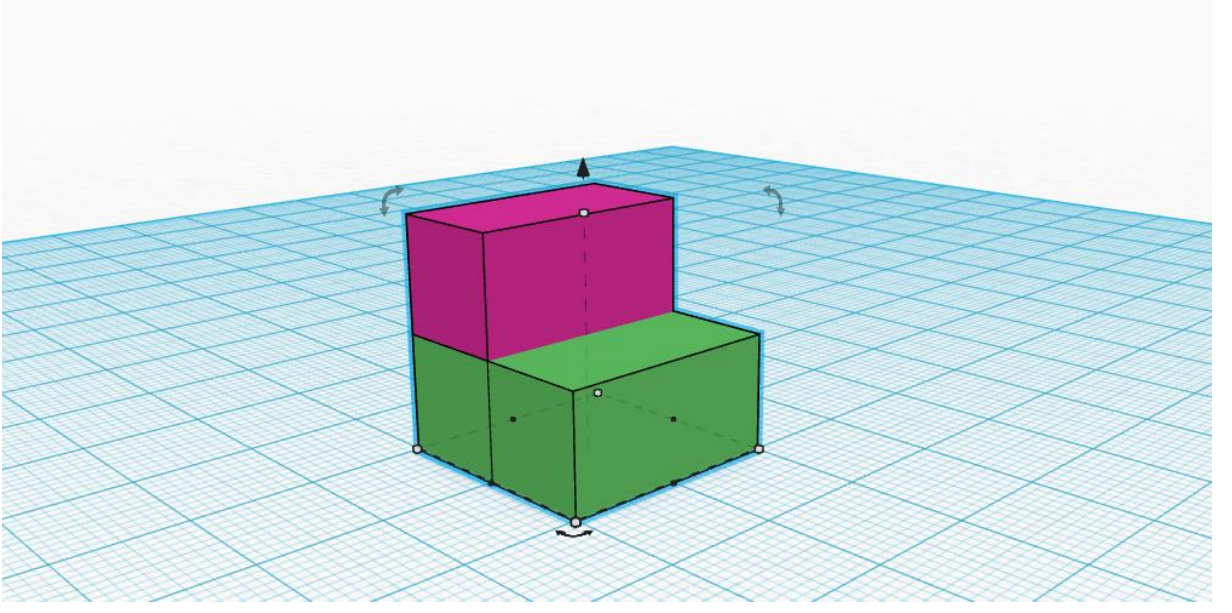
(0, 1, 0), (1, 2, 2) (x₁, y₁, z₁), (x₂, y₂, z₂).

İşlem adımı 2 olarak kayıt edilir.

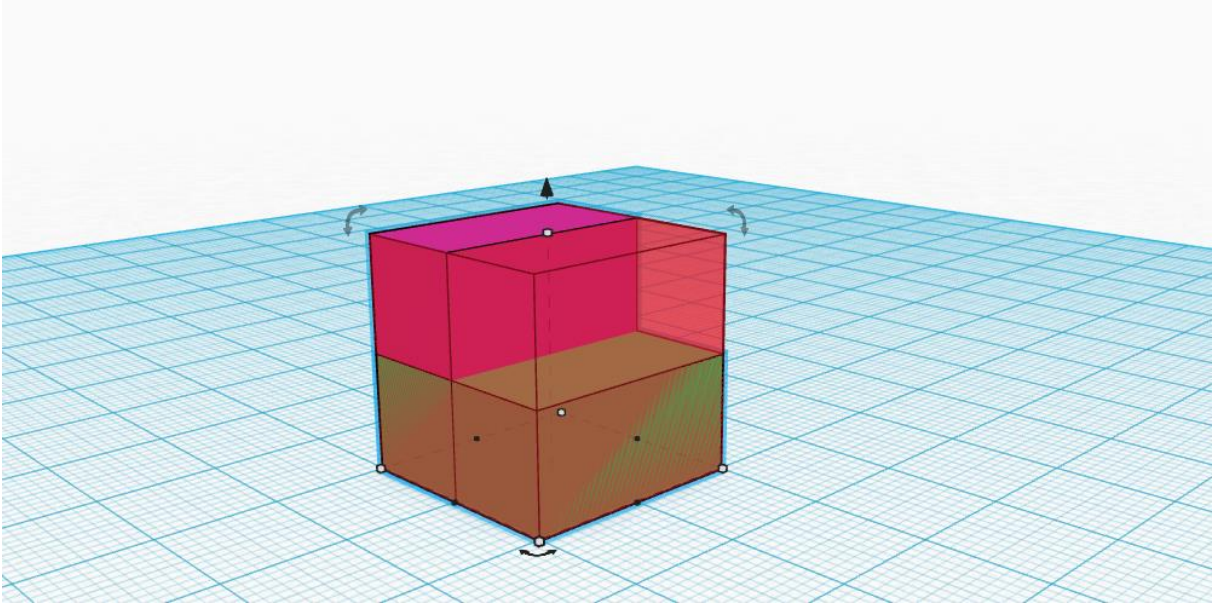
Bu adımdan sonra konteyner içine yerleştirilebilecek maksimum ağırlık : 20kg

(80-40-20=20KG)

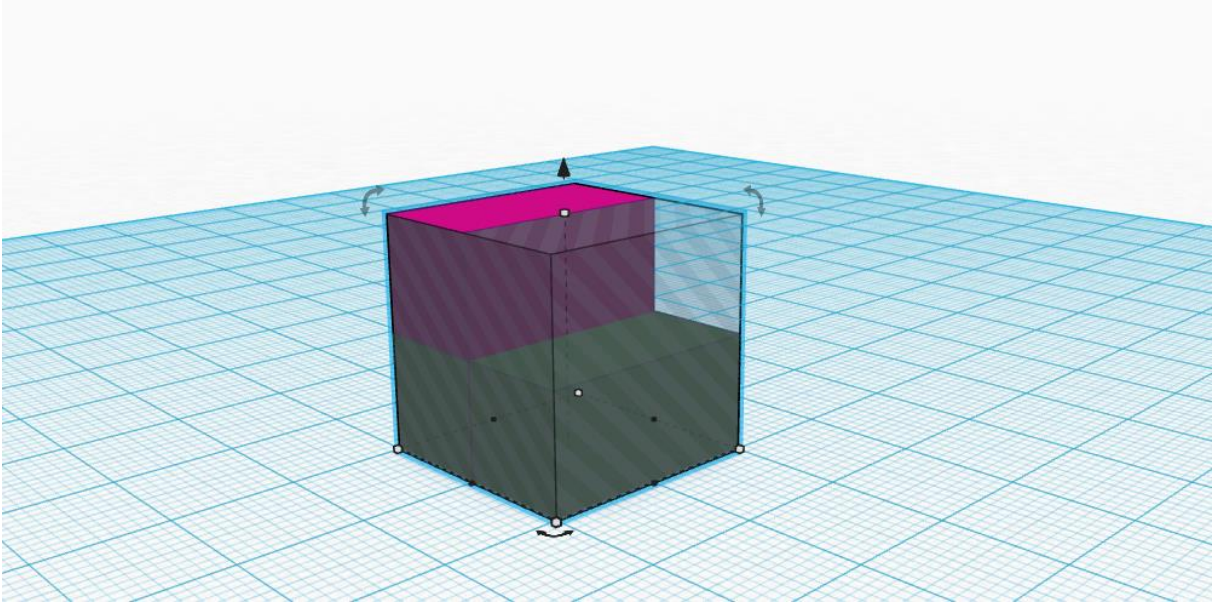
2 numaralı kutunun yerleşimi aşağıdaki şekillerde ayrıntılı olarak gösterilmiştir:



Şekil 49. B Tipinde Tanımlı Kutu, A Tipinde Kutunun Üzerine Yerleştiriliyor



Şekil 50. B Tipinde Tanımlı Kutunun Yerleştirilmesi Gösteriliyor



Şekil 51. B Tipinde Tanımlı Kutunun Konteyner İçine Yerleşimi Konteyner Gölgelemlenmesi İle Gösteriliyor

Yukarıdaki ölçüt karşılaştırmasının tek düzlemde farklı olmasından dolayı, oluşacak yeni boşluk tanımı 1 adettir. Başka bir açıklama ile kutu hacimsel olarak boşluk tanımını x düzleminde tamamen dolduramamış, y ve z düzleminlerinde tamamen doldurmuştur. Bu boşluk tanımının hesaplanması aşağıdaki gibidir:

Olasılık 1 (Possibility 1)

Olasılık 1 dahilinde tanımlı boşluk tanımı 1

Daha iyi anlaşılması adına bu değerleri tekrar yazacak olursak:

Kutu genişlik değeri ile boşluk tanımının genişlik değeri bir birine eşit değildir ($1 \neq 2$)

Kutu yükseklik değeri ile boşluk tanımının yükseklik değeri bir birine eşittir. ($1=1$)

Kutu derinlik değeri ile boşluk tanımının derinlik değeri bir birine eşittir. ($2=2$)

Ve yeni oluşan boşluk tanımının hesabı:

$$\text{New spaceLength} = \text{oldspaceLength} - \text{boxlength} = 1$$

$$\text{New spaceHeight} = \text{oldspaceheight} = 1$$

$$\text{New spaceWidth} = \text{oldspaceWidth} = 2$$

$$\text{New space x1} = \text{oldspace x1} + \text{boxlength} = 1$$

$$\text{New space y1} = \text{oldspace y1} = 1$$

$$\text{New space z1} = \text{oldspace z1} = 0$$

New space x2 = New space x1 + New spacelength = 2

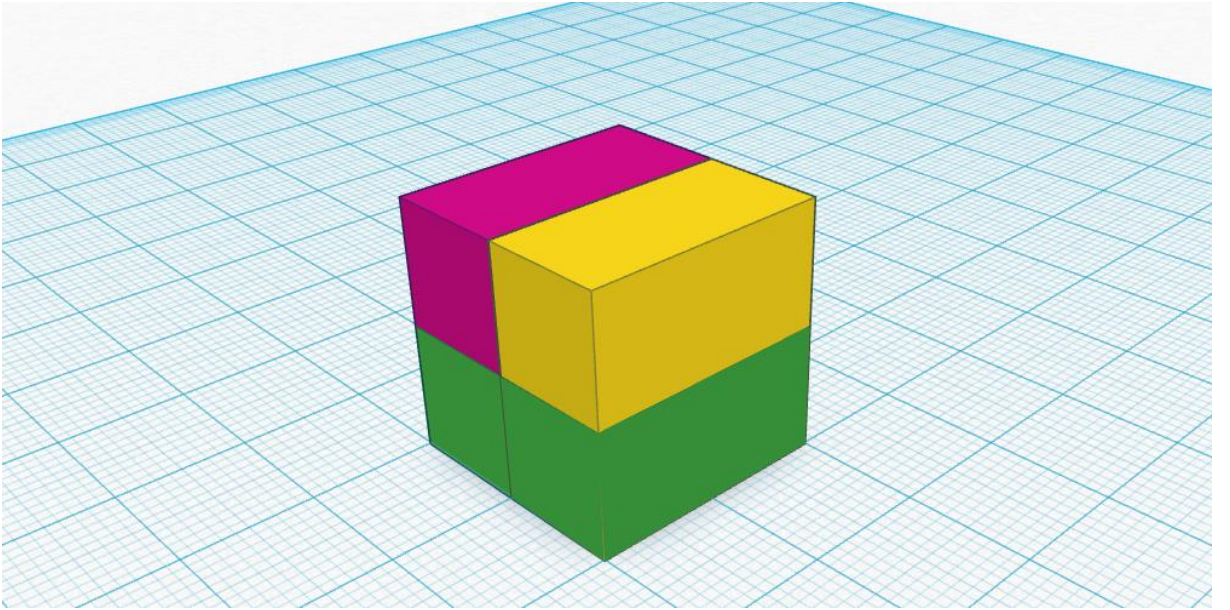
New space y2 = New space y1 + New spaceheight = 2

New space z2 = New space z1 + New spacewidth = 2

Hesaplamalar sonucunda B tipinde 2 numaralı kutu konteyner içine yerleştirildikten sonra konteyner içinde oluşan 1 adet boşluk tanımına ait tüm değerler aşağıdaki gibidir:

1x1x2, (1, 1, 0), (2, 2, 2) (x1, y1, z1), (x2, y2, z2)

Bu boşluk tanımı aşağıdaki şekilde sarı renki alan ile gösterilmiştir:



Şekil 52. B Tipinde Tanımlı Kutunun Konteyner İçine Yerleştirilmesinden Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Web uygulaması konteyner içinde oluşan yeni boşluk tanımlarını sıradaki kutuyu yerleştirmeye geçmeden kontrol eder. Bu işlemin amacı boşluk tanımlarını birleştirmektir. Bu sayede konteyner içine yerleştirilecek kutu listesinden daha fazla kutuyu konteyner içine yerleştirebilir miyiz sorgusunu hesaplar. Daha fazla kutunun yerleşebileceği boşluk tanımı belirlenir. Geleneksel noktada boşluk tanım sayısı 1 adet olduğu için bu işlem bu aşamada geçilir.

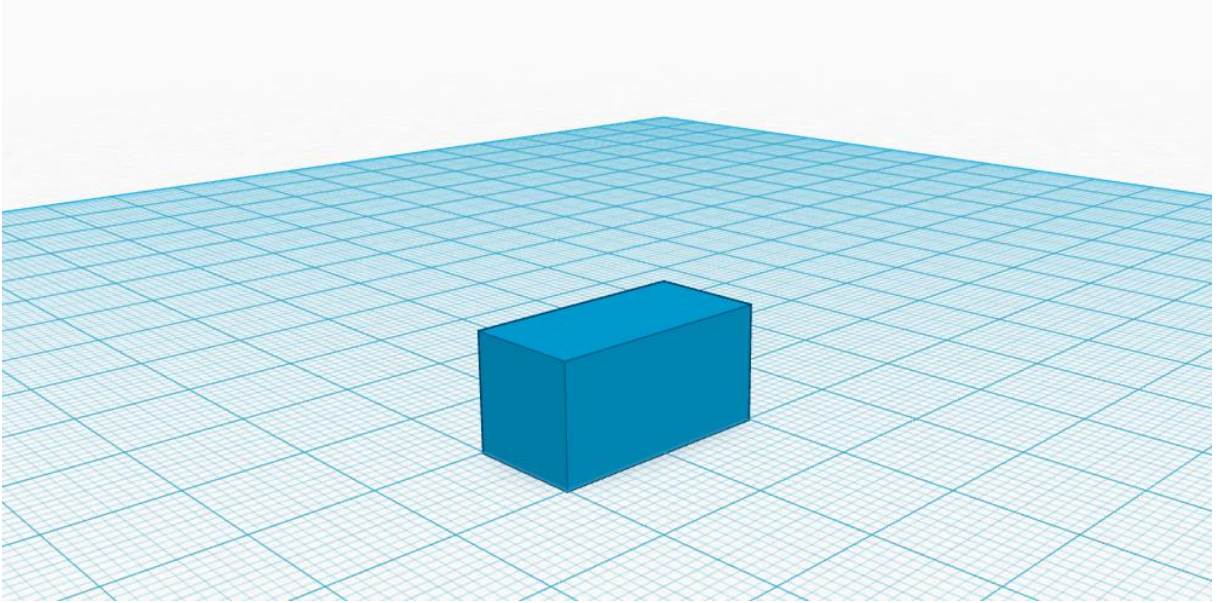
Boşluk Tanımlarının Birleştirilme Sona Erer

İşlem 8:Konteyner İçine Yerleştirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seçilir

Konteyner içine yerleştirilecek kutu listesine tekrar dönülür. Her kutu için aynı olan işlemler dizisine tekrar başlanır.

Listede sırada olan kutu seçilir. Bu kutu aşağıdaki şekilde gösterilen kutudur.

Kutu tipi C:



Şekil 53. C Tipinde Tanımlı Kutu Konteyner İçine Yerleştirilmek İçin Seçiliyor

Genişlik değeri: 1 birim (metre)

Yükseklik değeri: 1 birim (metre)

Derinlik değeri: 2 birim (metre)

Kutunun ağırlığı: 30 kg

Bu kutu bu aşamada geçilir. Çünkü bu kutunun ağırlık değeri konteyner için bu aşamada fazladır.

Konteynerin taşıyabileceği maksimum ağırlık değeri: 80 kg

An itibarı ile konteyner içindeki kutuların ağırlık değeri: 40kg + 20kg=60 kg

3. sırada olan bu kutunun ağırlık değeri : 30 kg.

$60 \text{ kg} + 30 \text{ kg} > 80 \text{ kg}$.

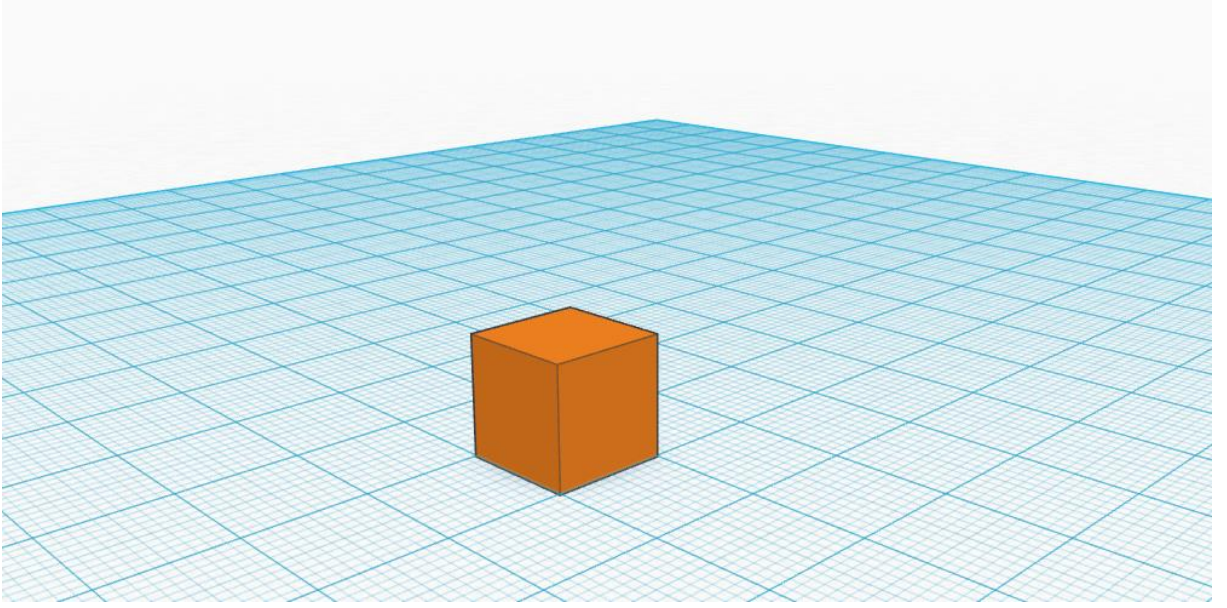
Web uygulamasında uygulanan bir diğer prensip bu aşamada gözlemlenmiş olunur. İlk amaç işlem halinde olan konteyneri doldurmaktır. An itibarı ile konteyner içine hacim ya da ağırlık değerlerinden dolayı yerleşemeyen kutular işlem halinde olan

konteyner doldurulduktan sonra yerleştirilecektir. Bu aşamada aşağıdaki adıma tekrar dönülür.

İşlem 9:Konteyner İçine Yerleştirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seçilir

Listede sırada olan kutu seçilir. Bu kutu aşağıdaki şekilde gösterilen kutudur.

Kutu Tipi D:



Şekil 54. D Tipinde Tanımlı 4 Numaralı Kutu Konteyner İçine Yerleştirilmek İçin Listedden Seçiliyor

Genişlik değeri: 1 birim (metre)

Yükseklik değeri: 1 birim (metre)

Derinlik değeri: 1 birim (metre)

Kutunun ağırlığı: 10 kg

Sırası: 4

İşlem 10:Seçilen kutunun döndürülmesi gerekip gerekmediği sorgulanır

Seçilen kutu küp olduğu için bu işlem adımı geçilir. Şekli küp olan kutunun döndürülmesi bir anlam ifade etmemektedir.

İşlem 11:Seçilen Kutu Konteyner İçine Yerleştiriliyor

An itibarı ile konteyner içindeki boşluk tanımları okunur. Boşluk tanım sayısı 1 adettir. Boşluk tanımının değerleri aşağıdaki gibidir:

Boşluk tanımının genişliği: 1 birim,

Boşluk tanımının yüksekliği: 1 birim,

Boşluk tanımının derinliği: 2 birim,

(1, 1, 0), (2, 2, 2) (x₁, y₁, z₁), (x₂, y₂, z₂)

Seçilen kutunun değerleri okunur: 1x1x1 (genişlik, yükseklik, derinlik), 10kg

Bu değerler kıyaslanır.

Boşluk tanımının genişliği ile kutunun genişlik değeri bir birine eşittir. (1=1)

Boşluk tanımının yüksekliği ile kutunun yükseklik değeri bir birine eşittir. (1=1)

Boşluk tanımının derinliği ile kutunun derinlik değeri bir birine eşit değildir. (2≠1)

Görüldüğü üzere farklılık tek düzlemde dir. (derinlik ya da z ekseni). Kutu aşağıda gösterilen kordinatlara yerleştirilir

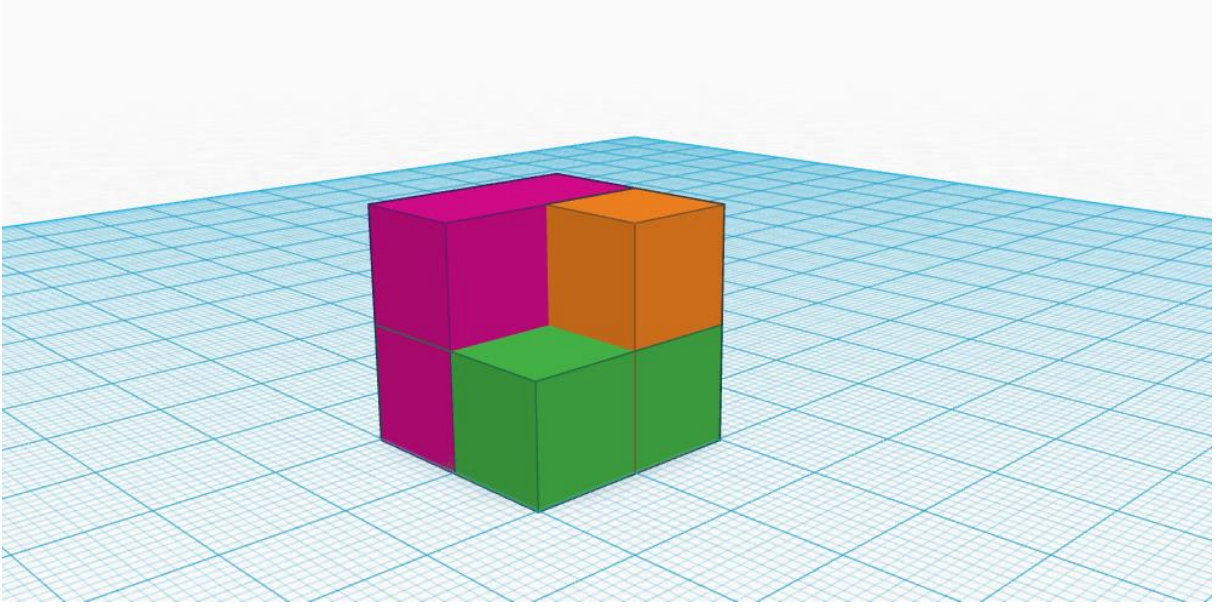
(1, 1, 0), (2, 2, 1) (x₁, y₁, z₁), (x₂, y₂, z₂)

İşlem adımı 3 olarak kayıt edilir.

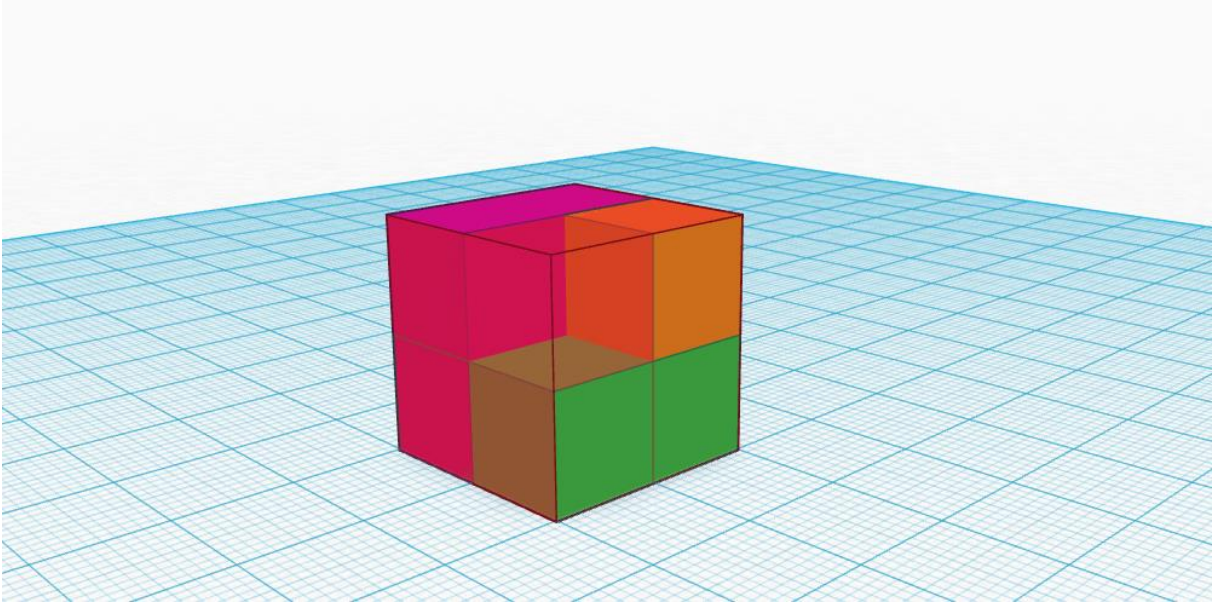
Konteyner içine yerleştirilecek kutunun maksimum ağırlık değeri 10 kg olarak güncellenir.

(80-40-20-10=10kg)

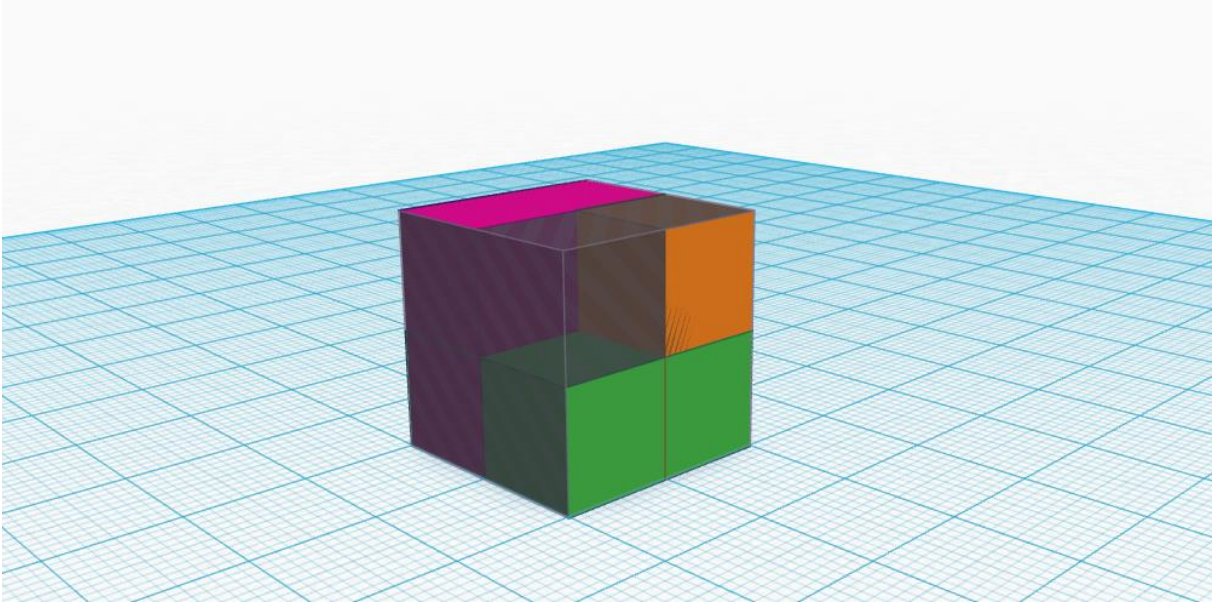
4 numaralı kutunun yerleşimi aşağıdaki şekillerde ayrıntılı olarak gösterilmiştir:



Şekil 55. D Tipinde Tanımlı 4 Numaralı Kutu 1 Numaralı Kutunun Üzerine, 2 Numaralı Kutun İse Önüne Yerleştiriliyor



Şekil 56. D Tipinde Tanımlı 4 Numaralı Kutunun Konteynere Yerleştirilmesi Gösteriliyor



**Şekil 57. D Tipinde Tanımlı 4 Numaralı Kutunun Konteynere Yerleştirilmesi
Konteyner Gölgeleştirilmesi İle Gösteriliyor**

4 numaralı kutu konteyner içine yerleştirildikten sonra konteyner içinde oluşan son boşluk tanımı hesaplanır. Yukarıda yazılan yorumları bu hesap öncesinde tekrar gözden geçirecek olursak:

Boşluk tanımına ait genişlik, yükseklik, derinlik ölçüleri ile kutuya ait genişlik, yükseklik, derinlik ölçüleri karşılaştırılır. Farklılık tek derinlik yani z kordinat düzlemindedir. Bu yüzden konteyner içinde oluşacak yeni boşluk tanımı sayısı 1'dir.

Olasılık 1 (Possibility 1)

Olasılık 1 dahilinde tanımlı boşluk tanımı 1

$$\text{New spaceLength} = \text{oldspacelength} = 1$$

$$\text{New spaceHeight} = \text{oldspaceheight} = 1$$

$$\text{New spacewidth} = \text{oldspacewidth} - \text{boxwidth} = 1$$

$$\text{New space x1} = \text{oldspace x1} = 1$$

$$\text{New space y1} = \text{oldspace y1} = 1$$

$$\text{New space z1} = \text{oldspace z1} + \text{boxwidth} = 1$$

$$\text{New space x2} = \text{New space x1} + \text{New spacelength} = 2$$

New space y2 = New space y1 + New spaceheight = 2

New space z2 = New space z1 + New spacewidth = 2

Boşluk tanım sayısı 1'dir. Ve bu yeni boşluk tanımına ait değerler aşağıdaki gibidir:

yeni boşluk tanımını için genişlik değeri: 1,

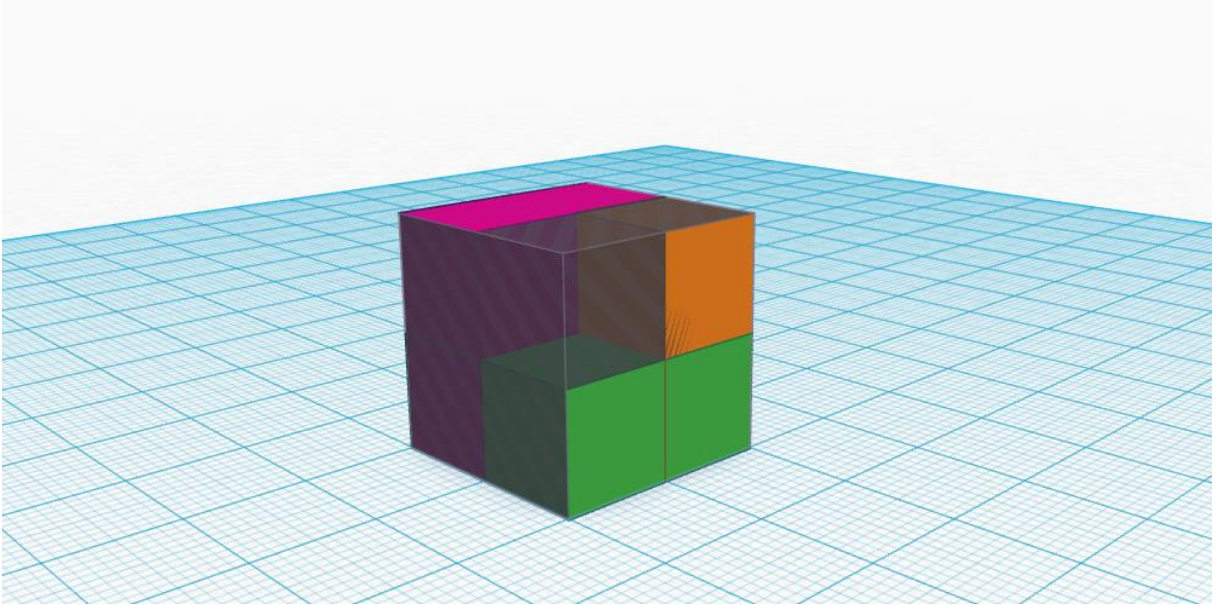
yeni boşluk tanımını için yükseklik değeri:1,

yeni boşluk tanımını için derinlik değeri:1,

yeni boşluk tanımını ifade eden x, y, z kordinat noktaları:

1x1x1, (1, 1, 1), (2, 2, 2) (x1, y1, z1), (x2, y2, z2)

Aşağıdaki resimde boş kalan alan D tipinde 4 numaralı kutunun konteynere yerleşiminden sonraki konteyner içinde kalan 1 adet boşluk tanımını göstermektedir:



Şekil 58. D Tipinde Tanımlı 4 Numaralı Kutunun Konteyner İçine Yerleşiminden Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Web uygulaması konteyner içinde oluşan yeni boşluk tanımlarını sıradaki kutuyu yerleştirmeye geçmeden kontrol eder. Bu işlemin amacı boşluk tanımlarını birleştirmektir. Bu sayede konteyner içine yerleştirilecek kutu listesinden daha fazla kutuyu konteyner içine yerleştirebilir miyiz sorgusunu hesaplar. Daha fazla kutunun yerleşebileceği boşluk tanımı belirlenir. Gelineen noktada boşluk tanım sayısı 1 adet olduğu için bu işlem bu aşamada geçilir.

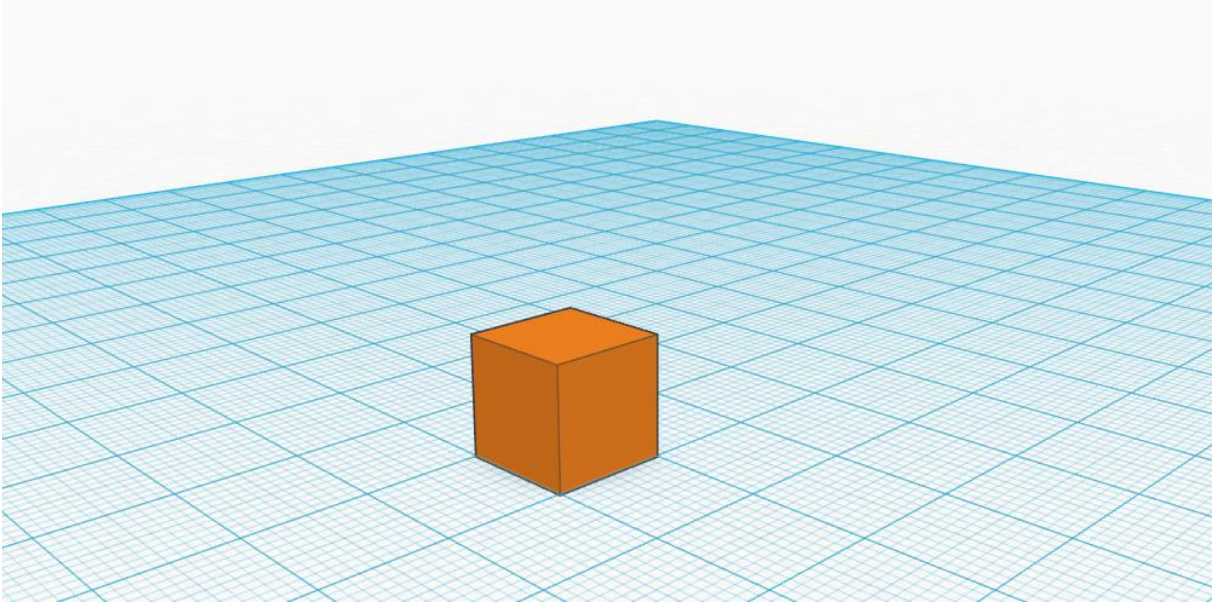
Boşluk Tanımlarının Birleştirilme İşlemi Sona Erer

İşlem 12:Konteyner İçine Yerleştirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seçilir

Konteyner içine yerleştirilecek kutu listesine tekrar dönülür. Her kutu için aynı olan işlemler dizisine tekrar başlanır.

Listede sırada olan kutu seçilir. Bu kutu aşağıdaki şekilde gösterilen kutudur.

Kutu Tipi D:



Şekil 59. D Tipinde Tanımlı 5 Numaralı Kutu Konteynere Yerleştirilmek İçin Listeden Seçiliyor

Genişlik değeri: 1 birim (metre)

Yükseklik değeri: 1 birim (metre)

Derinlik değeri: 1 birim (metre)

Kutunun ağırlığı: 10 kg

Sırası: 5

İşlem 13:Seçilen Kutunun Döndürülmesi Gerekip Gerekmediği Sorgulanır

Seçilen kutu küp olduğu için bu işlem adımı geçilir. Şekli küp olan kutunun döndürülmesi bir anlam ifade etmemektedir.

İşlem 14:Seçilen Kutu Konteyner İçine Yerleştiriliyor

An itibarı ile konteyner içindeki boşluk tanımları okunur. Boşluk tanım sayısı 1 adettir. Boşluk tanımının değerleri aşağıdaki gibidir:

Boşluk tanımının genişliği: 1 birim,

Boşluk tanımının yüksekliği: 1 birim,

Boşluk tanımının derinliği: 1 birim,

(1, 1, 1), (2, 2, 2) (x₁, y₁, z₁), (x₂, y₂, z₂)

Seçilen kutunun değerleri okunur: 1x1x1 (genişlik, yükseklik, derinlik)

Bu deęerler kıyaslanır;

Boşluk tanımının genişlięi ile kutunun genişlik deęeri bir birine eşittir. (1=1)

Boşluk tanımının yükseklięi ile kutunun yükseklik deęeri bir birine eşittir. (1=1)

Boşluk tanımının derinlięi ile kutunun derinlik deęeri bir birine eşittir. (1=1)

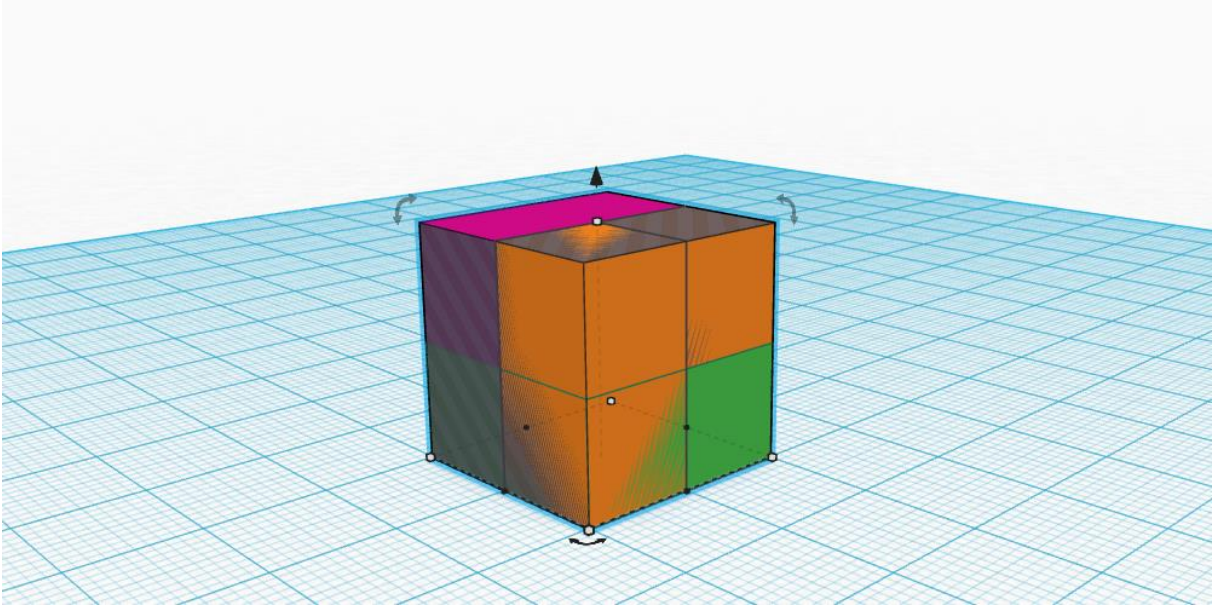
Görüldüęü üzere herhangi bir farklılık yoktur. Bu durum boşluk tanımı ile kutuya ait hacim ve kordinat deęerlerinin birebir aynı olduęunu ifade eder. Bu nedenle kutu bu noktaya yerleştirilir.

(1, 1, 1), (2, 2, 2) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

İşlem adımı 4 olarak kayıt edilir.

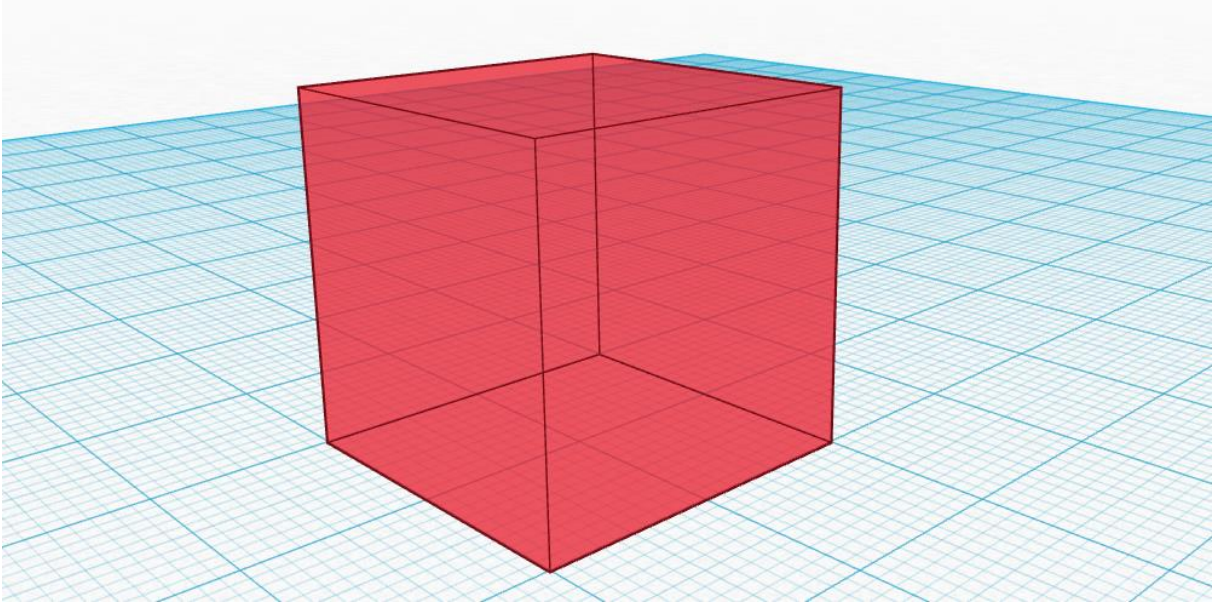
Bu konteyner hacim ve aęırlık olarak %100 verimlilikte kullanılmıřtır. Bundan sonraki ařama için yeni konteyner nesnesi oluşturulacaktır.

5 numaralı kutunun konteyner iine yerleřimi ařaęıdaki řekilde gsterilmiřtir:



řekil 60. D Tipinde Tanımlı 5 Numaralı Kutu Konteynere Yerleřtiriliyor

Sisteme tanımlı konteyner deęerleri okunur. Bu problem için tanımlı konteyner sayısı 1 adettir. Konteyner iine yerleřtirilecek kutu sayısı 0 deęerinden büyük olduęu için sisteme tanımlı konteynerden bir adet daha oluşturulur. Sonraki işlemler için kullanılacak konteyner bu konteynerdir. Yeni tanımlanan konteyner için benzersiz olan konteyner ID numarası 2 deęeri ile yeni tanımlanan konteynere atanır. Ařaęıdaki resimde yeni oluşturulan konteyner gsterilmektedir:



Şekil 61. Problemede Tanımlı Konteynerin İçinde Oluşan Boşluk Tanımı (İçeride Herhangi Bir Kutu Yok İken)

Bir sonraki konteyner içine, kutu yerleştirmeye geçmeden önce ilk konteyner kutu yüklenmesi kısaca özetlenecek olursa sıralama aşağıdaki gibidir.

5.10.4. Konteyner 1'in Yüklenmesi Kısaca Özetleniyor

Adım 1: Konteyner ait tüm değerler okunur, $2 \times 2 \times 2$

Adım 2: Konteyner içindeki boşluk tanım sayısı belirlenir: 1 Adet

Adım 3: Konteyner içindeki boşluk tanımının (1.boşluk tanımı) değerleri okunur: $2 \times 2 \times 2$ (0, 0, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Adım 4: Konteyner içine yerleştirilecek kutu listesinden ilk sırada bulunan kutu seçilir. Seçilen kutu, A tipinde tanımlı $2 \times 2 \times 1$ ölçülerinde, 40kg ağırlındaki 1 numaralı kutudur.

Adım 5: Seçilen 1 numaralı kutunun döndürülüp döndürülmeyeceğine yapılan hesaplamalar sonucu karar verilir; kutu döndürülecektir. Kutuya ait yeni ölçüler değişmiş olur; $2 \times 1 \times 2$.

Adım 6: Seçilen 1 numaralı kutu yapılan hesaplamalar sonucunda aşağıdaki koordinatlara yerleştirilir:

(0, 0, 0), (2, 1, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Adım 7: İşlem adımı 1 olarak atanır. Bu adımdan sonra konteyner içine yerleştirilebilecek maksimum ağırlık 40 kg'dir.

Adım 8: 1 numaralı kutunun konteyner içine yerleştirilmesinden sonra konteyner içinde yeni boşluk tanımı hesaplanır: $2 \times 1 \times 2$ (0, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Adım 9: Konteyner içinde an itibarı ile var olan boşluk tanım ya da tanımlarının birleştirilip birleştirilemeyeceği hesaplanır. Boşluk tanım sayısı 1 olduğu için bu adım geçilir.

Adım 10: Konteyner içine yerleştirilecek kutu listesinden ilk sırada bulunan kutu seçilir. Seçilen kutu, B tipinde tanımlı 1x2x1 ölçülerinde, 20kg ağırlındaki 2 numaralı kutudur.

Adım 11: Seçilen 2 numaralı kutunun döndürülüp döndürülmeyeceğine yapılan hesaplamalar sonucu karar verilir; kutu döndürülecektir. Kutuya ait yeni ölçüler değişmiş olur; 1x2x1.

Adım 12: Seçilen 2 numaralı kutu yapılan hesaplamalar sonucunda aşağıdaki koordinatlara yerleştirilir:

(0, 1, 0), (1, 2, 2) $(x_1, y_1, z_1), (x_2, y_2, z_2)$.

Adım 13: İşlem adımı 2 olarak atanır. Bu adımdan sonra konteyner içine yerleştirilebilecek maksimum ağırlık 20 kg'dir.

Adım 14: 2 numaralı kutunun konteyner içine yerleştirilmesinden sonra konteyner içinde yeni boşluk tanımı hesaplanır: 1x1x2, (1, 1, 0), (2, 2, 2) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

Adım 15: Konteyner içinde an itibarı ile var olan boşluk tanım ya da tanımlarının birleştirilip birleştirilemeyeceği hesaplanır. Boşluk tanım sayısı 1 olduğu için bu adım geçilir.

Adım 16: Konteyner içine yerleştirilecek kutu listesinden ilk sırada bulunan kutu seçilir. Seçilen kutu, C tipinde tanımlı 1x1x2 ölçülerinde, 30kg ağırlındaki 3 numaralı kutudur. Bu kutu ağırlığı yüzünden bu konteyner içine giremeyecektir. Bu kutu bu adımda geçilir.

Adım 17: Konteyner içine yerleştirilecek kutu listesinden ilk sırada bulunan kutu seçilir. Seçilen kutu, D tipinde tanımlı 1x1x1 ölçülerinde, 10kg ağırlındaki 4 numaralı kutudur.

Adım 18: Seçilen 4 numaralı kutunun döndürülüp döndürülmeyeceğine yapılan hesaplamalar sonucu karar verilir; kutu küp olduğu için bu adım geçilir.

Adım 19: Seçilen 4 numaralı kutu yapılan hesaplamalar sonucunda aşağıdaki koordinatlara yerleştirilir:

(1, 1, 0), (2, 2, 1) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

Adım 20: İşlem adımı 3 olarak atanır. Bu adımdan sonra konteyner içine yerleştirilebilecek maksimum ağırlık 10 kg'dir.

Adım 21: 4 numaralı kutunun konteyner içine yerleştirilmesinden sonra konteyner içinde yeni boşluk tanımı hesaplanır: 1x1x1, (1, 1, 1), (2, 2, 2) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

Adım 22: Konteyner içinde an itibarı ile var olan boşluk tanım ya da tanımlarının birleştirilip birleştirilemeyeceği hesaplanır. Boşluk tanım sayısı 1 olduğu için bu adım geçilir.

Adım 23: Konteyner içine yerleştirilecek kutu listesinden ilk sırada bulunan kutu seçilir. Seçilen kutu, D tipinde tanımlı 1x1x1 ölçülerinde, 10kg ağırlındaki 5 numaralı kutudur.

Adım 24: Seçilen 5 numaralı kutunun döndürülüp döndürülmeyeceğine yapılan hesaplamalar sonucu karar verilir; kutu küp olduğu için bu adım geçilir.

Adım 25: Seçilen 5 numaralı kutu yapılan hesaplamalar sonucunda aşağıdaki koordinatlara yerleştirilir:

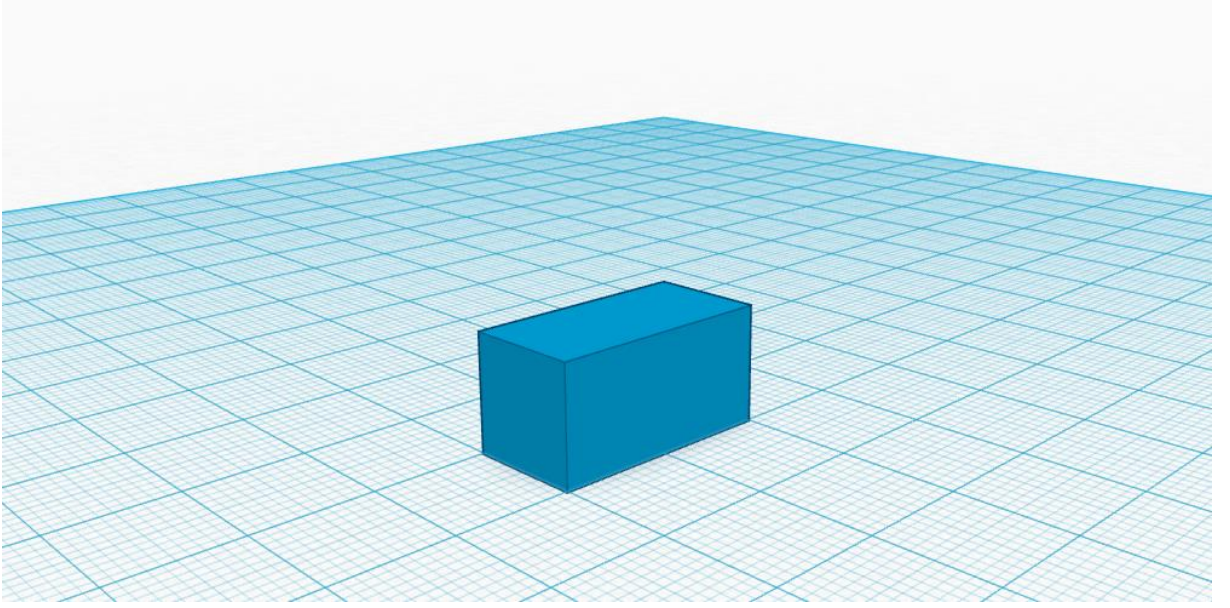
(1, 1, 1), (2, 2, 2) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

Adım 26: İşlem adımı 5 olarak atanır. Bu adımdan sonra konteyner içine yerleştirilebilecek maksimum ağırlık 0 kg'dir. İlk konteyner hacim ve ağırlık olarak %100 verimlilikte kullanılmıştır.

İşlem 15:Konteyner İçine Yerleştirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seçilir (İkinci Konteyner İçin İşlemler Başlar)

Konteyner içine yerleştirilecek kutu listesine tekrar dönülür. Her kutu için aynı olan işlemler dizisine tekrar başlanır. Listede sırada olan kutu seçilir. Bu kutu, ağırlığı nedeni ile ilk konteynere yerleştirilemeyen aşağıdaki şekilde gösterilen kutudur.

Kutu tipi C:



**Şekil 62. C Tipinde Tanımlanmış Kutu 2.Konteynere Yerleştirilmek Üzere
Listeden Seçiliyor**

Genişlik değeri: 1 birim (metre)

Yükseklik değeri: 1 birim (metre)

Derinlik değeri: 2 birim (metre)

Kutunun ağırlığı: 30 kg

Kutunun sıra numarası: 3

İşlem 16:Seçilen Kutunun Döndürülmesi Gerekip Gerekmediği Sorgulanır

Bu kutunun yerleştirileceği konteynerin numarası 2'dir . (konteyner ID). Konteyner içindeki boşluk tanımları belirlenir. Henüz konteyner içine herhangi bir kutu yerleşmediğinden dolayı konteyner içindeki boşluk tanımı konteynerin hacmine eşittir.

Konteyner ölçüleri: 2x2x2 (genişlik, yükseklik, derinlik)

Boşluk tanımının sayısı :1

Boşluk tanımının ölçüleri: 2x2x2 (genişlik, yükseklik, derinlik)

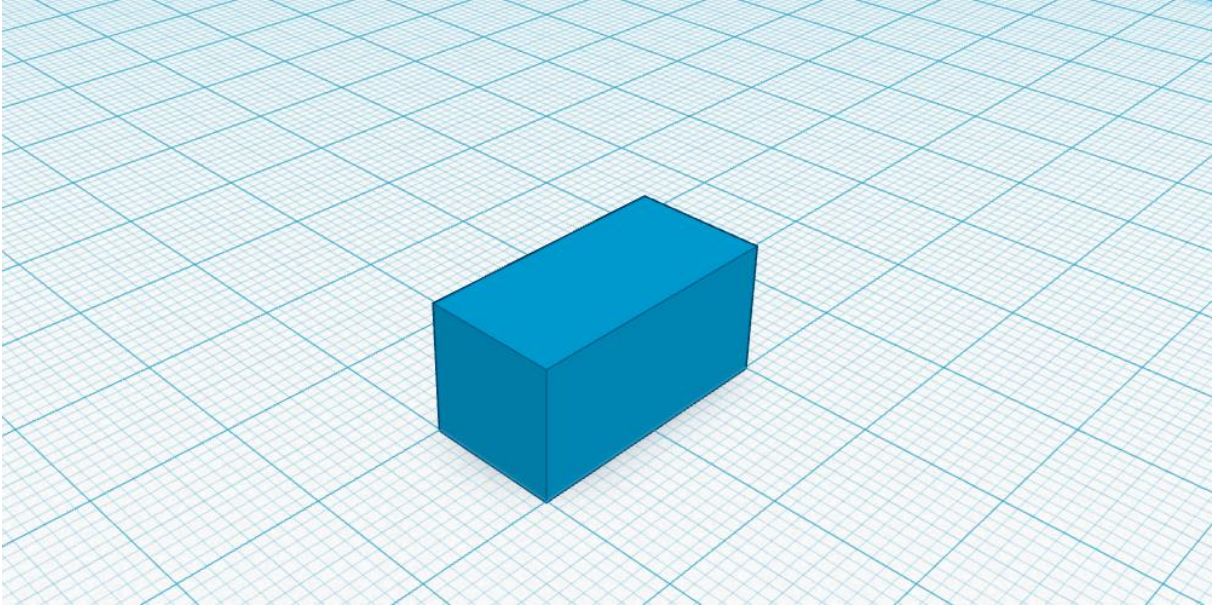
Seçilen kutunun ölçüleri:1x1x2 (genişlik, yükseklik, derinlik)

Seçilen kutunun ağırlığı: 30kg.

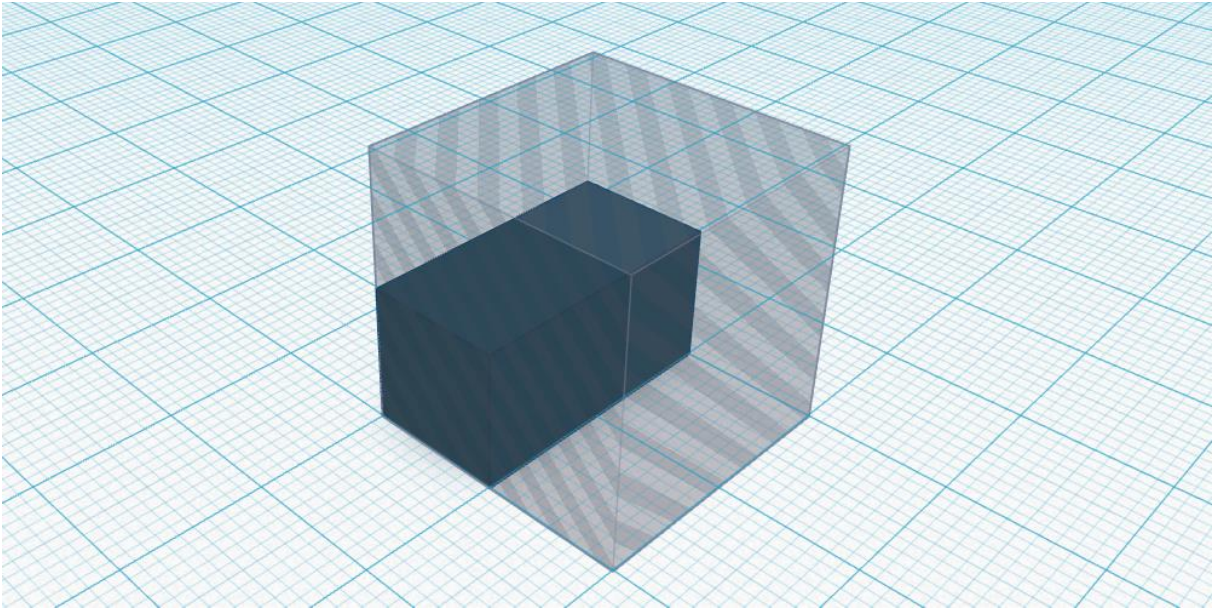
Seçilen kutunun sıra numarası: 3

Seçili kutunun konteyner içinde hangi şekilde olabileceğini ifade eden 6 pozisyon:

1 numaralı pozisyon(kombinasyon)1x1x2,

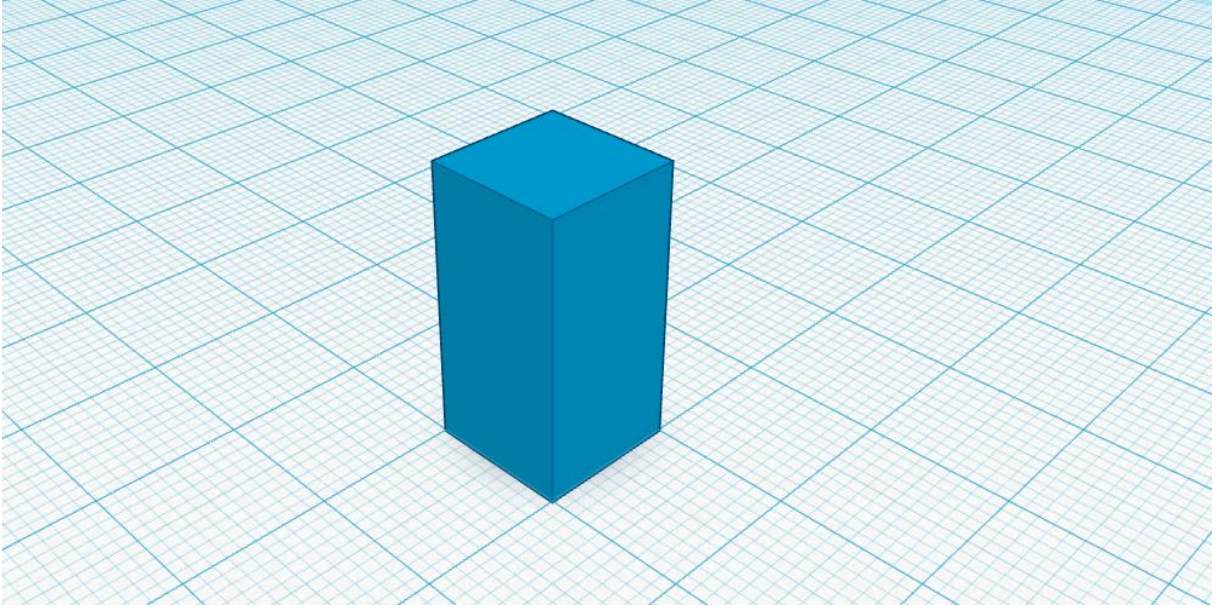


Şekil 63. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon

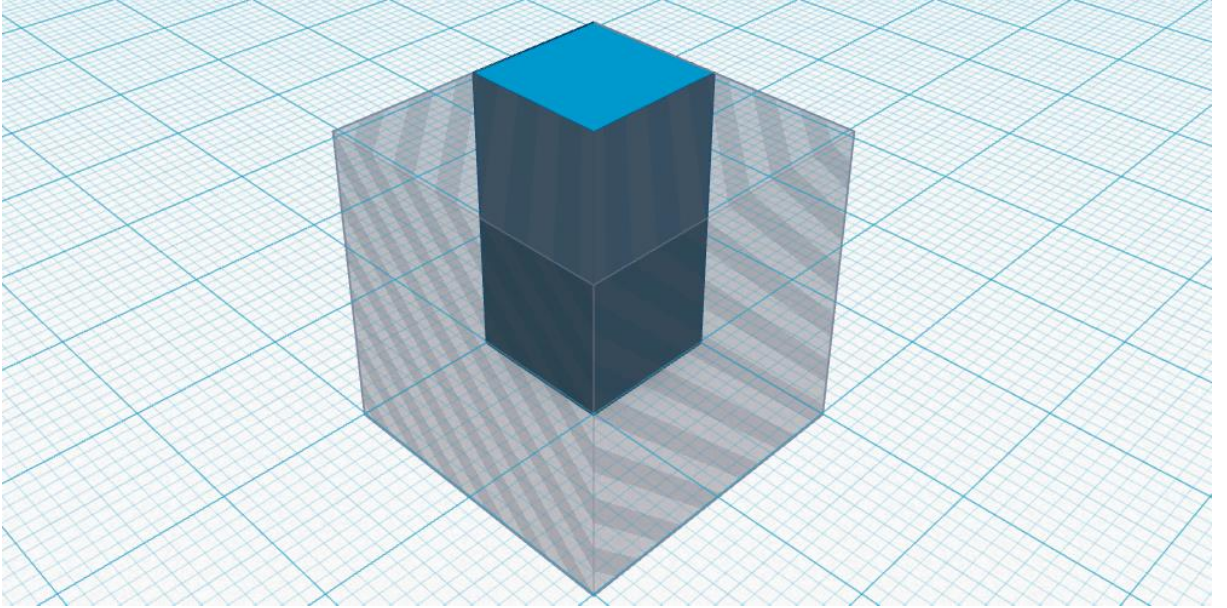


Şekil 64. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 1 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor)

2 numaralı pozisyon (kombinasyon) 1x2x1,

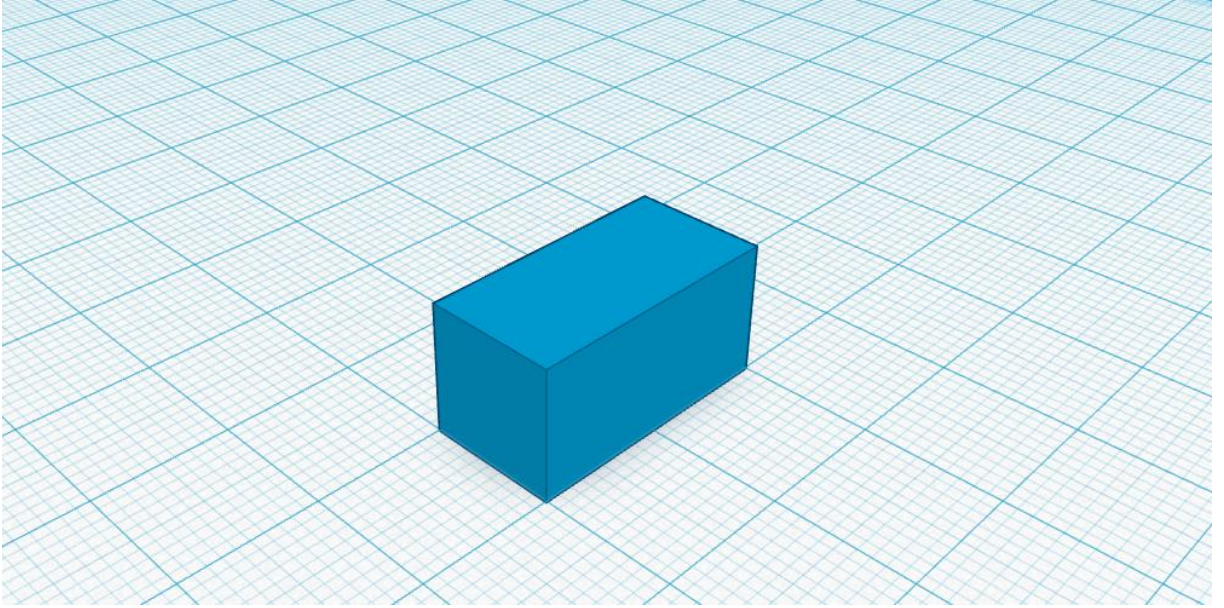


Şekil 65. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon

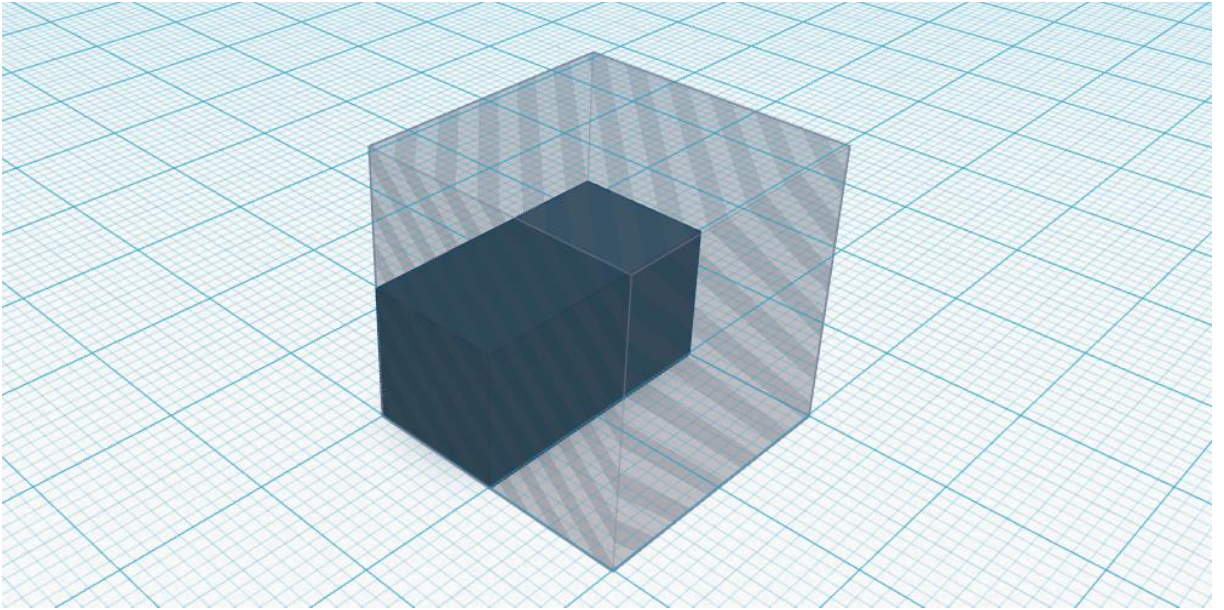


Şekil 66. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 2 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor)

3 numaralı pozisyon (kombinasyon) 1x1x2,

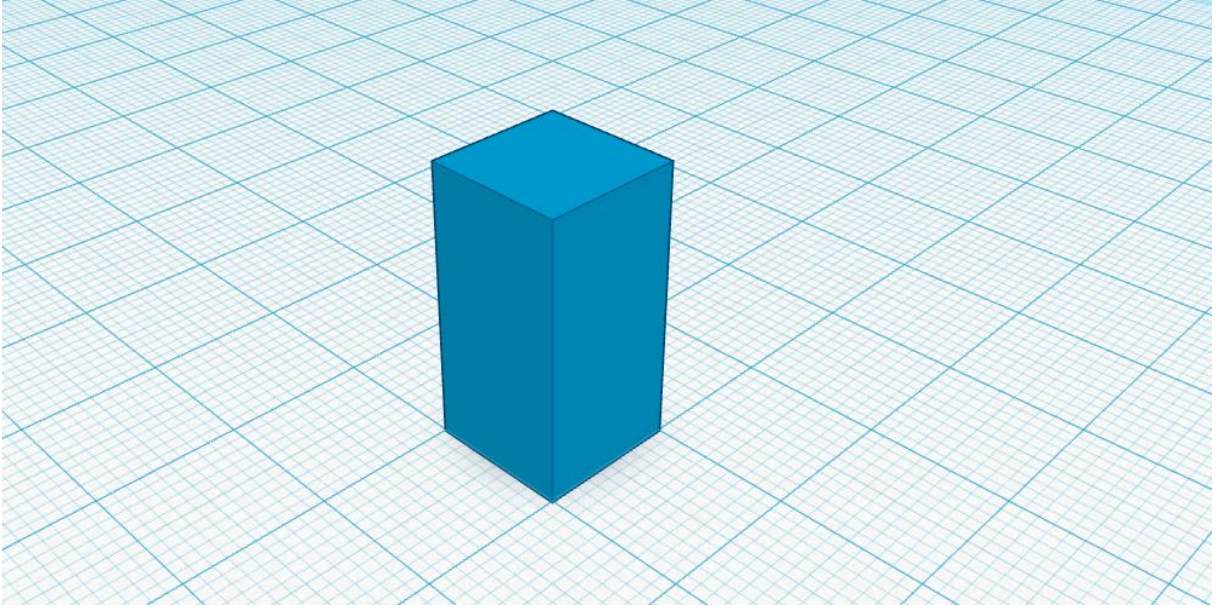


Şekil 67. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon

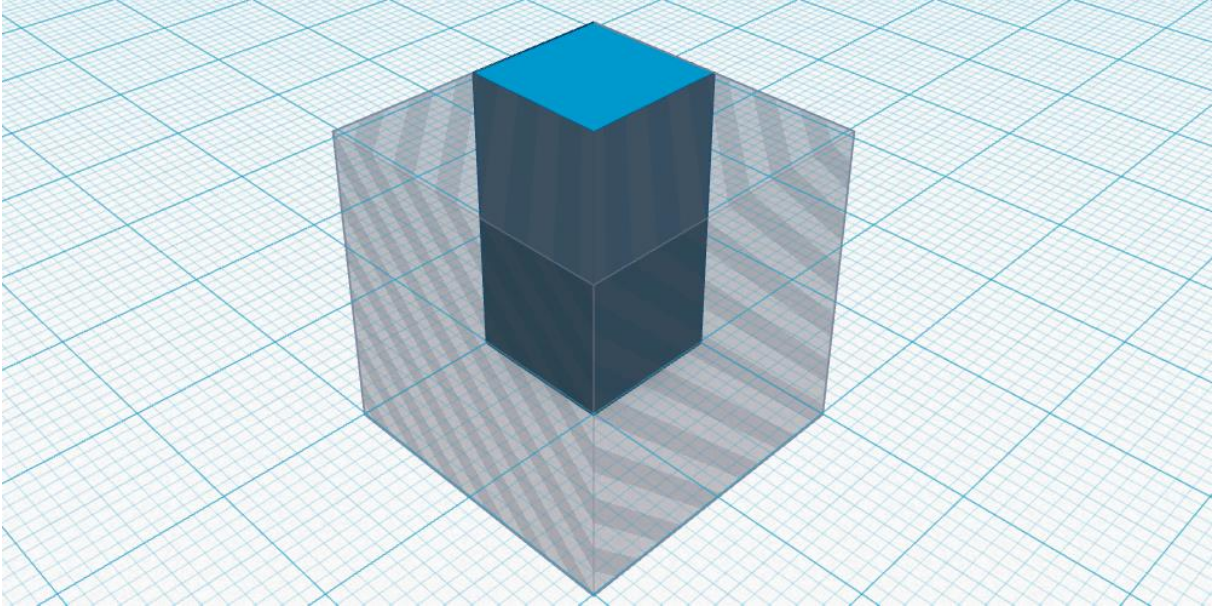


Şekil 68. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 3 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor)

4 numaralı pozisyon(kombinasyon) 1x2x1,

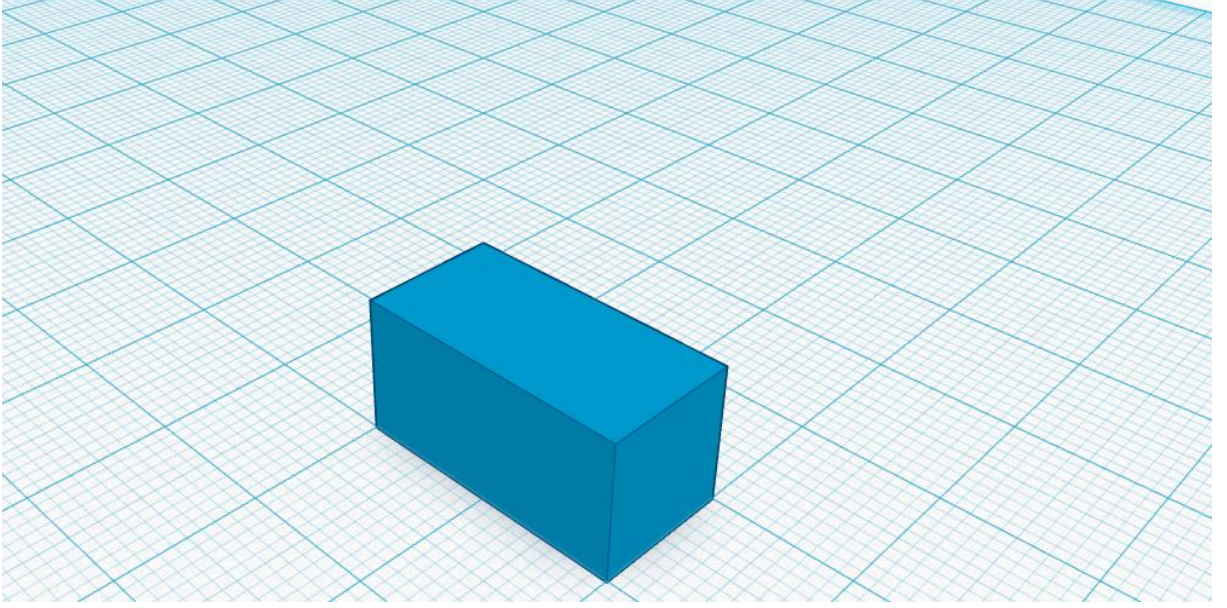


Şekil 69. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon

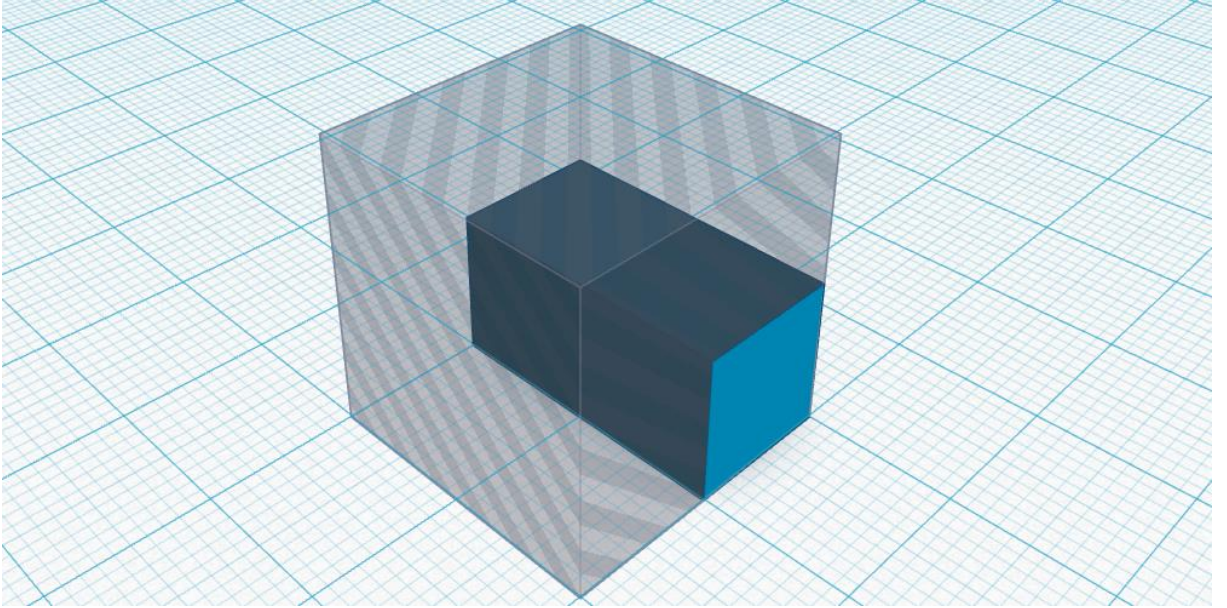


Şekil 70. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 4 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor)

5 numaralı pozisyon (kombinasyon) 2x1x1,

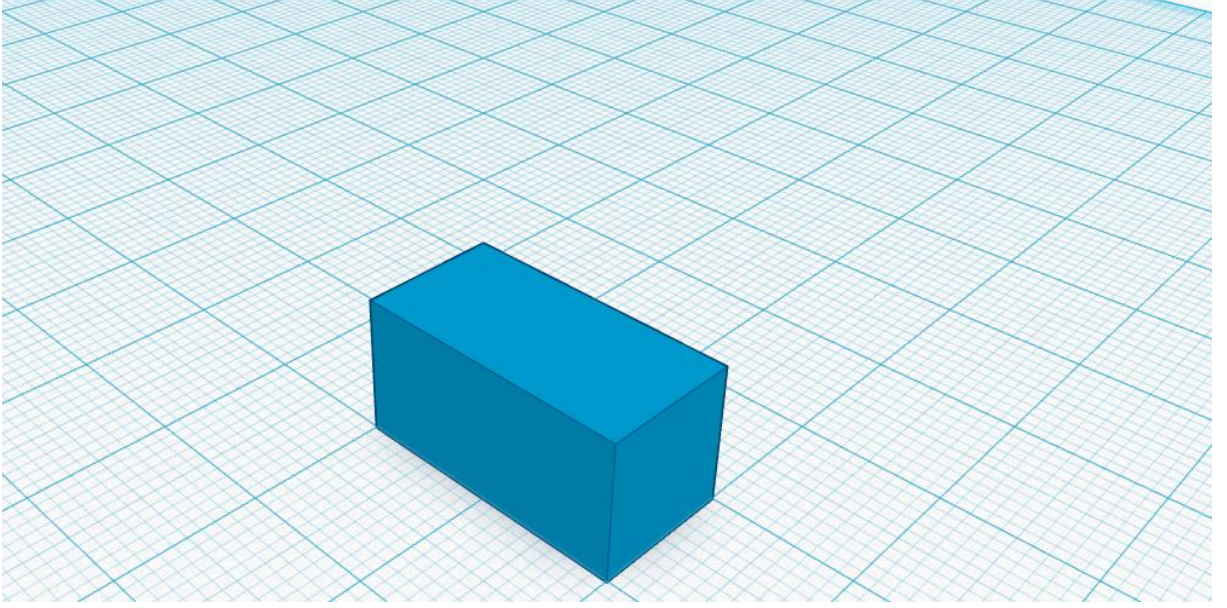


Şekil 71. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon

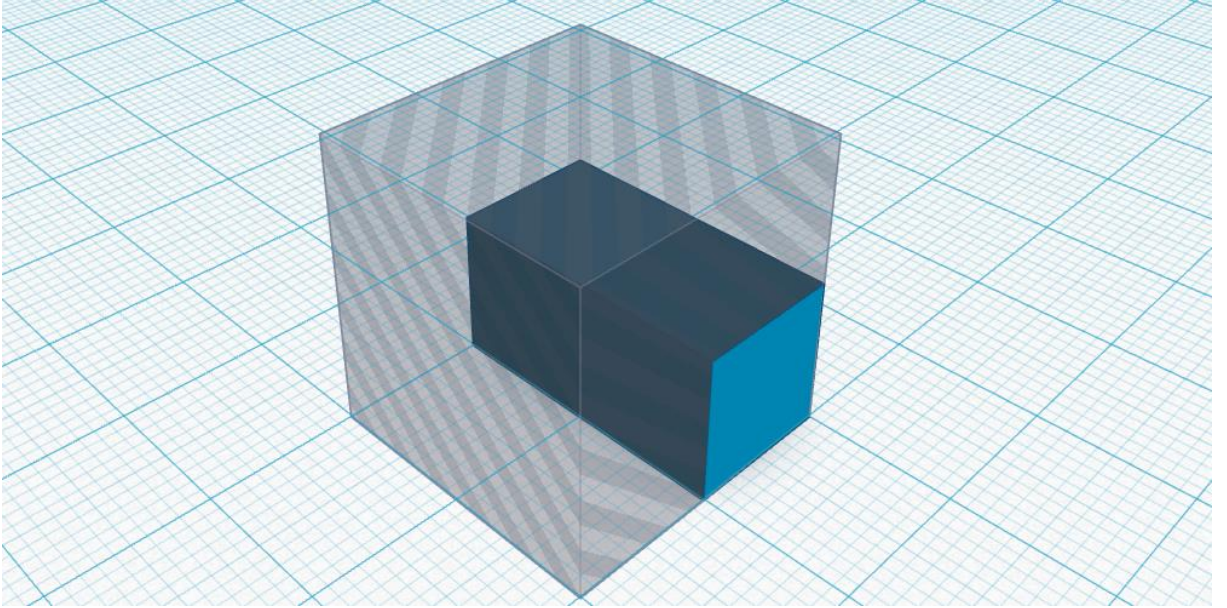


Şekil 72. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 5 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor)

6 numaralı pozisyon(kombinasyon) 2x1x1,



Şekil 73. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon



Şekil 74. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğini Gösteren 6 Numaralı Kombinasyon (Konteyner Gölgeleştirilmesi İle Gösteriliyor)

Problemin çözümün prensibine göre seçilecek kombinasyon yükseklik (y_1) değeri en az olan kombinasyon olmalıdır. Bu nedenle 1x1x2 kombinasyonu seçilir.

Kombinasyon tanımı: $1 \times 1 \times 2$,

Seçilen kutunun ölçüleri: $1 \times 1 \times 2$,

Bu değerler aynı olması kutunun döndürülmesine gerek kalmadığını ifade eder.

İşlem 17:Seçilen Kutu Konteyner İçine Yerleştiriliyor

Bu adımda 3 numaralı kutu konteyner içine yerleştirilecektir. Bir önceki adımda konteyner içindeki boşluk tanımı belirlenmişti: $2 \times 2 \times 2$, $(0, 0, 0)$, $(2, 2, 2)$ (x_1, y_1, z_1) , (x_2, y_2, z_2)

Seçilen kutunun değerleri: $1 \times 1 \times 2$,

Kutu ağırlığı: 30kg.

Seçilen kutu değerleri ile boşluk tanımının değerleri karşılaştırılır:

Boşluk tanımına ait genişlik değeri ile kutunun genişlik değeri bir birine eşit değildir. $(2 \neq 1)$

Boşluk tanımına ait yükseklik değeri ile kutunun yükseklik değeri birbirine eşit değildir. $(2 \neq 1)$

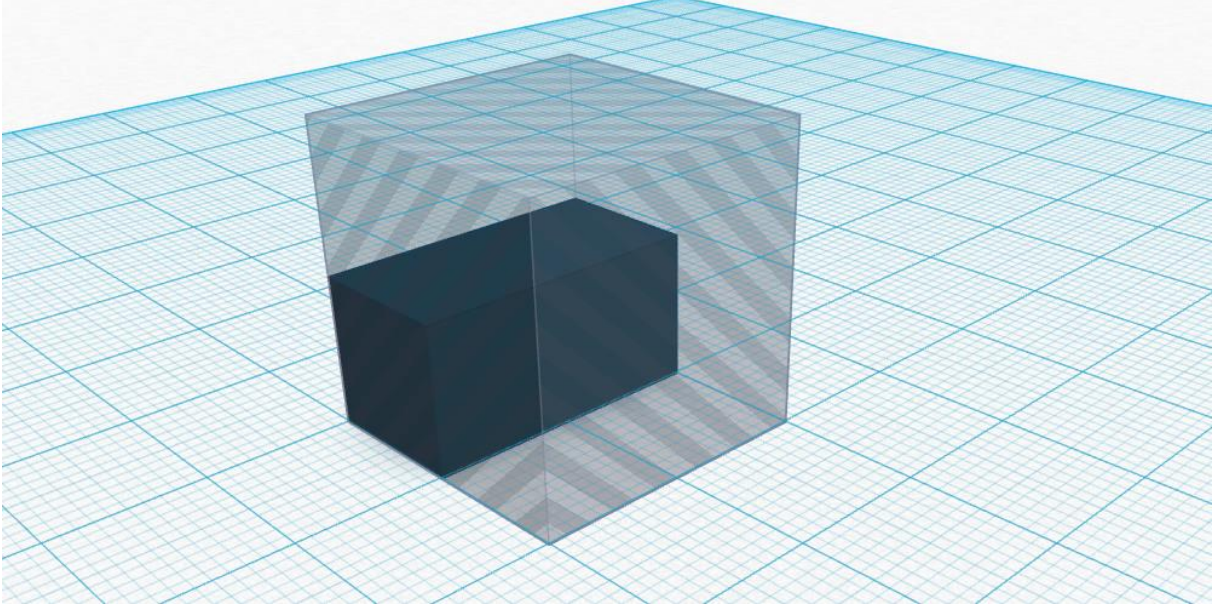
Boşluk tanımına ait derinlik değeri ile kutunun derinlik değeri birbirine eşittir. $(1=1)$

Seçilen kutu şu koordinatlara yerleştirilir: $(0, 0, 0)$, $(1, 1, 2)$ (x_1, y_1, z_1) , (x_2, y_2, z_2)

Bu adımdan sonra konteyner içine yerleştirilebilecek maksimum ağırlık : $80\text{kg} - 30\text{kg} = 50\text{kg}$ 'dır.

İşlem adımı kayıt edilir. $(1, \text{konteynerID}:2, \text{kutuID}:3)$

Kıyaslama sonucunda iki düzlemde fark olduğu ortaya çıkar. (genişlik ve yükseklik). Bu durum 2 farklı olasılığı ifade eder, bu iki olasılık da 2 farklı boşluk tanımı oluşturur. Kutunun konteyner içine yerleştirildiği pozisyonu gösterecek olursak:



Şekil 75. C Tipinde Tanımlı Kutunun Ne Şekilde Konteyner İçine Yerleştirileceğine Karar Veriliyor

Olasılık 1 (Possibility 1) (3 numaralı C tipinde kutu konteynere yerleştikten sonra oluşan olasılık 1 tanımı)

Kutunun genişlik değeri (1) \neq Boşluk tanımının genişlik değeri (2)

1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımının hesaplanması:

New space 1 Length = oldspacelength - boxlength = 1

New space 1 Height = oldspaceheight = 2

New space 1 Width = oldspaceWidth = 2

New space 1 x1 = oldspace x1 + boxlength = 1

New space 1 y1 = oldspace y1 = 0

New space 1 z1 = oldspace z1 = 0

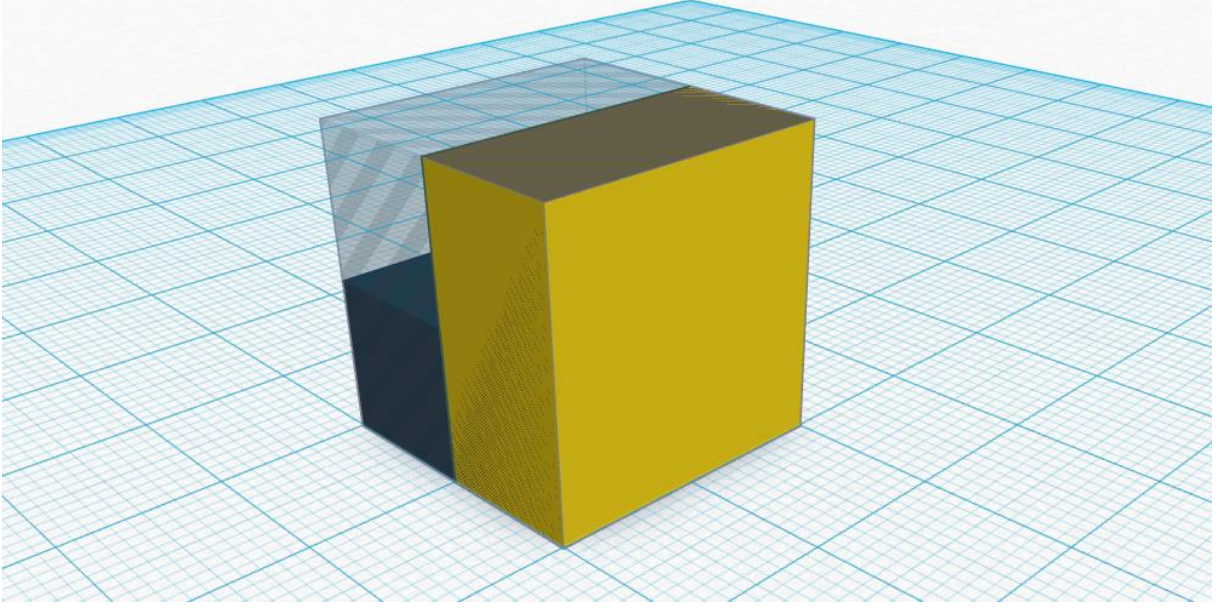
New space 1 x2 = New space 1 x1 + New space 1 length = 2

New space 1 y2 = New space 1 y1 + New space 1 height = 2

New space 1 z2 = New space 1 z1 + New space 1 width = 2

Sonuç: 1x2x2 (genişlik, yükseklik, derinlik) (1, 0, 0), (2, 2, 2) (x1, y1, z1), (x2, y2, z2)

Aşağıdaki resimde, sarı renkle gösterilen alan 1 numaralı olasılık dahilinde oluşan 1 numaralı boşluk tanımını ifade eder.



Şekil 76. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı)

1 numaralı olasılık tanımı dahilinde oluşan 2 numaralı boşluk tanımının hesaplanması:

$$\text{New space 2 Length} = \text{boxlength} = 1$$

$$\text{New space 2 Height} = \text{oldspaceheight} - \text{boxheight} = 1$$

$$\text{New space 2 width} = \text{oldspacewidth} = 2$$

$$\text{New space 2 } x_1 = \text{oldspace } x_1 = 0$$

$$\text{New space 2 } y_1 = \text{oldspace } y_1 + \text{boxheight} = 1$$

$$\text{New space 2 } z_1 = \text{oldspace } z_1 = 0$$

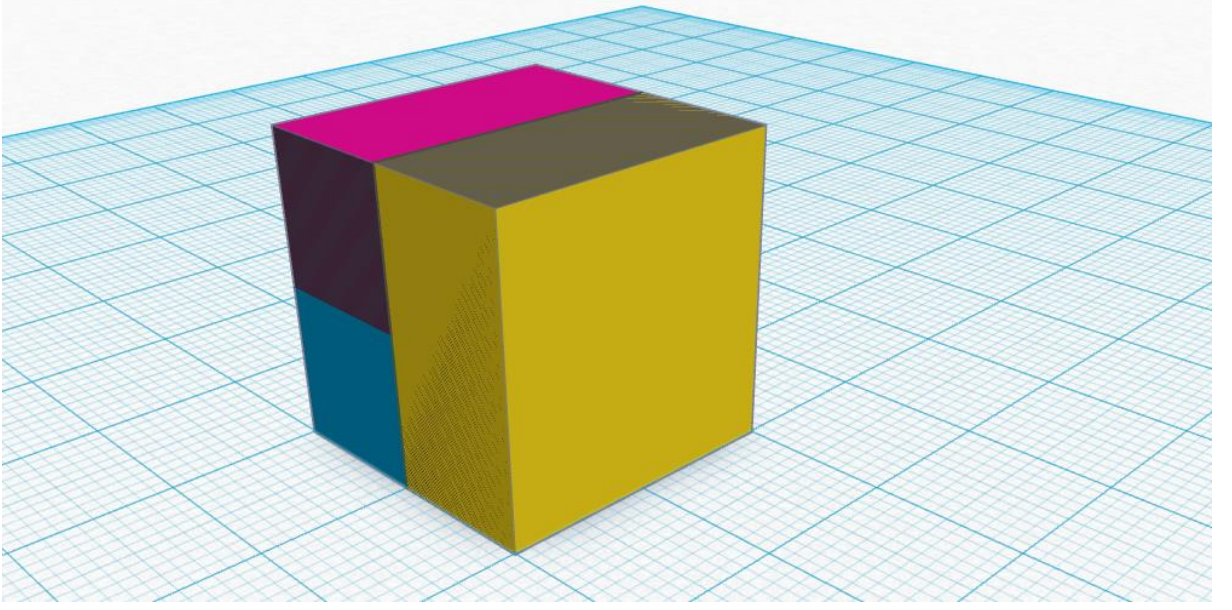
$$\text{New space 2 } x_2 = \text{New space 2 } x_1 + \text{New space 2 length} = 1$$

$$\text{New space 2 } y_2 = \text{New space 2 } y_1 + \text{New space 2 height} = 2$$

$$\text{New space 2 } z_2 = \text{New space 2 } z_1 + \text{New space 2 width} = 2$$

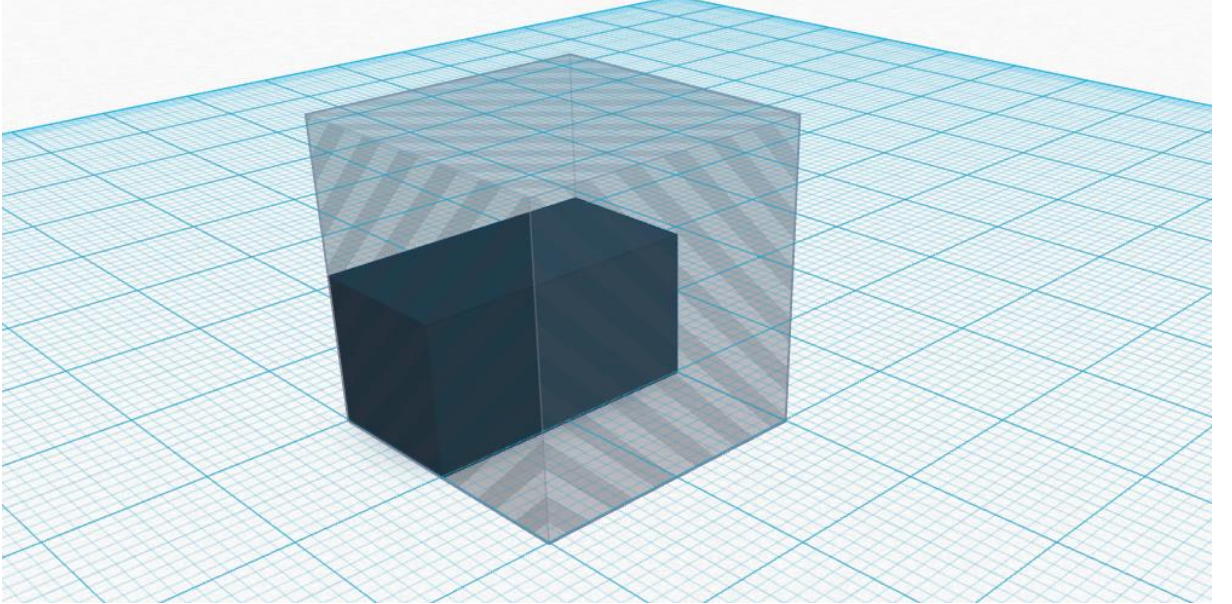
Sonuç: 1x1x2 (genişlik, yükseklik, derinlik) (0, 1, 0), (1, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Aşağıdaki resimde, pembe renkle gösterilen alan 1 numaralı olasılık dahilinde oluşan 2 numaralı boşluk tanımını ifade eder.



Şekil 77. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 2 numaralı boşluk tanımı)

2 numaralı olasılığı hesaplamadan önce kutunun konteyner içindeki pozisyonunu tekrar gösterecek olursak



Şekil 78. C Tipinde Tanımlı Kutunun Konteynere Yerleşmiş Hali Gösteriliyor

Olasılık 2 (Possibility 2) (3 numaralı C tipinde kutu konteynere yerleştikten sonra oluşan olasılık 1 tanımı)

Kutunun yükseklik değeri (1) \neq Boşluk tanımının yükseklik değeri (2)

2 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımının hesaplanması:

$$\text{New space 1 Length} = \text{oldspacelength} = 2$$

$$\text{New space 1 Height} = \text{oldspaceheight} - \text{boxheight} = 1$$

$$\text{New space 1 width} = \text{oldspacewidth} = 2$$

$$\text{New space 1 } x1 = \text{oldspace } x1 = 0$$

$$\text{New space 1 } y1 = \text{oldspace } y1 + \text{boxheight} = 1$$

$$\text{New space 1 } z1 = \text{oldspace } z1 = 0$$

$$\text{New space 1 } x2 = \text{New space 1 } x1 + \text{New space 1 length} = 2$$

$$\text{New space 1 } y2 = \text{New space 1 } y1 + \text{New space 1 height} = 2$$

$$\text{New space 1 } z2 = \text{New space 1 } z1 + \text{New space 1 width} = 2$$

Box length: 1 is not equal to spacelength: 2

Sonuç: 2x1x2 (genişlik, yükseklik, derinlik) (0, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

2 numaralı olasılık tanımı dahilinde oluşan 2 numaralı boşluk tanımının hesaplanması:

$$\text{New space 2 Length} = \text{oldspacelength} - \text{boxlength} = 1$$

$$\text{New space 2 Height} = \text{boxheight} = 1$$

$$\text{New space 2 Width} = \text{oldspaceWidth} = 2$$

$$\text{New space 2 } x1 = \text{oldspace } x1 + \text{boxlength} = 1$$

$$\text{New space 2 } y1 = \text{oldspace } y1 = 0$$

$$\text{New space 2 } z1 = \text{oldspace } z1 = 0$$

$$\text{New space 2 } x2 = \text{New space 2 } x1 + \text{New space 2 length} = 2$$

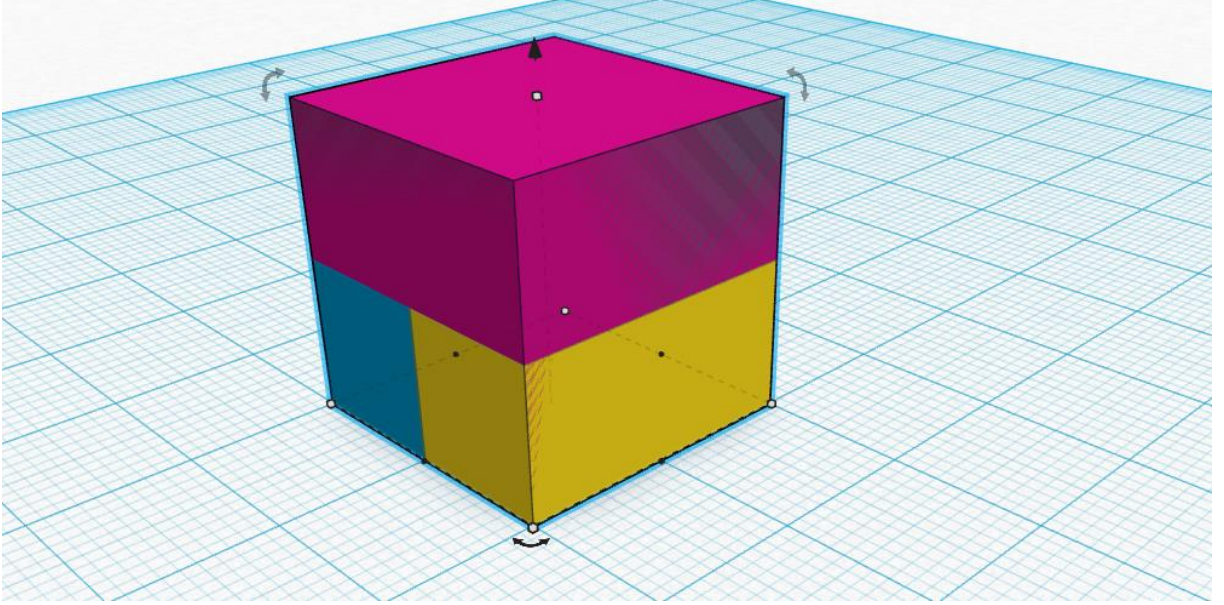
$$\text{New space 2 } y2 = \text{New space 2 } y1 + \text{New space 2 height} = 1$$

$$\text{New space 2 } z2 = \text{New space 2 } z1 + \text{New space 2 width} = 2$$

Sonuç: 1x1x2 (genişlik, yükseklik, derinlik) (1, 0, 0), (2, 1, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Aşağıdaki resimde pembe renkle gösterilen alan, 2 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı iken, 2x1x2 (0, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2) sarı renkle gösterilen alan 2 numaralı olasılık tanımı dahilinde oluşan 2 numaralı

boşluk tanımıdır $(1 \times 1 \times 2)$, $(1, 0, 0)$, $(2, 1, 2)$ (x_1, y_1, z_1) , (x_2, y_2, z_2)



Şekil 79. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor (2 numaralı olasılık tanımı dahilinde oluşan 1 ve 2 numaralı boşluk tanımları)

Bu aşamada web uygulaması olasılık 1 ve 2 için hesaplamalar yapar. Bu hesaplamalarda geride kalan kutuların konteyner içindeki boşluk tanımlarına yerleştirildikleri farz edilerek işlemlere devam edilir. En fazla kutuyu konteyner içine alabilecek olan boşluk tanımı seçilecektir.

İşlem 18: Sanal İşlem Başlar (Virtual Process Started)

Olasılık 1 ve Olasılık 2 kaç tane kutu içine alabilir bu hesaplanacaktır.

1 numaralı olasılık için:

An itibarı ile konteyner içine koyulabilecek maksimumum ağırlık : 50 kg (80kg - 30kg)

1 numaralı olasılığın oluşturduğu boşluk tanım sayısı : 2 adet

1 numaralı olasılık altında oluşan 1 numaralı boşluk tanımın ölçüleri: $1 \times 2 \times 2$

1 numaralı olasılık altında oluşan 2 numaralı boşluk tanımının ölçüleri: $1 \times 1 \times 2$

Konteyner içine yerleştirilmek üzere bekleyen kutu sayısı: 2

1. Kutu:

Kutu No:6, D tipinde tanımlı kutu

Kutu No 6 için ölçüler: 1x1x1 (genişlik, yükseklik, derinlik)

Kutu no 6 için ağırlık: 10kg

2. Kutu:

Kutu No:7, D tipinde tanımlı kutu

Kutu No 7 için ölçüler: 1x1x1 (genişlik, yükseklik, derinlik)

Kutu no 7 için ağırlık: 10kg

Problemin çözüm prensibine göre ilk olarak yükseklik değeri (y_1) en düşük olan boşluk tanımı ele alınacaktır.

Seçilen kutu, 6 numaralı kutu. 1x1x1, 10kg

Kutunun konteyner içindeki pozisyonu belirlenecektir. Bir başka deyişle kutunun döndürülmesi gerekiyor mu sorgusu yapılır. Kutu küp olduğu için döndürülme adımı geçilir.

İşlem 19: Sanal İşlem Başlar - Seviye 1 (Virtual Process Started - Level 1)

Bu işlem 1 numaralı olasılık için hesaplanmaktadır.

Konteyner içine yerleşebilecek maksimum ağırlık: 50kg.

1 numaralı olasılığın içerdiği boşluk tanımları 2 adet idi. Bunlar ;

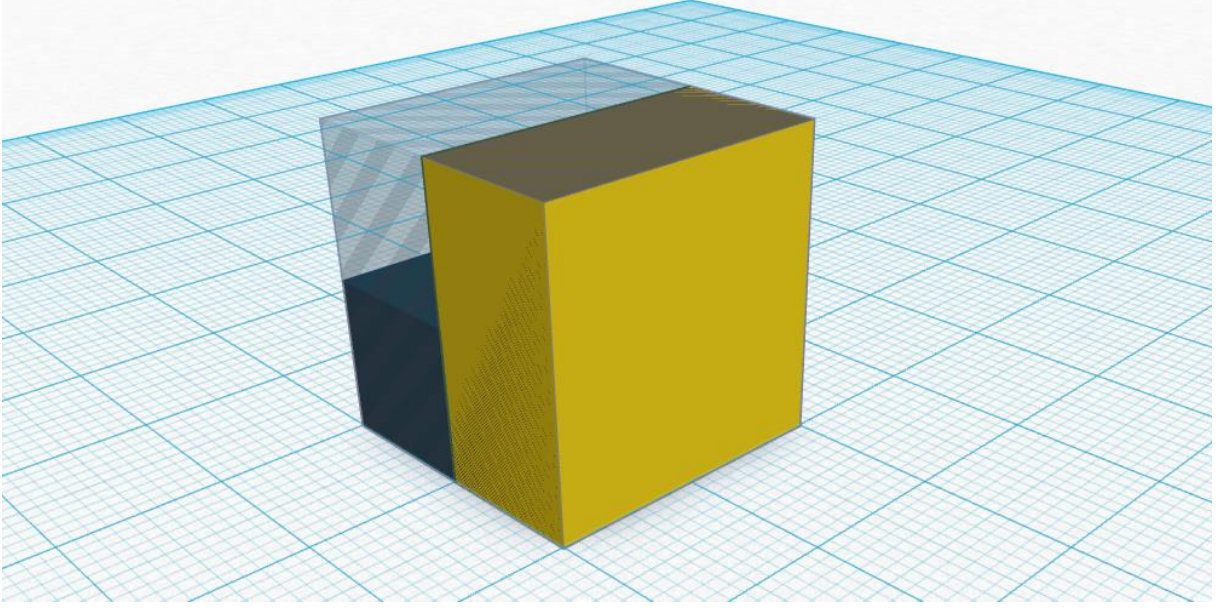
1 numaralı yeni boşluk tanımı:

1x2x2 (genişlik, yükseklik, derinlik) (1, 0, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

2 numaralı yeni boşluk tanımı:

1x1x2 (genişlik, yükseklik, derinlik) (0, 1, 0), (1, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Bu iki boşluk tanımından y_1 değeri küçük olan ele alınır. Dolayısı ile seçilen 1 numaralı boşluk tanımıdır. 1 Numaralı boşluk tanımı aşağıdaki resimde sarı renk ile gösterilen alandır.

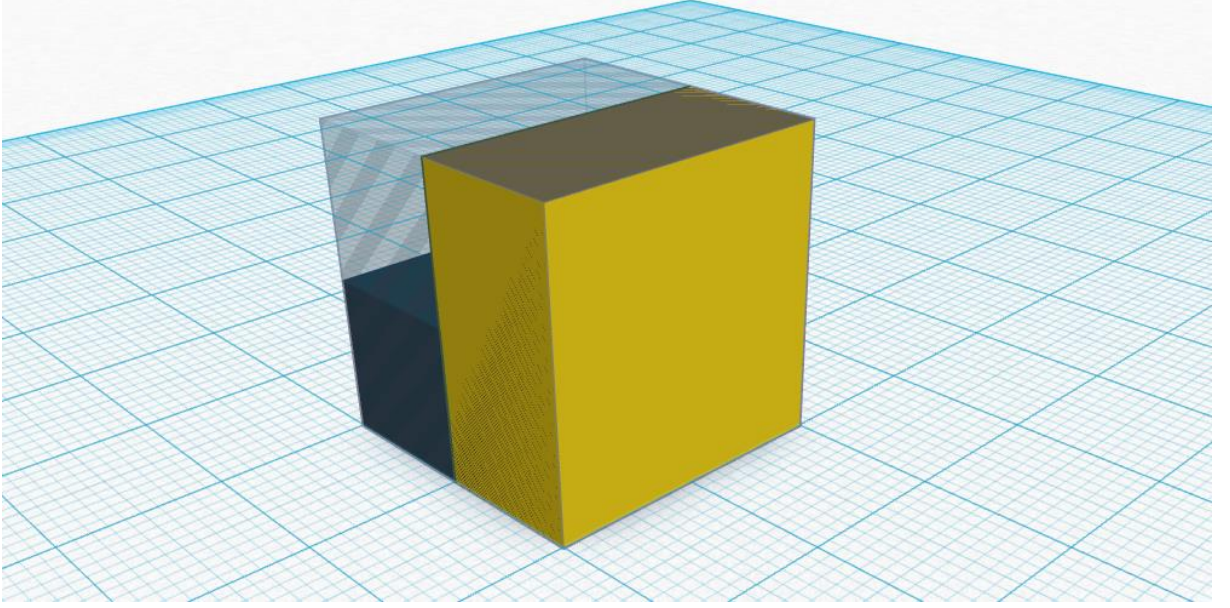


Şekil 80. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı)

Kutu ile bu boşluk tanımının tüm ölçütleri kıyaslanır:

Bu arada daha sonraki adımların anlaşılır olması adına şu açıklama önemlidir:

Resimde sarı renk ile gösterilen 1 numaralı boşluk tanımı ele alınırsa, (1x2x2) bir sonraki aşamada bu boşluk tanımı iki farklı olasılığı oluşturacaktır. Bu her iki olasılık da iki farklı yeni boşluk tanımını oluşturacaktır. Bunlardan biri seçilecektir. Bu işlemin sonunda yukarıdaki 2 numaralı boşluk tanımı ile 1 numaralı boşluk tanımının oluşturacağı 2 farklı boşluk tanımı toplamda 3 boşluk tanımı ifade eder.



Şekil 81. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı)

Seçilen boşluk tanımı (1x2x2, sarı ile gösterilen boşluk tanımı) ile kutunun ölçüleri karşılaştırılır:

Seçilen boşluk tanımına ait genişlik değeri kutunun genişlik değeri ile eşittir. (1=1)

Seçilen boşluk tanımına ait yükseklik değeri kutunun yükseklik değeri ile eşit değildir. (2≠1)

Seçilen boşluk tanımına ait derinlik değeri kutunun derinlik değeri ile eşit değildir. (2≠1)

Kıyaslama sonucunda iki düzlemde farklılık gözlemlenir. (yükseklik ve derinlik) Bu durum iki yeni farklı olasılığı, her olasılık da iki farklı boşluk tanımını oluşturur.

Gelinen aşama şu şekilde açıklanabilir:

Olasılık1

Olasılık2 (An itibarı ile hesaplamalar Olasılık 1 üzerinden devam etmektedir)

Olasılık1.BoşlukTanımı1

Olasılık1.BoşlukTanımı1. Olasılık1

Olasılık1.BoşlukTanımı1. Olasılık1.BoşlukTanımı1

Olasılık1.BoşlukTanımı1. Olasılık1.BoşlukTanımı2

Olasılık1.BoşlukTanımı1.Olasılık2

Olasılık1.BoşlukTanımı1. Olasılık2.BoşlukTanımı1**Olasılık1.BoşlukTanımı1. Olasılık2.BoşlukTanımı2**

Yeni oluşan olasılık değerleri hesaplanır

Olasılık1.BoşlukTanımı1. Olasılık1

Kutu yüksekliği boşluk tanımının yüksekliği ile farklıdır.

Olasılık1.BoşlukTanımı1. Olasılık1.BoşlukTanımı1 değerinin hesaplanması

$$\text{New space 1 Length} = \text{oldspacelength} = 1$$

$$\text{New space 1 Height} = \text{oldspaceheight} - \text{boxheight} = 1$$

$$\text{New space 1 width} = \text{oldspacewidth} = 2$$

$$\text{New space 1 } x_1 = \text{oldspace } x_1 = 1$$

$$\text{New space 1 } y_1 = \text{oldspace } y_1 + \text{boxheight} = 1$$

$$\text{New space 1 } z_1 = \text{oldspace } z_1 = 0$$

$$\text{New space 1 } x_2 = \text{New space 1 } x_1 + \text{New space 1 length} = 2$$

$$\text{New space 1 } y_2 = \text{New space 1 } y_1 + \text{New space 1 height} = 2$$

$$\text{New space 1 } z_2 = \text{New space 1 } z_1 + \text{New space 1 width} = 2$$

Sonuç: $1 \times 1 \times 2$ (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Olasılık1.BoşlukTanımı1. Olasılık1.BoşlukTanımı2 değerinin hesaplanması

$$\text{New space 2 Length} = \text{oldspacelength} = 1$$

$$\text{New space 2 Height} = \text{boxheight} = 1$$

$$\text{New space 2 width} = \text{oldspacewidth} - \text{boxwidth} = 1$$

$$\text{New space 2 } x_1 = \text{oldspace } x_1 = 1$$

$$\text{New space 2 } y_1 = \text{oldspace } y_1 = 0$$

$$\text{New space 2 } z_1 = \text{oldspace } z_1 + \text{boxwidth} = 1$$

$$\text{New space 2 } x_2 = \text{New space 2 } x_1 + \text{New space 2 length} = 2$$

$$\text{New space 2 } y_2 = \text{New space 2 } y_1 + \text{New space 2 height} = 1$$

$$\text{New space 2 } z_2 = \text{New space 2 } z_1 + \text{New space 2 width} = 2$$

Sonuç: $1 \times 1 \times 1$ (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 1, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Olasılık1.BoşlukTanımı1. Olasılık2

Kutu derinliği boşluk tanımının derinliği ile farklıdır

Olasılık1.BoşlukTanımı1. Olasılık2.BoşlukTanımı1

New space 1 Length = oldspacelength = 1

New space 1 Height = oldspaceheight = 2

New space 1 width = oldspacewidth - boxwidth = 1

New space 1 x1 = oldspace x1 = 1

New space 1 y1 = oldspace y1 = 0

New space 1 z1 = oldspace z1 + boxwidth = 1

New space 1 x2 = New space 1 x1 + New space 1 length = 2

New space 1 y2 = New space 1 y1 + New space 1 height = 2

New space 1 z2 = New space 1 z1 + New space 1 width = 2

Sonuç: 1x2x1 (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 2, 2) (x₁, y₁, z₁), (x₂, y₂, z₂)

Olasılık1.BoşlukTanımı1. Olasılık2.BoşlukTanımı2

New space 2 Length = oldspacelength = 1

New space 2 Height = oldspaceheight - boxheight = 1

New space 2 width = boxwidth = 1

New space 2 x1 = oldspace x1 = 1

New space 2 y1 = oldspace y1 + boxheight = 1

New space 2 z1 = oldspace z1 = 0

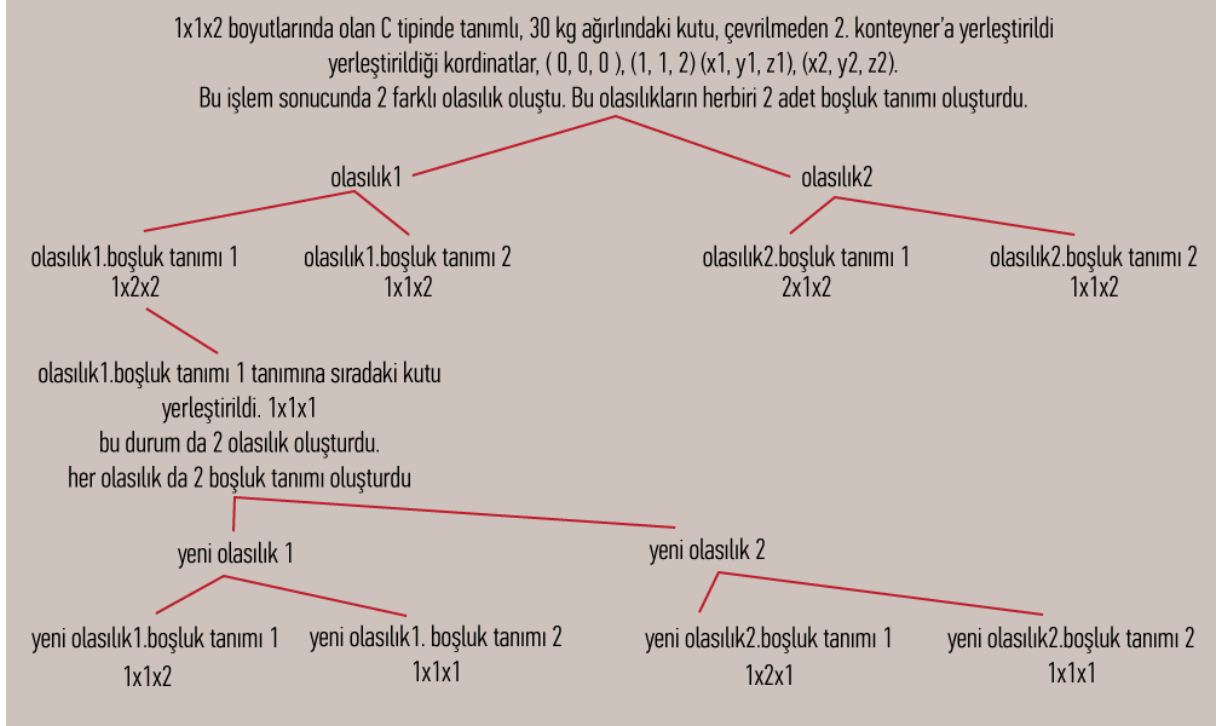
New space 2 x2 = New space 2 x1 + New space 2 length = 2

New space 2 y2 = New space 2 y1 + New space 2 height = 2

New space 2 z2 = New space 2 z1 + New space 2 width = 1

Sonuç: 1x1x1 (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 1) (x₁, y₁, z₁), (x₂, y₂, z₂)

İkinci konteyner'r ilk kutu yerleştirilmesinden bu aşamaya kadar geçen süreç aşağıdaki resimde ifade edilmiştir:



Şekil 82. İkinci Konteynere iki kutu yerleştikten sonra oluşan boşluk tanımlarını gösteriliyor

Gelinen noktada toplam 3 boşluk tanımlı vardır.

1 - Olasılık1.BoşlukTanımı2

1x1x2 (genişlik, yükseklik, derinlik) (0, 1, 0), (1, 2, 2) (x1, y1, z1), (x2, y2, z2)

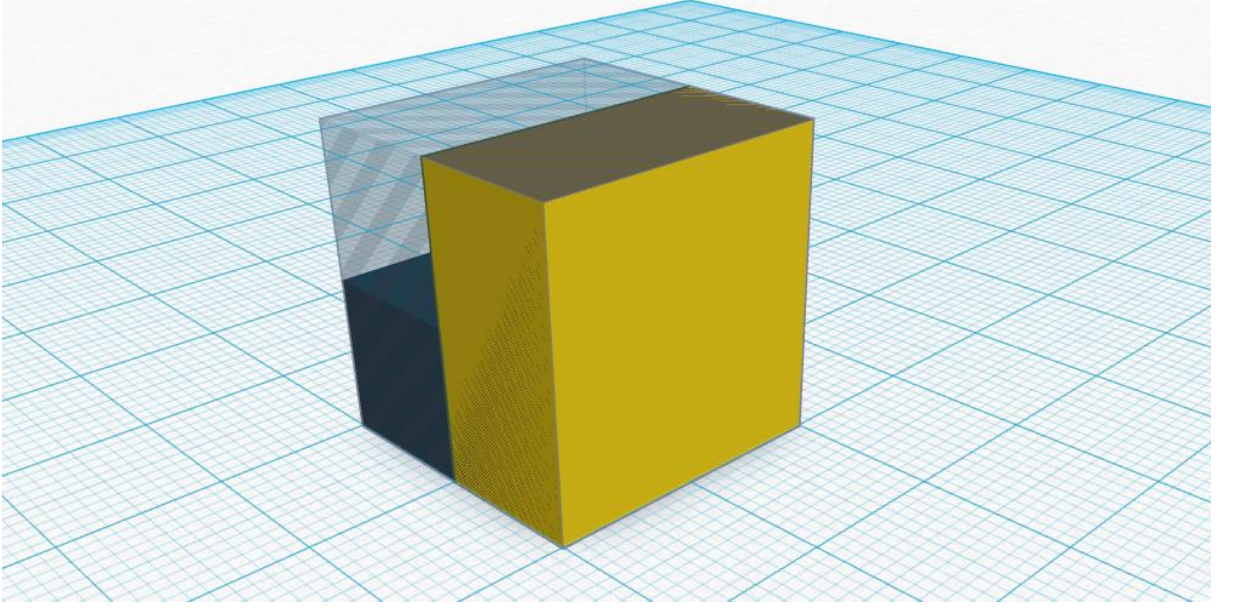
2 - Olasılık1.BoşlukTanımı1. Olasılık1.BoşlukTanımı1

1x1x2 (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 2) (x1, y1, z1), (x2, y2, z2)

3 - Olasılık1.BoşlukTanımı1. Olasılık1.BoşlukTanımı2

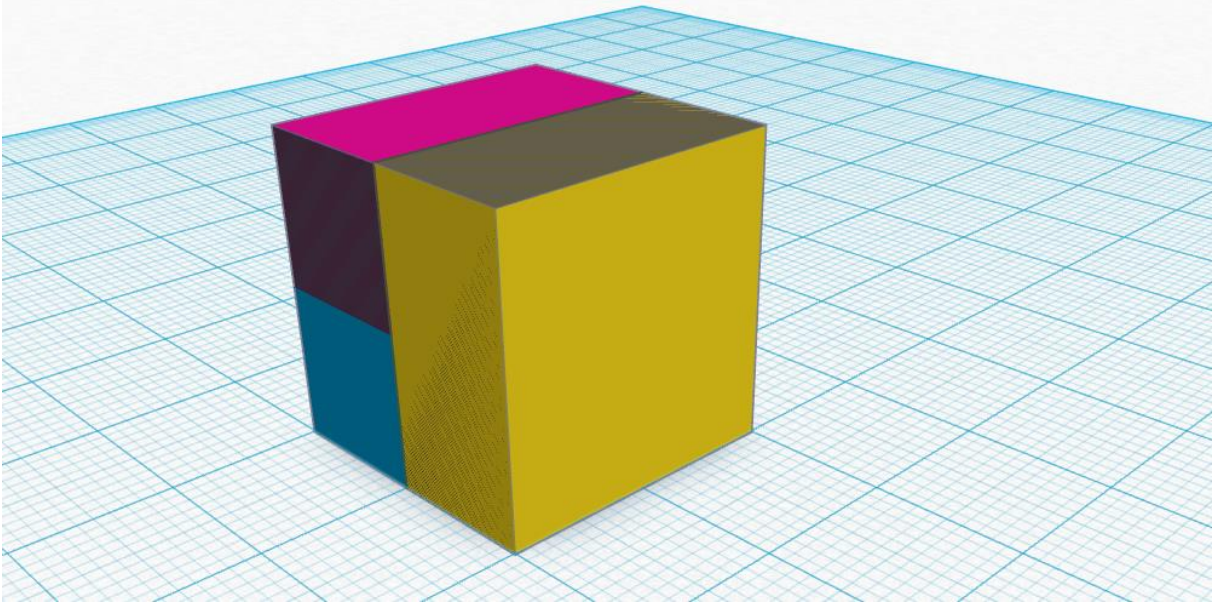
1x1x1 (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 1, 2) (x1, y1, z1), (x2, y2, z2)

2 ve 3 numarada belirtilen boşluk aşağıda sarı olarak gösterilen alanın içindedir.



Şekil 83. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımı Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 numaralı boşluk tanımı)

1 numarada belirtilen boşluk tanımı ise aşağıdaki resimde pembe renkte gösterilen alandır.



Şekil 84. C Tipinde Tanımlı Kutunun Konteynere Yerleştirildikten Sonra Konteyner İçinde Oluşan Boşluk Tanımları Gösteriliyor (1 numaralı olasılık tanımı dahilinde oluşan 1 ve 2 numaralı boşluk tanımları)

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Bu durumda boşluk tanımlarının birleştirilme ihtimali de olabilir.Çünkü birden fazla boşluk tanımı vardır.

Gelinen aşamada konteyner içindeki boşluk tanım sayısı 3 adettir. Sıradaki işlem her boşluk tanımı seçimi ve mümkün ise diğer boşluk tanımları ile birleştirmesi olacaktır.

Boşluk tanımı 1 seçilir, boşluk tanımı 2 ve boşluk tanımı 3 ile karşılaştırılır:

Boşluk tanımı 1'in y1 değeri boşluk tanımı 3'ün y1 değeri ile aynıdır.

Boşluk tanımı 1'in yükseklik boyutu boşluk tanımı 3'ün yükseklik değeri ile aynı değildir.

Boşluk tanımı 3'ün x2 değeri boşluk tanımı 1'in x1 değeri ile aynı değildir.

Boşluk tanımı 1'in z1 değeri boşluk tanımı boşluk tanımı 3'ün z1 değeri ile aynıdır.

Boşluk tanımı 3'ün z2 değeri boşluk tanımı 1'in z2 değeri ile eşittir.

Boşluk tanımı 1 ile boşluk tanımı 3 birleştirilir ve yeni bir boşluk tanımı oluşur.

yeni boşluk tanımına ait genişlik değeri=boşluk tanımı 1 genişlik değeri + boşluk tanımı 3 genişlik değeri =2

yeni boşluk değeri yükseklik değeri = boşluk tanımı 1'e ait yükseklik değeri =1

yeni boşluk değeri derinlik değeri = boşluk tanımı 1'e ait derinlik değeri=2

yeni boşluk tanımına ait x1 değeri= boşluk tanımı 3'e ait x1 değeri=0,

yeni boşluk tanımına ait y1 değeri= boşluk tanımı 3'e ait y1 değeri=1,

yeni boşluk tanımına ait z1 değeri = boşluk tanımı 3'e ait z1 değeri=0,

yeni boşluk tanımına ait x2 değeri = boşluk tanımı 1'e ait x1 değeri + yeni boşluk tanımı genişlik değeri =2,

yeni boşluk tanımına ait y2 değeri = boşluk tanımı 1'e ait y1 değeri + yeni boşluk tanımı yükseklik değeri=2,

yeni boşluk tanımına ait z2 değeri = boşluk tanımı 1'e ait z1 değeri + yeni boşluk tanımı derinlik değeri=2,

Sonuç olarak boşluk tanımı 1 ile boşluk tanımı 3 birleştirilerek yeni bir boşluk tanımı oluşturuldu: $2 \times 1 \times 2$, (0, 1, 0), (2, 2, 2) (x_1, y_1, z_1) , (x_2, y_2, z_2)

Boşluk Tanımlarının Birleştirilme Sona Erer

İşlem 20:Özyinelemeli fonksiyonu ifade eden sanal işlem adımı başlar

Bu aşamada geride kalan kutular boşluk tanımlarına yerleştirilmeleri denenecektir. En fazla kutuyu içeriye alacak olan seçilir.

Olasılık 1 için

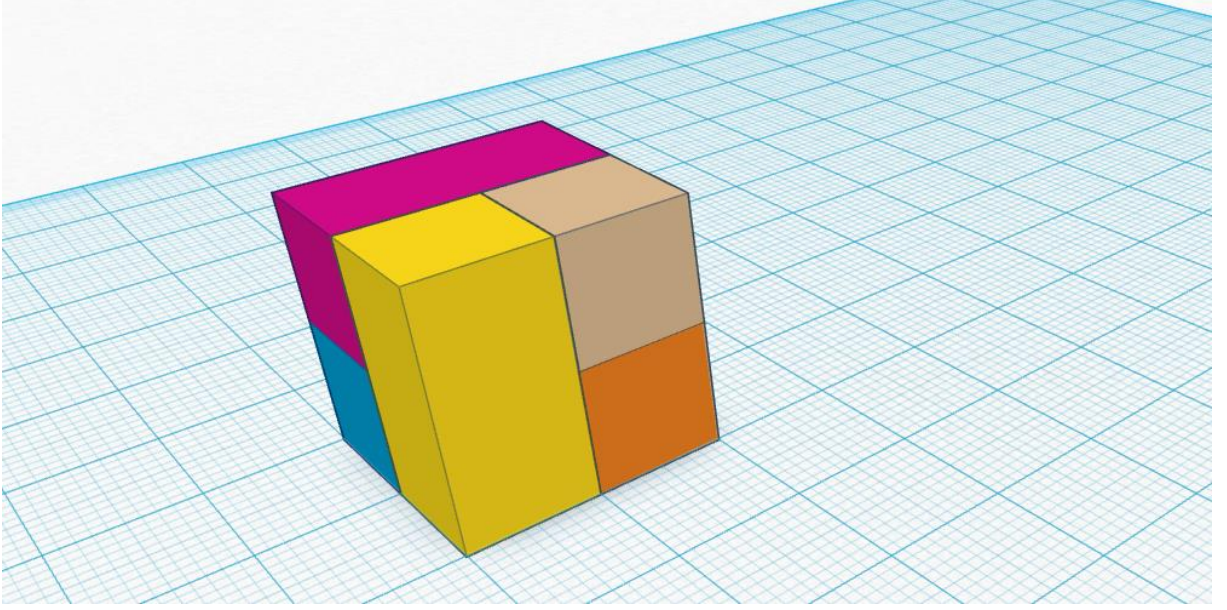
Gelinen adımda konteyner içine yerleşebilecek maksimum ağırlık: 50kg.

Konteyner içindeki boşluk tanımı sayısı 3.

1 Numaralı boşluk tanımı: $1 \times 2 \times 1$ (aşağıdaki resimde sarı olarak renklendirilmiş alan)

2 Numaralı boşluk tanımı: $1 \times 1 \times 1$ (aşağıdaki resimde kahverengi renk ile renklendirilmiş alan)

3 Numaralı boşluk tanımı $1 \times 1 \times 2$ (aşağıdaki resimde pembe renk ile renklendirilmiş alan)



Şekil 85. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor

Kutu: 1x1x1, ağırlık 10kg.

Bütün boşluk tanımları sıralanır. İlk etapta yükseklik değeri en düşük olan seçilmeye çalışılır. Yükseklik değerinden kasıt y_1 koordinat düzleminin değeridir. Konteyner içine yerleştirilecek listeden sıradaki kutu (en büyük hacimde olan) seçilir. Bu kutu boşluk tanımlarına yerleştirilebilir mi bu kontrol edilir.

Listedeki sıradaki kutu seçilir; 1x1x1, 10kg.

Bu kutu küp olduğundan, kutuyu çevirmek gerekir mi sorgusu adımı geçilir.

İşlem 21: Sanal İşlem Başlar - Seviye 2 (Virtual Process Started - Level 2)

Olasılığa ait detaylar:

Konteyner içine yüklenebilecek maksimum ağırlık değeri: 40kg

Konteyner içindeki boşluk tanımları: 3 adet

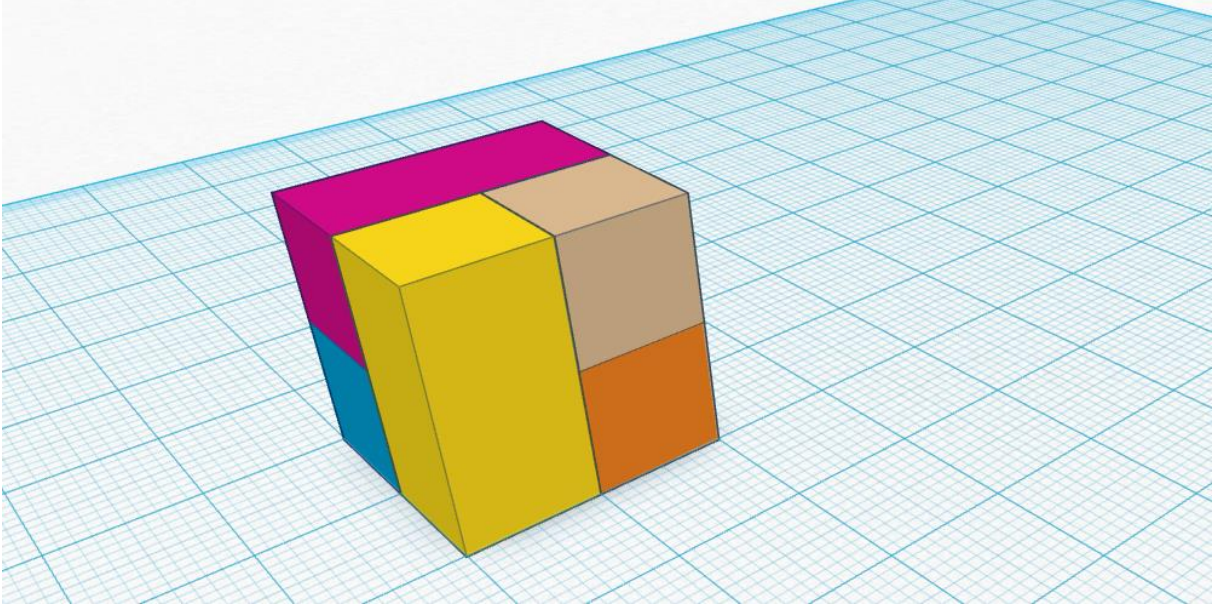
Boşluk tanımı 1: 1x2x1

Boşluk tanımı 2: 1x1x1

Boşluk tanımı 3: 1x1x2

İşlemden ele alınan kutu: 1x1x1 (genişlik, yükseklik, derinlik), 10kg

Boşluk tanımı 1 seçilir. Çünkü y_1 değeri en düşük olan boşluk tanımı boşluk tanımı 1 olarak isimlendirilen boşluk tanımıdır. Seçilen boşluk tanımı aşağıdaki resimde sarı alanla gösterilmiştir:



Şekil 86. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor

Boşluk tanımı 1'e ait genişlik değeri kutunun genişlik değeri ile eşittir. ($1=1$)

Boşluk tanımı 1'e ait yükseklik değeri kutunun yükseklik değeri ile eşit değildir. ($2 \neq 1$)

Boşluk tanımı 1'e ait derinlik değeri kutunun derinlik değeri ile eşittir. ($1=1$)

Kıyaslama sonucunda tek düzlemde farklılığın olduğu gözlenir. (yükseklik ya da y eksenini)

Bu sonuç da 1 olasılığı ifade eder. Yani sıradaki kutu belirtilen alana koyulduğu zaman oluşacak boşluk tanımı olasılığı 1 adettir. Ve Bu olasılık 1 adet boşluk tanımı içerir.

Olasılık 1 dahilinde oluşan 1 numaralı boşluk tanımının tüm değerlerinin hesaplanması :

$$\text{New spaceLength} = \text{oldspacelength} = 1$$

$$\text{New spaceHeight} = \text{oldspaceheight} - \text{boxheight} = 1$$

$$\text{New spacewidth} = \text{oldspacewidth} = 1$$

$$\text{New space x1} = \text{oldspace x1} = 1$$

$$\text{New space y1} = \text{oldspace y1} + \text{boxheight} = 1$$

$$\text{New space z1} = \text{oldspace z1} = 1$$

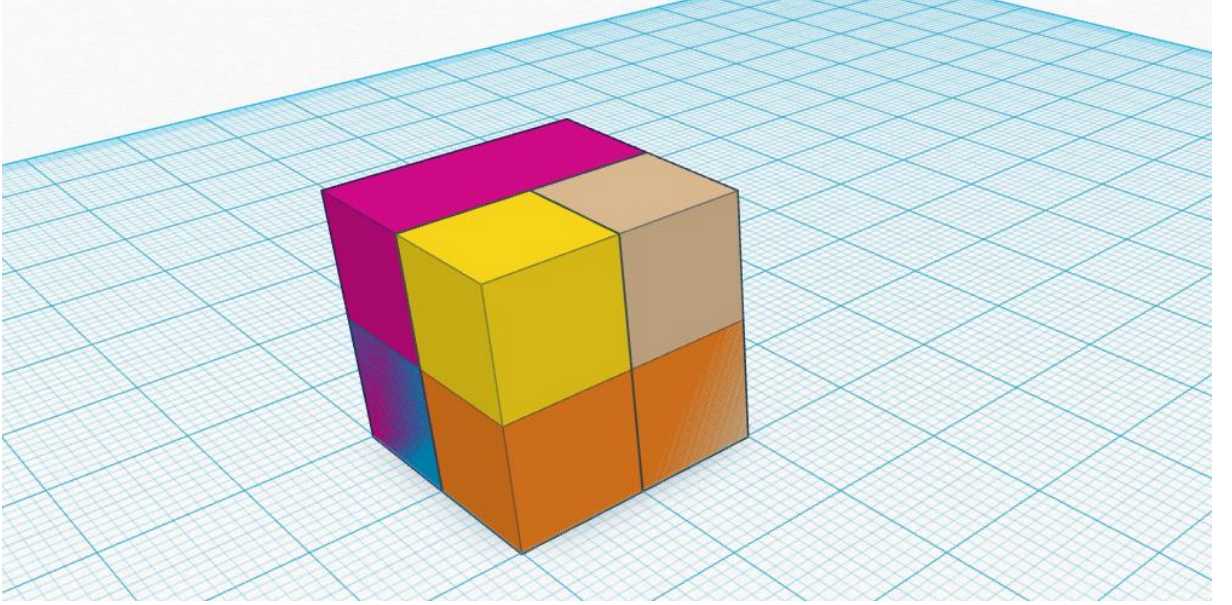
$$\text{New space x2} = \text{New space x1} + \text{New spacelength} = 2$$

New space y2 = New space y1 + New spaceheight = 2

New space z2 = New space z1 + New spacewidth = 2

Sonuç: 1x1x1 (genişlik, yükseklik, derinlik) (1, 1, 1), (2, 2, 2) (x1, y1, z1), (x2, y2, z2)

Bu alan aşağıdaki resimde sarı ile gösterilmiştir.



Şekil 87. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Kutuların yerleşiminden sonra oluşan boşluk tanımları daha büyük boşluk tanımlarının oluşması için daha fazla kutuyu içine alması için birleştirilebilir mi? Bu adımda bu soruya cevap aranır.

Konteyner Olasılığı 1 için

An itibarı ile konteyner içinde boşluk tanım sayısı:3

İlk boşluk tanımı seçilir. Boşluk tanımı 2 ile karşılaştırılır.

Boşluk tanımı 1'e ait y1 değeri boşluk tanımı 2'ye ait y1 değeri ile aynıdır.

Boşluk tanımı 1'e ait yükseklik değeri boşluk tanımı 2'ye ait yükseklik değeri ile aynıdır.

Boşluk tanımı 2'e ait z2 değeri boşluk tanımı 1'ye ait z1 değeri ile aynıdır.

Boşluk tanımı 1'e ait x1 değeri boşluk tanımı 2'ye ait x1 değeri ile aynıdır.

Boşluk tanımı 2'e ait x2 değeri boşluk tanımı 2'ye ait x2 değeri ile aynıdır.

Bu durumda boşluk tanımı 1 ile boşluk tanımı 2 birleştirilebilir. Yeni boşluk tanımına ait değerler hesaplanacak olursa:

New Space Length = Space No. 1's Length = 1

New Space Height = Space No. 1's Height = 1

New Space Width = Space No. 1's Width + Space No. 2's Width = 2

New Space x1 = Space No. 2's x1 = 1

New Space y1 = Space No. 2's y1 = 1

New Space z1 = Space No. 2's z1 = 0

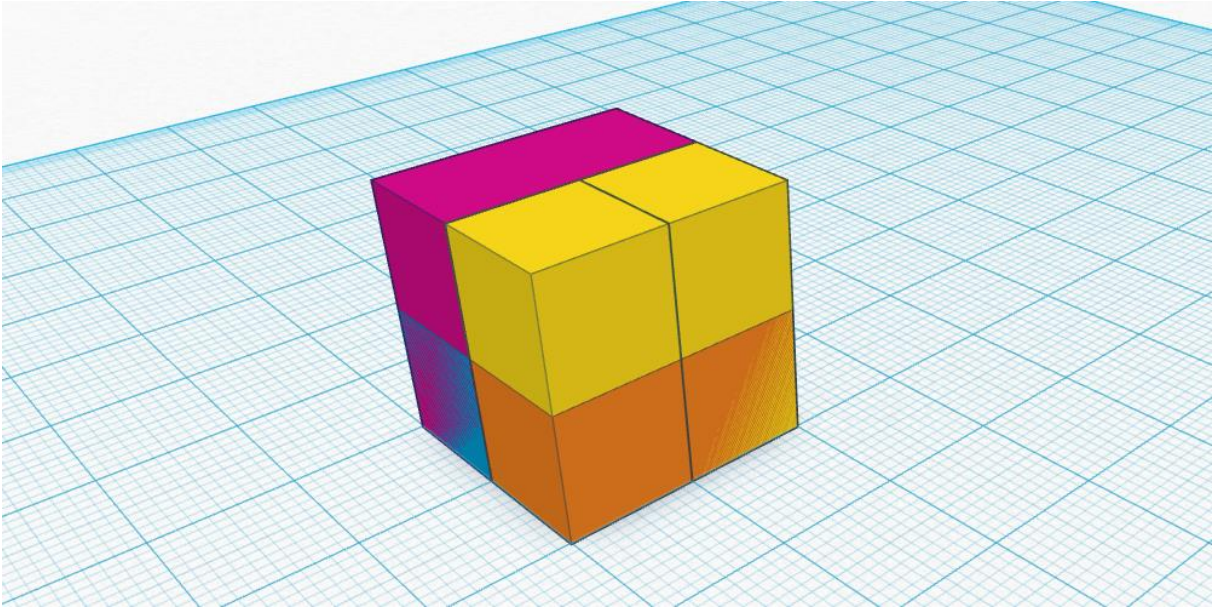
New Space x2 = New Space's x1 + New Space's Length = 2

New Space y2 = New Space's y1 + New Space's Height = 2

New Space z2 = New Space's z1 + New Space's Width = 2

Sonuç: 1x1x2 (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 2) (x1, y1, z1), (x2, y2, z2)

Birleştirilen alan aşağıdaki resimde sarı renk ile gösterilmiştir:



Şekil 88. Konteyner İçinde Oluşan Boşluk Tanımlarının Birleştirilmesi Gösteriliyor

Boşluk Tanımlarının Birleştirilme İşlemi Sona Erer

İşlem 22.: Sanal İşlem Sona Erer - Seviye 2 (Virtual Process Started - Level 2)

Olasılık 2 için

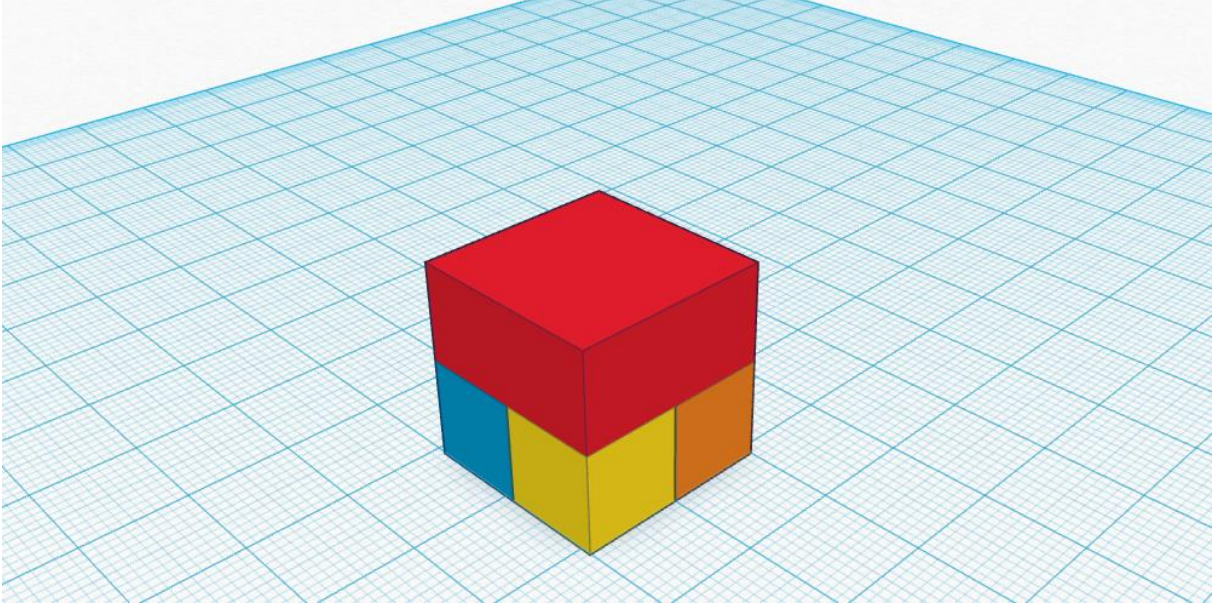
Konteyner içine yerleştirilebilecek maksimum ağırlık: 40 kg.

Olasılık 2 dahilinde boşluk tanım sayısı:2

1 numaralı boşluk tanımı sayısı: 2x1x2 (genişlik, yükseklik, derinlik)

2 numaralı boşluk tanımı sayısı: 1x1x1(genişlik, yükseklik, derinlik)

Aşağıdaki resimde 1 Numaralı boşluk tanımı (2x1x2) kırmızı ile renklendirilmiş iken 2 numaralı boşluk tanımı (1x1x1) ise sarı ile renklendirilmiştir



Şekil 89. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor

Yerleştirilecek kutuya ait tüm değerler: 1x1x1 (genişlik, yükseklik, derinlik), 10kg y₁ değeri en düşük boşluk tanımı seçilir.

Kutu çevrilecek mi? Bu sorgu yapılır. Kutu küp olduğu için bu adım geçilir.

Kutunun konteynere eklendiği varsayılarak aşağıdaki işlemlere başlanır:

İşlem 23:Sanal İşlem Başlar - Seviye 2 (Virtual Process Started - Level 2)

Boşluk tanım değerleri ve kutuya ait tüm değerler tekrar okunur.

1 numaralı boşluk tanımı sayısı: 2x1x2 (genişlik, yükseklik, derinlik)

2 numaralı boşluk tanımı sayısı: 1x1x1(genişlik, yükseklik, derinlik)

Yerleştirilecek kutuya ait tüm değerler: 1x1x1 (genişlik, yükseklik, derinlik), 10kg

y₁ değeri en düşük boşluk tanımı seçilir (2 numaralı boşluk tanımı) Tüm değerlerin eşit olduğu gözlemlenir:

Kutuya ait genişlik değeri ile boşluk tanımı 2'ye ait genişlik değeri.

Kutuya ait yükseklik değeri ile boşluk tanımı 2'ye ait yükseklik değeri.

Kutuya ait derinlik değeri ile boşluk tanımı 2'ye ait derinlik değeri.

Boşluk tanımlarını birleştirme işlemi tüm değerlerin eşit olmasından sebep başladığı gibi sonuçsuz olarak sona erer.

İşlem 24:Sanal İşlem Sona Erer - Seviye 2 (Virtual Process Started - Level 2)

Gelinen aşamada olasılık 1 ve olasılık 2'nin geriye kalan kutular içinden kaç tane içine alabilecekleri hesaplanmış olur.

Olasılık 1'in içine alabileceği kutu sayısı 1

Olasılık 2'nin içine alabileceği kutu sayısı 1.

İşlem 25:Özyinelemeli Fonksiyonu İfade Eden Sanal İşlem Adımı Sona Erer

İşlem 26:Sanal İşlem Sona Erer - Seviye 1 (Virtual Process Started - Level 1)

Tekrar tüm boşluk tanımlarını sıralanır.

y₁ değeri en düşük boşluk tanımı seçilecektir.

Listede sırada bulunan kutu seçilir: 1x1x1 (genişlik, yükseklik, derinlik)

Kutu küp olduğu için kutunun çevrilme işlem adımı geçilir.

Kutunun konteyner içine yerleştirildiği farz edilir

İşlem 27:Sanal İşlem Başlar - Seviye 1 (Virtual Process Started - Level 1)

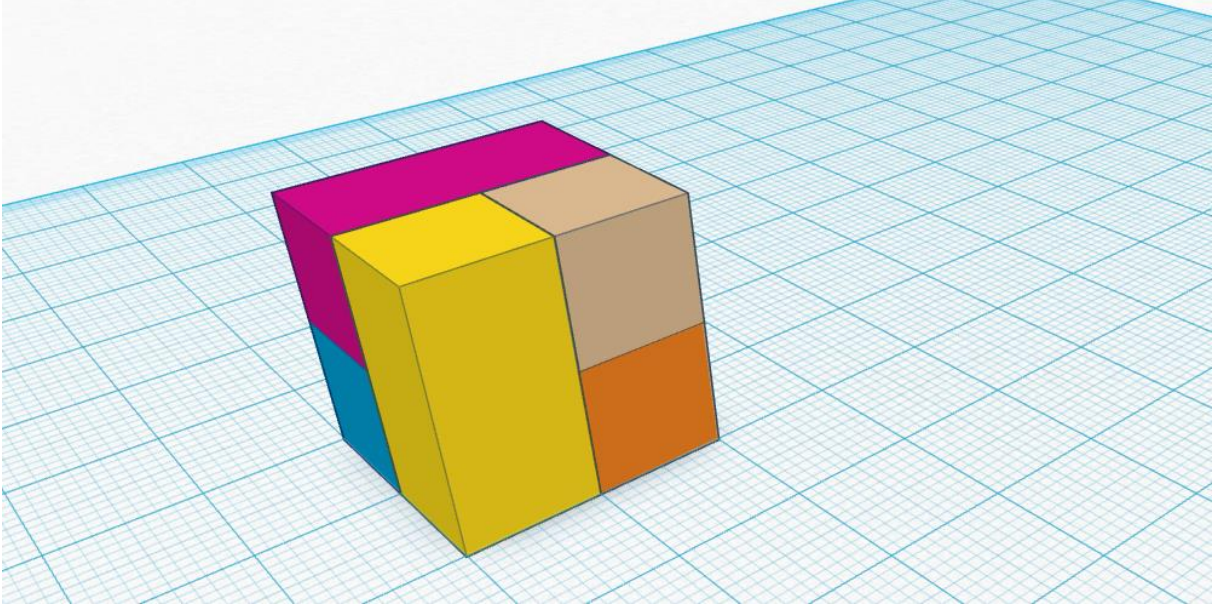
Konteyner içine yüklenebilecek maksimum ağırlık 50 kg'dır.

Konteyner içindeki tüm boşluk tanımları 3 tanedir.

1 Numaralı boşluk tanımı: 1x2x1 (aşağıdaki resimde sarı olarak renklendirilmiş alan)

2 Numaralı boşluk tanımı:1x1x1 (aşağıdaki resimde kahverengi renk ile renklendirilmiş alan)

3 Numaralı boşluk tanımı 1x1x2 (aşağıdaki resimde pembe renk ile renklendirilmiş alan)



Şekil 90. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor

y_1 değeri en düşük boşluk tanımı seçilir: boşluk tanımı 1.

Kutu ile boşluk tanımı 1'e ait tüm değerler kıyaslanır:

Tek düzlemde fark olduğu tespit edilir. (y koordinat düzlemi) Bu fark 1 olasılığı, 1 olasılık da 1 boşluk tanımını ifade eder. Boşluk tanımına ait tüm değerlerin hesaplanması:

$$\text{New spaceLength} = \text{oldspacelength} = 1$$

$$\text{New spaceHeight} = \text{oldspaceheight} - \text{boxheight} = 1$$

$$\text{New spacewidth} = \text{oldspacewidth} = 1$$

$$\text{New space } x1 = \text{oldspace } x1 = 1$$

$$\text{New space } y1 = \text{oldspace } y1 + \text{boxheight} = 1$$

$$\text{New space } z1 = \text{oldspace } z1 = 1$$

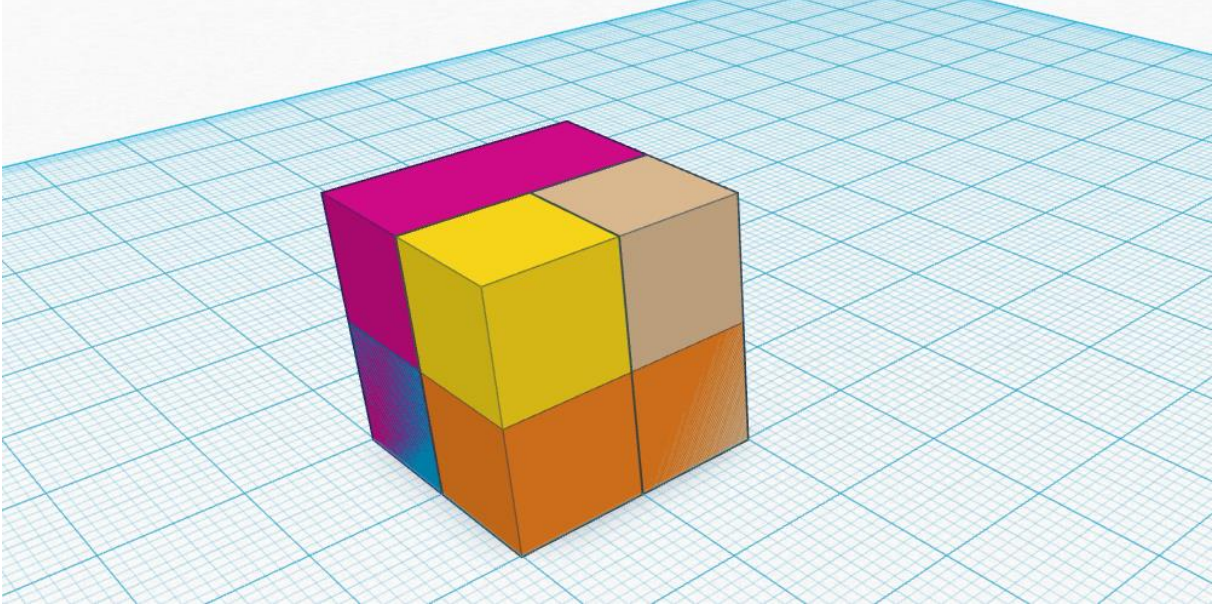
$$\text{New space } x2 = \text{New space } x1 + \text{New spacelength} = 2$$

$$\text{New space } y2 = \text{New space } y1 + \text{New spaceheight} = 2$$

$$\text{New space } z2 = \text{New space } z1 + \text{New spacewidth} = 2$$

Sonuç: 1x1x1 (genişlik, yükseklik, derinlik) (1, 1, 1), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Bu alan aşağıdaki resimde sarı ile gösterilmiştir.



Şekil 91. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Gelinen aşamada konteyner içindeki boşluk tanım sayısı 3 adettir. Sıradaki işlem her boşluk tanımı seçimi ve mümkün ise diğer boşluk tanımları ile birleştirilmesi olacaktır.

Boşluk tanımı 1 seçilir, boşluk tanımı 2 ile karşılaştırılır.

Boşluk tanımı 1 $y_1 =$ boşluk tanımı 2 y_1

Boşluk tanımı 1 yükseklik = boşluk tanımı 2 yükseklik

Boşluk tanımı 2 $z_2 =$ boşluk tanımı 1 z_1

Boşluk tanımı 1 $x_2 =$ boşluk tanımı 1 x_1

Boşluk tanımı 2 $x_2 =$ boşluk tanımı 1 x_2

Boşluk tanımı 1 ile boşluk tanımı 2 birleştirilir. Yeni boşluk tanımının tüm değerlerinin hesaplanması:

New Space Length = Space No. 1's Length = 1

New Space Height = Space No. 1's Height = 1

New Space Width = Space No. 1's Width + Space No. 2's Width = 2

New Space $x_1 =$ Space No. 2's $x_1 = 1$

New Space $y_1 =$ Space No. 2's $y_1 = 1$

New Space z1 = Space No. 2's z1 = 0

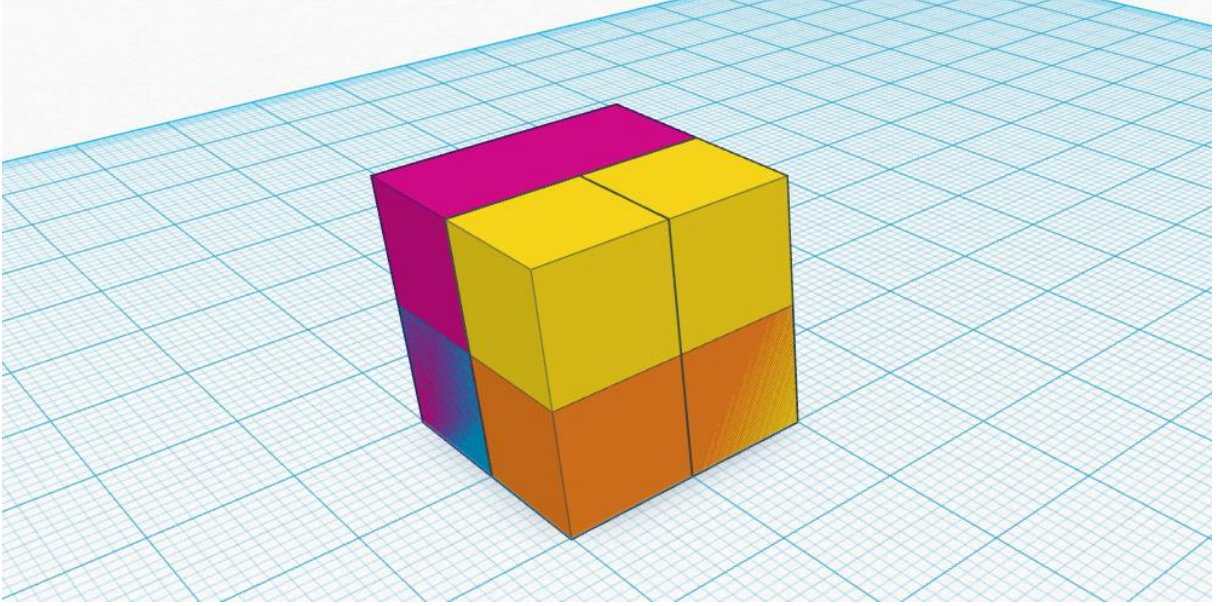
New Space x2 = New Space's x1 + New Space'sLength = 2

New Space y2 = New Space's y1 + New Space'sHeight = 2

New Space z2 = New Space's z1 + New Space'sWidth = 2

Sonuç: 1x1x2 (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 2) (x₁, y₁, z₁), (x₂, y₂, z₂)

Birleştirilen alan aşağıdaki resimde sarı renk ile gösterilmiştir:



Şekil 92. Konteyner İçinde Oluşan Boşluk Tanımlarının Birleştirilmesi Gösteriliyor

Boşluk Tanımlarının Birleştirilme İşlemi Sona Erer

İşlem 28:Sanal İşlem Sona Erer- Seviye 1 (Virtual Process Started - Level 1)

Sonuç olarak, Olasılık 1, listede olan kutulardan 2 kutuyu içine alabilir.

Olasılık 2 için*

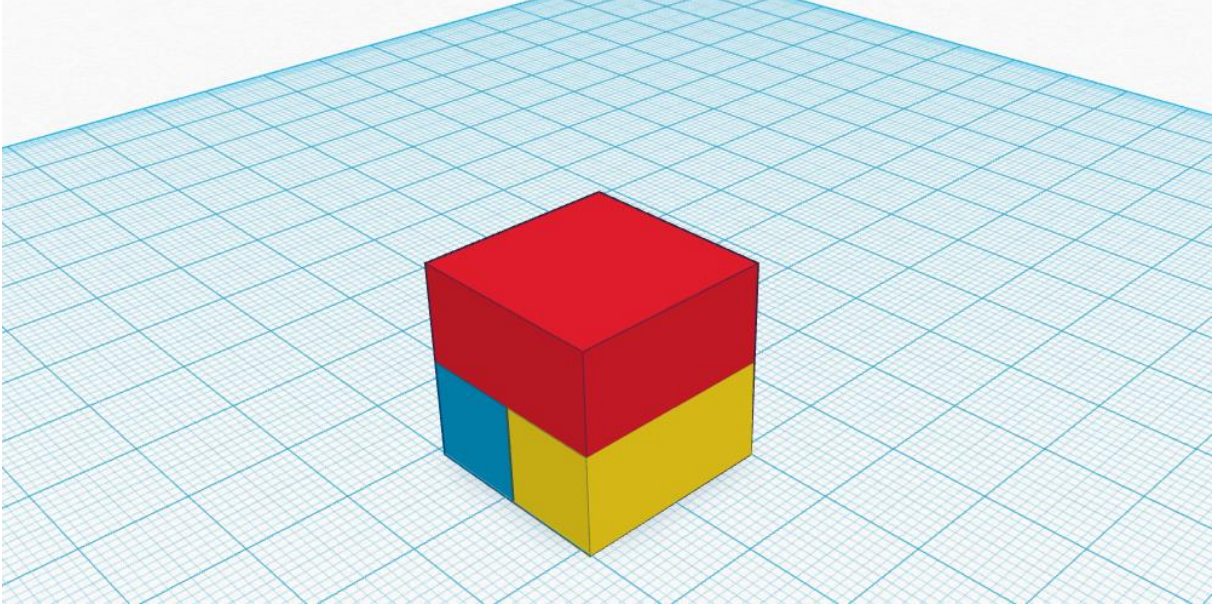
Olasılık 2 dahilinde boşluk tanım sayısı:2

Konteyner içine koyabileceğimiz maksimum ağırlık :50kg

1 Numaralı boşluk tanımı:2x1x2 (genişlik, yükseklik, derinlik)

2 Numaralı boşluk tanımı:1x1x2 (genişlik, yükseklik, derinlik)

Aşağıdaki resimde 1 numaralı boşluk tanımı kırmızı renk ile 2 numaralı boşluk tanımı ise sarı renk ile gösterilmiştir:



Şekil 93. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor

Konteyner içine yerleştirilecek: 1x1x1, 10Kg,

Konteyner içine yerleştirilecek: 1x1x1, 10Kg,

Boşluk tanımlarından y_1 değeri minimum olan boşluk tanımı seçilecektir.

Kutunun çevrilmesi gerekiyor mu sorgusu bu adımda kutu küp olduğu için geçilir.

İşlem 29:Sanal İşlem Başlar - Seviye 1 (Virtual Process Started - Level 1)

Konteyner için yüklenebilecek maksimum ağırlık 50kg

Boşluk tanım sayısı: 2

1 Numaralı boşluk tanımı:2x1x2 (genişlik, yükseklik, derinlik)

2 Numaralı boşluk tanımı:1x1x2 (genişlik, yükseklik, derinlik)

Konteyner içine yerleştirilecek: 1x1x1, 10Kg,

Seçilen boşluk tanımı: 2 numaralı boşluk tanımıdır. çünkü y_1 değeri en düşük olan boşluk tanımı 2 numaralı boşluk tanımıdır.

Boşluk tanımı ile kutunun tüm değerleri karşılaştırılır:

kutuya ait genişlik = boşluk tanımı 2 'ye ait genişlik değeri (1=1)

kutuya ait yükseklik = boşluk tanımı 2 'ye ait yükseklik değeri (1=1)

kutuya ait derinlik \neq boşluk tanımı 2 'ye ait derinlik değeri (1 \neq 2)

Tek düzlemde farklılık (z koordinat düzlemi) 1 adet olasılığı, 1 olasılık da 1 adet boşluk tanımını ifade eder. Boşluk tanımına ait değerlerin hesaplanması:

Bu olasılık tanımı ve olasılık tanımı dahilindeki boşluk tanımının hesaplanması:

$$\text{New spaceLength} = \text{oldspacelength} = 1$$

$$\text{New spaceHeight} = \text{oldspaceheight} = 1$$

$$\text{New spacewidth} = \text{oldspacewidth} - \text{boxwidth} = 1$$

$$\text{New space x1} = \text{oldspace x1} = 1$$

$$\text{New space y1} = \text{oldspace y1} = 0$$

$$\text{New space z1} = \text{oldspace z1} + \text{boxwidth} = 1$$

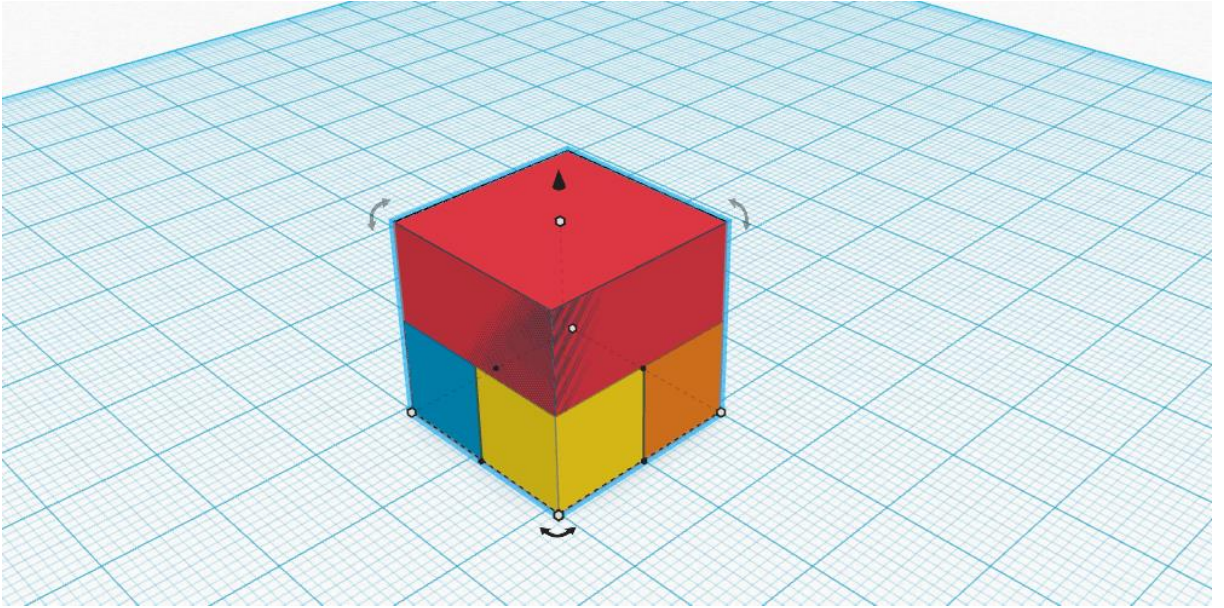
$$\text{New space x2} = \text{New space x1} + \text{New spacelength} = 2$$

$$\text{New space y2} = \text{New space y1} + \text{New spaceheight} = 1$$

$$\text{New space z2} = \text{New space z1} + \text{New spacewidth} = 2$$

Sonuç: 1x1x1 (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 1, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Yeni boşluk tanımı aşağıdaki resimde sarı alan ile gösterilmiştir.



**Şekil 94. Konteyner İçinde Oluşan 2 Boşluk Tanımı Konteyner
Gölgeleştirilmesi İle Gösteriliyor**

Gelinen noktada kutu tüm olasılık değerlerine yerleştirilecek ve hangisi en verimli ise o seçilecektir. Fakat 1 tane olasılık durumu olduğundan,

İşlem 30:Özyinelemeli fonksiyonu ifade eden sanal işlem adımı başlar

Özyinelemeli sana işlem tek olasılıktan dolayı sona erer.

İşlem 31:Özyinelemeli fonksiyonu ifade eden sanal işlem adımı sona erer

Özyinelemeli sana işlem tek olasılıktan dolayı sona erer.

İşlem 32:Sanal İşlem Sona Erer- Seviye 1 (Virtual Process Started - Level 1)

Konteyner içindeki tüm boşluk tanımları tekrar okunur. Bu boşluk tanımlarından y₁ değeri en düşük değere sahip olan boşluk tanımı seçilecektir.

Sıradaki kutuya ait tüm değerler tekrar okunur:1x1x1 (genişlik, yükseklik, derinlik), 10kg

Kutunun çevrilmesi gerekiyor mu? sorgusu kutu küp olduğu için bu adımda geçilir. Kutunun konteyner içine yerleştirildiği farz edilerek işleme başlanır.

İşlem 33: Sanal İşlem Başlar - Seviye 1 (Virtual Process Started - Level 1)

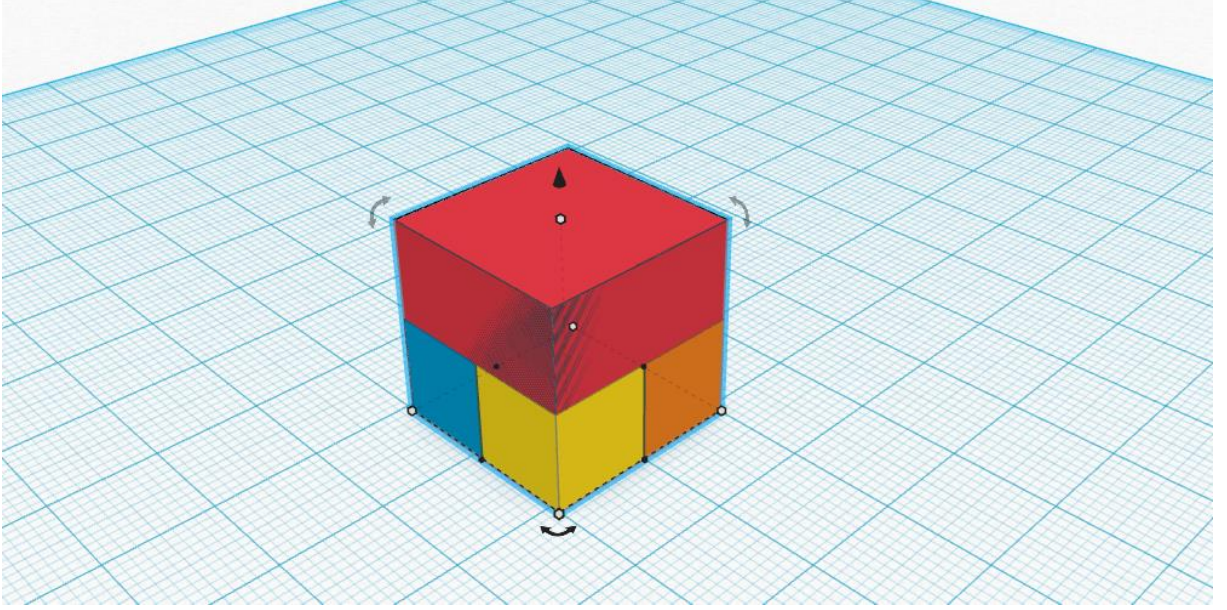
Kutu içine yerleştirilebilecek maksimum ağırlık: 50kg,

Boşluk tanımları:

1 numaralı boşluk tanımı: 1x1x1(genişlik, yükseklik, derinlik)

2 numaralı boşluk tanımı: 2x1x2(genişlik, yükseklik, derinlik)

Aşağıdaki resimde sarı renkle gösterilmiş alan 1 numaralı boşluk tanımını ifade ederken kırmızı renkle gösterilmiş alan ise 2 numaralı boşluk tanımını ifade eder.



**Şekil 95. Konteyner İçinde Oluşan 2 Boşluk Tanımı Konteyner
Gölgelendirilmesi İle Gösteriliyor**

Kutuya dair ölçüler ve özellikler: 1x1x1(genişlik, yükseklik, derinlik), 10kg,

Boşluk tanımlarından y_1 değeri en düşük değere sahip olan boşluk tanımı seçilir:
Seçilen boşluk tanımı 1 numaralı boşluk tanımıdır: 1x1x1(genişlik, yükseklik, derinlik)

Kutu değerleri ile seçilen boşluk tanımı değerleri kıyaslanır:

kutuya ait genişlik = boşluk tanımı 2 'ye ait genişlik değeri (1=1)

kutuya ait yükseklik = boşluk tanımı 2 'ye ait yükseklik değeri (1=1)

kutuya ait derinlik= boşluk tanımı 2 'ye ait derinlik değeri (1=1)

İşlem 34:Sanal İşlem Sona Erer - Seviye 1 (Virtual Process Started - Level 1)

2 numaralı olasılık dahilinde konteyner içine listeden girebilecek kutu sayısı 2 olarak hesaplanır.

İşlem 35: Sanal İşlem Başlar (Virtual Process Ended)

İşlem 36:Konteyner İçine Yerleştirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seçilir

Seçilen kutu, 6 numaralı kutudur. Ölçüleri: 1x1x1 (genişlik, yükseklik, derinlik), ağırlık 10kg.

İşlem 37:Seçilen Kutunun Döndürülmesi Gerekip Gerekmediği Sorgulanır

Bu sorgu bu adımda geçilir, çünkü kutu küp şeklindedir.

İşlem 38:Seçilen Kutu Konteyner İçine Yerleştiriliyor

6 numaralı kutunun konteyner içine yerleştirildiği farz edilerek işleme başlanır. Bu arada bu kutu ağırlık yönü ile ele alındığında konteyner içine koyulmasına bir engel yoktur.

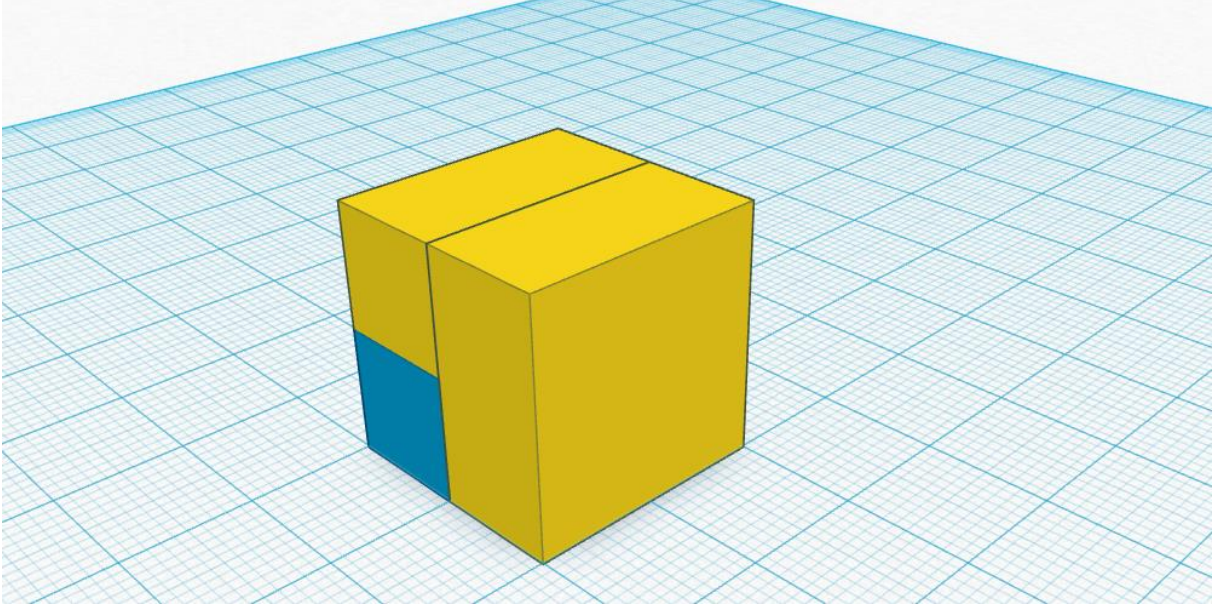
İşlem 39:Konteyner İçindeki Boşluk Tanımlamaları Okunur

Konteyner içindeki boşluk tanımları:

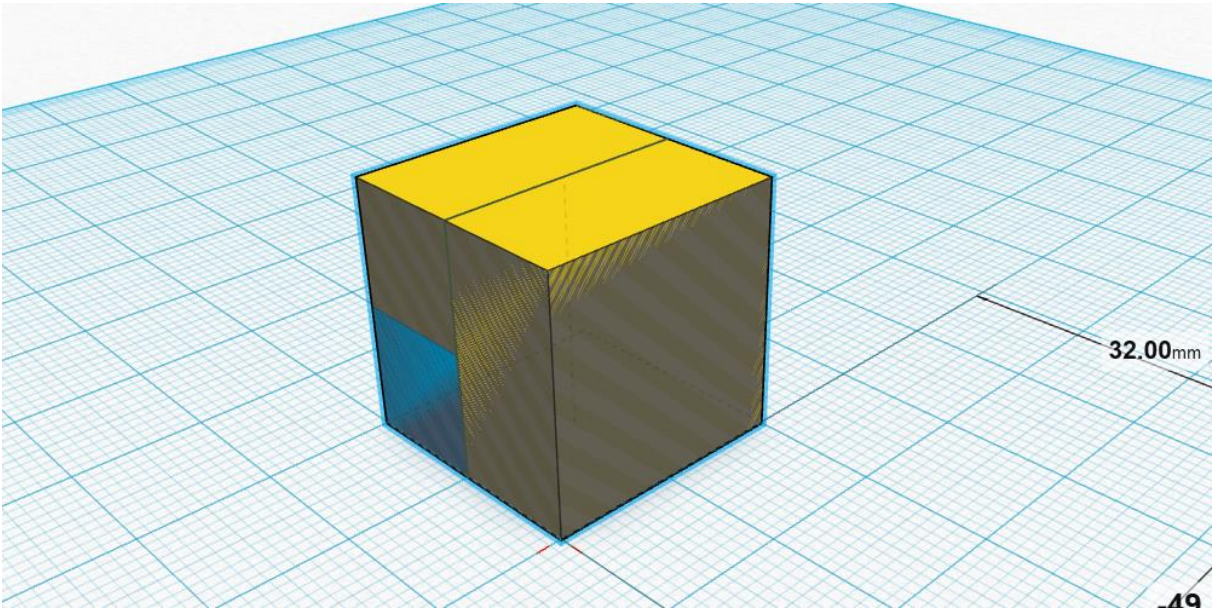
1 numaralı boşluk tanımı: $1 \times 2 \times 2$ (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 1, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

2 numaralı boşluk tanımı: $1 \times 1 \times 2$ (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 1, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

1 ve 2 numaralı boşluk tanımları aşağıdaki şekilde gösterilmiştir:



Şekil 96. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor



Şekil 97. Konteyner İçinde Oluşan 2 Boşluk Tanımı Konteyner Gölgeleştirilmesi İle Gösteriliyor

Mavi renkte olan alan c tipli kutu üzerindeki boşluk tanımı 2 numaralı boşluk tanımı iken diğeri 1 numaralı boşluk tanımıdır. Problemin çözüm prensibine dayanarak y_1 değeri düşük olan boşluk tanımı seçilecektir. Bu nedenle seçilen boşluk tanımı 1 numaralı boşluk tanımıdır. 1 Numaralı boşluk tanımının tüm ölçüt değerleri 6 numaralı kutu ile karşılaştırılır.

Kutuya ait genişlik = boşluk tanımı 1 'ye ait genişlik değeri (1=1)

Kutuya ait yükseklik \neq boşluk tanımı 1 'ye ait yükseklik değeri (1 \neq 2)

Kutuya ait derinlik \neq boşluk tanımı 1 'ye ait derinlik değeri (1 \neq 2)

Kıyaslama yapıldığında iki düzlemde farklılık olduğu tespit edilir. (y ve z kordinat düzlemleri) Bu durum iki farklı olasılığı, bu iki farklı olasılık da ikişer tane farklı boşluk tanımını ifade eder. Bu yeni boşluk tanımları kutunun aşağıdaki kordinatlara koyulmasından sonra oluşan boşluk tanım olasılıklarıdır:

(1, 0, 0), (2, 1, 1) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

Bu durumda kutu içine yerleştirilebilecek ağırlık birimi: 40 kg'dır.

1 Numaralı Olasılık

1 Numaralı olasılık altında tanımlı 1 numaralı boşluk tanımının hesaplanması

New space 1 Length = oldspacelength = 1

New space 1 Height = oldspaceheight - boxheight = 1

New space 1 width = oldspacewidth = 2

New space 1 x1 = oldspace x1 = 0

New space 1 y1 = oldspace y1 + boxheight = 1

New space 1 z1 = oldspace z1 = 0

New space 1 x2 = New space 1 x1 + New space 1 length = 2

New space 1 y2 = New space 1 y1 + New space 1 height = 2

New space 1 z2 = New space 1 z1 + New space 1 width = 2

Sonuç: 1x1x2 (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 2, 2) $(x_1, y_1, z_1), (x_2, y_2, z_2)$

1 Numaralı olasılık altında tanımlı 2 numaralı boşluk tanımının hesaplanması

New space 2 Length = oldspacelength = 1

New space 2 Height = boxheight = 1

New space 2 width = oldspacewidth - boxwidth = 1

New space 2 x1 = oldspace x1 = 1

New space 2 y1 = oldspace y1 = 0

New space 2 z1 = oldspace z1 + boxwidth = 1

$$\text{New space 2 } x_2 = \text{New space 2 } x_1 + \text{New space 2 length} = 2$$

$$\text{New space 2 } y_2 = \text{New space 2 } y_1 + \text{New space 2 height} = 1$$

$$\text{New space 2 } z_2 = \text{New space 2 } z_1 + \text{New space 2 width} = 2$$

Sonuç: 1x1x1 (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 1, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

2 Numaralı Olasılık

2 Numaralı olasılık altında tanımlı 1 numaralı boşluk tanımının hesaplanması

$$\text{New space 1 Length} = \text{oldspacelength} = 1$$

$$\text{New space 1 Height} = \text{oldspaceheight} = 2$$

$$\text{New space 1 width} = \text{oldspacewidth} - \text{boxwidth} = 1$$

$$\text{New space 1 } x_1 = \text{oldspace } x_1 = 1$$

$$\text{New space 1 } y_1 = \text{oldspace } y_1 = 0$$

$$\text{New space 1 } z_1 = \text{oldspace } z_1 + \text{boxwidth} = 1$$

$$\text{New space 1 } x_2 = \text{New space 1 } x_1 + \text{New space 1 length} = 2$$

$$\text{New space 1 } y_2 = \text{New space 1 } y_1 + \text{New space 1 height} = 2$$

$$\text{New space 1 } z_2 = \text{New space 1 } z_1 + \text{New space 1 width} = 2$$

Sonuç: 1x2x1 (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

2 Numaralı olasılık altında tanımlı 2 numaralı boşluk tanımının hesaplanması

$$\text{New space 2 Length} = \text{oldspacelength} = 1$$

$$\text{New space 2 Height} = \text{oldspaceheight} - \text{boxheight} = 1$$

$$\text{New space 2 width} = \text{boxwidth} = 1$$

$$\text{New space 2 } x_1 = \text{oldspace } x_1 = 1$$

$$\text{New space 2 } y_1 = \text{oldspace } y_1 + \text{boxheight} = 1$$

$$\text{New space 2 } z_1 = \text{oldspace } z_1 = 0$$

$$\text{New space 2 } x_2 = \text{New space 2 } x_1 + \text{New space 2 length} = 2$$

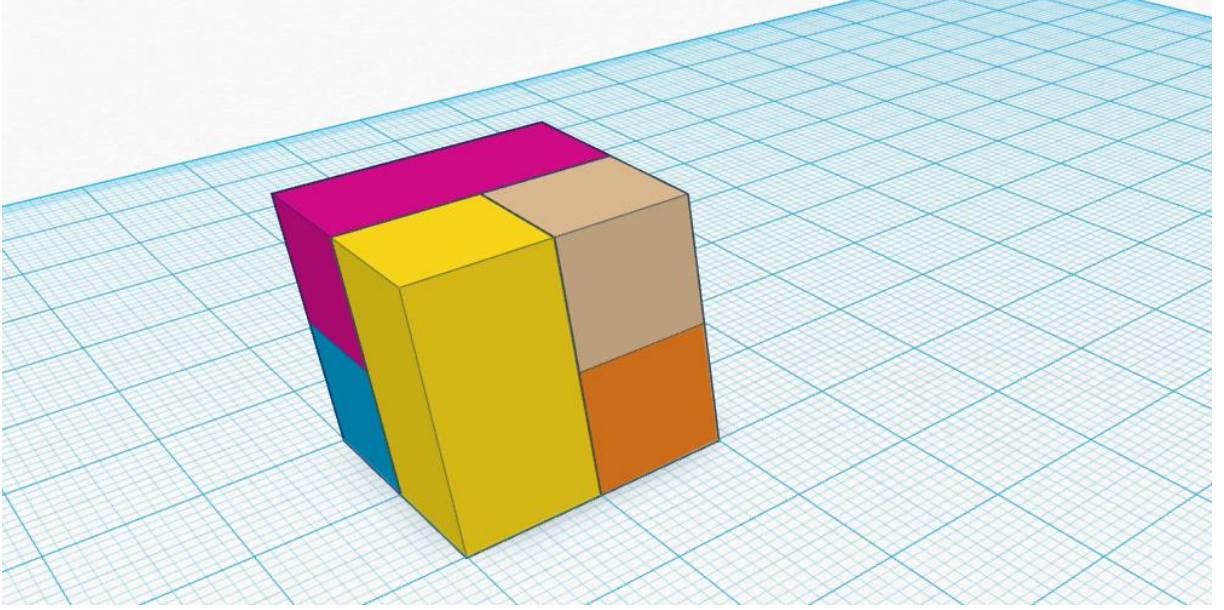
$$\text{New space 2 } y_2 = \text{New space 2 } y_1 + \text{New space 2 height} = 2$$

$$\text{New space 2 } z_2 = \text{New space 2 } z_1 + \text{New space 2 width} = 1$$

Sonuç: 1x1x1 (genişlik, yükseklik, derinlik) (1, 1,0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Bu adımda bu boşluk tanımlarının birleştirilme aşamasına geçilir. Bu işlemin amacı daha büyük boşluk tanımları oluşturabilmek, bu sayede listede bulunan kutulardan daha fazlasını içeri alabilmek.



Şekil 98. Konteyner İçinde Oluşan 3 Boşluk Tanımı Gösteriliyor

Yukarıdaki resimde pembe renkle gösterilmiş alan boşluk tanımı 1'i ifade etmektedir. Sarı renkle gösterilmiş alan ise boşluk tanımı 3'ü ifade etmektedir. Kahve rengi alan ise boşluk tanımı 4'ü ifade etmektedir. Kutunun bulunduğu alan ise boşluk tanımı 2'yi ifade etmektedir.

1 Numaralı ihtimal dahilinde olabilecek konteyner

Toplam boşluk tanımı 3 adettir. Bunlar sarı, pembe ve kahverengi renkleri ile yukarıdaki şekilde gösterilmiş alanlardır.

Boşluk tanımı 1 seçilir, boşluk tanımı 3 ile karşılaştırılır.

$$\text{Boşluk tanımı 1 } y_1 = \text{boşluk tanımı 3 } y_1$$

$$\text{Boşluk tanımı 1 yükseklik} = \text{boşluk tanımı 3 yükseklik}$$

$$\text{Boşluk tanımı 3 } x_2 = \text{boşluk tanımı 1 } x_1$$

$$\text{Boşluk tanımı 1 } z_1 = \text{boşluk tanımı 3 } z_1$$

$$\text{Boşluk tanımı 3 } z_2 = \text{boşluk tanımı 1 } z_2$$

Boşluk tanımı 1 ile boşluk tanımı 3 birleştirilir. Yeni boşluk tanımının tüm değerlerinin hesaplanması:

$$\text{New Space Length} = \text{Space No. 1's Length} + \text{Space No. 3's Length} = 2$$

$$\text{New Space Height} = \text{Space No. 1's Height} = 1$$

$$\text{New Space Width} = \text{Space No. 1's Width} = 2$$

$$\text{New Space } x_1 = \text{Space No. 3's } x_1 = 0$$

$$\text{New Space } y_1 = \text{Space No. 3's } y_1 = 1$$

$$\text{New Space } z_1 = \text{Space No. 3's } z_1 = 0$$

$$\text{New Space } x_2 = \text{New Space's } x_1 + \text{New Space's Length} = 2$$

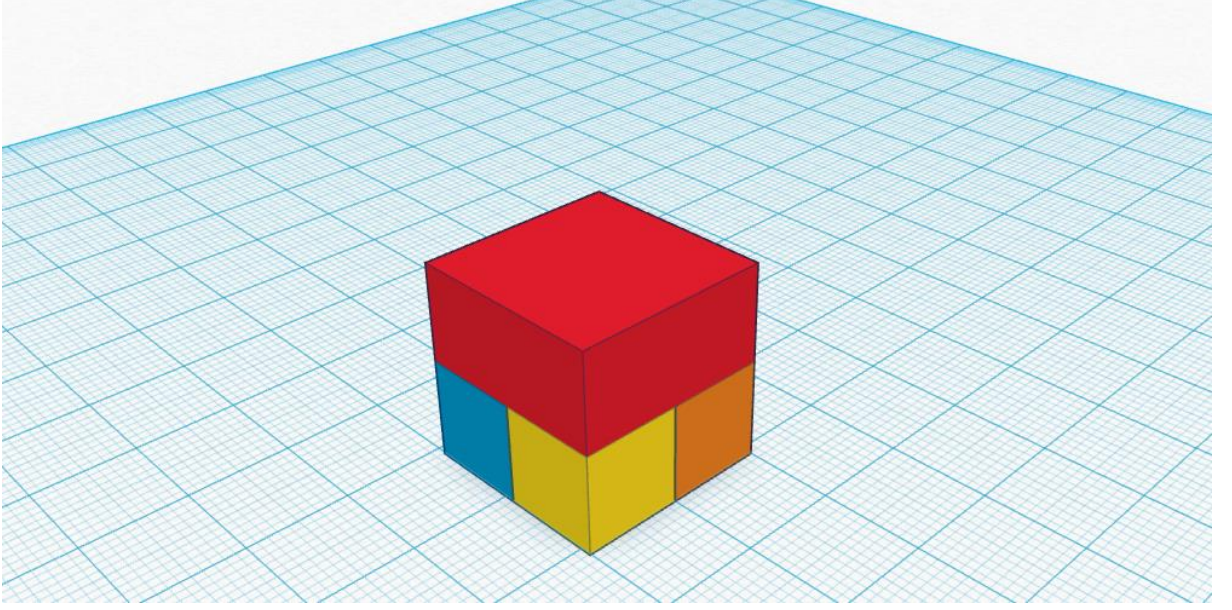
$$\text{New Space } y_2 = \text{New Space's } y_1 + \text{New Space's Height} = 2$$

$$\text{New Space } z_2 = \text{New Space's } z_1 + \text{New Space's Width} = 2$$

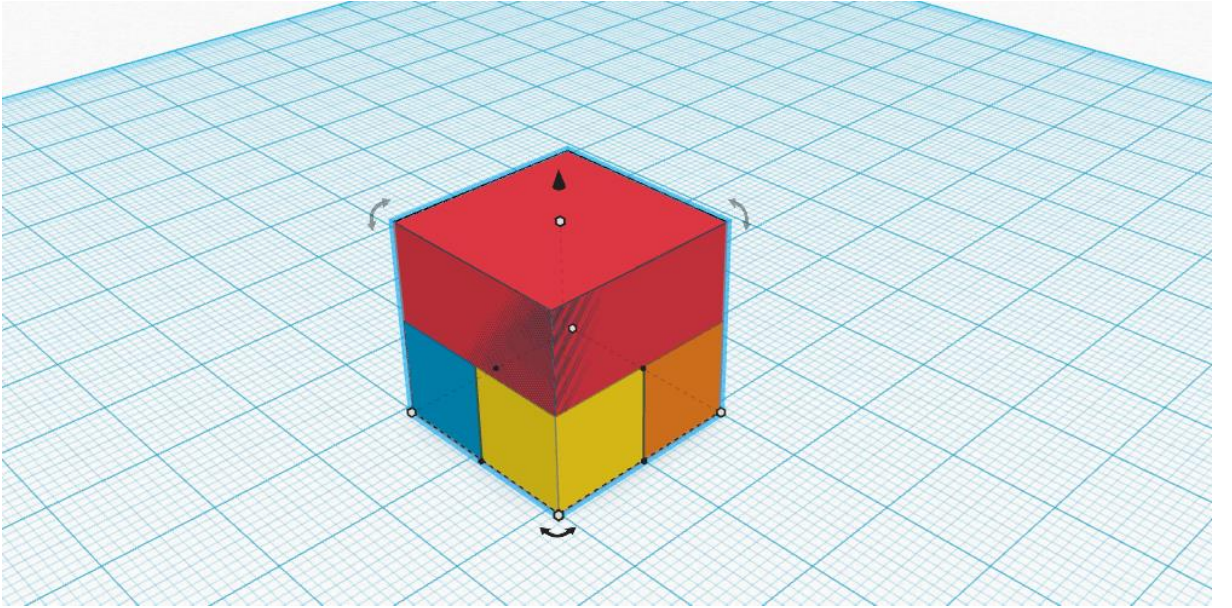
Sonuç: 2x1x2 (genişlik, yükseklik, derinlik) (0, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

1 ve 3 numaralı boşluk tanımının birleşimi sonucunda oluşan yeni boşluk tanımı aşağıdaki resimde kırmızı alanda gösterilmiştir:

Diğer alanlar da renkler ile açıklanacak olursa, mavi renk ile gösterilen c tipli 3 numaralı kutudur, turuncu renkle gösterilen d tipli 6 numaralı kutudur, sarı renk ile gösterilen ise bir önceki adımda oluşan 4 numaralı boşluk tanımıdır.



Şekil 99. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor



Şekil 100. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor

Bu adımdan sonra kutu listesinden kutuları konteynere koymuş farzederek hesaplamalar yapacağız. Hangi seçenek en fazla kutuyu içeri alabiliyor bunu hesaplayacağız.

İşlem 40: Sanal İşlem Başlar (Virtual Process Started)

Konteyner içine girebilecek ağırlık: 40kg

Konteyner içindeki boşluk tanımı: 3

1 numaralı boşluk tanımı :1x2x1

2 numaralı boşluk tanımı: 1x1x1

3 numaralı boşluk tanımı:1x1x2

Kutu:1x1x1, 10kg

Kutu küp olduğu için kutu çevrilmeli mi sorgusu bu adımda geçilir.

İşlem 41: Sanal İşlem Başlar - Seviye 1 (Virtual Process Started - Level 1)

Boşluk tanımlarından y_1 değeri en düşük olan seçilir. Bu durumda 1 numaralı boşluk tanımı seçilir.

Boşluk tanımına ait değerler kutu değerleri ile kıyaslanır:

Kutuya ait genişlik = boşluk tanımı 1 'ye ait genişlik değeri (1=1)

Kutuya ait yükseklik \neq boşluk tanımı 1 'ye ait yükseklik değeri (1 \neq 2)

Kutuya ait derinlik = boşluk tanımı 1 'ye ait derinlik değeri (1=1)

Gelinen noktada tek düzlemde farklılık gözlemlenir. Bu da y kordinat düzlemidir.

Bu durumda bir olasılık ve bu olasılığın içerdiği bir boşluk tanımı oluşur.

Olasılık 1

Olasılık 1 dahilindeki yeni boşluk tanımı

New spaceLength = oldspacelength = 1

New spaceHeight = oldspaceheight - boxheight = 1

New spacewidth = oldspacewidth = 1

New space x1 = oldspace x1 = 1

New space y1 = oldspace y1 + boxheight = 1

New space z1 = oldspace z1 = 1

New space x2 = New space x1 + New spacelength = 2

New space y2 = New space y1 + New spaceheight = 2

New space z2 = New space z1 + New spacewidth = 2

Sonuç: 1x1x1 (genişlik, yükseklik, derinlik) (1, 1,1), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Bu boşluk tanımı ile beraber diğer iki boşluk tanımı daha büyük alan yaratmak için birleştirilebilir mi? Bu sorgu için işlemler başlar:

Boşluk tanımı1 seçilir, boşluk tanımı 2 ile mukayese edilir:

Boşluk tanımı 1 $y_1 =$ boşluk tanımı 2 y_1

Boşluk tanımı 1 yükseklik = boşluk tanımı 2 yükseklik

Boşluk tanımı 2 $z_2 =$ boşluk tanımı 1 z_1

Boşluk tanımı 1 $x_1 =$ boşluk tanımı 2 x_1

Boşluk tanımı 2 $x_2 =$ boşluk tanımı 1 x_2

Boşluk tanımı 1 ile boşluk tanımı 2 birleştirilir. Yeni boşluk tanımının tüm değerlerinin hesaplanması:

New Space Length = Space No. 1's Length = 1

New Space Height = Space No. 1's Height = 1

New Space Width = Space No. 1's Width + Space No. 2's Width = 2

New Space $x_1 =$ Space No. 2's $x_1 = 1$

New Space $y_1 =$ Space No. 2's $y_1 = 1$

New Space $z_1 =$ Space No. 2's $z_1 = 0$

New Space $x_2 =$ New Space's $x_1 +$ New Space's Length = 2

New Space $y_2 =$ New Space's $y_1 +$ New Space's Height = 2

New Space $z_2 =$ New Space's $z_1 +$ New Space's Width = 2

Sonuç: $1 \times 1 \times 2$ (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Boşluk Tanımlarının Birleştirilme İşlemi Sona Erer

İşlem 42: Sanal İşlem Sona Erer- Seviye 1 (Virtual Process Ended- Level 1)

Sonuç olarak bu boşluk tanımının konteyner içine alabileceği kutu sayısı 1 adettir.

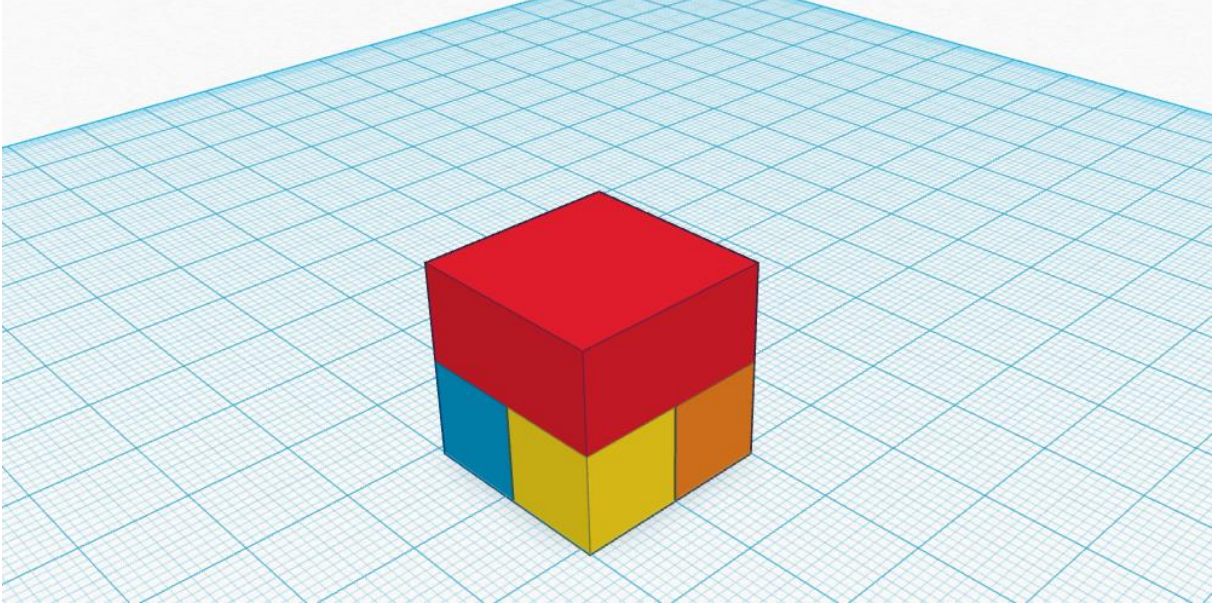
Olasılık 2

Olasılık 2 dahilindeki boşluk tanımları

1 Numaralı boşluk tanımı: $2 \times 1 \times 2$ (genişlik, yükseklik, derinlik)

2 Numaralı boşluk tanımı: $1 \times 1 \times 1$ (genişlik, yükseklik, derinlik)

Aşağıdaki resimde kırmızı olarak gösterilen alan 1 numaralı boşluk tanımını ifade ederken, sarı renkte gösterilen alan ise 2 numaralı boşluk tanımını ifade etmektedir.



Şekil 101. Konteyner İçinde Oluşan 2 Boşluk Tanımı Gösteriliyor

Konteyner içine alınabilecek ağırlık:40kg

Kutu:1x1x1, 10 kg

Boşluk tanımlarından y_1 değeri en düşük olan seçilir: Seçilen 2 numaralı boşluktur.

İşlem 43: Seçilen Kutunun Döndürülmesi Gerekip Gerekmediği Sorgulanır

Kutu çevrilecek mi sorgusu bu adımda geçilir çünkü kutunun şekli küptür.

İşlem 44:Sanal İşlem Başlar - Seviye 1 (Virtual Process Started - Level 1)

Boşluk tanımı ile seçili kutunun tüm ölçütleri karşılaştırılır. Karşılaştırma sonucu tüm değerler birbirine eşit olduğu için bu boşluk tanımına girebilecek kutu sayı adeti 1 tanedir.

Olasılık 1 dahilinde oluşan boşluk tanımına girebilecek kutu sayısı ise 2 adet idi.

İşlem 45: Sanal İşlem Sona Erer- Seviye 1 (Virtual Process Ended- Level 1)

İşlem 46:Sanal İşlem Sona Erer (Virtual Process Ended)

İşlem 47:Konteyner İçine Yerleştirilecek Kutu Listesinin İlk Sırasında Olan Kutu Seçilir

Konteyner içene yerleşecek kutu listesinde bulunan sıradaki kutu seçilir.

Seçilen kutu 7 numaralı D tipinde kutudur: 1x1x1(genişlik, yükseklik, derinlik), 10kg.

İşlem 48:Seçilen Kutunun Döndürülmesi Gerekip Gerekmediği Sorgulanır

Kutunun çevrilmesi gerekiyor mu sorgusu bu adımda geçilir çünkü kutunun şekli küptür.

İşlem 49: Seçilen Kutu Konteyner İçine Yerleştiriliyor

Kutunun konteyner içine yerleştirildiği varsayılarak hesaplamalara geçilir.

Kutu aşağıdaki pozisyona yerleştirilir:

(1, 0, 1), (2, 1, 2) (x₁, y₁, z₁), (x₂, y₂, z₂)

Konteyner içine yerleştirilebilecek ağırlık miktarı bu adımdan sonra 30'dur.

İşlem 50:Konteyner İçindeki Boşluk Tanımlamaları Okunur

Konteyner içindeki boşluk tanımları:

1 Numaralı boşluk tanımı:1x2x1 (genişlik, yükseklik, derinlik) (1, 0, 1), (2, 2, 2)
(x₁, y₁, z₁), (x₂, y₂, z₂)

2 Numaralı boşluk tanımı:1x1x1 (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 1)
(x₁, y₁, z₁), (x₂, y₂, z₂)

3 Numaralı boşluk tanımı:1x1x2 (genişlik, yükseklik, derinlik) (0, 1, 0), (1, 2, 2)
(x₁, y₁, z₁), (x₂, y₂, z₂)

Boşluk tanımlarından y₁ değeri en düşük olan seçilir: 2 numaralı boşluktur. Seçilen boşluk tanımı ile kutunun ölçütleri karşılaştırılır. Karşılaştırma sonucu tek düzlemde fark olduğu için 1 tane olasılık, 1 olasılık için ise 1 tane boşluk tanımı oluşur.

Kutunun yerleştirilmesinden sonra oluşan bu boşluk tanımının tüm değerlerinin hesabı aşağıdaki gibidir:

New spaceLength = oldspacelength = 1

New spaceHeight = oldspaceheight - boxheight = 1

New spacewidth = oldspacewidth = 1

New space x1 = oldspace x1 = 1

New space y1 = oldspace y1 + boxheight = 1

New space z1 = oldspace z1 = 1

New space x2 = New space x1 + New spacelength = 2

New space y2 = New space y1 + New spaceheight = 2

New space z2 = New space z1 + New spacewidth = 2

Sonuç:1x1x1 (genişlik, yükseklik, derinlik) (1, 1, 1), (2, 2, 2) (x₁, y₁, z₁), (x₂, y₂, z₂)

Boşluk Tanımlarının Birleştirilme İşlemi Başlar

Gelinen durumda konteyner içindeki boşluk tanım sayısı:3 adettir. Boşluk tanımı 1 seçilir 2 numaralı boşluk tanımı ile mukayese edilir :

Boşluk tanımı 1 $y_1 =$ boşluk tanımı 2 y_1

Boşluk tanımı 1 yükseklik = boşluk tanımı 2 yükseklik

Boşluk tanımı 2 $z_2 =$ boşluk tanımı 1 z_1

Boşluk tanımı 1 $x_1 =$ boşluk tanımı 2 x_1

Boşluk tanımı 2 $x_2 =$ boşluk tanımı 1 x_2

Boşluk tanımı 1 ile boşluk tanımı 2 birleştirilir. Yeni boşluk tanımının tüm değerlerinin hesaplanması:

Boşluk tanımı 1 ile boşluk tanımı 2 birleştirilir. Yeni boşluk tanımının tüm değerlerinin hesaplanması:

New Space Length = Space No. 1's Length = 1

New Space Height = Space No. 1's Height = 1

New Space Width = Space No. 1's Width + Space No. 2's Width = 2

New Space $x_1 =$ Space No. 2's $x_1 = 1$

New Space $y_1 =$ Space No. 2's $y_1 = 1$

New Space $z_1 =$ Space No. 2's $z_1 = 0$

New Space $x_2 =$ New Space's $x_1 +$ New Space'sLength = 2

New Space $y_2 =$ New Space's $y_1 +$ New Space'sHeight = 2

New Space $z_2 =$ New Space's $z_1 +$ New Space'sWidth = 2

Sonuç:1x1x2 (genişlik, yükseklik, derinlik) (1, 1, 0), (2, 2, 2) (x_1, y_1, z_1), (x_2, y_2, z_2)

Boşluk Tanımlarının Birleştirilme İşlemi Sona Erer

İşlem 51: Konteyner İçine Yerleştirilecek Kutu Listesinde Kutu Kalmadığı Tespit Edilir

İşlem 52:Tüm Sonuçlar Ekranaya Yazdırılır

İşlem 53: Konteyner İçine Yerleştirilemeyen Kutular

10.Sıradaki kutu: G tipinde tanımlanan 10 numaralı kutu (81kg, 2x2x2, 8 m³)

9.Sıradaki kutu: F tipinde tanımlanan 9 numaralı kutu (40kg, 2x2x3, 12 m³)

8.Sıradaki kutu: E tipinde tanımlanan 8 numaralı kutu (40kg, 5x1x1, 5 m³)

İşlem 54:Konteyner İçine Yerleştirilebilen Kutular

A tipinde tanımlı 1 numaralı kutu,

40 KG, 2x2x1, 4 m³

İşlem sırası: 1,

Yerleştirildiği Konteyner Numarası: 1

(x₁, y₁, z₁), (x₂, y₂, z₂) koordinatları: (0, 0, 0), (2, 1, 2)

Kutu çevrildi mi? Evet. (isRotate:true)

B tipinde tanımlı 2 numaralı kutu,

1x2x1, 2 m³

20 KG, İşlem sırası:2,

Yerleştirildiği Konteyner Numarası: 1

(x₁, y₁, z₁), (x₂, y₂, z₂) koordinatları: (0, 1, 0), (1, 2, 2)

Kutu çevrildi mi? Evet. (isRotate:true)

C tipinde tanımlı 3 numaralı kutu,

30KG, 1x1x2, 4 m³

İşlem sırası:5,

Yerleştirildiği Konteyner Numarası:2

(x₁, y₁, z₁), (x₂, y₂, z₂) koordinatları: (0, 0, 0), (1, 1, 2)

Kutu çevrildi mi? Hayır. (isRotate:false)

D tipinde tanımlı 4 numaralı kutu,

10 KG, 1x1x1, 1 m³

İşlem sırası:3,

Yerleştirildiği Konteyner Numarası: 1

(x₁, y₁, z₁), (x₂, y₂, z₂) koordinatları: (1, 1, 0), (2, 2, 1)

Kutu çevrildi mi? Hayır. (isRotate:false)

D tipinde tanımlı 5 numaralı kutu,

10 KG, 1x1x1, 1 m³

İşlem sırası:4,

Yerleştirildiği Konteyner Numarası: 1

(x₁, y₁, z₁), (x₂, y₂, z₂) koordinatları: (1, 1, 1), (2, 2, 2)

Kutu çevrildi mi? Hayır. (isRotate:false)

D tipinde tanımlı 6 numaralı kutu,

10 KG, 1x1x1, 1 m³

İşlem sırası:6,

Yerleştirildiği Konteyner Numarası:2

(x₁, y₁, z₁), (x₂, y₂, z₂) koordinatları: (1, 0, 0), (2, 1, 1)

Kutu çevrildi mi? Hayır. (isRotate:false)

D tipinde tanımlı 7 numaralı kutu,

10 KG, 1x1x1, 1 m³

İşlem sırası:7,

Yerleştirildiği Konteyner Numarası:2

(x₁, y₁, z₁), (x₂, y₂, z₂) koordinatları: (1, 0, 1), (2, 1, 2)

Kutu çevrildi mi? Hayır. (isRotate:false)

İşlem 55: Konteynerler İçin Sonuç Değerleri

1 Numaralı Konteyner,

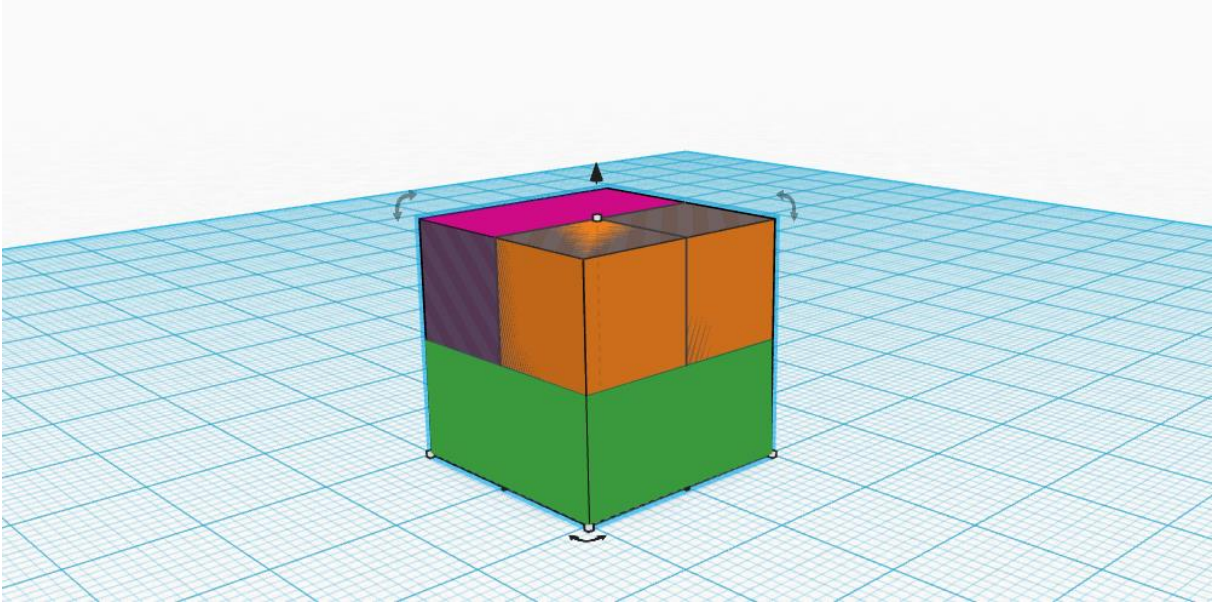
Max. 80 KG, 2x2x2,

Konteyner İçindeki Kutu Sayısı: 4

Hacim Yönünden Verimlilik : %100

Ağırlık Yönünden Verimlilik : %100

Konteyner İçindeki Kutuların Toplam Ağırlığı: 80 KG



Şekil 102. 1 Numaralı Konteynerin Son Görüntüsü

2 Numaralı Konteyner,

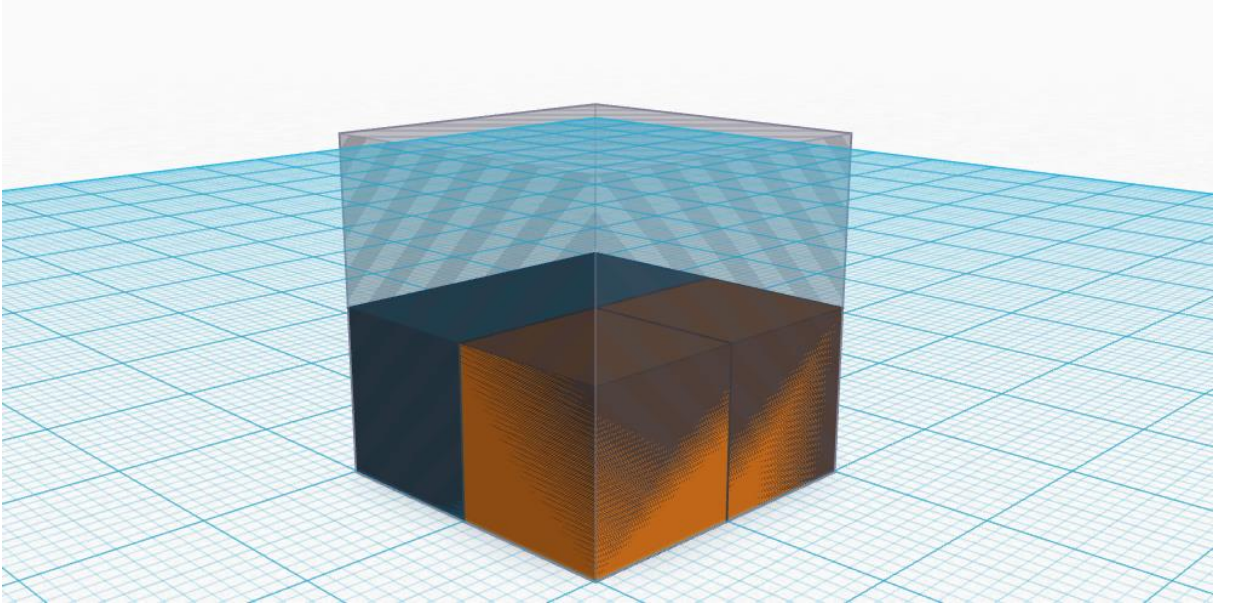
Max. 80 KG, 2x2x2,

Konteyner İçindeki Kutu Sayısı: 3

Hacim Yönünden Verimlilik : %50

Ağırlık Yönünden Verimlilik : %62,5

Konteyner İçindeki Kutuların Toplam Ağırlığı: 50 KG



Şekil 103. 2 Numaralı Konteynerin Son Görüntüsü

SONUÇ

Gerek ülkemizde gerekse dünya çapında, geçen her saniye ivme kazanan lojistik sektörü için yapılan iyileştirme çalışmaları oldukça fazladır. Yapılan iyileştirme çalışmaları lojistik sektörünün etkilendiği süreçleri kapsar.

Bu tez çalışmasında bu süreçlerden en önemli olan taşıma maliyetinin azaltılması ele alınmıştır. Taşıma maliyetinin azaltılması ürün ya da hizmet taşınmasının en büyük verimlilik ile yapılması anlamına gelmektedir. Bu bağlamda ortaya çıkan problem, konteyner yükleme problemi olarak adlandırılmaktadır. Bu problemin çözümünde birbirinden farklı bir çok yaklaşım denenmiştir. Son zamanlarda bu problemin çözümü için sunulan yazılımların çıktıkları yakın değerler içerse de bir birinden farklı sonuçlar verebilmektedir. Bu farklılığın sebebi problemin çözümüne ait farklı yaklaşımdır.

Bu tez çalışmasında probleme çözümü için kabul edilen 3 temel prensip aşağıdaki gibidir;

1- Konteyner içine girecek kutu listesinden ilk olarak en büyük hacimde olan kutular ile işleme başla

2 - Konteyner içindeki alanı doldurulurken ilk olarak doldurmaya başladığın zemini doldurmayı hedefle

3 - Oluşacak boşluk tanımlarının(spaces) bölümlere ayrılması ya da birleştirilmesi. Bu işleme yapılan büyük hesaplamalar sonunda karar verilir. Konteyner içine henüz yerleştirilmeyen kutular kombinasyon hesabına dahil edilerek, boşluk tanımının ayrılma ya da birleştirilme işlemi gerçekleştirilir.

Kodlanan web uygulaması bu 3 temel yaklaşıma bağlı olarak hesaplamalar yapar. Web kullanıcısı tarafından web ara yüzüne girilen değerler kontrol edilir. Herhangi bir değer yazılım web uygulaması için uygun değil ise, hatanın ne olduğu web kullanıcısına gösterilir. Web uygulamasına girilen değerler problem tanımına uygun ise, paketle eylemi web kullanıcısı tarafından başlatılabilir. İlk adımda sisteme tanımlanan konteyner ya da konteynerler tüm ölçütleri ile sistem tarafından okunur (değerlendirilir). Daha sonra sisteme tanımlanan tüm kutu ya da kutular tüm ölçütleri

ile yine sistem tarafından okunur. (değerlendirilir). Konteyner içine hacminden dolayı yerleştirilemeyecek herhangi bir kutu var ise, konteyner içine yerleştirilemeyecek kutu listesine eklenir. Yine aynı şekilde, konteyner içine herhangi bir ölçütü yüzünden yerleştirilemeyecek kutu var ise bir önceki adımda oluşturulan liste güncellenir. Bu liste dışında kalan tüm kutular ise, konteyner içine yerleştirilecek kutular adında bir listeye eklenir. Problemin çözümündeki temel prensibe bağlı kalmak için bu liste hacmi en büyük olandan en küçük olana olacak şekilde tekrar güncellenir.

Birden fazla konteyner seçilmiş ise, en büyük kutuyu hacim olarak içine alabilecek konteyner ilk adımda ilk işlem için seçilir. Konteyner içine yerleştirilecek kutu listesinin ilk sırasındaki kutu yani hacim olarak en büyük hacme sahip olan kutu seçilen konteyner içine yerleştirilmek için ilk kutu olarak seçilir.

Konteyner içindeki boşluk tanımına ait tüm değerler sistem tarafından okunur. Kutu yerleştirilmeden önceki boşluk tanımı ile kutunun ölçütleri karşılaştırılır. Bu karşılaştırmalar analitik düzlemde tanımlanan x, y, z eksenlerine göre yapılır. (genişlik, yükseklik, derinlik). Bir sonraki aşamaya geçmeden önce ise kutunun kendi eksenleri etrafında çevrilip çevrilmeyeceği hesaplanır. Küp şeklindeki kutular bu hesaplama tabii tutulmaz. Analitik düzlemde yapılan karşılaştırmalar sonucu yapılan hesaplamalara göre kutu konteyner içinde bir noktaya yerleştirilir. Kutu yerleştirildikten sonra konteyner içinde oluşan yeni boşluk tanımları sistem tarafından belirlenir. Boşluk tanımlarının parçalara ayrılması ya da birleştirilmesi yine yapılan hesaplamalar sonucunda olan bir eylemdir. Bu hesaplamalar oldukça uzun olan ve içinde kombinasyonların olduğu hesaplamalardır. Bu hesaplamaların sonucunda boşluk tanımları tekrar güncellenir. Konteyner içine yerleştirilecek kutuların listesinde 2. sırada bulunan kutu seçilir. Kutu içinde oluşan son boşluk tanımlarının her biri ile kutunun ölçüleri x, y, z analitik düzleminde çeşitli açılardan karşılaştırılır, hangi boşluk tanımının henüz yerleştirilmeyen kutular da hesaba katılarak kaç kutuyu içine alabileceği hesaplanır. Bu hesaplamada en yüksek sayıda kutuyu içine alabilecek olan boşluk tanımı 2. kutunun yerleşeceği alanı belirtir. Bu şekilde, konteyner içine yerleşecek kutu listesindeki tüm kutular tek tek aynı işleme tabii tutulur. Tüm kutular işleme alındıktan sonra ise, her kutu tek tek pozisyonları ile birlikte web kullanıcılarına rapor edilir.

Kodlanmış olan web uygulaması ile sisteme tanımlanan konteynerin iç hacmi olabildiğince yüksek verimle kullanılmış olur. Bu sayede ürün ya da hizmetlerin taşınma maliyeti düşürülmüş olur. Düşürülen bu maliyet taşımacılık sektöründe büyük bir önem teşkil etmektedir.

Tez çalışması dahilinde kodlanmış olan web uygulamasının benzerleri taşımacılık sektöründe oldukça yaygın olarak kullanılmaktadır. Tanımlanan problem, benzer uygulamalarda aynı sonuçlar verebildiği gibi farklı sonuçlar da verebilmektedir. Farklı sonuçların oluştuğu problemde iç hacim diğer uygulamalara bazı denemelerde daha yüksek verimlilikte sonuç vermiştir, bu da tez çalışması sonucunda kodlanan web uygulamasının başarılı olduğunu göstermektedir. Tez çalışmasında kodlanan web uygulaması sektörde yaygın olarak kullanılan web uygulaması ile aynı örneklerle test edilmiştir. (<https://3dbinpacking.com/>)

3dBinPacking.com web sayfasında yer alan arayüze problemin tanıtılması:

The screenshot displays two side-by-side forms for configuring a 3D bin packing problem. The 'Bins' form on the left has columns for Id, Width, Height, Depth, and Weight. It contains two rows of input fields with values 0, 3, 3, 3, 0 and 1, 1, 1, 1, 0. Below the fields are 'Add bin' and 'Remove bin' buttons. The 'Items' form on the right has columns for Id, Width, Height, Depth, Qty, Weight, and Vert. rot. It contains two rows of input fields with values 0, 1, 1, 1, 28, 0, checked and 1, 2, 2, 2, 4, 0, checked. Below the fields are 'Add item' and 'Remove item' buttons.

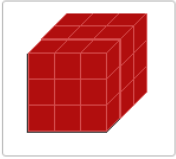
Id	Width	Height	Depth	Weight
0	3	3	3	0
1	1	1	1	0

Id	Width	Height	Depth	Qty	Weight	Vert. rot.
0	1	1	1	28	0	<input checked="" type="checkbox"/>
1	2	2	2	4	0	<input checked="" type="checkbox"/>

Şekil 104. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(1)

Problemin 3dBinPacking.com web sayfasında yer alan çıktısı: (1)

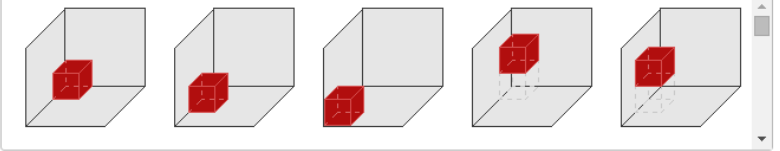
Bin Id: 0



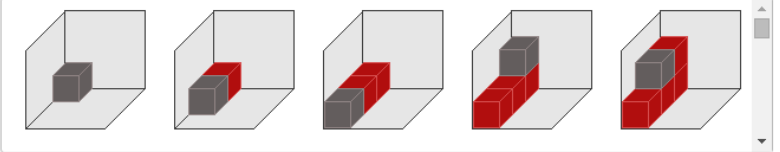
Dimensions:
Width: 3
Height: 3
Depth: 3
Weight: 0

Packed items: 27
Space taken: 100 %
Weight taken: 100 %
The packing time: 0.0205 (s)
The time of generating images: 0.02286 (s)

Images presenting the individual items inside the bin.
[Click to expand/hide](#)

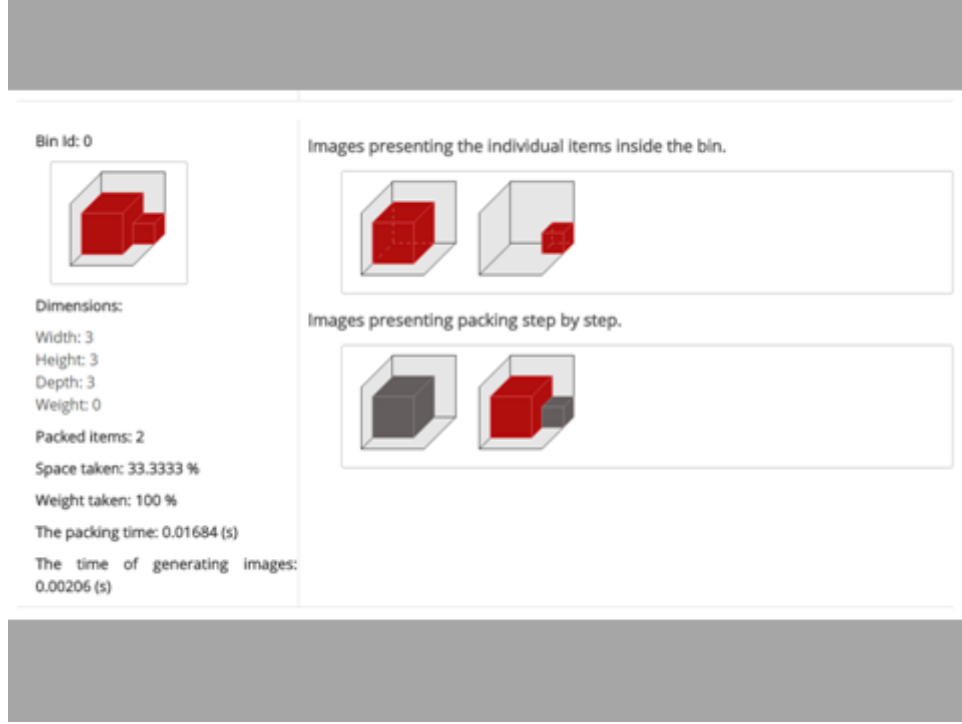


Images presenting packing step by step.
[Click to expand/hide](#)



**Şekil 105. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması
Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(2)**

Problemin 3dBinPacking.com web sayfasında yer alan çıktısı: (2)



Şekil 106. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(3)

Problemin 3dBinPacking.com web sayfasında yer alan çıktısı: (3)

Bin Id: 0



Dimensions:
Width: 3
Height: 3
Depth: 3
Weight: 0
Packed items: 1
Space taken: 29.6296 %
Weight taken: 100 %
The packing time: 0.0167 (s)
The time of generating images:
0.00134 (s)

Images presenting the individual items inside the bin.

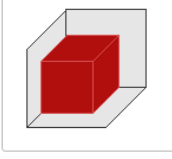
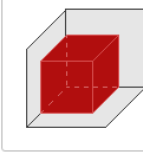
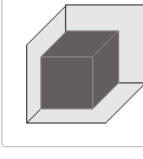


Images presenting packing step by step.



**Şekil 107. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması
Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(4)**

Problemin 3dBinPacking.com web sayfasında yer alan çıktısı: (4)

<p>Bin Id: 0</p>  <p>Dimensions: Width: 3 Height: 3 Depth: 3 Weight: 0</p> <p>Packed items: 1</p> <p>Space taken: 29.6296 %</p> <p>Weight taken: 100 %</p> <p>The packing time: 0.01678 (s)</p> <p>The time of generating images: 0.00134 (s)</p>	<p>Images presenting the individual items inside the bin.</p>  <p>Images presenting packing step by step.</p> 
--	--

**Şekil 108. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması
Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(5)**

Problemin 3dBinPacking.com web sayfasında yer alan çıktısı: (5)



Şekil 109. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dahilinde Kodlanan Web Uygulamasının Karşılaştırılması(6)

Şekillerde de görüldüğü üzere, 3dBinPacking.com web sayfası üzerinde sunulan uygulamada tanımlanan problem için gerekli konteyner sayısı 5 adettir.

Problemin tez çalışması dahilinde kodlanan web sayfasında yer alan çıktısı:

Output

Containers			
ContainerID:1	ContainerID:2	ContainerID:3	ContainerID:4
Length:3	Length:3	Length:3	Length:3
Height:3	Height:3	Height:3	Height:3
Width:3	Width:3	Width:3	Width:3
NumberOfBoxes:20	NumberOfBoxes:10	NumberOfBoxes:1	NumberOfBoxes:1
SpaceConsumed:100%	SpaceConsumed:63%	SpaceConsumed:29,6%	SpaceConsumed:29,6%

Şekil 110. 3dBinPacking.com Web Sayfasındaki Uygulama İle Tez Çalışması Dâhilinde Kodlanan Web Uygulamasının Karşılaştırılması(7)

Şekil 110.'da görüldüğü üzere, tez çalışması dahilinde kodlanan programın aynı problem için gerekli gördüğü konteyner sayısı 5 değil 4'dür.

Bunun dışında konteyner yükleme problemi için yaygın olarak kullanılan test verileri de vardır. Bu test verileri literatürde, Bischoff ve Ratcliff'in 1995, Davies ve Bischoff'ın ise 1999 yılında yayınladıkları makalede bulunan BR serisi test verileridir. Bu test verileri konteyner yükleme problemi ile alakalı olan makalelerde yaygın olarak kullanılmıştır. Toplam 1500 adet problem ayrı olarak 15 setten oluşmaktadır. Anlaşılacağı üzere her set 100 problem içermektedir. 15 seti birbirinden ayıran setlerde kullanılan farklı tipte kutu sayılarıdır. Kutu tip sayıları BR1 testlerinden BR15 testlerine doğru artmaktadır. Ve ortalama her kutu tipi toplamda yaklaşık 130 kutuyu ifade eder. BR1- BR15, kutu tip sayıları ve tip başına düşen kutu sayısı aşağıdaki Şekil.104' de gösterilmiştir.

Test	Kutu Tip Sayısı	Tip Düşen Sayısı	Başına Kutu
BR1	3	46,5	
BR2	5	28,0	
BR3	7	16,9	
BR4	10	13,2	
BR5	12	10,6	
BR6	15	8,9	
BR7	20	6,5	
BR8	30	4,6	
BR9	40	3,2	
BR10	50	2,6	
BR11	60	2,1	
BR12	70	1,8	
BR13	80	1,6	
BR14	90	1,4	
BR15	100	1,2	

Şekil 111. BR Serisi Test Verilerinin İncelenmesi

Çizelgede gösterilen BR4 testi örneklendirme ile açıklanacaktır. Eğer,

BR4 testinde yer alan farklı tipte kutu sayısı 10 adettir. Yaklaşık olarak tip başına düşen kutu adet sayısı ise 13,2'dir. Bunun anlamı bu test setinde kabaca $10 \times 13 = 130$ adet kutu bulunmaktadır. Kutu tip sayısı ve adeti arttıkça problemin zorluk derecesi de artmaktadır. Test sonuçlarının değerlendirilmesi ise şu şekilde yapılmıştır:

Problemler Őu Őekilde tanımlanmıŐtır: Kutuların konteyner hacimlerine oranı minimum %95 olmalıdır. SonuŐ ise konteyner iŐine giren kutuların hacimlerinin tım kutuların hacmine oranı olarak hesaplanmaktadır.

Őekil 111'de belirtilen verilerden BR1 ve BR7 de dahil olmak üzere BR1- BR7 arası veriler QR- Library'den [11]. BR15 de dahil olmak üzere BR7 – BR15 arası veriler ise Mr.WenŐi Huang'dan alınmıŐtır. [12, 13]

BR1 tipinde tanımlanan probleme dair sonuŐlar yapılmıŐ, aŐaŐıdaki sonuŐlar karŐılaŐtırılmıŐtır.

H.B. sonuŐ : 81.76

H.BR. sonuŐ: 83.79

GA.GB. sonuŐ: 85.80

TS.BG. sonuŐ: 93.23

H.T. sonuŐ: 89.90

HGA.BG. sonuŐ: 87.81

H.M. sonuŐ: 88.75

PTS.BG. sonuŐ : 93.52

BS.KJ. sonuŐ: 93.65

Bu tez ŐalıŐmasının sonucu:

BR1 Tipinden tanımlanan rnek problem:

1 Adet konteyner ve 3 ayrı kutu tipi, toplamda yaklaŐık 139 kutu.

Container				
Length	Height	Width	Weight	
6	7	8	139	
Add		Remove		

Box				
Length	Height	Width	Quantity	Weight
1	1	1	45	1
2	1	1	47	1
2	2	1	47	1
Add		Remove		

Şekil 112. BR1 Testinin, Tez Çalışması Dâhilinde Kodlanan Web Uygulamasının Ara Yüzüne Tanımlanması

Konteyner

Konteyner ölçüsü: 8x7x8 (genişlik, yükseklik, derinlik) (metre)

Konteyner Hacmi: 336 m³,

Konteyner Adet: 1 tane.

Konteynerin taşıyabileceği maksimum ağırlık miktarı: 139 KG.

A tipinde tanımlı kutu

A tipinde tanımlı kutunun ölçüleri: 1x1x1 (genişlik, yükseklik, derinlik) (metre)

A tipinde kutu hacmi: 1m³

A tipinde kutu adet: 45 adet

A tipinde kutuların toplam hacmi : 45m³

A tipinde kutu ağırlığı: 1 KG

B tipinde tanımlı kutu

B tipinde tanımlı kutunun ölçüleri: 2x1x1 (genişlik, yükseklik, derinlik) (metre)

B tipinde kutu hacmi: 2m³

B tipinde kutu adet: 47 adet

B tipinde kutuların toplam hacmi : 94m³

B tipinde kutu ağırlığı: 1 KG

C tipinde tanımlı kutu

C tipinde tanımlı kutunun ölçüleri: 2x2x1 (genişlik, yükseklik, derinlik) (metre)

C tipinde kutu hacmi: 4m³

C tipinde kutu adet: 47 adet

C tipinde kutuların toplam hacmi: 188m³

C tipinde kutu ağırlığı: 1 KG

Kutuların kapladığı toplam hacim: 188+94+45=327 m³ . Bu değer konteynerin hacmin(648 m³) %95'inden büyük.

Tam oran, Kutuların toplam hacmi, konteyner hacminin %97.32'si kadarıdır. Dolayısı ile bu veriler test için uygundur.

Toplam kutu sayısı: 139. Bu sayı da test verisi için uygundur. Her tip için gerekli olan kutu sayısı yaklaşık olarak 46.5

Kutuların tamamının ağırlığı: 139 KG

Toplam kutu sayısı / konteyner için yerleşebilen kutu sayısı: 139 / 139

İç Hacim verimlilik: %97,3

Ağırlık verimlilik: %100.

Bu sonuç BR1 test için oldukça iyidir.

Output

Containers
ContainerID:1
Length:6
Height:7
Width:8
NumberOfBoxes:139
SpaceConsumed:97,3%
TotalWeight:139
WeightConsumed:100%

Şekil 113. BR1 Testinin, Tez Çalışması Dâhilinde Kodlanan Web Uygulaması İle Çözümü

Kodlanan web uygulamasının BR1 testi için sonuç değerleri ayrıntı olarak tez çalışmasına ek olarak eklenmiştir. (bkz. Ek-1)

EKLER

Ek 1. BR1 Testlerinin, Tez Çalışması Dâhilinde Kodlanan Web Uygulaması İle Çözümünün Çıktısı

BoxID:93	BoxID:94	BoxID:95	BoxID:96	BoxID:97
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:1	ProcessStepNumber:2	ProcessStepNumber:3	ProcessStepNumber:4	ProcessStepNumber:5
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:0	x1:0	x1:0	x1:0	x1:2
y1:0	y1:0	y1:0	y1:0	y1:0
z1:0	z1:2	z1:4	z1:6	z1:4
x2:2	x2:2	x2:2	x2:2	x2:4
y2:1	y2:1	y2:1	y2:1	y2:1
z2:2	z2:4	z2:6	z2:8	z2:6
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:98	BoxID:99	BoxID:100	BoxID:101	BoxID:102
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:6	ProcessStepNumber:7	ProcessStepNumber:8	ProcessStepNumber:9	ProcessStepNumber:10
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:2	x1:2	x1:4	x1:4	x1:4
y1:0	y1:0	y1:0	y1:0	y1:0
z1:0	z1:2	z1:0	z1:2	z1:4
x2:4	x2:4	x2:6	x2:6	x2:6
y2:1	y2:1	y2:1	y2:1	y2:1
z2:2	z2:4	z2:2	z2:4	z2:6
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:103	BoxID:104	BoxID:105	BoxID:106	BoxID:107
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:11	ProcessStepNumber:12	ProcessStepNumber:13	ProcessStepNumber:14	ProcessStepNumber:15
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:2	x1:4	x1:0	x1:0	x1:0
y1:0	y1:0	y1:1	y1:1	y1:1
z1:6	z1:6	z1:0	z1:2	z1:4
x2:4	x2:6	x2:2	x2:2	x2:2
y2:1	y2:1	y2:2	y2:2	y2:2
z2:8	z2:8	z2:2	z2:4	z2:6
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:108	BoxID:109	BoxID:110	BoxID:111	BoxID:112
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:16	ProcessStepNumber:17	ProcessStepNumber:18	ProcessStepNumber:19	ProcessStepNumber:20
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:0	x1:2	x1:2	x1:2	x1:4
y1:1	y1:1	y1:1	y1:1	y1:1
z1:6	z1:4	z1:0	z1:2	z1:0
x2:2	x2:4	x2:4	x2:4	x2:6
y2:2	y2:2	y2:2	y2:2	y2:2
z2:8	z2:6	z2:2	z2:4	z2:2
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:113	BoxID:114	BoxID:115	BoxID:116	BoxID:117
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:21	ProcessStepNumber:22	ProcessStepNumber:23	ProcessStepNumber:24	ProcessStepNumber:25
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:4	x1:4	x1:2	x1:4	x1:0
y1:1	y1:1	y1:1	y1:1	y1:2
z1:2	z1:4	z1:6	z1:6	z1:0
x2:6	x2:6	x2:4	x2:6	x2:2
y2:2	y2:2	y2:2	y2:2	y2:3
z2:4	z2:6	z2:8	z2:8	z2:2
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:118	BoxID:119	BoxID:120	BoxID:121	BoxID:122
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:26	ProcessStepNumber:27	ProcessStepNumber:28	ProcessStepNumber:29	ProcessStepNumber:30
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:0	x1:0	x1:0	x1:2	x1:2
y1:2	y1:2	y1:2	y1:2	y1:2
z1:2	z1:4	z1:6	z1:4	z1:0
x2:2	x2:2	x2:2	x2:4	x2:4
y2:3	y2:3	y2:3	y2:3	y2:3
z2:4	z2:6	z2:8	z2:6	z2:2
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:123	BoxID:124	BoxID:125	BoxID:126	BoxID:127
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:31	ProcessStepNumber:32	ProcessStepNumber:33	ProcessStepNumber:34	ProcessStepNumber:35
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:2	x1:4	x1:4	x1:4	x1:2
y1:2	y1:2	y1:2	y1:2	y1:2
z1:2	z1:0	z1:2	z1:4	z1:6
x2:4	x2:6	x2:6	x2:6	x2:4
y2:3	y2:3	y2:3	y2:3	y2:3
z2:4	z2:2	z2:4	z2:6	z2:8
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:128	BoxID:129	BoxID:130	BoxID:131	BoxID:132
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:36	ProcessStepNumber:37	ProcessStepNumber:38	ProcessStepNumber:39	ProcessStepNumber:40
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:4	x1:0	x1:0	x1:0	x1:0
y1:2	y1:3	y1:3	y1:3	y1:3
z1:6	z1:0	z1:2	z1:4	z1:6
x2:6	x2:2	x2:2	x2:2	x2:2
y2:3	y2:4	y2:4	y2:4	y2:4
z2:8	z2:2	z2:4	z2:6	z2:8
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:133	BoxID:134	BoxID:135	BoxID:136	BoxID:137
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:41	ProcessStepNumber:42	ProcessStepNumber:43	ProcessStepNumber:44	ProcessStepNumber:45
TypeID:3	TypeID:3	TypeID:3	TypeID:3	TypeID:3
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:2	Height:2	Height:2
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:4	Volume:4	Volume:4
isRotated:True	isRotated:True	isRotated:True	isRotated:True	isRotated:True
x1:2	x1:2	x1:2	x1:4	x1:4
y1:3	y1:3	y1:3	y1:3	y1:3
z1:4	z1:0	z1:2	z1:0	z1:2
x2:4	x2:4	x2:4	x2:6	x2:6
y2:4	y2:4	y2:4	y2:4	y2:4
z2:6	z2:2	z2:4	z2:2	z2:4
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:138	BoxID:139	BoxID:46	BoxID:47	BoxID:48
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:46	ProcessStepNumber:47	ProcessStepNumber:48	ProcessStepNumber:49	ProcessStepNumber:50
TypeID:3	TypeID:3	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:2	Height:2	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:4	Volume:4	Volume:2	Volume:2	Volume:2
isRotated:True	isRotated:True	isRotated:False	isRotated:False	isRotated:False
x1:4	x1:2	x1:4	x1:4	x1:0
y1:3	y1:3	y1:3	y1:3	y1:4
z1:4	z1:6	z1:6	z1:7	z1:0
x2:6	x2:4	x2:6	x2:6	x2:2
y2:4	y2:4	y2:4	y2:4	y2:5
z2:6	z2:8	z2:7	z2:8	z2:1
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:49	BoxID:50	BoxID:51	BoxID:52	BoxID:53
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:51	ProcessStepNumber:52	ProcessStepNumber:53	ProcessStepNumber:54	ProcessStepNumber:55
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:0	x1:0	x1:0	x1:0	x1:0
y1:4	y1:4	y1:4	y1:4	y1:4
z1:1	z1:2	z1:3	z1:4	z1:5
x2:2	x2:2	x2:2	x2:2	x2:2
y2:5	y2:5	y2:5	y2:5	y2:5
z2:2	z2:3	z2:4	z2:5	z2:6
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:54	BoxID:55	BoxID:56	BoxID:57	BoxID:58
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:56	ProcessStepNumber:57	ProcessStepNumber:58	ProcessStepNumber:59	ProcessStepNumber:60
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:0	x1:0	x1:2	x1:2	x1:4
y1:4	y1:4	y1:4	y1:4	y1:4
z1:6	z1:7	z1:6	z1:7	z1:6
x2:2	x2:2	x2:4	x2:4	x2:6
y2:5	y2:5	y2:5	y2:5	y2:5
z2:7	z2:8	z2:7	z2:8	z2:7
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:59	BoxID:60	BoxID:61	BoxID:62	BoxID:63
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:61	ProcessStepNumber:62	ProcessStepNumber:63	ProcessStepNumber:64	ProcessStepNumber:65
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:2	x1:2	x1:4	x1:2	x1:2
y1:4	y1:4	y1:4	y1:4	y1:4
z1:4	z1:5	z1:4	z1:2	z1:0
x2:4	x2:4	x2:6	x2:4	x2:4
y2:5	y2:5	y2:5	y2:5	y2:5
z2:5	z2:6	z2:5	z2:3	z2:1
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:64	BoxID:65	BoxID:66	BoxID:67	BoxID:68
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:66	ProcessStepNumber:67	ProcessStepNumber:68	ProcessStepNumber:69	ProcessStepNumber:70
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:2	x1:4	x1:4	x1:4	x1:2
y1:4	y1:4	y1:4	y1:4	y1:4
z1:1	z1:0	z1:1	z1:2	z1:3
x2:4	x2:6	x2:6	x2:6	x2:4
y2:5	y2:5	y2:5	y2:5	y2:5
z2:2	z2:1	z2:2	z2:3	z2:4
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:69	BoxID:70	BoxID:71	BoxID:72	BoxID:73
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:71	ProcessStepNumber:72	ProcessStepNumber:73	ProcessStepNumber:74	ProcessStepNumber:75
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:4	x1:4	x1:4	x1:0	x1:0
y1:4	y1:4	y1:4	y1:5	y1:5
z1:3	z1:5	z1:7	z1:0	z1:1
x2:6	x2:6	x2:6	x2:2	x2:2
y2:5	y2:5	y2:5	y2:6	y2:6
z2:4	z2:6	z2:8	z2:1	z2:2
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:74	BoxID:75	BoxID:76	BoxID:77	BoxID:78
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:76	ProcessStepNumber:77	ProcessStepNumber:78	ProcessStepNumber:79	ProcessStepNumber:80
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:0	x1:0	x1:0	x1:0	x1:0
y1:5	y1:5	y1:5	y1:5	y1:5
z1:2	z1:3	z1:4	z1:5	z1:6
x2:2	x2:2	x2:2	x2:2	x2:2
y2:6	y2:6	y2:6	y2:6	y2:6
z2:3	z2:4	z2:5	z2:6	z2:7
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:79	BoxID:80	BoxID:81	BoxID:82	BoxID:83
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:81	ProcessStepNumber:82	ProcessStepNumber:83	ProcessStepNumber:84	ProcessStepNumber:85
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:0	x1:2	x1:2	x1:4	x1:2
y1:5	y1:5	y1:5	y1:5	y1:5
z1:7	z1:6	z1:7	z1:6	z1:4
x2:2	x2:4	x2:4	x2:6	x2:4
y2:6	y2:6	y2:6	y2:6	y2:6
z2:8	z2:7	z2:8	z2:7	z2:5
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:84	BoxID:85	BoxID:86	BoxID:87	BoxID:88
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:86	ProcessStepNumber:87	ProcessStepNumber:88	ProcessStepNumber:89	ProcessStepNumber:90
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:2
Length:2	Length:2	Length:2	Length:2	Length:2
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:2
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:2	x1:4	x1:2	x1:2	x1:2
y1:5	y1:5	y1:5	y1:5	y1:5
z1:5	z1:4	z1:2	z1:0	z1:1
x2:4	x2:6	x2:4	x2:4	x2:4
y2:6	y2:6	y2:6	y2:6	y2:6
z2:6	z2:5	z2:3	z2:1	z2:2
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:89	BoxID:90	BoxID:91	BoxID:92	BoxID:1
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:91	ProcessStepNumber:92	ProcessStepNumber:93	ProcessStepNumber:94	ProcessStepNumber:95
TypeID:2	TypeID:2	TypeID:2	TypeID:2	TypeID:1
Length:2	Length:2	Length:2	Length:2	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:2	Volume:2	Volume:2	Volume:2	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:4	x1:4	x1:4	x1:2	x1:4
y1:5	y1:5	y1:5	y1:5	y1:5
z1:0	z1:1	z1:2	z1:3	z1:3
x2:6	x2:6	x2:6	x2:4	x2:5
y2:6	y2:6	y2:6	y2:6	y2:6
z2:1	z2:2	z2:3	z2:4	z2:4
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:2	BoxID:3	BoxID:4	BoxID:5	BoxID:6
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:96	ProcessStepNumber:97	ProcessStepNumber:98	ProcessStepNumber:99	ProcessStepNumber:100
TypeID:1	TypeID:1	TypeID:1	TypeID:1	TypeID:1
Length:1	Length:1	Length:1	Length:1	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:1	Volume:1	Volume:1	Volume:1	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:5	x1:4	x1:5	x1:4	x1:5
y1:5	y1:5	y1:5	y1:5	y1:5
z1:3	z1:5	z1:5	z1:7	z1:7
x2:6	x2:5	x2:6	x2:5	x2:6
y2:6	y2:6	y2:6	y2:6	y2:6
z2:4	z2:6	z2:6	z2:8	z2:8
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:7	BoxID:8	BoxID:9	BoxID:10	BoxID:11
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:101	ProcessStepNumber:102	ProcessStepNumber:103	ProcessStepNumber:104	ProcessStepNumber:105
TypeID:1	TypeID:1	TypeID:1	TypeID:1	TypeID:1
Length:1	Length:1	Length:1	Length:1	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:1	Volume:1	Volume:1	Volume:1	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:0	x1:0	x1:0	x1:0	x1:0
y1:6	y1:6	y1:6	y1:6	y1:6
z1:0	z1:1	z1:2	z1:3	z1:4
x2:1	x2:1	x2:1	x2:1	x2:1
y2:7	y2:7	y2:7	y2:7	y2:7
z2:1	z2:2	z2:3	z2:4	z2:5
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:12	BoxID:13	BoxID:14	BoxID:15	BoxID:16
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:106	ProcessStepNumber:107	ProcessStepNumber:108	ProcessStepNumber:109	ProcessStepNumber:110
TypeID:1	TypeID:1	TypeID:1	TypeID:1	TypeID:1
Length:1	Length:1	Length:1	Length:1	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:1	Volume:1	Volume:1	Volume:1	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:0	x1:0	x1:0	x1:1	x1:1
y1:6	y1:6	y1:6	y1:6	y1:6
z1:5	z1:6	z1:7	z1:6	z1:7
x2:1	x2:1	x2:1	x2:2	x2:2
y2:7	y2:7	y2:7	y2:7	y2:7
z2:6	z2:7	z2:8	z2:7	z2:8
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:17	BoxID:18	BoxID:19	BoxID:20	BoxID:21
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:111	ProcessStepNumber:112	ProcessStepNumber:113	ProcessStepNumber:114	ProcessStepNumber:115
TypeID:1	TypeID:1	TypeID:1	TypeID:1	TypeID:1
Length:1	Length:1	Length:1	Length:1	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:1	Volume:1	Volume:1	Volume:1	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:2	x1:2	x1:3	x1:3	x1:4
y1:6	y1:6	y1:6	y1:6	y1:6
z1:6	z1:7	z1:6	z1:7	z1:6
x2:3	x2:3	x2:4	x2:4	x2:5
y2:7	y2:7	y2:7	y2:7	y2:7
z2:7	z2:8	z2:7	z2:8	z2:7
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:22	BoxID:23	BoxID:24	BoxID:25	BoxID:26
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:116	ProcessStepNumber:117	ProcessStepNumber:118	ProcessStepNumber:119	ProcessStepNumber:120
TypeID:1	TypeID:1	TypeID:1	TypeID:1	TypeID:1
Length:1	Length:1	Length:1	Length:1	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:1	Volume:1	Volume:1	Volume:1	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:4	x1:5	x1:1	x1:1	x1:2
y1:6	y1:6	y1:6	y1:6	y1:6
z1:7	z1:6	z1:4	z1:5	z1:4
x2:5	x2:6	x2:2	x2:2	x2:3
y2:7	y2:7	y2:7	y2:7	y2:7
z2:8	z2:7	z2:5	z2:6	z2:5
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:27	BoxID:28	BoxID:29	BoxID:30	BoxID:31
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:121	ProcessStepNumber:122	ProcessStepNumber:123	ProcessStepNumber:124	ProcessStepNumber:125
TypeID:1	TypeID:1	TypeID:1	TypeID:1	TypeID:1
Length:1	Length:1	Length:1	Length:1	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:1	Volume:1	Volume:1	Volume:1	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:2	x1:3	x1:3	x1:4	x1:4
y1:6	y1:6	y1:6	y1:6	y1:6
z1:5	z1:4	z1:5	z1:4	z1:5
x2:3	x2:4	x2:4	x2:5	x2:5
y2:7	y2:7	y2:7	y2:7	y2:7
z2:6	z2:5	z2:6	z2:5	z2:6
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

BoxID:32	BoxID:33	BoxID:34	BoxID:35	BoxID:36
ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1	ContainerId:1
ProcessStepNumber:126	ProcessStepNumber:127	ProcessStepNumber:128	ProcessStepNumber:129	ProcessStepNumber:130
TypeID:1	TypeID:1	TypeID:1	TypeID:1	TypeID:1
Length:1	Length:1	Length:1	Length:1	Length:1
Height:1	Height:1	Height:1	Height:1	Height:1
Width:1	Width:1	Width:1	Width:1	Width:1
Volume:1	Volume:1	Volume:1	Volume:1	Volume:1
isRotated:False	isRotated:False	isRotated:False	isRotated:False	isRotated:False
x1:5	x1:1	x1:1	x1:2	x1:2
y1:6	y1:6	y1:6	y1:6	y1:6
z1:4	z1:2	z1:3	z1:2	z1:3
x2:6	x2:2	x2:2	x2:3	x2:3
y2:7	y2:7	y2:7	y2:7	y2:7
z2:5	z2:3	z2:4	z2:3	z2:4
Weight:1	Weight:1	Weight:1	Weight:1	Weight:1

ÖZGEÇMİŞ

04 Eylül 1984 tarihi Ankara doğumluyum. İlkokul, ortaokul ve liseyi Samsun'da tamamladım. Samsun Milli Piyango Anadolu Lisesi Mezunuyum. Daha sonra Amerika Birleşik Devletleri, New York Eyaletinde, New York Mercy College'de Computer Science and Mathematics lisans bölümünü tamamladım. 2012'nin Mart ayında Beykent Üniversitesi'nin Bilgisayar Mühendisliği bölümünde yüksek lisansa başladım. Yüksek lisans süreci boyunca veri yapıları ve algoritmalar, veri madenciliği ve nesne tabanlı programlama konularına ilgim arttı. Hobilerim arasında ise yazılım ile uğraşmak, çizim yapmak, Tarih alanı ile alakalı araştırma yapmak, Fenerbahçe Futbol Kulübünün iç ve dış sahada oynadığı maçlara gitmek vardır.

Selim SELÇUK