EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(YÜKSEK LİSANS TEZİ)

ASAL SAYI ÖRÜNTÜLERİ ve GOLDBACH SANISI ÜZERİNE BİR ÇALIŞMA

Özgü CAN

Uluslararası Bilgisayar Anabilim Dalı
Bilim Dalı Kodu:619.02.04

Sunuş Tarihi : 10.09.2002

Tez Danışmanı : Doç. Dr. Mehmet Emin DALKILIÇ

Bornova - İZMİR

Sayın **Özgü CAN** tarafından YÜKSEK LİSANS Tezi olarak sunulan **"Asal Sayı Örüntüleri ve Goldbach Sanısı Üzerine Bir Çalışma"** adlı bu çalışma, "Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin" ve Enstitü yönergesinin ilgili hükümleri dikkate alınarak

Jüri Başkanı	; Doç. Dr. M. Emin DALKILIÇ
Raportör	; Yrd. Doç. Dr. Mustafa İNCEOĞLU
Üye	; Doç. Dr. Bahar ALAKENT
Üye	; ....
Üye	; ....

tarafından değerlendirilmiş olup, yapılan Tez Savunma Sınavında aday oy ..birliği... ile başarılı bulunmuştur.

ÖZET

ASAL SAYI ÖRÜNTÜLERİ VE GOLDBACH SANISI ÜZERİNE BİR ÇALIŞMA

CAN, Özgü
Yüksek Lisans Tezi, Uluslararası Bilgisayar Enstitüsü
Tez Yöneticisi: Doç. Dr. Mehmet Emin DALKILIÇ
Eylül 2002, 137 sayfa

Bu tezde, her dönem bilim adamlarının ilgisini çekmiş olan asal sayıları daha iyi anlamak için asal sayı örüntüleri ve 1742 yılında Christian Goldbach tarafından ortaya atılan ve dörtden büyük her çift sayının iki asal sayının toplamı olarak yazılabileceğini ifade eden Goldbach Sanısı üzerinde çalışılmıştır.

Konu ile ilgili literatür taranmış, verilen bir *n* çift sayısının bir Goldbach Çiftini veren en küçük asal sayıyı ifade eden mevcut *g(n)* fonksiyonuna alternatif olmak üzere *n/2* merkez olarak alındığında, *n*'nin bir Goldbach Çiftini veren en yakın simetrik asal çiftinin merkeze uzaklığını veren yeni bir fonksiyon (*e(n)*) geliştirilmiş ve bu fonksiyonlar karşılaştırılmıştır. Goldbach Sanısı'nın verilen bir aralıkta doğrulanması için kullanılan mevcut yöntem incelenmiş, biri bu yöntemin modifikasyonu diğeri ise tümüyle yeni bir yöntem olmak üzere iki farklı yeni yöntem geliştirilerek kodlanmış ve bu üç yöntem birbirleriyle karşılaştırılmıştır. Ayrıca verilen bir *n* çift sayısı için bütün Goldbach Çiftlerinin sayısını veren *f(n)* fonksiyonu üzerinde çeşitli uygulamalar gerçekleştirilmiştir. Uygulamalar UNIX işletim sitemi altında GAP ve C ortamlarında gerçekleştirilmiştir.

Anahtar sözcükler: Asal sayı, Asal sayı örüntüleri, Goldbach Sanısı, GAP.

ABSTRACT

AN INVESTIGATION ON PRIME NUMBER PATTERNS AND GOLDBACH CONJECTURE

CAN, Özgü
MSc, International Computer Institute
Supervisor: Assoc. Prof. Dr. Mehmet Emin DALKILIÇ
September 2002, 137 pages

Prime numbers have attracted scientists throughout history. In this thesis, prime number patterns and the Goldbach Conjecture which brought up in 1742 by Christian Goldbach stating that every even number greater than four can be written as a sum of two prime numbers have been studied to better understand the prime numbers.

The relevant literature has been searched and a new function (*e(n)*) which gives the nearest symmetrical Goldbach Partition of *n* (*n* even and greater than two) by taking *n/2* center, has been devoloped as an alternative to the existing *g(n)* function which gives the smallest prime in *n*'s Goldbach partitions and these two functions are compared. The current method in use for verifying the conjecture in a given interval has been examined and two new methods one is modified version of the existing method and the other is completely new have been developed, coded and these methods are compared. In addition, several applications have been carried out on the *f(n)* function which determines all Goldbach Partitions for a given even number, *n*. All applications have been developed on GAP and C environments under UNIX operating system.

Keywords: Prime number, Prime Number Patterns, Goldbach Conjecture, GAP.

TEŞEKKÜR

Bu tez çalışması süresince, deneyimleri ve önerilerinden yararlandığım danışmanım sayın Doç. Dr. Mehmet Emin DALKILIÇ'a, tecrübelerini ve bilgilerini paylaşan Arş. Grv. Enis KARAASLAN'a, destekleri için Ege Üniversitesi Network Yönetim Grubu'ndaki iş arkadaşlarıma, her zaman yanımda olan ve beni destekleyen annem ve babama teşekkür ederim.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (devam)

İÇİNDEKİLER (devam)

İÇİNDEKİLER (devam)

ŞEKİLLER DİZİNİ

ŞEKİLLER DİZİNİ (devam)

ÇİZELGELER DİZİNİ

ÇİZELGELER DİZİNİ (devam)

Çizelge Sayfa

ÇİZELGELER DİZİNİ (devam)

1. GİRİŞ

Her dönemde bilim insanlarını teorik açıdan cezbeden asal sayılar, günümüzde elektronik güvenlik protokolleri ve açık anahtar şifreleme gibi kritik uygulamaların merkezinde yer almaktadır. Bu nedenle hem teorik hem uygulama açısından asal sayılar üzerinde yoğun olarak çalışılmaktadır. Bu tez çalışmasında asal sayı örüntülerinin *(patterns)* daha iyi anlaşılması, bu alanda yapılan çalışmalara katkı sağlanması ve bu alandaki son gelişmelerin derlenmesi hedeflenmiştir.

Asal sayı örüntülerini daha iyi anlamak için bilim çevrelerinde doğruluğu kabul edilen fakat henüz doğruluğu matematiksel olarak kanıtlanmamış olan Goldbach Sanısı *(Goldbach Conjecture)* üzerinde çalışılmıştır. Goldbach Sanısı, 1742 yılında Christian Goldbach tarafından ortaya atılmıştır ve 4'ten büyük her çift sayının iki asal sayının toplamı olarak ifade edilebileceğini belirtir.

Tezin ilk iki bölümünde asal sayılar, asal sayılar ile ilgili tanımlamalar, Goldbach Sanısı ve bugüne kadar bilim adamları tarafından yapılmış olan çalışmalar hakkında bilgi verilmiştir. Takip eden diğer üç bölüm bu tez çalışması sırasında asal sayı örüntüleri ve Goldbach Sanısı ile ilgili yapılan çalışmaları anlatmaktadır.

Bu tez kapsamında yapılan çalışmalar üç ana başlık altında toplanabilir:

1. **g(n) ve e(n) fonksiyonları:** $g(n)$ fonksiyonu ile bir çift sayının, n, Goldbach Çiftini oluşturan en küçük asal sayısı belirlenir. $n/2$ merkez alındığında, n'nin Goldbach Çiftini veren en yakın simetrik asal çiftinin merkeze uzaklığı $e(n)$'dir. Çeşitli aralıklar için bu fonksiyonların değerlerine bakılmış ve karşılaştırmalar yapılmıştır.

2. **Goldbach Sanısı'nın verilen bir aralıkta doğrulanması:** Verilen bir aralıkta Goldbach Sanısı'nın doğrulanması ile ilgili olarak 3 yönteme bakılmıştır. Bunlardan biri Deshouillers ve te Riele tarafından geliştirilen ve $g(n)$ fonksiyonunun bulunmasına dayalı olan yöntemdir. Deshouillers ve te Riele'in teoremi geliştirilerek ikinci bir yöntem geliştirilmiştir. Üçüncü yöntem ise tez çalışması sırasında geliştirilen ve $e(n)$ fonksiyonuna dayalı olan MID yöntemidir.

3. **Verilen bir aralıkta tüm Goldbach Çiftlerinin bulunması:** $f(n)$ fonksiyonu n çift sayısının tüm Goldbach Çiftlerinin adedini tespit etmektedir. Bu uygulamada, verilen bir sayı aralığındaki tüm n çift sayıları için $f(n)$ değerlerinin bulunması gerçekleştirilmiştir. Elde edilen değerler üzerinde çeşitli çalışmalar yapılmış ve asal sayı örüntüleri olup olmadığına bakılmıştır.

Uygulamaların çoğu UNIX işletim sistemi altında GAP *(Gruplar, Algoritmalar ve Programlama; Groups, Algorithms and Programming)* ortamında kodlanmıştır. Ayrıca C programlama dili de kullanılmış ve yine UNIX işletim sistemi altında çalıştırılmıştır.

2. ASAL SAYILAR

Sadece kendisine ve bire bölünebilen, birden büyük pozitif tam sayılara *asal sayılar* denir. Asal olmayan sayılar bileşik sayılardır. Asal sayılar ve özellikleri kapsamlı olarak ilk kez antik Yunanlı matematikçiler tarafından çalışılmıştır. M.Ö. 500-300 yılları arasında Pythagoras okulunun matematikçileri asal sayılar üzerinde çalışmışlar ve asal sayıların temellerini keşfetmişlerdir (O'Connor and Robertson, 2001). Euclid, asal sayılar hakkında bir çok önemli sonucu ve aritmetiğin ana teoremini kanıtlamıştır.

Aşağıda 1000'den küçük tüm asal sayılar (168 adet) listelenmektedir:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997.

Asal sayılar ile ilgili bazı bilgiler şöyledir:

- ✓ En küçük asal sayı olan 2, tek çift asal sayıdır.
- ✓ 5'ten büyük hiç bir asal sayı 5 ile bitmez.
- ✓ 0 ve 1 asal sayı olarak kabul edilmez.
- ✓ 0 ve 1 dışındaki herhangi bir sayı, ya bileşik sayıdır ya da asal sayıdır.
- ✓ En büyük asal sayı 14 Kasım 2001 tarihinde Michael Cameron, George Woltman, Scott Kurowski ve arkadaşları tarafından bulunan 4053946 basamaklı $2^{13466917}$-1'dir (Lucile ve Gallot, 2002).

Asal sayılar konusunda cevap bekleyen bir çok problem bulunmaktadır. Henüz çözülememiş bazı problemler şöyledir (O'Connor ve Robertson, 2001):

- ➢ Birbirinden uzaklığı 2 olan bir çok asal sayı olduğunu belirten İkiz Asallar Sanısı *(Twin Primes Conjecture)*
- ➢ Dörtten büyük her çift sayının iki asal sayının toplamı olarak ifade edilebileceğini belirten Goldbach Sanısı *(Goldbach's Conjecture)*
- ➢ n^2 ve $(n+1)^2$ arasında her zaman bir asal sayı var mıdır?
- ➢ $0 \leq n \leq 40$ için n^2-n+41 asal sayıdır. Bu formda başka asal sayılar var mıdır? Aynı soru $0 \leq n \leq 79$ için $n^2-79n+1601$'in asal sayı olması durumu için de geçerlidir.

İnternet üzerinde asal sayılar ile ilgili ayrıntılı bilgi içeren siteler (örneğin: http://www.utm.edu/research/primes/) mevcuttur.

2.1 Tanımlar

Bu bölümde sayı kuramı ve asal sayılar ile ilgili bazı tanımlara yer verilmiştir.

Aritmetiğin Ana Teoremi *(The Fundamental Theorem of Arithmetic)*:

M.Ö. 350 yılında, Euclid, Elementler (*The Elements*) kitabında "Asal sayı olmayan bir n sayısının asal çarpanları var mıdır? " sorusunu evet olarak cevaplamaktadır.

$n > 1$ şeklindeki her tam sayı, asal sayıların çarpımı olarak tek bir şekilde ifade edilebilir. Aritmetiğin ana teoremine göre bütün pozitif tam sayılar asal sayıların çarpımı şeklinde çarpanlarına ayrılmaktadır.

Örnek 2.1:

$$10 = 2 \times 5$$

$$399 = 3 \times 7 \times 19$$

$$9745 = 5 \times 1949$$

$$13079 = 11 \times 29 \times 41$$

Bileşik Sayı *(Composite Number)*:

Birden büyük, iki veya daha fazla çarpanı olan pozitif tamsayılara ***bileşik sayı*** denir. Bileşik sayılar asal olmayan sayılardır. Matematikçiler 1'i ne asal sayı ne de bileşik sayı olarak kabul etmektedirler.

Örneğin 28 tane elmanız varsa bunları 2 tane 14'lük gruba ya da 4 tane 7'lik veya 7 tane 4'lü gruba vs. ayırabilirsiniz. Eğer bir elma daha eklerseniz yani toplam 29 elmanız olursa bunu eşit olarak bölemezsiniz. Bu durumda 29'un asal, 28'in ise bileşik sayı olduğu görülmektedir.

Asal Çarpanlara Ayırma *(Prime Factorization)*:

Bir sayının asal sayı mı bileşik sayı mı olduğuna karar vermek için o sayının asal çarpanları bulunur. Birden fazla tekrar eden asal faktörler üs şeklinde yazılabilir. Asal çarpanlara ayırma genellikle en büyük ortak böleni (OBEB) veya en küçük ortak katı(OKEK) bulmada kullanılır.

$$n = p_1^{e1} \times p_2^{e2} \times p_3^{e3} \times \ldots\times p_k^{ek} \quad \text{(Formül 2.1)}$$

n : pozitif tam sayı

p_i : asal sayı

e_i : asal sayının üssü

k : pozitif tam sayının faktörlerinin sayısı

Örnek 2.2:

$$96 \div \mathbf{2} = 48$$
$$48 \div \mathbf{2} = 24$$
$$24 \div \mathbf{2} = 12$$
$$12 \div \mathbf{2} = 6$$
$$6 \div \mathbf{2} = 3$$
$$3 \div \mathbf{3} = 1$$
$$96 = 2 \times 2 \times 2 \times 2 \times 2 \times 3 = 2^5 \times 3$$

İkiz Asallar *(Twin Primes)*:

Aralarındaki fark iki olan asal sayılara *ikiz asallar* denir. İkiz asallar *(p, p+2)* şeklinde ifade edilir.

Çizelge 2.1'de n çift sayısı için $n \mp 1$ formunda ilk birkaç ikiz asal görülmektedir.

Çizelge 2.1 n çift sayısı için $n \mp 1$ formundaki ilk birkaç ikiz asal

n	İkiz Asallar
4	(3, 5)
6	(5, 7)
12	(11, 13)
18	(17, 19)
30	(29, 31)
42	(41, 43)
60	(59, 61)
72	(71, 73)

(3, 5) çifti hariç bütün ikiz asal sayılar $6n \mp 1$ formundadır.

Örnek 2.3:

$$n = 1 \rightarrow 6n \mp 1 = (5,\ 7)$$

$$n = 2 \rightarrow 6n \mp 1 = (11,\ 13)$$

$$n = 3 \rightarrow 6n \mp 1 = (17,\ 19)$$

....

En büyük ikiz asal sayı çifti olan 32220 basamaklı $318032361 \times 2^{107001} \mp 1$ sayısı David Underbakke ve Phil Carmody tarafından 17 Mayıs 2001 tarihinde bulunmuştur (Lucile ve Gallot, 2002). 10^{11}'e kadar yaklaşık olarak 224376048 tane ikiz asal sayı vardır (Anderson ve Bell, 1997).

Henüz kanıtlanamamış olan İkiz Asallar Sanısı *(Twin Primes Conjecture)* sonsuz sayıda ikiz asal sayı çifti olduğunu belirtir.

Üçüz Asallar *(Prime Triplets)*:

Üçüz asallar *(p, p+2, p+4)* şeklinde ifade edilir. Bu formda sadece bir tane üçüz asal vardır: (3, 5, 7).

Eğer fark 6'ya çıkarılırsa *(p, p+2, p+6)* veya *(p, p+4, p+6)* formları elde edilir. Bu formlardan elde edilen üçüz asallar şöyledir: (5, 7, 11), (7, 11, 13), (11, 13, 17), (13, 17, 19), (17, 19, 23), (37, 41, 43), ...

Kesin olarak kanıtlanmamış olmasına rağmen üçüz asalların da sonsuz sayıda olduğu tahmin edilmektedir.

Dördüz Asallar *(Prime Quadruplets)*:

Dördüz asallar *(p, p+2, p+6, p+8)* formunda ifade edilir. Örneğin: (5, 7, 11, 13), (11, 13, 17, 19), (101, 103, 107, 109), (191, 193, 197, 199), (821, 823, 827, 829), ...

5'ten büyük bütün asal sayılar 1, 3, 7 veya 9'dan biri ile biter ve büyük dördüz asal sayılarda bu dört asal sayı her zaman aynı 10'lu blokta bulunur. 50 basamaklı en küçük dördüz asallar G. John Stevens tarafından 1995 yılında bulunmuştur (Forbes, 2000):

100000000000000000000000000000000000000000000058537891,

100000000000000000000000000000000000000000000058537893,

100000000000000000000000000000000000000000000058537897,

100000000000000000000000000000000000000000000058537899.

Mükemmel Sayılar (Perfect Numbers):

Bir pozitif tam sayı, çarpanlarının toplamına eşit ise o sayı mükemmel sayıdır. Ne kadar mükemmel sayı olduğu veya mükemmel tek sayı olup olmadığı bilinmemektedir. Bilinen bütün mükemmel sayılar çift sayıdır ve 6 ya da 8 ile bitmektedir.

Örnek 2.4:

6 = 1+2+3

28 = 1+2+4+7+14

496 = 1+2+4+8+16+31+62+124+248

8128 = 1+2+4+8+16+32+64+127+254+508+1016+2032+4064

Euclid'in teoremi, eğer $2^n - 1$ asal sayı ise $2^{n-1} \times (2^n - 1)$'in mükemmel sayı olduğunu söyler. Çizelge 2.2'de ilk beş mükemmel asal sayının çarpanları görülmektedir.

Çizelge 2.2 İlk beş mükemmel sayı asal sayının çarpanlarına ayrılması (Stepney, 2002)

Mükemmel Sayı	Asal Çarpanları
6	$2 \times 3 = 2^1 \times (2^2-1)$
28	$2^2 \times 7 = 2^2 \times (2^3-1)$
496	$2^4 \times 31 = 2^4 \times (2^5-1)$
8128	$2^6 \times 127 = 2^6 \times (2^7-1)$
33550336	$2^{12} \times 8191 = 2^{12} \times (2^{13}-1)$

Fermat'ın Küçük Teoremi *(Fermat's Little Theorem)*:

Pierre de Fermat, mükemmel sayıları araştırırken Fermat Teoremini buldu. Bu teorem genellikle sayı kuramında büyük asal sayıları test ederken kullanılır. Bu teoreme göre:

p → asal sayı

a → çarpanlarından biri p olmayan tam sayı $(obeb(a,p)=1)$ olsun,

$a^{p-1} \equiv 1 (\mathrm{mod}\, p)$' dir. (Formül 2.2)

Asallığı test edilmek istenen bir p sayısı için $a^{p-1} \equiv 1 (\mathrm{mod}\, p)$ ise sayı olası asaldır *(pseudoprime)*, değilse kesinlikle asal sayı değildir.

256 bit rasgele bir sayının bu testi geçip asal olmama olasılığı 10^{22}'de 1'dir (Rivest, 1991).

Riemann Hipotezi *(Riemann Hypothesis)* :

Asal sayıların dağılımında herhangi bir düzen yoktur. Alman matematikçi G.F.B. Riemann (1826-1866) asal sayıların sıklığının Riemann Zeta fonksiyonu, $\zeta(s)$, ile ilişkili olduğunu belirtir. Riemann, asal sayıların dağılımı ile ilgilenirken Euler'in zeta fonksiyonunu genişletmiştir. Riemann, zeta fonksiyonunun -2, -4, -6 ...' da önemsiz sıfırların yer aldığını ve bütün önemli sıfırların simetrik olarak Re(s) = 1/2 çizgisinde olduğunu belirtir. Riemann Hipotezi bütün önemli sıfırların bu çizgide olduğunu belirtir.

$$\zeta(s) = 1 + \frac{1}{2^s} + \frac{1}{3^s} + \frac{1}{4^s} + \frac{1}{5^s} + \ldots = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

1901 yılında von Koch, Riemann Hipotezi'nin aşağıdaki ifadeye eşit olduğunu göstermiştir (Caldwell,2002):

$$\pi(x) = Li(x) + O(x^{1/2} \log x) \quad \text{(Formül 2.3)}$$

π(x) ise *x*'e eşit ya da *x*'ten küçük asalların sayısı, *Li(x)* logaritmik integral, *ei* ise üstel integraldir.

$$Li(x) = \int_2^x \frac{du}{\ln u} = li(x) - li(2) \approx li(x) - 1.04516 = ei(\ln x)$$

Asal sayı teoremi ile ilgili ilişkisinden dolayı Riemann Hipotezi asal sayı teorileri içinde en önemli varsayımlardan biridir.

2.2 Asal Sayı Teoremi

2300 yıl önce Euclid, asal sayıların sonsuz sayıda olduğunu kanıtladı. Bu durumda akla şu soru geliyor :

x sayısından küçük kaç asal sayı vardır?

Asal Sayı Teoremi bu soru ile ilgilidir. Asal Sayı Teoremi 1791 yılında, henüz 14 yaşındayken, Gauss tarafından varsayım olarak ortaya atıldı. Bu teoreme eşit başka bir ifade 1798 yılında Legendre tarafından yayınlandı. Asal Sayı Teoremini kanıtlamak için ilk gerçek adım 1850 yılında Chebyshev tarafından atıldı. Asal Sayı Teoremi 1896 yılında, Charles de la Vallee Poussin ve Jacques Hadamard tarafından aynı anda ve birbirlerinden bağımsız olarak kanıtlandı.

Bu teorem rasgele bir x sayısının asal olması olasılığının yaklaşık olarak $1/\log_e x$ olduğunu belirtir. ($\log_e x = \ln x$ şeklinde de ifade edilebilmektedir.)

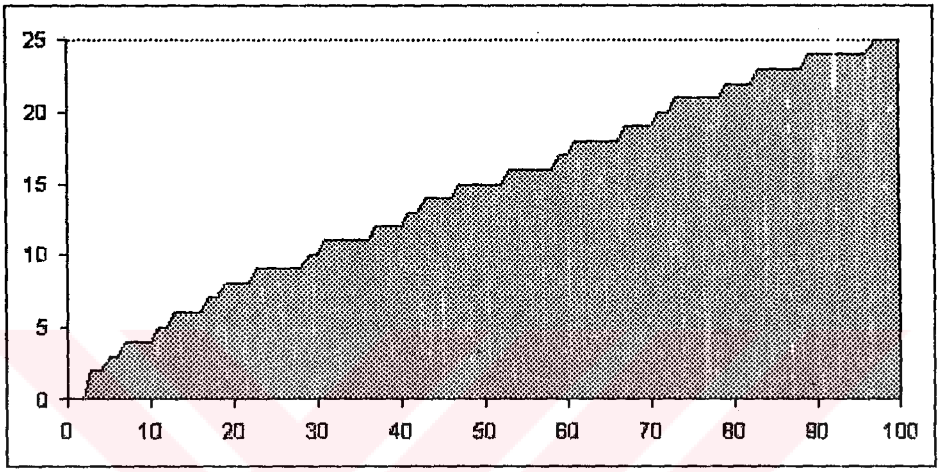
$\pi(x)$ = x'e eşit ya da x'ten küçük asalların sayısı

Örnek 2.5:

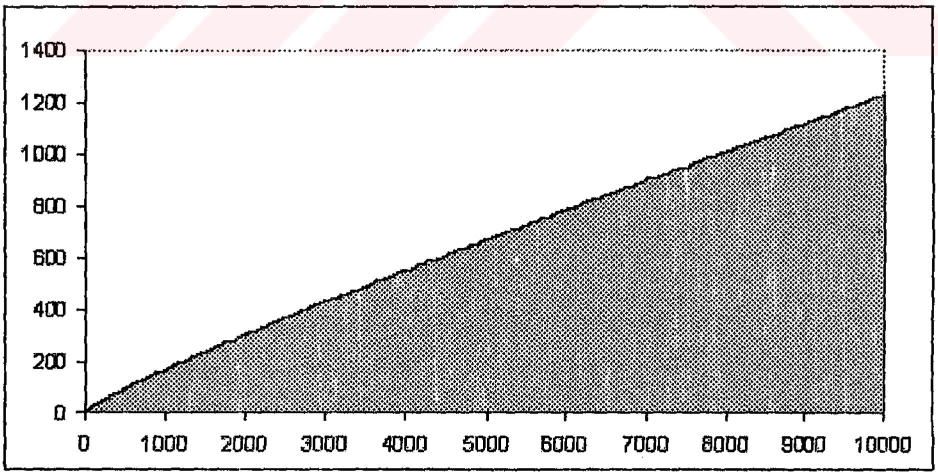
25'den küçük asal sayılar = 2, 3, 5, 7, 11, 13, 17, 19, 23

Bu durumda; $\pi(3) = 2$, $\pi(10) = 4$, $\pi(25) = 9$

Şekil 2.1 $\pi(x)$'in küçük değerler için ne kadar düzensiz olduğunu göstermektedir. Aralığı biraz daha büyüttüğümüzde Şekil 2.2'de yer alan grafik elde edilmektedir. Çizelge 2.3 $x = [10, 4\times10^{22}]$ için $\pi(x)$ değerlerini göstermektedir.



Şekil 2.1 $0\leq x\leq 100$ için $\pi(x)$ grafiği



Şekil 2.2 $0\leq x\leq 10000$ için $\pi(x)$ grafiği

Çizelge 2.3 $x = 4 \times 10^{22}$'ye kadar olan $\pi(x)$ değerleri (Caldwell, 2002)

x	$\pi(x)$
10	4
100	25
1,000	168
10,000	1,229
100,000	9,592
1,000,000	78,498
10,000,000	664,579
100,000,000	5,761,455
1,000,000,000	50,847,534
10,000,000,000	455,052,511
100,000,000,000	4,118,054,813
1,000,000,000,000	37,607,912,018
10,000,000,000,000	346,065,536,839
100,000,000,000,000	3,204,941,750,802
1,000,000,000,000,000	29,844,570,422,669
10,000,000,000,000,000	279,238,341,033,925
100,000,000,000,000,000	2,623,557,157,654,233
1,000,000,000,000,000,000	24,739,954,287,740,860
10,000,000,000,000,000,000	234,057,667,276,344,607
100,000,000,000,000,000,000	2,220,819,602,560,918,840
1,000,000,000,000,000,000,000	21,127,269,486,018,731,928
10,000,000,000,000,000,000,000	201,467,286,689,315,906,290
15,000,000,000,000,000,000,000	299,751,248,358,699,805,270
20,000,000,000,000,000,000,000	397,382,840,070,993,192,736
40,000,000,000,000,000,000,000	783,964,159,847,056,303,858

x'i geçmeyen asalların sayısı $\pi(x)$, $x/\ln x$'e asimptotiktir.

$$\pi(x) \sim x/\ln x$$

Başka bir deyişle;

$$\lim_{x\to\infty} \frac{\pi(x)\ln x}{x} \to 1 \quad \text{(Formül 2.3)}$$

"$a(x)$, $b(x)$'e asimptotik" ya da "$a(x) \sim b(x)$" ile belirtilen bu iki ifadede x sonsuza yaklaşırken $a(x)/b(x)$ oranının 1'e yaklaştığı görülmektedir.

Çizelge 2.4'te 10^{10}'a kadar olan bazı sayılar için elde edilen değerler yer almaktadır.

Çizelge 2.4 Asal sayı teoremi ile ilgili 10^{10}'a kadar olan bazı sayılar için elde edilen değerler (The University of Sheffield, Department of Pure Mathematics, 1999)

x	$\pi(x)$	$x/\ln(x)$	$\pi(x)\ln(x)/x$
10	4	4.3	0.921034
10^2	25	21.7	1.15129
10^3	168	144.8	1.1605
10^4	1229	1085.7	1.13195
10^5	9592	8685.9	1.10432
10^6	78498	72382.4	1.08449
10^7	664579	620421	1.07117
10^8	5761455	5428681	1.0613
10^9	50847534	48254942	1.05373
10^{10}	455052511	434294482	1.0478

x bir asal sayı ise bir sonraki asal sayıya olan ortalama uzaklık yaklaşık olarak $\ln x$'tir. Asal sayı teoremi, asal sayıların dağılımı hakkında kısıtlı da olsa bir fikir vermektedir.

2.3 Asal Sayıların Dağılımı

Sayı kuramındaki en klasik ve uzun süre çözülemeyen problem asal sayı dağılımıdır. Belirli bir dağılım bulmak için yapılan çalışmalar başarıya ulaşamamıştır.

Asal sayılar, tam sayılar arasına düzensiz bir şekilde dağılmıştır. İlk birkaç asal sayıya baktığımızda (2, 3, 5, 7, 11, 13) aralarında düzenli boşluklar *(gaps)* olmadığı görülmektedir ve bu dizinin nasıl devam edeceği tahmin edilemez. Matematikçiler yıllardır asal sayılar üzerinde çalışarak bir örüntü *(pattern)* bulmaya çalışmaktadırlar.

Bir asal sayıyı takip eden bileşik sayıların adedine *asal boşluğun uzunluğu* denir. Örneğin 2, 3, 5 ve 7'den sonra gelen asal aralıkları sırası ile 0, 1, 1 ve 3'tür.

Sayılar büyüdükçe asal sayıların görülme sıklığı azalmaktadır.

100'ün altında 25 asal sayı = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

1000 – 1100 arasında 16 asal sayı = [1009, 1013, 1019, 1021, 1031, 1033, 1039, 1049, 1051, 1061, 1063, 1069, 1087, 1091, 1093, 1097]

10000 – 10100 arasında 11 asal sayı = [10007, 10009, 10037, 10039, 10061, 10067, 10069, 10079, 10091, 10093, 10099]

100000 – 100100 arasında 6 asal sayı = [100003, 100019, 100043, 100049, 100057, 100069] bulunmaktadır.

Fakat bu azalmanın belirli bir düzeni yoktur. Örneğin 17200 ile 17300 arasında sadece 9 tane asal sayı (17203, 17207, 17209, 17231, 17239, 17257, 17291, 17293, 17299) varken, 170200 ve 170300 arasında

12 tane asal sayı (170207, 170213, 170227, 170231, 170239, 170243, 170249, 170263, 170267, 170279, 170293, 170299) vardır.

2.4 Asallık Testleri

Verilen bir tam sayının asal olup olmadığını belirlemek için asallık testlerinden yararlanılır. Bir n tamsayısının asal sayı olması için 2 ile $\sqrt{n}$ arasında hiçbir böleninin olmaması gerekir. Küçük sayılar için bu yöntemi uygulamakta bir sorun yoktur fakat sayılar büyüdükçe hesaplama işlemi zorlaşmaktadır. Bu nedenle büyük sayıların asal olup olmadıklarını anlamak için daha gelişmiş asallık deneyleri gerekmektedir. Asallık testleri iki gruba ayrılır:

1. Kesin *(Deterministic)* Asallık Deneyleri
2. Olası *(Probabilistic)* Asallık Deneyleri

2.4.1 Kesin *(Deterministic)* Asallık Deneyleri

Kesin Asallık Deneyleri ile bir sayının asal sayı olup olmadığını kesin olarak belirlemek mümkündür. Bu tür yöntemler genellikle çarpanlara ayırmaya dayanmaktadır.

En çok kullanılan yöntemlerin bazıları şunlardır :

1. Cyclotomic Ring Deneyi
2. Elliptic Curve Deneyi
3. Lucas-Lehmer Deneyi

Kesin Asallık Deneyleri büyük sayılar için çok zaman gerektirdiğinden pratik değildir.

2.4.2 Olası *(Probabilistic)* Asallık Deneyleri

Kesin Asallık Deneylerine göre daha hızlı olmasından dolayı büyük sayılar için Olası Asallık Deneyleri kullanılır. Olası Asallık Deneyleri testi geçen sayının yüksek olasılıkla asal olduğunu kanıtlar. Olası Asallık Deneylerinde asal sayı üretmek için öncelikle *n* bitlik bir rastsal sayı üretilir ve asallık deneyine tabi tutularak istenildiği kadar küçültülebilen bir hata payı ile sayının asal olup olmadığına bakılır. Olası asal kararı için, geçilmesi gerekli testlerin sayısı arttırıldıkça hata payı küçülür.

En çok kullanılan Olası Asallık Deneylerinin bazıları şunlardır :

1. Miller&Rabin Deneyi
2. Fermat Deneyi
3. Lehmann Deneyi
4. Solovay Strassen Deneyi

2.5 Eratosthenes Kalburu

Asal sayıları nasıl elde ederiz? Bu soru matematikçilerin hala cevap aradıkları bir sorudur. En basit method M.Ö. 300'de Eratosthenes tarafından geliştirildi. Eratosthenes "Eratosthenes Kalburu (Sieve of Eratosthenes)" olarak adlandırılan asal sayıları bulan bir algoritma geliştirdi. Eratosthenes Kalburu bileşik sayıları eler ve asal sayıları bırakır. Eratosthenes Kalburu şöyledir :

1 den büyük, n'e eşit veya küçük tam sayıların listesi yapılır. n'in kareköküne eşit ya da küçük olan bütün asal sayıların katları olan sayılar çıkarılır. Geriye kalan sayılar asal sayılardır..

Önce n sayısına kadar olan bütün sayılar yazılır. Daha sonra sırası ile 2'nin, 3'ün, 5'in, .. p'nin katları olan sayılar elenir. Bu işlem p sayısı, n sayısının kare kökünden küçük ($p \leq \sqrt{n}$) olana kadar tekrar eder. Elenmeyen sayılar n sayısına kadar olan asal sayılardır.

Örnek 2.6:

50'den küçük asal sayıları Eratosthenes Kalburu yöntemi ile bulunması

1. Aşama:

Önce 2 ve 50 arasındaki sayılar listelenir.

2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41
42	43	44	45	46	47	48	49	50	

2.Aşama:

İlk sayı olan 2 asal sayı olduğundan iki kalır, ikinin katları olan sayılar elenir.

2	3	**4**	5	**6**	7	**8**	9	**10**	11
12	13	**14**	15	**16**	17	**18**	19	**20**	21
22	23	**24**	25	**26**	27	**28**	29	**30**	31
32	33	**34**	35	**36**	37	**38**	39	**40**	41
42	43	**44**	45	**46**	47	**48**	49	**50**	

3.Aşama:

Daha sonra gelen ilk sayı olan 3 ilk tek asal sayıdır. Bu nedenle üç kalır, üçün katları olan sayılar elenir.

2	3	**4**	5	**6**	7	**8**	**9**	**10**	11
12	13	**14**	**15**	**16**	17	**18**	19	**20**	**21**
22	23	**24**	25	**26**	**27**	**28**	29	**30**	31
32	**33**	**34**	35	**36**	37	**38**	**39**	**40**	41
42	43	**44**	**45**	**46**	47	**48**	49	**50**	

4. Aşama:

Bir sonraki kalan sayı ikinci tek asal sayı olan 5'tir. Yine beş kalır, beşin katları elenir.

2	3	**4**	5	**6**	7	**8**	**9**	**10**	11
12	13	**14**	**15**	**16**	17	**18**	19	**20**	**21**
22	23	**24**	**25**	**26**	**27**	**28**	29	**30**	31
32	**33**	**34**	**35**	**36**	37	**38**	**39**	**40**	41
42	43	**44**	**45**	**46**	47	**48**	49	**50**	

5. Aşama:

Bir sonraki kalan sayı 7'dir . Bu adımda da yedi kalır, yedinin katları elenir.

2	3	**4**	5	**6**	7	**8**	**9**	**10**	11
12	13	**14**	**15**	**16**	17	**18**	19	**20**	**21**
22	23	**24**	**25**	**26**	**27**	**28**	29	**30**	31
32	**33**	**34**	**35**	**36**	37	**38**	**39**	**40**	41
42	43	**44**	**45**	**46**	47	**48**	**49**	**50**	

6. Aşama:

Bir sonraki kalan olan 11 sayısı 50'nin karekökünden ($\sqrt{50}\approx7.07$) büyük olduğu için işleme daha fazla devam edilmez ve geriye kalan sayılar asal sayıdır denilir.

2	3	**4**	5	**6**	7	**8**	**9**	**10**	11
12	13	**14**	**15**	**16**	17	**18**	19	**20**	**21**
22	23	**24**	**25**	**26**	**27**	**28**	29	**30**	31

32	**33**	**34**	**35**	**36**	37	**38**	**39**	**40**	41
42	43	**44**	**45**	**46**	47	**48**	**49**	**50**	

Bu durumda, Eratosthenes Kalburu yöntemi ile bulduğumuz 50'den küçük asal sayılar şunlardır:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47

O($n\sqrt{n}$) zaman karmaşıklığına sahip olan bu yöntem, belirli bir aralıktaki asal sayıları kesin olarak bulur. Fakat sayılar büyüdükçe daha fazla zaman alır.

2.6 Asal Sayı Uygulamaları

Asal sayılar matematiğin temel taşlarından biridir ve yüzyıllardır matematikçilerin ilgisini çeken bir konu olmuştur. Daha küçük çarpanlarına ayrılamaz olması asal sayıları özel kılar. 13 asal sayıdır ve 13 ile 1'in çarpımıdır. 13'ü başka şekilde ifade edemezsiniz. Ama 12 asal değildir ve birkaç şekilde ifade edilebilir: 2×6 veya 4×3.

Asal sayıların kriptografi ve bilgisayar sistemleri güvenliği ile ilgili uygulamaları vardır. Bankalar, ATM makinalarında ve internet üzerinden yapılan işlemlerinde asal sayılara dayanan güvenlik protokolleri kullanırlar. Asal sayı uygulamaları güvenli web sayfaları, e-ticaret ve gizlilik gerektiren e-mail işlemlerinde kullanılır.

Son zamanlarda hesaplanmış olan büyük asal sayılar matematiksel olarak merak edilen sayılardır. Asal sayı bulma yöntemlerinin bir çok pratik faydası vardır. Örneğin Slowinski ve Gage tarafından geliştirilen "asal sayı bulma *(prime finder)*" programı Cray Research[1] tarafından bütün yeni süperbilgisayar sistemlerinin kalite testinde kullanılmaktadır (Caldwell, 2002). Programın çekirdek elemanı bir sayının karesini bulmayı içeren bir rutindir. Bu işlem devam ettiğinde sonuçta büyük sayıların çarpımını içerir. Slowinski bu testin bilgisayar için tam bir işkence testi olduğunu ifade eder. Slowinski'ye göre: "Bu program *(prime finder)* bir sistemin bütün elemanlarını, işlemcinin mantığından belleğe kadar derleyiciyi, işletim sistemini ve çok işlemli sistemleri *(multitasking systems)* test eder. Çok işlemcili yüksek performanslı sistemler için sistemin bütün verinin nerede olduğunu izleme kabiliyeti için harika bir testtir."

CRAY T94 sistemi test edilirken yeni bir asal sayı bulunduğunda program sistemin bir merkezi işlemci ünitesi üzerinde 6 saate yakın çalışmıştır.

Slowinski, asal sayı bulma programının performansını arttırmada kullanılan tekniklerin gerçek dünya problemleri ile ilgilenen programların (örneğin hava durumu tahmini) performansını arttırmada da kullanılabileceğini belirtiyor.

---

[1] Cray Research, Silicon Graphics şirketinin bir kurumudur. Müşterilerin sorunlarına çözüm bulmak için temel süperhesaplama araçları (*supercomputing tools*) ve servisleri geliştirmektedir.

3. GOLDBACH SANISI

Bu bölümde Christian Goldbach, Goldbach Sanısı ve Goldbach Sanısı ile ilgili yapılan çalışmalar hakkında bilgi verilmektedir.

3.1 Christian Goldbach

Christian Goldbach, 18 Mart 1690 tarihinde Königsberg, Prussia (bugün Kaliningrad/Rusya)'da doğdu. 1725 yılında 35 yaşında iken Saint Petersburg Academy of Sciences'da matematik profesörü oldu (Gagnon, 2001).

Goldbach daha çok Sayılar Kuramı ve analiz ile uğraşmıştır. Sayılar kuramı ile ilgili önemli çalışmalar yapmış, bunların büyük bir kısmında Leonard Euler ile haberleşmiştir. En çok, hala açık bir soru olan Goldbach Sanısı ile hatırlanır. Aslında bu sanıyı Goldbach'tan önce ortaya çıkaran Descartes idi fakat Goldbach'ın fazla tanınmış bir matematikçi olmamasından dolayı sanıya Goldbach'ın adı verilmiştir (Brooke Weston CTC, 1999).

4'ten büyük her çift sayının iki asal sayının toplamı şeklinde ifade edilebildiğini söyleyen Goldbach Sanısı ilk kez 7 Haziran 1742 tarihinde Goldbach'ın Moskova'dan Berlin'deki Euler'e yazdığı mektupta geçer. Bu ifade kanıtlanması en zor sanılardan biri olarak gösterilmektedir. Euler, 30 Haziran 1742'de Goldbach'a yazdığı mektupta şu cevabı vermiştir: "... her sayı iki asal sayının toplamıdır, hemen hemen doğru olan bir teorem düşünüyorum ama onu ispatlayamıyorum" (Gagnon, 2001). Euler sanıyı ispatlamak için oldukça uğraşmış fakat başaramamıştır.

Christian Goldbach, 20 Kasım 1764 tarihinde Moskova (Rusya)'da öldü. Hayatı boyunca sanısını yazılı şekilde göremedi. İlk yazı ölümünden 6 yıl sonra 1770 yılında İngiltere'de Edward Waring tarafından yayınlandı. Daha sonra 1843 yılında Euler'in torunu tarafından Goldbach ve Euler arasında Almanca ve Latince olarak yazılan mektuplar yayınlandı.

Goldbach Sanısı'nın doğru olup olmadığı hala ispatlanamamıştır.

3.2 Goldbach Sanısı

1742 yılında Christian Goldbach, Leonard Euler'e yazdığı mektupta aşağıdaki fikrini iletir:

Her tek doğal $n \geq 9$ sayısı üç tek asal sayının toplamı olarak yazılabilir.

(Goldbach's odd Conjecture)

Goldbach, Euler'in bunu kanıtlayacağını düşünüyordu ama Euler bu sanıyı sadece basitleştirdi ve bugünkü bilinen şekline getirdi. Euler sanıyı iki kısıma böldü:

1. Her çift doğal $n \geq 4$ sayısı iki asal sayının toplamı olarak yazılabilir. *(Binary Goldbach Conjecture=Goldbach Sanısı)*

Örnek 3.1:

$6 = 3 + 3$

$8 = 3 + 5$

$10 = 3 + 7 = 5 + 5$

$12 = 5 + 7$

$14 = 3 + 11 = 7 + 4$

$16 = 3 + 13 = 5 + 11$

$18 = 5 + 13 = 7 + 11$

$20 = 3 + 17 = 7 + 13$

2. Her tek sayı, üç asal sayının toplamıdır. *(Ternary Goldbach Conjecture=Üçlü Goldbach Sanısı)*

Örnek 3.2:

$9 = 3 + 3 + 3$

$11 = 3 + 3 + 5$

$13 = 3 + 3 + 7 = 3 + 5 + 5$

$15 = 3 + 5 + 7 = 5 + 5 + 5$

$17 = 3 + 3 + 11 = 3 + 7 + 7 = 5 + 5 + 7$

İkinci Goldbach Sanısı 1937 yılında Rus Matemetikçi Ivan Vinogradov tarafından genelleştirilmiş Riemann Hipotezi'nin doğru olduğu kabulu ile ispatlanmıştır (Doxiadis, 2000). Vinogradov, yeteri kadar büyük her tek tam sayının üç tek asal sayının toplamı şeklinde ifade edilebileceğini kanıtlamıştır.

İlk Goldbach Sanısı oldukça büyük çift tam sayılar için vektör bilgisayarların yardımı ile kontrol edildi ama hala matematiksel anlamda ispatlanamamıştır.

Goldbach Sanısı'nı ispata en çok yaklaşan teorem 1966 yılında Çinli matematikçi Chen Jing-Run'ın teoremi idi (Doxiadis, 2000). Bu teoreme göre:

İkiden büyük her çift sayı, bir asal ve en fazla iki çarpanı olan bir bileşik sayının toplamına eşittir.

Bu teorem Goldbach Sanısı'na yakın görünse de aralarında geçişi sağlayacak belirgin bir adım yoktur.

Matematikçiler ve diğer araştırmacılar sanıyı büyük çift sayılar için çözmeye çalışmaktadırlar.

- 1998 yılında Amsterdam National Research Institute'dan Herman te Riele bir Cray süperbilgisayarında 10^{14}'e kadar olan bütün çift sayıları ve 10^{300}'e kadar olan bazı örnek büyük çift sayıları kontrol etti (Deshouillers,te Riele and Saouter, 1998).
- Giessen Üniversitesi Enformatik Enstitüsü'nden Jörg Richstein 4×10^{14}'e kadar sanıyı kanıtlamıştır (Richstein, 1999).

20 Mart 2000 tarihinde Faber ve Faber Yayınevi, Apostolos Doxiadis'in Petros Amca ve Goldbach Sanısı *(Uncle Petros and Goldbach's Conjecture)* adlı kitabını yayınlamış ve sanıyı ispatlayan kişiye bir milyon dolar ödül vereceklerini açıklamışlardır (Silva, 2002). Ödülü almak için 18 yaşını doldurmuş olmak ve İngiltere ya da

Amerikan vatandaşı olmak gerekmektedir. Matematik Komitesi sanının doğru ya da yanlış olduğu ile ilgili ispatı 31 Aralık 2020 tarihine kadar kabul edecektir (WWW Virtual School Pre-Algebra, 2000).

İnternet üzerinde Goldbach Sanısı ile ilgili çalışmaların tartışıldığı eGruplar mevcuttur. goldbach@egroups.com ve Goldbac-conjecture@egroups.com adreslerine email atarak bu eGrupların bazılarına üye olabilirsiniz.

3.3. Goldbach Sanısı ile İlgili Yapılan Çalışmalar

1855 yılında Desboves Goldbach Sanısı'nı 10000'e kadar kontrol etmiştir. Cunningham 1906 yılında özel formlardaki sayıları kontrol etmiştir. Dijital bilgisayarların yardımı ile bu limitler kısa zamanda arttırılmıştır. Çizelge 3.1 Goldbach Sanısı ile ilgili yapılan çalışmalardan elde edilen sonuçları göstermektedir.

Çizelge 3.1 Goldbach Sanısı ile ilgili yapılan çalışmalar (Richstein, 1999)

Kontrol eden	Tarih	Limit
Desboves	1885	10^4
N. Pipping	1940	10^5
M.K. Shen	1964	3.3×10^7
M.L. Stein, P.R. Stein	1965	10^8
A. Granville, J.V.D. Lune, H.J.J. te Riele	1989	2×10^{10}
M. Sinisalo	1993	4×10^{11}
J.-M. Deshouillers, H.J.J. te Riele, Y. Saouter	1998	10^{14}
J. Richstein	1998	4×10^{14}

1930 yılında Shnirelman, her doğal sayının 20'den fazla olmayacak şekilde asal sayıların toplamı şeklinde ifade edilebileceğini kanıtlayarak konu ile ilgili ilk büyük buluşu yapmıştır. 1937 yılında Vinagradov yeteri kadar büyük her tek sayının, üç asal sayının toplamı şeklinde ifade edilebileceğini kanıtlamıştır. 1966 yılında Chen yeteri kadar büyük her çift doğal sayının bir asal sayı ve en fazla iki asal sayının çarpımına eşit olan bir sayının toplamına eşit olduğunu kanıtlamıştır. 1977 yılında Pogorzelski Goldbach Sanısı' nı ispat ettiğini iddia etmiş fakat kanıtı kabul edilmemiştir (Weisstein, 1999). 1995 yılında Ramare her çift sayının altı veya daha az sayıda asal sayının toplamı şeklinde ifade edilebileceğini belirtmiştir. Aynı yıl Kaniecki, Riemann Hipotezi'nin doğru olduğunu kabul ederek her tek sayının en fazla beş asal sayının toplamı şeklinde ifade edilebileceğini göstermiştir (Woon, 2000).

Matti Sinisalo, 1993 yılında IBM 3083 üzerinde 130 CPU saati harcayarak Goldbach Sanısı'nı 4×10^{11}'e kadar kontrol etmiştir. Sinisalo bu işlem için bir bit dizisi ve Eratosthenes Kalburu'nu kullanmıştır (Herkommer, 2002). Yapılan işlem Algoritma 3.1'de yer almaktadır.

Algoritma 3.1 Goldbach Sanısı'nı kontrol eden algoritma

1. *bir çift sayı al, n*
2. $for(i = 3; i < n/2; i++)$ *asalları bul*
3. $if\ p = asal\_sayı\ then\ fark = n - p$
4. $if\ fark = asal\_sayı\ then\ (p, n-p)$ *Goldbach çiftidir*

Mark Herkommer web sayfasında (http://home.flash.net/~mherk/goldbach.htm) Goldbach Sanısı'nın doğru olduğunu gösteren bir çözüm yayınlamıştır (Herkommer, 2002). Bu çözüm aşağıdaki gibidir:

$n[0]$ çift sayı olsun:

$$n[0] = p[0] + q[0]$$

ve

$$n[1] = q[0] - p[0] \text{ olsun}$$

$p[0]$ ve $q[0]$ farklı tek asal sayılardır, $p[0] < q[0]$.

$$n[1] = p[1] + q[1]$$

ve

$$n[2] = q[1] - p[1] \text{ olsun}$$

$p[1]$ ve $q[1]$ farklı tek asal sayılardır, $p[1] < q[1]$.

Bu şekilde;

$n[m] = p[m] + q[m]$

ve

$n[m+1] = q[m] - p[m]$ olana kadar devam edilir.

$p[m]$ ve $q[m]$ farklı tek asal sayılar olmayabilir, $p[m] \leq q[m]$, ve $n[m+1] = 0, 2$ *veya* 4.

$n[0]$ şu şekilde tekrar oluşturulabilir:

$n[0] = 2 \times p[0] + 2 \times p[1] + 2 \times p[2] + \ldots + 2 \times p[m] + (0$ *veya* 2 *veya* 4)

Bu durumda $n[0]/2$ sınırlı sayıda asal sayı serisinin toplamına ek olarak keyfi olarak 0,1 ya da 2'nin eklenmesinden oluşur.

2 asal sayı olduğundan aşağıdaki ifade söylenebilir:

$n[0]/2 = p[0] + p[1] + p[2] + \ldots + p[m] + (0$ *ya da* 1)

Buradaki soru şöyledir; sınırlı sayıda asal sayı serisinin toplamına 0 veya 1'in eklenmesi ile temsil edilemeyecek $n[0]/2$ şeklinde bir sayı var mıdır?

İkiz asal sayıların mevcut olduğu bilindiğine göre, 0 ya da 1 keyfi olarak eklendiğinden, ve her çift aralık $(p, 2 \times p)$ en az bir asal sayı içerdiğinden burada cevap olarak HAYIR verilebilir.

Mark Herkommer: "Bu durumda Goldbach Sanısı doğru olarak kanıtlanmış mıdır?" der ve cevabını da şu şekilde verir: "... hayır, pek değil ama çok yaklaşmış görünüyoruz .. sadece biraz daha çalışma gerekiyor..”

Algoritma uygulandığında elde edilen sonuçlar şöyledir:

$n = 30$ için;

$$30 = 7 + 23$$
$$16 = 3 + 13$$
$$10 = 3 + 7$$
$$4 = 2 + 2$$

$n = 98$ için;

$$98 = 19 + 79$$
$$60 = 7 + 53$$
$$46 = 3 + 43$$
$$40 = 3 + 37$$
$$34 = 3 + 31$$
$$28 = 5 + 23$$
$$18 = 5 + 13$$
$$8 = 3 + 5$$
$$2$$

Fakat bu çözümde yapılmış olan bir hata vardır. Çözümün başında kabul edilen bir $n[0]$ çift sayısı alınması ve $n[0] = p[0] + q[0]$ olarak kabul edilmesi hatalıdır çünkü zaten Goldbach Sanısı ile ispatlanmaya çalışılan ifade budur. Mark Herkommer ispatlanmaya çalışılan fikri başta doğru kabul ederek çözümüne hatalı başlamıştır.

Jörg Richstein, Goldbach Sanısı'nı 4×10^{14}'e kadar kontrol etmiştir (Richstein, 1999). Richstein'in yaklaşımı onu bir çift sayının kaç farklı şekilde iki asal sayının toplamı olarak ifade edilebileceğini araştırmaya itmiştir. Program çeşitli iş istasyonlarına dağıtılmıştır (Peterson, 2000). Her bir iş istasyonu farklı bir sayı aralığı için hesaplamaları gerçekleştirmiştir.Eğer bir çift sayı için bir asal sayı çifti bulunamazsa o zaman sanının doğru olmadığı ortaya çıkacaktır.

Richstein, 500 milyona kadar olan çift sayılar için asal sayı çiftlerini bulmuştur. Fransa'daki Recherche Enstitüsü'nden Yannick Saouter aynı uygulamayı farklı bir teknik ile 128 milyona kadar gerçekleştirmiştir (Richstein, 2000).

Çizelge 3.2 bazı sayılar için Goldbach çiftleri sayısını göstermektedir.

Çizelge 3.2 Bazı sayılar için Goldbach çifti sayısı (Peterson, 2000)

Sayı	Goldbach Çifti Sayısı
10	2
100	6
1,000	28
10,000	127
100,000	810
1,000,000	5,402
10,000,000	38,807
100,000,000	291,400

Çift sayıların bu değerleri dalgalanma göstermektedir. Genel olarak çift sayılar büyüdükçe asal sayı çiftlerinin[2] sayısı da artmaktadır. Fakat bu her zaman geçerli olmayabilir. Örneğin 30030 için 905 asal sayı çifti bulunurken komşuları olan 30028'in 237, 30032'nin ise 225 asal sayı çifti bulunmaktadır.

---

[2] Burada asal sayı çifti ve Goldbach Çifti ifadeleri aynı anlamdadır.

4. *g(n)* ve *e(n)* FONKSİYONLARI

Bu bölümde, uygulanan g(n), e(n) fonksiyonları ve elde edilen sonuçlar anlatılmaktadır. Uygulamalar, Unix işletim sistemi altında GAP yazılım ortamında gerçekleştirilmiştir.

4.1 Uygulamalarda Kullanılan Unix Makinalarının Konfigürasyonu

Uygulamalarda Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü makina parkında yer alan bodrum, bayindir, bergama, buca, urla, foca, didim ve dikili makinaları kullanılmıştır. Makinalar 400 Mhz Ultra-Sparc IIi işlemcili ve 128 MB RAM özelliklerine sahip Ultra 5 iş istasyonlarıdır. İşletim sistemleri Solaris 5.8'dir.

4.2 GAP Ortamı

GAP kısaltmasının açılımı Gruplar, Algoritmalar ve Programlama (Groups, Algorithms and Programming) şeklindedir. GAP'ın özellikleri şöyledir:

- GAP, kullanıcıya bir çok matematiksel fonksiyon ve yüksek basamaklı aritmetik işlemlerde kolaylık sağlar.

- Unix, Linux, Macintosh ve Windows işletim sistemlerinin her biri ile kullanılabilir.

- Açık ve eklentiler ile genişletilebilecek bir sistemdir. Kullanıcı GAP dili ile yazdığı programını, herhangi bir GAP fonksiyonunu çalıştırdığı gibi çalıştırabilmektedir. Programlar paylaşım paketi *(share package)* şeklinde diğer kullanıcılar ile paylaşılabilmektedir.

- GAP, internetten ücretsiz olarak temin edilebilir. http://www-gap.dcs.st-and.ac.uk/~gap/Info4/distrib.html

- Kullanıcıların GAP ile ilgili konuları tartıştıkları bir GAP forumu mevcuttur. `miles@dcs.st-and.ac.uk` adresinden GAP listesine üye olduktan sonra `gap-forum@dcs.st-and.ac.uk` adresine mesajlarınızı gönderebilirsiniz.

4.2.1 GAP Fonksiyonları

Uygulamalarda kullanılan GAP fonksiyonları Çizelge 4.1'de açıklanmıştır.

Çizelge 4.1 Bazı GAP Fonksiyonları

Fonksiyon	**İşlevi**
Add(*list*,*n*)	*n*'yi listenin sonuna ekler.
Append(*list1*,*list2*)	İkinci listeyi birinci listenin sonuna ekler.
Difference(*list1*,*list2*)	İki listeyi birbiri ile karşılaştırıp farklı olan elemanlarını listeler.
IsEvenInt(*n*)	*n*'nin çift sayı olup olmadığını kontrol eder. Çift sayı ise 'true', değilse 'false' döndürür.
IsPrimeInt(*n*)	*n*'nin asal sayı olup olmadığını kontrol eder. Eğer *n* bileşik sayı ise 'false', asal sayı ise 'true' döndürür. 10^{13}'e kadar olan sayılar için kesin sonuç verir. $n>10^{13}$ için sonucun kesin olmadığı şeklinde uyarı mesajı verir.
Length(*list*)	Listedeki eleman sayısını döndürür.
NextPrimeInt(*n*)	*n*'den büyük ilk asal sayıyı verir.
PrevPrimeInt(*n*)	*n*'den küçük en büyük asal sayıyı verir.
Sort(*list*)	Listenin elemanlarını artan biçimde sıralar.
Runtime()	İşlemin harcadığı zamanı *milisaniye işlemci zamanı* cinsinden verir.
Unbind(*n*)	*n*'nin değerini sıfırlar.

4.3 g(n) Fonksiyonu

g(n) fonksiyonu ile 4'ten büyük bir *n* çift sayısı için, $p_1 \leq p_2$

olmak üzere, $n = p_1 + p_2$ eşitliğini sağlayan en küçük p_1 asal sayısı belirlenmektedir.

Örnek 4.1:

10 için ilk Goldbach çifti = 3 + 7 → g(10) = 3

100 için ilk Goldbach çifti = 3 + 97 → g(100) = 3

1000 için ilk Goldbach çifti = 3 + 997 → g(1000) = 3

10000 için ilk Goldbach çifti = 59 + 9941→ g(10000) = 59

100000 için ilk Goldbach çifti = 11 + 99989 → g(100000) = 11

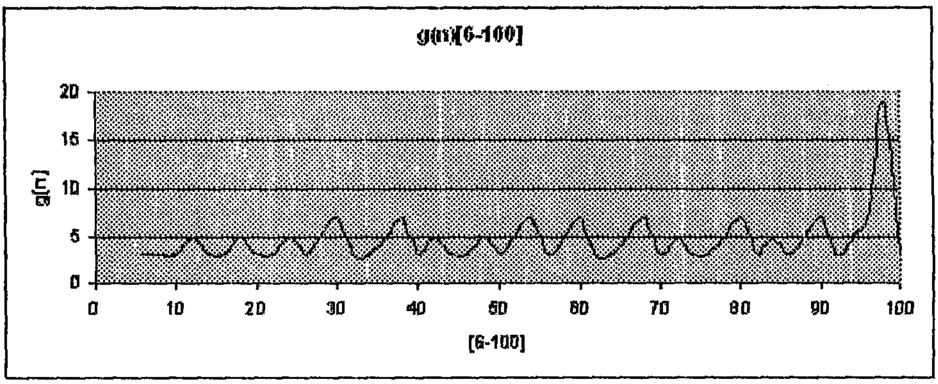
1000000 için ilk Goldbach çifti = 17 + 999983 → g(1000000) = 17

100'e kadar olan ($n > 4$) çift sayılar için elde edilen *g(n)* değerleri Çizelge 4.2'de yer almaktadır.

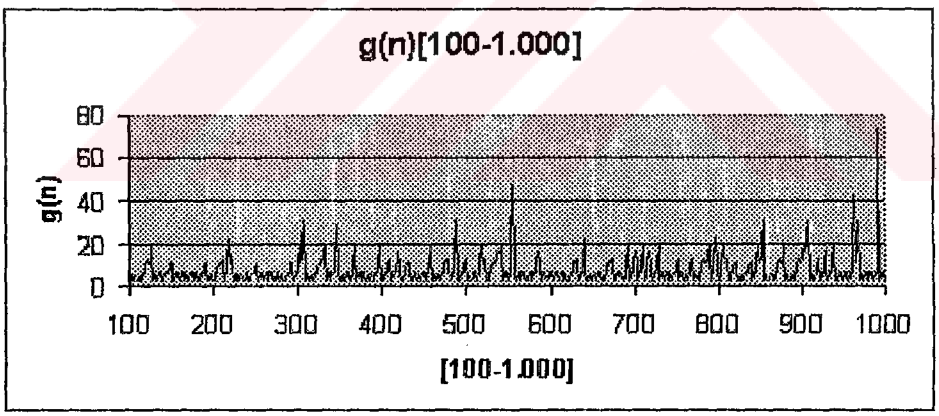
[6, 100], [100, 1000], [1000, 10000], [10000, 50000] ve [50000, 100000] aralıklarındaki *g(n)* değerlerinin grafikleri Şekil 4.1,…,4.5'te verilmiştir.

Çizelge 4.2 [6, 100] aralığındaki çift sayılar için *g(n)* değerleri

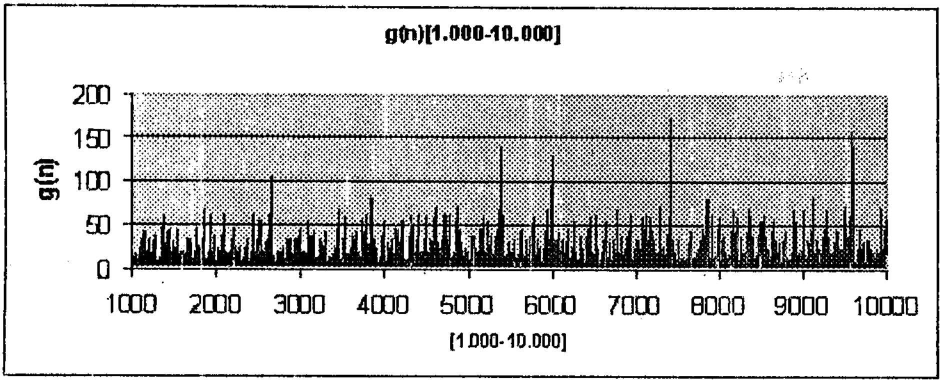
Sayı	g(n)	Sayı	g(n)
6	3	54	7
8	3	56	3
10	3	58	5
12	5	60	7
14	3	62	3
16	3	64	3
18	5	66	5
20	3	68	7
22	3	70	3
24	5	72	5
26	3	74	3
28	5	76	3
30	7	78	5
32	3	80	7
34	3	82	3
36	5	84	5
38	7	86	3
40	3	88	5
42	5	90	7
44	3	92	3
46	3	94	5
48	5	96	7
50	3	98	19
52	5	100	3



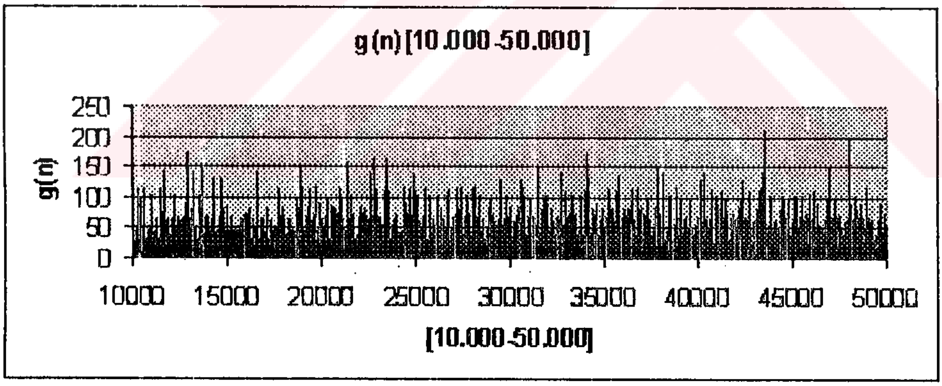
Şekil 4.1 [6, 100] aralığı için *g(n)* grafiği



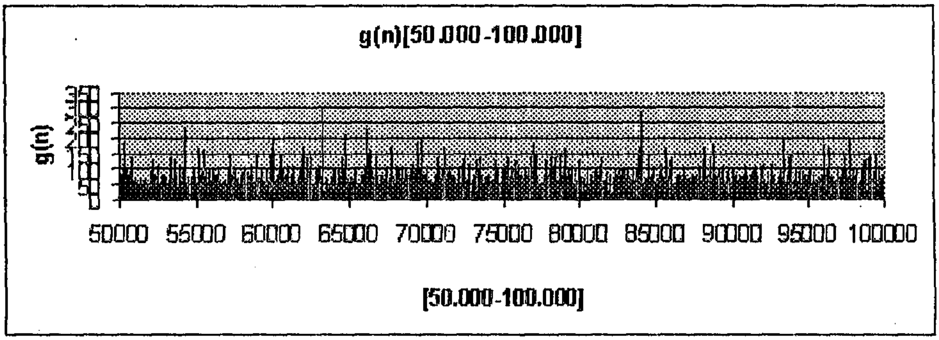
Şekil 4.2 [100, 1000] aralığı için *g(n)* grafiği



Şekil 4.3 [1000, 10000] aralığı için *g(n)* grafiği



Şekil 4.4 [10000, 50000] aralığı için *g(n)* grafiği



Şekil 4.5 [50000, 100000] aralığı için $g(n)$ grafiği

$g(n)$ değerleri incelendiğinde n değeri arttıkça ortalama olarak $g(n)$ değerinin de artmakta olduğu görülmektedir. Çizelge 4.3, $g(n)$'in bazı aralıklar bazında ve bu aralıkların tümündeki, [6,100000], ortalamasını göstermektedir. Aralıklar büyüdükçe $g(n)$ ortalaması da artmaktadır.

Çizelge 4.3 $g(n)$ Değerlerinin Ortalamaları

g(n)	Ortalama
[6, 100]	4,6250
[100, 1000]	7,3415
[1000, 10000]	11,2604
[10000, 50000]	14,4834
[50000, 100000]	16,2162
Genel Ortalama [6, 100000]	10,7853

Aralık değişimlerine göre $g(n)$ grafikleri bazı noktalarda görülen büyük sıçramalar dışında birbirine benzemektedir. Örneğin Şekil 4.1'de g(98)'de büyük bir sıçrama fark edilmektedir. g(98) = 19 iken komşuları g(96) = 7, g(100) = 3 değerlerini almaktadır. Şekil 4.2'de ki sıçramalardan birinin olduğu noktada g(992) = 73, komşuları g(990) = 7 ve g(994) = 3 değerlerini alır. Şekil 4.5'te yer alan sıçramalardan biri g(63274) noktasındadır. g(63272) = 31, sıçramanın olduğu nokta olan g(63274) = 293 ve komşusu g(63276) = 29 değerlerini almaktadır. Bu sıçramalarda belirli bir düzen görülmemektedir.

Çizelge 4.4, n<1000000 için o ana kadar görülen $g(n)$ değerlerinden daha büyük olan değerlerin ilk görüldükleri yerleri, n değerlerini, belirtmektedir. Bu n değerleri yukarıda bahsedilen sıçramaların gerçekleştiği noktalardır. n değeri arttıkça en son sütunda yer alan $g(n)/n$ oranının azaldığı görülmektedir.

Çizelge 4.4 n<1000000 için $g(n)$ değerleri

n	g(n)	n – g(n)	g(n) / n
6	3	3	0.500000000
12	5	7	0.416666667
30	7	23	0.233333333
98	19	79	0.193877551
220	23	197	0.104545455
308	31	277	0.100649351
556	47	509	0.084532374
992	73	919	0.073588710
2642	103	2539	0.038985617
5372	139	5233	0.025874907
7426	173	7253	0.023296526
43532	211	43321	0.004847009
54244	233	54011	0.004295406
63274	293	62981	0.004630654
113672	313	113359	0.002753536
128168	331	127837	0.002582548
194428	359	194069	0.001846442
194470	383	194087	0.001969455
413572	389	413183	0.000940586
503222	523	502699	0.001039303

4.4 e(n) Fonksiyonu

$n = 2k,\ k \in Z,\ k > 2$ olmak üzere $n/2$ değeri merkez alındığında en yakın simetrik asal sayı çiftinin merkeze uzaklığı $e(n)$'dir. $e(n)$ fonksiyonu $n/2$ değerinden ne kadar uzakta ilk simetrik asal sayı çiftinin bulunduğunu göstermektedir.

p_1 ve p_2 asal sayı, n dörtten büyük bir çift sayı olmak üzere $n = p_1 + p_2$'yi sağlayacak şekilde en küçük $e(n)$ değeri bulunmaktadır.

$p_1 = \frac{n}{2} - e(n)$, $p_2 = \frac{n}{2} + e(n)$ formülü ile bulunmaktadır. Algoritma 4.1 $e(n)$ fonksiyonun hesaplanmasında kullanılan algoritmadır.

Algoritma 4.1 $e(n)$ fonksiyonunun algoritması

1. *bir çift sayı al, n*
2. $sayaç = 0$
3. $if\,(n/2) = asal\_sayı$ *then* $sayaç = sayaç + 1$
4. $p_1 = (n/2) - sayaç$, $p_2 = (n/2) + sayaç$
5. $if\,(p_1 = asal\_sayı$ *ve* $p_2 = asal\_sayı)$ *then* $e(n) = sayaç$
6. *else* $sayaç = sayaç + 2$, $p_1 = p_1 - 2$, $p_2 = p_2 + 2$ *goto* 5

Örnek 4.2:

$n = 10$ için;

1. n = 10
2. $sayaç = 0$
3. 10/2 = 5, 5 asal sayı olduğundan e(10) = 0'dır.

$n = 20$ için;

1. n = 20
2. sayaç = 0
3. 20/2 = 10

4. 10 asal sayı olmadığı için $p_1 = 9$, $p_2 = 11$
5. p_2 asal sayı ama p_1 asal sayı değil
6. $e(n) = 3$, $p_1 = 7$, $p_2 = 11$

   p_1 ve p_2 asal sayı olduklarından e(20) = 3' tür.

n = 100 için; e(100) = 3

n = 1000 için; e(1000) = 9

n = 10000 için; e(10000) = 81

n = 100000 için; e(100000) = 123

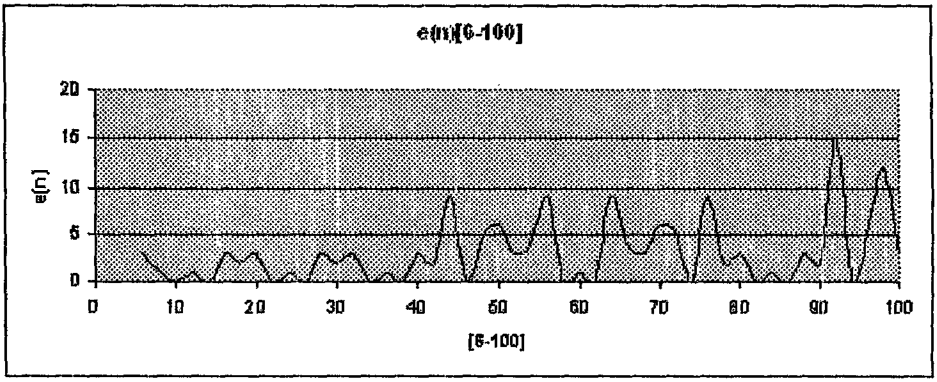
n = 1000000 için; e(1000000) = 273

4'ten büyük 100'e kadar olan çift sayılar için elde edilen $e(n)$ değerleri Çizelge 4.5'de verilmiştir.

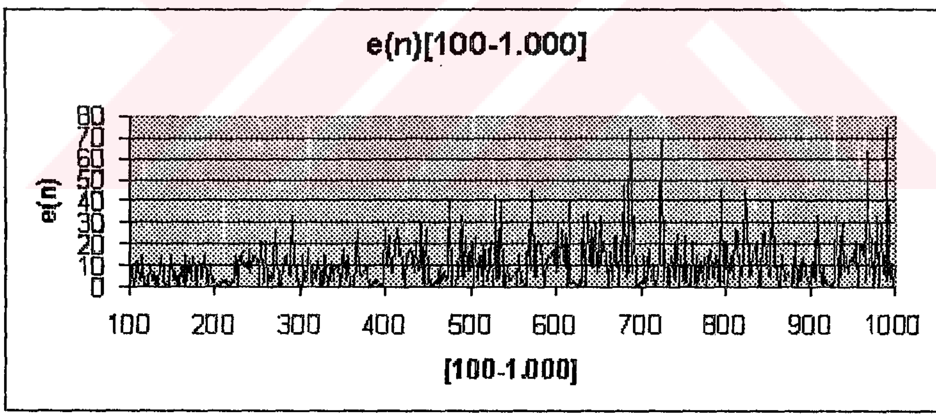
[6, 100], [100, 1000], [1000, 10000], [10000, 50000] ve [50000, 100000] aralıklarındaki $e(n)$ değerlerinin grafikleri Şekil 4.6,..., 4.10'da görülmektedir.

Çizelge 4.5 [6, 100] aralığındaki çift sayılar için *e(n)* değerleri

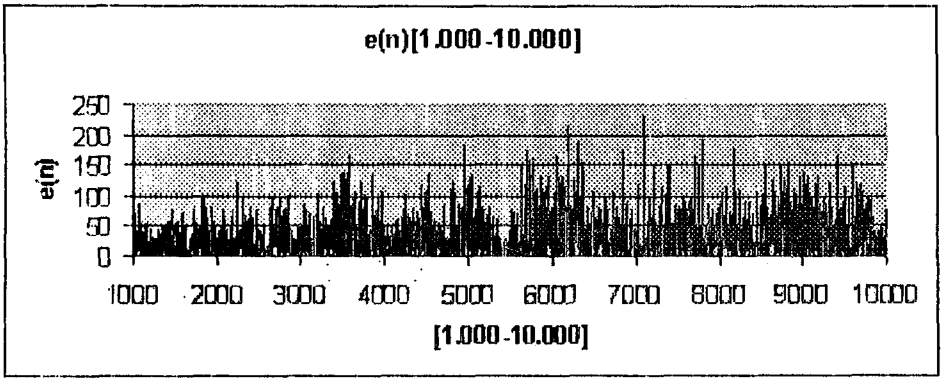
Sayı	e(n)	Sayı	e(n)
6	0	54	4
8	1	56	9
10	0	58	0
12	1	60	1
14	0	62	0
16	3	64	9
18	2	66	4
20	3	68	3
22	0	70	6
24	1	72	5
26	0	74	0
28	3	76	9
30	2	78	2
32	3	80	3
34	0	82	0
36	1	84	1
38	0	86	0
40	3	88	3
42	2	90	2
44	9	92	15
46	0	94	0
48	5	96	5
50	6	98	12
52	3	100	3



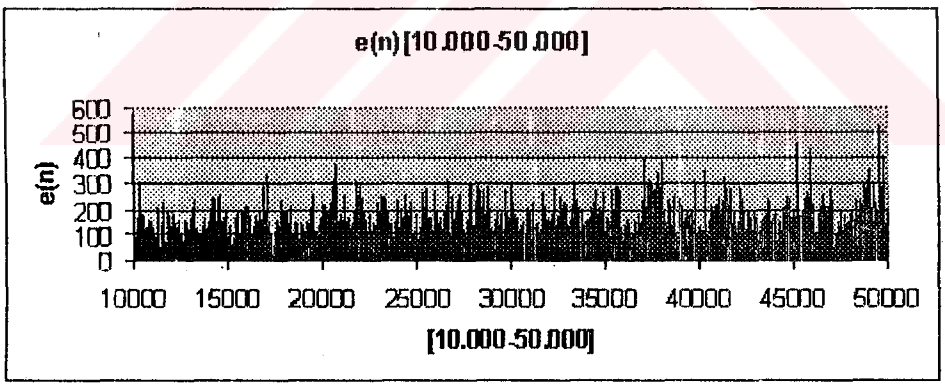
Şekil 4.6 [6, 100] aralığı için *e(n)* grafiği



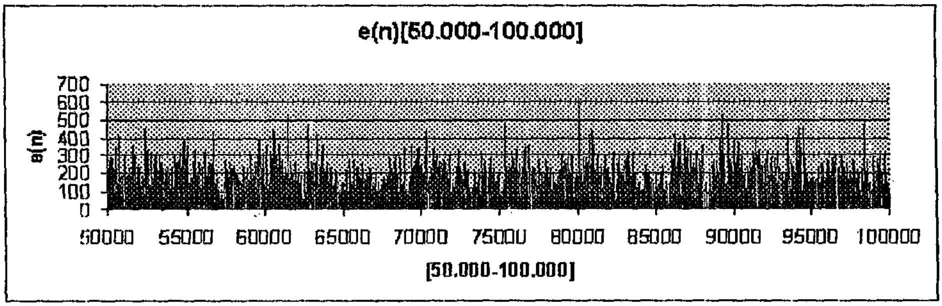
Şekil 4.7 [100, 1000] aralığı için *e(n)* grafiği



Şekil 4.8 [1000, 10000] aralığı için *e(n)* grafiği



Şekil 4.9 [10000, 50000] aralığı için *e(n)* grafiği



Şekil 4.10 [50000, 100000] aralığı için $e(n)$ grafiği

$e(n)$ grafiklerinin temel karakteristiği n değeri arttıkça ortalama olarak $e(n)$ değerinin de artmasıdır. Grafiklerin ortalamaları ve genel ortalama Çizelge 4.6'da yer almaktadır. Aralıktaki sayılar büyüdükçe $e(n)$ ortalaması da artmaktadır.

Çizelge 4.6 $e(n)$ değerlerinin ortalamaları

e(n)	Ortalama
[6, 100]	3
[100, 1000]	9,9778
[1000, 10000]	24,1215
[10000, 50000]	38,6731
[50000, 100000]	48,1257
Genel Ortalama [6, 100000]	24,7796

$n = 2p$, p asal sayı, durumunda $e(n) = 0$ durumu tüm asal sayılar için geçerlidir. Bu durum grafiklerde en dip vadi değerlerini oluşturmaktadır. Örneğin Şekil 4.6'da yer alan e(94) = 0 ve e(46) = 0 dip vadi değerlerindendir.

Grafiklerden bazı $e(n)$ değerleri için büyük atlamalar olduğu görülmektedir. Örneğin Şekil 4.6'da e(92) = 15 değerini alırken komşuları e(90) = 2 ve e(94) = 0'dır. Şekil 4.7'ye bakıldığında e(688) = 75 iken komşuları olan e(686) = 5 ve e(690) = 8 değerini almaktadır. Şekil 4.10'da e(80144) = 621 değerini alırken komşuları olan e(80142) = 40 ve e(80144) = 90'dır.

Aralık değişimlerine göre $e(n)$ grafikleri karşılaştırıldıkların da aralarında belirgin farklılıklar gözlenmemektedir. Genel olarak grafiklerin seyiri aynıdır. $n/2$ değerinin asal sayı olduğunu belirten sıfır noktaları ve büyük artışların gerçekleştiği tepe noktaları dışında $e(n)$

grafiklerinde büyük değişimler gerçekleşmemektedir.

$n<1000000$ için o ana kadar görülen $e(n)$ değerlerinden daha büyük olan değerlerin ilk görüldükleri yerler Çizelge 4.7'de görülmektedir. Bu değerler yukarıda bahsedilen büyük atlamaların olduğu noktalara denk gelmektedir.

Çizelge 4.7 n<1000000 için özel $e(n)$ değerleri

n	e(n)	Goldbach Çifti	e(n)/n
8	1	(3,5)	0,125000000
16	3	(5,11)	0,187500000
44	9	(13,31)	0,204545455
92	15	(31,61)	0,163043478
242	18	(103,139)	0,074380165
256	21	(107,149)	0,082031250
272	27	(109,163)	0,099264706
292	33	(113,179)	0,113013699
476	39	(199,277)	0,081932773
530	42	(223,307)	0,079245283
572	45	(241,331)	0,078671329
682	48	(293,389)	0,070381232
688	75	(269,419)	0,109011628
1052	87	(439,613)	0,082699620
1808	93	(811,997)	0,051438053
2228	117	(997,1231)	0,052513465
3382	120	(1571,1811)	0,035481963
3472	135	(1601,1871)	0,038882488
3502	138	(1613,1889)	0,039406054
3562	168	(1613,1949)	0,047164514
4952	183	(2293,2659)	0,036954766
6194	210	(2887,3307)	0,033903778
7102	228	(3323,3779)	0,032103633
10262	300	(4831,5431)	0,029234067
17008	333	(8171,8837)	0,019579022
20684	369	(9973,10711)	0,017839876
37052	393	(18133,18919)	0,010606715
45128	453	(22111,23017)	0,010038114
49552	525	(24251,25301)	0,010594931
80144	621	(39451,40693)	0,007748553
137414	720	(67987,69427)	0,005239641
251806	810	(125093,126713)	0,003216762
349826	846	(174067,175759)	0,002418345
362534	1086	(180181,182353)	0,002995581
742856	1281	(370147,372709)	0,001724426

4.5 g(n) ve e(n) Fonksiyonlarının Karşılaştırılması

$g(n)$ ve $e(n)$ fonksiyonlarının $n > 4$ olmak üzere 1000'e kadar olan çift sayılar için elde edilen değerlerinin ortalamalarına bakılmıştır. Bu karşılaştırma Çizelge 4.8'de görülebilir.

Çizelge 4.8'deki aralıklarda genel olarak $g(n)$ ve $e(n)$ ortalamaları birbirine zıt bir şekilde değişmektedir. $g(n)$ ortalaması arttığı zaman $e(n)$ ortalaması düşmekte, $g(n)$ ortalaması azaldığı zaman $e(n)$ ortalaması artmaktadır. Artan n değerlerine karşı $g(n)$ artışı $e(n)$ artışına kıyasla daha yumuşaktır.

[6, 500] aralığında her iki fonksiyonunda ortalaması genel ortalamanın ve [500, 1000] aralığındaki ortalamanın altındadır.

Çizelge 4.8 [6, 1000] aralığında $g(n)$ ve $e(n)$ ortalamaları

Aralık	Ortalama g(n)	Ortalama e(n)	Aralık	Ortalama g(n)	Ortalama e(n)
6-20	3,50	1,63	502-520	6,80	10,70
22-40	4,40	1,30	522-540	10,00	12,70
42-60	4,60	3,90	542-560	11,00	4,70
62-80	4,40	4,10	562-580	4,80	13,10
82-100	6,00	4,10	582-600	7,00	12,90
102-120	4,20	5,90	602-620	4,40	15,70
122-140	8,20	4,70	622-640	8,80	11,30
142-160	5,80	4,30	642-660	5,20	16,70
162-180	5,00	6,70	662-680	6,40	11,30
182-200	5,40	6,70	682-700	7,80	17,90
202-220	8,80	1,30	702-720	8,80	6,90
222-240	5,80	9,50	722-740	6,40	15,30
242-260	5,80	12,50	742-760	5,40	11,30
262-280	4,80	5,50	762-780	5,60	6,70
282-300	6,00	9,50	782-800	10,80	13,30
302-320	9,60	6,70	802-820	8,40	11,50
322-340	8,60	6,90	822-840	6,20	14,90
342-360	7,80	6,50	842-860	10,00	13,70
362-380	6,40	8,90	862-880	8,20	4,30
382-400	6,00	3,70	882-900	6,80	7,30
402-420	7,60	14,50	902-920	12,20	8,50
422-440	5,80	12,30	922-940	8,00	8,90
442-460	6,00	7,70	942-960	5,00	11,90
462-480	6,80	6,90	962-980	12,40	19,10
482-500	8,20	12,90	982-1000	11,80	15,50
Ortalama	6,22	6,75	**Ortalama**	7,93	11,84

g(n) Genel Ortalama [6,1000]

7,08

e(n) Genel Ortalama [6,1000]

9,29

1000'e kadar olan çift sayılar için, $n > 4$, $g(n)$ ve $e(n)$ fonksiyonlarında bakılan bir diğer karşılaştırmada şöyledir:

$e(n) = g(n)$ olduğu durum → 0

$e(n) < g(n)$ olduğu durum → 1

$e(n) > g(n)$ olduğu durum → 2 olarak belirtilmiştir.

[6, 100] aralığındaki çift sayılar için , bu işlem sonucunda elde edilen değerler:

$e(n) = g(n)$ olduğu durumların sayısı : 7

$e(n) < g(n)$ olduğu durumların sayısı : 34

$e(n) > g(n)$ olduğu durumların sayısı : 7

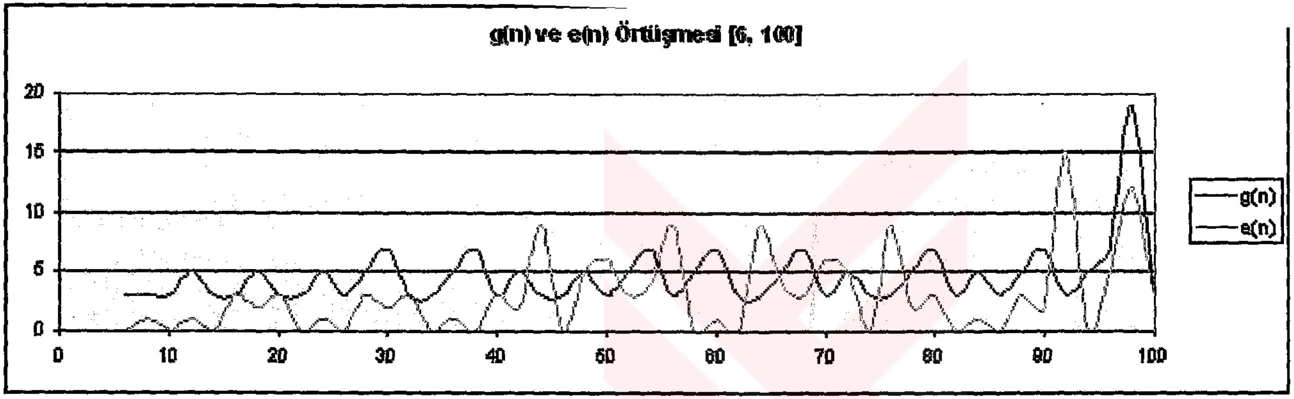
[6, 1000] aralığındaki çift sayılar için bu işlem sonucunda şu değerler elde edilmiştir:

$e(n) = g(n)$ olduğu durumların sayısı : 38

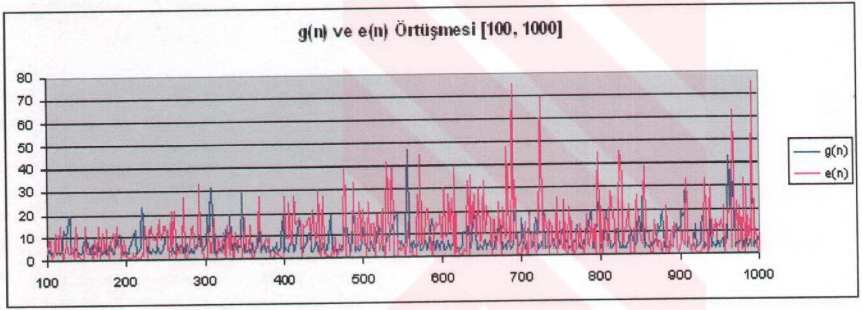
$e(n) < g(n)$ olduğu durumların sayısı : 230

$e(n) > g(n)$ olduğu durumların sayısı : 230 adettir.

Şekil 4.11 ve 4.12 $g(n)$ ve $e(n)$ fonksiyonlarının [6, 100] ve [100, 1000] aralıklarında ki değerlerinin üst üste bindirilmiş durumlarının grafikleridir.



Şekil 4.11 [6, 100] Aralığında *g(n)* ve *e(n)* Örtüşmesi



Şekil 4.12 [100, 1000] Aralığında *g(n)* ve *e(n)* Örtüşmesi

Şekil 4.11 ve 4.12'de bazen fonksiyonların birinin tepe yaptığı noktada diğerinin dibe vurduğu, bazen de ikisinin birden dibe vurduğu ya da tepe yaptıkları görülmektedir. Örneğin Şekil 4.11'de g(98) = 19 ve e(98) = 12, Şekil 4.12'de g(992)= 73 ve e(992) = 75 değerleri ile iki fonksiyonda tepe yapmaktadır. Şekil 4.11'de g(82) = 3 ve e(82) = 0, Şekil 4.12'de g(914) = 3 ve e(914) = 0 değerleri ile iki fonksiyonunda dibe vurduğu görülmektedir. Şekil 4.11'de g(38) = 7 ve e(38) = 0 değerleri ile $g(n)$ fonksiyonu tepe yaparken $e(n)$ fonksiyonu dibe vurmakta, Şekil 4.12'de ise g(824) = 3 ve e(824) = 45 değerleri ile $e(n)$ fonksiyonu tepe yapmakta $g(n)$ fonksiyonu ise dibe vurmaktadır. Sonuç olarak bu örtüşmelerde belirli bir desen görünmemektedir. Çizelge 4.9 fonksiyonların çalışma zamanlarını milisaniye işlemci zamanı (CPU) cinsinden göstermektedir. $g(n)$ ve $e(n)$ fonksiyonları, en kötü durum için, $O(n\sqrt{n})$ zaman karmaşıklığına sahiptir. m aralığında uzunluk için bu karmaşıklık değeri $O(mn\sqrt{n})$ olur.

Çizelge 4.9 Fonksiyonların çalışma zamanları

	$n=[6,10^2]$	$n=[6,10^3]$	$n=[6,10^4]$	$n=[6,10^5]$	$n=[6,10^6]$
$T_{e(n)}$	0	60	1490	28920	532550
$T_{g(n)}$	0	0	80	920	16374

$g(n)$ değerlerinin hesaplanması $e(n)$ değerlerinin hesaplanmasına göre daha az zaman almaktadır. Aralık değerleri büyüdükçe $g(n)$ fonksiyonu $e(n)$ fonksiyonundan daha hızlı hesaplanmaktadır.

$[6,10000]$ aralığı için elde edilen zaman ölçümlerini oranladığımızda; $\frac{T_{e(n)}}{T_{g(n)}} = \frac{1490}{80} = 18,62 \cong 18$, $e(n)$ fonksiyonu $g(n)$ fonksiyonundan yaklaşık olarak 18 kat daha yavaştır.

$[6,100000]$ aralığı için elde edilen zaman ölçümlerini oranladığımızda; $\frac{T_{e(n)}}{T_{g(n)}} = \frac{28920}{920} = 31,43 \cong 31$, $e(n)$ fonksiyonu $g(n)$ fonksiyonundan yaklaşık olarak 31 kat daha yavaştır.

Görüldüğü gibi aralık değerleri büyültüldükçe $g(n)$ ve $e(n)$ fonksiyonlarının hesaplanmasında geçen $\frac{T_{e(n)}}{T_{g(n)}}$ oranları da büyümektedir. Bunun nedeni; $g(n)$ fonksiyonunda n > 4 için Goldbach çiftini sağlayan en küçük asal sayıyı bulduğu zaman işlemin bitmesi, fakat $e(n)$ fonksiyonunda $n > 4$ için Goldbach çiftini verecek asal sayı $n/2$ merkezinden uzak ise işlemin daha fazla zaman harcamasıdır. Goldbach çiftini oluşturmaya aday değerler asal listesinden alınırken $e(n)$ için $n/2$ değerinden başlayarak birer birer azaltılıyor ve bu nedenle işlem daha fazla zaman alıyor. Ayrıca $e(n)$ fonksiyonunda asallığı test edilen her iki sayıda büyük, $g(n)$ fonksiyonunda ise sadece biri büyük'tür. Bu da asallık testlerinde $e(n)$ fonksiyonu aleyhine bir durumdur.

Örnek 4.3:

72 sayısının $g(n)$ ve $e(n)$ değerleri aynı olmasına rağmen ($g(72) = e(72) = 5$) sonuca ulaşmada geçilen adım sayısı $e(n)$ fonksiyonu için daha fazladır. Bu nedenle $e(n)$ fonksiyonu daha fazla zaman alıyor.

g(72) = 3 + 69 ✗

= 5 + 67 ✓ işlem 2. denemede bitmekte

e(72) = 35 + 37 ✗

= 34 + 38 ✗

= 33 + 39 ✗

= 32 + 40 ✗

= 31 + 41 ✓ işlem 5. denemede bitiyor

4.6 Sonuçların İrdelenmesi

- ✓ $g(n)$ fonksiyonundan elde edilen sonuçlar incelendiğinde büyük sıçramaların olduğu noktaların dışında herhangi bir örüntüye rastlanmamıştır. Sıçramaların olduğu noktalarda herhangi bir düzen görülmemektedir. Bu sıçrama noktaları o ana kadar görülen $g(n)$ değerlerinden daha büyük olan değerlerin ilk görüldükleri yerleri vermektedir.

- ✓ $e(n)$ fonksiyonu ile elde edilen sonuçlardan sıfır değerinin görüldüğü n noktasında $n/2$ değerinin asal sayı olduğu söylenebilir. $e(n)$ fonksiyonunun sonuçlarında $g(n)$ fonksiyonunununda olduğu gibi bazı değerlerde sıçramalar vardır. Sıçramaların olduğu noktalarda herhangi bir düzen yoktur. Bu sıçrama noktaları o ana kadar görülen $e(n)$ değerlerinden daha büyük olan değerlerin ilk görüldükleri yerleri vermektedir.

- ✓ İki fonksiyonun grafikleri örtüştürüldüğünde bazen fonksiyonların birinin tepe yaptığı noktada diğerinin dibe vurduğu, bazen de ikisinin birden dibe vurduğu ya da tepe yaptıkları görülmektedir.

- ✓ $e(n)$ fonksiyonunun genel ortalaması $g(n)$ fonksiyonun genel ortalamasından daha büyüktür.

- ✓ $e(n)$ fonksiyonunun hesaplanması $g(n)$ fonksiyonun hesaplanmasından daha fazla zaman almaktadır. Sonuç olarak $g(n)$ fonksiyonunun Goldbach Sanısı'nı verilen bir aralıkta doğrulama amaçlı kullanımında $e(n)$ fonksiyonundan daha hızlı olduğu görülmüştür.

5. GOLDBACH SANISI'NIN VERİLEN BİR ARALIKTA DOĞRULANMASI

Verilen bir aralık için Goldbach çiftlerinin bulunmasında üç yöntem incelenmiştir. Bu yöntemlerin bir kısmı geçen bölümde anlatılan $g(n)$ ve $e(n)$ fonksiyonlarına dayalıdır.

Yöntem 1: Deshouillers ve te Riele tarafından geliştirilen bir yöntemdir (Deshouillers, Effinger, te Riele ve Zinoviev, 1997). Bu yöntem, önceki bölümde anlatılan $g(n)$ fonksiyonunun bulunmasına dayanmaktadır.

Yöntem 2: 1. Yöntemin modifikasyonudur.

Yöntem 3: Bu tez çalışması sırasında geliştirilen MID yöntemidir. Geçen bölümde anlatılan $e(n)$ fonksiyonuna dayanmaktadır.

Uygulamalar UNIX işletim sistemi altında GAP yazılımı ile kodlanmıştır.

5.1 Yöntem 1 (Deshouillers ve te Riele)

Bu yöntem Deshouillers ve te Riele tarafından geliştirilmiştir. 4. bölümde anlatılan $g(n)$ fonksiyonuna ($g(n)$ fonksiyonu dörtten büyük bir n çift sayısı için, $p_1 \leq p_2$ olmak üzere, $n = p_1 + p_2$ eşitliğini sağlayan en küçük p_1 asal sayısına eşittir.) dayanan bir yöntemdir. Algoritma 5.1 Yöntem 1'in algoritmasıdır (Deshouillers, Effinger, te Riele ve Zinoviev, 1997). Bu algoritma, en kötü durum için, O(kn) + $a.g(n)$ zaman karmaşıklığına sahiptir. Burada O(kn) birinci aşamanın yani Algoritma 5.1'in ilk dört adımının, a değeri ilk aşamada bulunamayan Goldbach Çiftlerinin adedinin, $g(n)$ ise ikinci aşamanın karmaşıklığıdır.

Algoritma 5.1 Yöntem 1'in algoritması

1. *[a,b]* aralığı ve *k* sayısı belirle
2. *a*'dan önceki *k* asal sayıyı bul
3. *b-a*'ya kadar olan asal sayıları bul
4. 2. ve 3. adımdaki asal sayıları toplayarak *[a,b]* aralığındaki çift sayıları bul
5. Bulunamayan çift sayıları listele
6. Bulunamayan çift sayıların her biri için *b-a*'dan sonraki asal sayı ile farkını al ve farkın asal sayı olup olmadığına bak
   - asal sayı ise listeye ekle
   - asal sayı değilse bir sonraki sayıya geç

Algoritma 5.1'in GAP kodu aşağıda yer almaktadır:

```
yontem1:=function(alt,ust,k)
     local p,m,t1,t2,t,alt_asallar,n,fark,fark_asallar,
           r,l,q,numbers,w,y,z,toplam,x,cift_sayilar,
           bulunamayanlar, w2,w3,w4,alt2,ust2,alt3,fark2;
     p:=0;
     fark:=ust-alt;
     fark2:=ust-alt;
     alt_asallar:=[];
     fark_asallar:=[];
     numbers:=[2..fark];
     toplam:=[];
     cift_sayilar:=[];
     bulunamayanlar:=[];
     t1:=Runtime();
     m:=alt;
     alt2:=alt;
     ust2:=ust;
     alt3:=alt;
     repeat
          if alt mod 2 = 0 then
               Add(cift_sayilar,alt);
          fi;
           alt:=alt+1;
     until(alt=ust+1);
     Sort(cift_sayilar);
     Print("\nCift Sayilar = ",cift_sayilar,"\n");
     repeat
```

```
        n:=PrevPrimeInt(m);
        Add(alt_asallar,n);
        m:=n;
        p:=p+1;
    until(p=k);
    Sort(alt_asallar);
    Print("\nAlt_asallar = ",alt_asallar,"\n");
    for r in numbers do
        Add(fark_asallar,r);
        for l in numbers do
            if l mod r =0 then Unbind(numbers[l-1]);
            fi;
        od;
    od;
    Sort(fark_asallar);
    Print("\nFark_asallar = ",fark_asallar,"\n");
    for z in alt_asallar do
        for y in fark_asallar do
            w:=z+y;
            if(IsEvenInt(w)=true) then
                if (alt2-1<w and w<ust2+1) then
                    if w in toplam then Unbind(w);
                        else Add(toplam,w);
                    fi;
                fi;
            fi;
        od;
    od;
    t2:=Runtime();
    Sort(toplam);
    Print("\nToplam = ",toplam,"\n");
```

```
    bulunamayanlar:=Difference(cift_sayilar,toplam);
    Sort(bulunamayanlar);
    if(bulunamayanlar=0) then
        Print("\nBulunamayanlar = Verilen aralikta
              Goldbach Cifti bulunamayan sayi
              yoktur!","\n");
    else
        Print("\nBulunamayanlar =",bulunamayanlar,"\n");
            for w2 in bulunamayanlar do
                repeat
                w3:=NextPrimeInt(fark2);
                w4:=w2-w3;
                if(IsPrimeInt(w4)=true) then
                    Print("\n",w2,"= ",w4,"+",w3,"\n");
                    Unbind(w2);
                else
                    fark2:=w3;
                fi;
                until(IsPrimeInt(w4)=true);
                fark2:=fark;
            od;
        Print("\n Bu yontem ile verilen aralikta
              Goldbach Cifti bulunamayan sayi
              kalmamistir!", "\n");
    fi;
    t:=t2-t1;
    Print("\nTime elapsed ==> ",t," milisecond cpu time.
          \n");
end;
```

Bu yöntem ile; $a = 950$, $b = 1000$, $k = 10$ değerleri için elde edilen açıklama satırları ile örnek bir çıktı şöyledir:

```
gap> Read("yontem1.txt");
gap> yontem1(950,1000,10);
Cift Sayilar = [ 950, 952, 954, 956, 958, 960, 962, 964,
966, 968, 970, 972, 974, 976, 978, 980, 982, 984, 986,
988, 990, 992, 994, 996, 998, 1000 ]
// Cift Sayilar listesi [950,1000] aralığındaki Goldbach
// Çiftleri bulunacak olan çift sayıların listesidir.
Alt_asallar = [ 881, 883, 887, 907, 911, 919, 929, 937,
941, 947 ]
// Alt_asallar listesi 950 değerinden önceki k = 10 adet
// asal sayıyı içerir.
Fark_asallar = [ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31,
37, 41, 43, 47 ]
// Fark_asallar listesi, 1000-950 = 50, [2, 50]
// aralığındaki asal sayıları içermektedir.
Toplam = [ 950, 952, 954, 956, 958, 960, 962, 964, 966,
968, 970, 972, 974, 976, 978, 980, 982, 984, 988, 990, 994
]
// Alt_asallar ve Fark_asallar listelerindeki elemanlar
// toplandığında elde edilen [950,1000] aralığındaki
// değerler Toplam listesinde yer almaktadır.
Bulunamayanlar = [ 986, 992, 996, 998, 1000 ]
// Toplam listesinde yer almayan fakat Cift Sayilar
//    listesinde    bulunan    elemanlar,    yani    [950,1000]
aralığında
// Yöntem 1 ile Goldbach Çifteri bulunamayan sayılar
```

```
// Bulunamayanlar listesinde tutulur.
986 = 919 + 67
992 = 919 + 73
996 = 937 + 59
998 = 937 + 61
1000 = 947 + 53
Bu yontem ile verilen aralikta Goldbach Çifti bulunamayan
sayi kalmamistir!
Time elapsed ==> 40 milisecond cpu time.
gap>
```

5.2 Yöntem 2 (1. Yöntemin Modifikasyonu)

Yöntem 2, bu tez çalışması sırasında, Deshouillers ve te Riele'in yönteminin modifiye edilmiş şeklidir. Bu yöntemin algoritması Algoritma 5.2'de yer almaktadır. Bu algoritma, en kötü durum için, $O(kn) + a.g(n)$ zaman karmaşıklığına sahiptir. Burada $O(kn)$ birinci aşamanın yani Algoritma 5.2'nin ilk dört adımının, a değeri ilk aşamada bulunamayan Goldbach Çiftlerinin adedinin, $g(n)$ ise ikinci aşamanın karmaşıklığıdır.

Algoritma 5.2 Yöntem 2'nin Algoritması

1. *[a,b]* aralığı ve *k* sayısı belirle
2. *a*'dan önceki ve sonraki *k/2* asal sayıyı bul
3. *b-a*'ya kadar olan asal sayıları bul
4. 2. ve 3. adımdaki asal sayıları toplayarak *[a,b]* aralığındaki

çift sayıları bul

5. Bulunamayan çift sayıları listele
6. Bulunamayan çift sayıların her biri için *b-a*'dan sonraki asal sayı ile farkını al ve farkın asal sayı olup olmadığına bak
   - asal sayı ise listeye ekle
   - asal sayı değilse bir sonraki sayıya geç

Deshouillers ve te Riele algoritmalarının 2. adımında (Algoritma 5.1) sadece *a*'dan önceki *k* asal sayıya bakarken, Yöntem 2'de *a*'dan önceki ve sonraki $k/2$ asal sayıya bakılmaktadır.

Algoritma 5.2'nin GAP'ta kodlanmış hali şöyledir:

```
yontem2:=function(alt,ust,k)
     local p,m,t1,t2,t,alt_asallar,n,fark,fark_asallar,r,l,
           q, numbers,w,y,z, toplam,x,cift_sayilar,
           bulunamayanlar,ust_asallar,alt_ust_asallar,p2,
           m2,n2,alt2,ust2,fark2,w2,w4,w3;
    p:=0;
    p2:=0;
    fark:=ust-alt;
    fark2:=ust-alt;
    alt_asallar:=[];
    ust_asallar:=[];
    alt_ust_asallar:=[];
    fark_asallar:=[];
    numbers:=[2..fark];
    toplam:=[];
```

```
cift_sayilar:=[];
bulunamayanlar:=[];
t1:=Runtime();
m:=alt;
m2:=alt;
alt2:=alt;
ust2:=ust;
repeat
     if alt mod 2 = 0 then
          Add(cift_sayilar,alt);
     fi;
     alt:=alt+1;
until(alt=ust+1);

Sort(cift_sayilar);
Print("\nCift Sayilar = ",cift_sayilar,"\n");
repeat
     n:=PrevPrimeInt(m);
     Add(alt_asallar,n);
     m:=n;
     p:=p+1;
until(p=k/2);
Print("\nAlt_asallar = ",alt_asallar,"\n");
repeat
     n2:=NextPrimeInt(m2);
     Add(ust_asallar,n2);
     m2:=n2;
     p2:=p2+1;
until(p2=k/2);
Print("\nUst_asallar = ",ust_asallar,"\n");
Append(alt_ust_asallar,alt_asallar);
```

```
Append(alt_ust_asallar,ust_asallar);
Print("\nAlt_Ust_asallar = ",alt_ust_asallar,"\n");
for r in numbers do
     Add(fark_asallar,r);
     for l in numbers do
          if l mod r =0 then Unbind(numbers[l-1]);
          fi;
     od;
od;
Print("\nFark_asallar = ",fark_asallar,"\n");
for z in alt_ust_asallar do
     for y in fark_asallar do
          w:=z+y;
          if(IsEvenInt(w)=true) then
               if (alt2-1<w and w<ust2+1) then
                    if w in toplam then Unbind(w);
                         else Add(toplam,w);
                    fi;
               fi;
          fi;
     od;
od;
t2:=Runtime();
Print("\nToplam = ",toplam,"\n");
bulunamayanlar:=Difference(cift_sayilar,toplam);
Sort(bulunamayanlar);
if(bulunamayanlar=0) then
   Print("\nBulunamayanlar = Verilen aralikta
         Goldbach Cifti bulunamayan sayi
         yoktur!","\n");
else
```

```
        Print("\nBulunamayanlar =",bulunamayanlar,"\n");
        for w2 in bulunamayanlar do
             repeat
                  w3:=NextPrimeInt(fark2);
                  w4:=w2-w3;
                  if(IsPrimeInt(w4)=true) then
                       Print("\n",w2," = ",w4,"+",w3,"\n");
                       Unbind(w2);
                  else
                       fark2:=w3;
                  fi;
             until(IsPrimeInt(w4)=true);
             fark2:=fark;
        od;
        Print("\nVerilen aralikta Goldbach Cifti
              bulunamayan sayi kalmamistir!","\n");
        fi;
     t:=t2-t1;
     Print("\nTime elapsed ==> ",t," milisecond cpu time.
            \n");
end;
```

Bu yöntem ile $a = 950$, $b = 1000$, $k = 10$ değerleri için elde edilen örnek bir çıktı açıklama satırları ile şöyledir:

```
gap> Read("yontem2.txt");
gap> yontem2(950,1000,10);
Cift Sayilar = [ 950, 952, 954, 956, 958, 960, 962, 964,
966, 968, 970, 972, 974, 976, 978, 980, 982, 984, 986,
988, 990, 992, 994, 996, 998, 1000 ]
// [950,1000] aralığındaki çift sayılar Cift Sayilar
```

```
// listesinde tutulur.
Alt_asallar = [ 947, 941, 937, 929, 919 ]
// Alt_asallar listesi 950'den önceki, 10/2 = 5, 5 asal
// sayıyı içerir.
Ust_asallar = [ 953, 967, 971, 977, 983 ]
// Ust_asallar listesi 950'den sonraki, 10/2 = 5, 5 asal
// sayıyı içerir.
Alt_Ust_asallar = [ 947, 941, 937, 929, 919, 953, 967,
971, 977, 983 ]
// Alt_asallar ve Ust_asallar listesinin birleşimi
// Alt_Ust_asallar listesidir.
Fark_asallar = [ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31,
37, 41, 43, 47]
// Fark_asallar listesi, 1000-950 = 50, [2, 50]
// aralığındaki asal sayıları içerir.
Toplam = [ 950, 952, 954, 958, 960, 964, 966, 970, 976,
978, 984, 988, 990, 994, 972, 982, 956, 968, 974, 980,
962, 996, 1000, 986, 998 ]
// Alt_ust_asallar ve Fark_asallar listelerinin elemanları
// toplandığında elde edilen ve [950,1000] aralığında
// bulunan çift sayılar Toplam listesinde yer alır.
Bulunamayanlar = [ 992 ]
// Cift Sayilar listesinde yer alan fakat Toplam
listesinde
// yer almayan sayılar Bulunamayanlar listesinde tutulur.
992 = 919 + 73
Verilen aralikta Goldbach Cifti bulunamayan sayi
kalmamistir!
Time elapsed ==> 20 milisecond cpu time.
gap>
```

5.3. Yöntem 3 (MID Yöntemi)

Bu tez çalışması sırasında geliştirilen bir yöntemdir. 4. bölümde anlatılan $e(n)$ fonksiyonuna dayanır. ($e(n)$ fonksiyonu n değeri için, $n/2$ merkez alındığında en yakın simetrik asal sayı çiftinin merkezden uzaklığını vermektedir.) MID Yöntemi'nin algoritması Algoritma 5.3'te yer almaktadır. Bu algoritma, en kötü durum için, $O(kn) + a.\ e(n)$ zaman karmaşıklığına sahiptir. Burada $O(kn)$ birinci aşamanın yani Algoritma 5.1'in ilk dört adımının, a değeri ilk aşamada bulunamayan Goldbach Çiftlerinin adedinin, $e(n)$ ise ikinci aşamanın karmaşıklığıdır.

Algoritma 5.3 MID Yöntemi'nin Algoritması

1. *[a,b]* aralığı ve *k* sayısı belirle
2. Asal sayı listesini oluştur
   - *[a/2,b/2]* aralığındaki asal sayıları bul
   - *a/2'* den önceki *k/2* asal sayıyı bul
   - *b/2*'den sonraki *k/2* asal sayıyı bul
3. Asal sayıları sırayla birbiri ile toplayarak *[a,b]* aralığındaki çift sayıları bul
4. Bulunamayan çift sayıları listele
5. Bulunamayan çift sayıların her biri için *a/2*'den önceki ve *b/2*'den sonraki asal sayılar ile farkını al ve farkın asal sayı olup olmadığına bak
   - asal sayı ise listeye ekle
   - asal sayı değil ise bir sonraki sayıya geç

MID Yönteminin GAP kodu şöyledir:

```
mid:=function(alt,ust,k)
     local p,m,t1,t2,t,alt_asallar,n,r,l,q,w,y,z,toplam,x,
           cift_sayilar,bulunamayanlar,ust_asallar,
           alt_ust_asallar,p2,m2,n2,alt2,ust2,w2,w4,w3,w5,
           ara_asallar,m3,m4,n3,i,leng,x1,y1;
     p:=0;
     p2:=0;
     alt_asallar:=[];
     ust_asallar:=[];
     ara_asallar:=[];
     alt_ust_asallar:=[];
     toplam:=[];
     cift_sayilar:=[];
     bulunamayanlar:=[];
     m:=alt/2;
     m2:=ust/2;
     m3:=alt/2;
     m4:=ust/2;
     alt2:=alt;
     ust2:=ust;
     t1:=Runtime();
     repeat
          if alt mod 2 = 0 then
               Add(cift_sayilar,alt);
          fi;
               alt:=alt+1;
     until(alt=ust+1);
     Sort(cift_sayilar);
     Print("\nCift Sayilar = ",cift_sayilar,"\n");
```

```
    repeat
        n:=PrevPrimeInt(m);
        Add(alt_asallar,n);
        m:=n;
        p:=p+1;
    until(p=k/2);
    Sort(alt_asallar);
    Print("\nAlt_asallar = ",alt_asallar,"\n");
    repeat
        n2:=NextPrimeInt(m2);
        Add(ust_asallar,n2);
        m2:=n2;
        p2:=p2+1;
    until(p2=k/2);
    Print("\nUst_asallar = ",ust_asallar,"\n");
    repeat
        n3:=NextPrimeInt(m3);
        m3:=n3;
        if n3<=m4 then
            Add(ara_asallar,n3);
        else
            Unbind(n3);
        fi;
    until(m3>=m4+1);
    Print("\nAra_asallar = ",ara_asallar,"\n");
    Append(alt_ust_asallar,alt_asallar);
    Append(alt_ust_asallar,ara_asallar);
    Append(alt_ust_asallar,ust_asallar);
    Print("\nAlt_Ust_asallar = ",alt_ust_asallar,"\n");
    leng:=Length(alt_ust_asallar);
    for i in [1..leng] do
```

```
    for w2 in alt_ust_asallar do
        w3:=alt_ust_asallar[i]+w2;
        if(IsEvenInt(w3)=true) then
            if (alt2-1<w3 and w3<ust2+1) then
                if w3 in toplam then Unbind(w3);
                    else Add(toplam,w3);
                fi;
            fi;
        fi;
    od;
od;
Sort(toplam);
Print("\nToplam = ",toplam,"\n");
bulunamayanlar:=Difference(cift_sayilar,toplam);
Sort(bulunamayanlar);
if(bulunamayanlar=0) then
    Print("\nBulunamayanlar = Verilen aralikta
           Goldbach Cifti bulunamayan sayi
           yoktur!","\n");
else
    Print("\nBulunamayanlar =",bulunamayanlar,"\n");
    for w4 in bulunamayanlar do
        w5:=w4/2;
        x1:=w5-1;
        y1:=w5+1;
        if IsPrimeInt(x1) and IsPrimeInt(y1) then
            Print("\n",w4," = ",x1,"+",y1);
        else
            repeat
                x1:=x1-1;
                y1:=y1+1;
```

```
                    until(IsPrimeInt(x1) and
                          IsPrimeInt(y1));
                    Print("\n",w4," = ",x1,"+",y1);
               fi;
          od;
          Print("\nVerilen aralikta Goldbach Cifti
                bulunamayan sayi kalmamistir!","\n");
     fi;
     t2:=Runtime();
     t:=t2-t1;
     Print("\nTime elapsed ==> ",t," milisecond cpu
           time.\n");
end;
```

Bu yöntem ile a = 950, b = 1000, k = 10 değerleri içinelde edilen örnek bir çıktı açıklama satırları ile şöyledir:

```
gap> Read("mid");
gap> mid(950,1000,10);
     Cift Sayilar = [ 950, 952, 954, 956, 958, 960, 962,
     964, 966, 968, 970, 972, 974, 976, 978, 980, 982,
     984, 986, 988, 990, 992, 994, 996, 998, 1000 ]
// [950,1000] aralığındaki çift sayılar Cift Sayilar
// listesinde tutulur.
Alt_asallar = [ 449, 457, 461, 463, 467 ]
// 950/2 = 475'ten önceki 10/2 = 5 asal sayı
Ust_asallar = [ 503, 509, 521, 523, 541 ]
// 1000/2 = 500'den sonraki 10/2 = 5 asal sayı
Ara_asallar = [ 479, 487, 491, 499 ]
// [950/2,1000/2] = [475,500] aralığındaki asal
// sayılar
```

```
Alt_Ust_asallar = [ 449, 457, 461, 463, 467, 479, 487,
491, 499, 503, 509, 521, 523, 541 ]
// Alt_asallar, Ust_asallar ve Ara_asallar
// listelerinin birleşimi Alt_Ust_asallar listesidir.
Toplam = [ 950, 952, 954, 956, 958, 960, 962, 964, 966,
970, 972, 974, 976, 978, 980, 982, 984, 986, 988, 990,
994, 996, 998, 1000 ]
// Alt_Ust_asallar listesindeki elemanların toplanması
// ile elde edilen sayılar Toplam listesinde tutulur.
Bulunamayanlar = [ 968, 992 ]
// Cift Sayilar listesinde yer alan ama Toplam
// listesinde yer almayan sayılar Bulunamayanlar'dır.
968 = 421 + 547
992 = 421 + 571
    Verilen aralikta Goldbach Cifti bulunamayan sayi
    kalmamistir!
Time elapsed ==> 10 milisecond cpu time.
gap>
```

5.4 Yöntemlerin Karşılaştırılması

Yöntem 1, 2 ve 3 çeşitli aralıklarda farklı *k* değerleri için test edilmiştir. Testlerde $k = 50, 100, 250$ ve 500 değerleri alınmıştır.

İlk etapta aralık uzunluğu 10000 olarak seçilmiştir. Buna göre seçilen aralık başlangıçları: 10^4, 10^5, 2×10^5, 5×10^5, 10^6, 2×10^6, 5×10^6, 10^7, 2×10^7, 5×10^7, 10^8, 2×10^8, 5×10^8, 10^9, 2×10^9, 5×10^9, 10^{10}'dur.

Bu değerler Yöntem1, 2 ve 3'e uygulanmıştır. Her yöntem için elde edilen bulunamayan sayıların adetleri ve yöntemlerin çalışma sürelerine ilişkin sonuçlar Çizelge 5.1 ve 5.2'de gösterilmiştir.

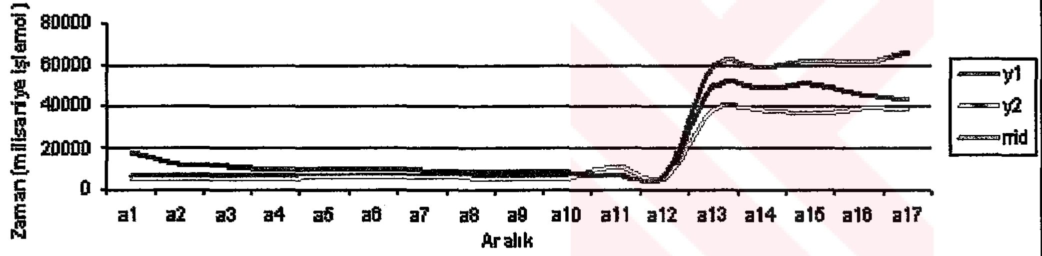
Bu çizelgelerde *y1* ifadesi ile Yöntem 1, *y2* ifadesi ile Yöntem 2 ve *mid* ifadesi ile Yöntem 3 belirtilmektedir. Zaman ölçümleri *milisaniye işlemci zamanı* cinsindendir.

		k = 50	k = 100	k = 250	k = 500
$[10^4, 2.10^4)$	y1	55	55	55	55
	y2	0	0	0	0
	mid	0	0	0	0
$[10^5, 10^5+10^4]$	y1	39	39	39	39
	y2	1	0	0	0
	mid	3	0	0	0
$[2.10^5, 2.10^5+10^4]$	y1	40	40	40	40
	y2	0	0	0	0
	mid	5	0	0	0
$[5.10^5, 5.10^5+10^4]$	y1	42	42	42	42
	y2	0	0	0	0
	mid	14	2	0	0
$[10^6, 10^6+10^4]$	y1	41	41	41	41
	y2	0	0	0	0
	mid	2	0	0	0
$[2.10^6, 2.10^6+10^4]$	y1	50	50	50	50
	y2	1	0	0	0
	mid	8	2	0	0
$[5.10^6, 5.10^6+10^4]$	y1	42	42	42	42
	y2	0	0	0	0
	mid	7	1	0	0
$[10^7, 10^7+10^4]$	y1	50	50	50	50
	y2	0	0	0	0
	mid	17	2	0	0
$[2.10^7, 2.10^7+10^4]$	y1	83	83	83	83
	y2	0	0	0	0
	mid	27	0	0	0

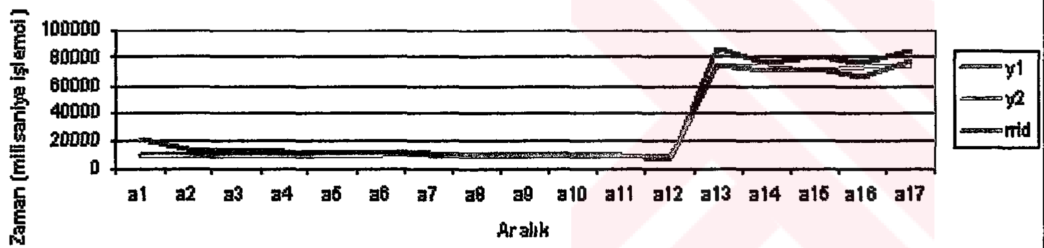
		k = 50	k = 100	k = 250	k = 500
$[5.10^7, 5.10^7+10^4]$	y1	61	61	61	61
	y2	0	0	0	0
	mid	10	0	0	0
$[10^8, 10^8+10^4]$	y1	56	56	56	56
	y2	0	0	0	0
	mid	9	2	0	0
$[2.10^8, 2.10^8+10^4]$	y1	44	43	43	43
	y2	1	0	0	0
	mid	29	0	0	0
$[5.10^8, 5.10^8+10^4]$	y1	78	78	78	78
	y2	0	0	0	0
	mid	44	6	0	0
$[10^9, 10^9+10^4]$	y1	87	87	87	87
	y2	0	0	0	0
	mid	30	6	0	0
$[2.10^9, 2.10^9+10^4]$	y1	57	57	57	57
	y2	0	0	0	0
	mid	39	5	0	0
$[5.10^9, 5.10^9+10^4]$	y1	75	75	75	75
	y2	0	0	0	0
	mid	68	15	0	0
$[10^{10}, 10^{10}+10^4]$	y1	70	70	70	70
	y2	2	0	0	0
	mid	46	3	0	0

Çizelge 5.2 10^4 aralık uzunluğunda farklı algoritma ve *k* değerleri için elde edilen çalışma zamanları

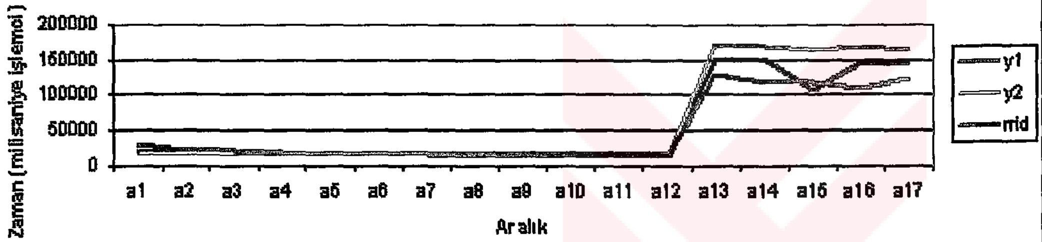
		k = 50	k = 100	k = 250	k = 500
$[10^4, 2.10^4]$	y1	6260	9290	20860	28480
	y2	5290	8110	16230	29200
	mid	17850	20950	29900	46060
$[10^5, 10^5+10^4]$	y1	6460	9330	18080	25310
	y2	5250	8130	16310	28200
	mid	12500	14880	22260	35990
$[2.10^5, 2.10^5+10^4]$	y1	6200	9180	18730	23980
	y2	5250	8100	16160	28100
	mid	11490	13670	20860	34080
$[5.10^5, 5.10^5+10^4]$	y1	6140	9170	18600	23630
	y2	5280	8060	16120	31760
	mid	10000	11860	18290	34080
$[10^6, 10^6+10^4]$	y1	6400	9350	16390	22730
	y2	5350	8170	16540	30560
	mid	8770	10720	17070	28110
$[2.10^6, 2.10^6+10^4]$	y1	6500	9420	16490	22750
	y2	5600	9560	16680	29730
	mid	9620	12010	18190	29820
$[5.10^6, 5.10^6+10^4]$	y1	6310	9780	16130	22450
	y2	5400	9170	16770	29020
	mid	9110	11130	17480	27950
$[10^7, 10^7+10^4]$	y1	6440	9370	15700	21890
	y2	5490	8890	16650	27410
	mid	8150	9390	15380	26210
$[2.10^7, 2.10^7+10^4]$	y1	6310	9170	15360	21450
	y2	5260	8070	16390	28790
	mid	8180	8060	14660	24740
$[5.10^7, 5.10^7+10^4]$	y1	6510	9450	12920	21720
	y2	5560	8330	16650	28400
	mid	7830	9670	15350	24960
$[10^8, 10^8+10^4]$	y1	6280	9290	14970	20630
	y2	11040	8290	16240	27910
	mid	6720	8280	12670	22280
$[2.10^8, 2.10^8+10^4]$	y1	6370	7570	15200	20490
	y2	5610	8370	17060	28640
	mid	7540	8070	13460	21740
$[5.10^8, 5.10^8+10^4]$	y1	49990	86070	149300	200[illegible]0
	y2	38110	73470	170880	2913[illegible]0
	mid	59610	75450	128760	2389[illegible]0
$[10^9, 10^9+10^4]$	y1	49260	77170	153510	1812[illegible]0
	y2	38190	72650	168080	2850[illegible]0
	mid	58620	70620	118230	226780
$[2.10^9, 2.10^9+10^4]$	y1	50840	81900	106160	164480
	y2	36490	70850	162180	277660
	mid	62360	72120	117430	205100
$[5.10^9, 5.10^9+10^4]$	y1	46220	77120	145170	173180
	y2	39170	74030	170140	288150
	mid	61020	65730	109160	191480
$[10^{10}, 10^{10}+10^4]$	y1	43580	84810	143170	159500
	y2	38420	72890	164530	272230
	mid	67140	77960	124930	206080



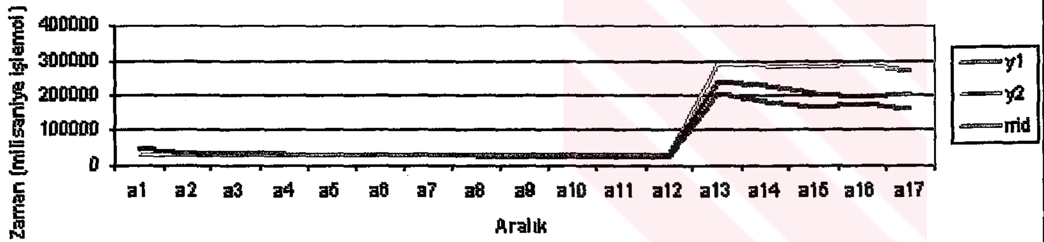
Şekil 5.1 k=50 için algoritmaların çalışma zamanları



Şekil 5.2 k=100 için algoritmaların çalışma zamanları



Şekil 5.3 k=250 için algoritmaların çalışma zamanları



Şekil 5.4 k=500 için algoritmaların çalışma zamanları

Şekil 5.1,..,5.4'te algoritmaların farklı k değerlerindeki çalışma zamanları ile ilgili grafikleri görülmektedir. Bu grafiklerde ki aralık başlangıç değerleri şöyledir;

$a1 = 10^4$	$a6 = 2\times10^6$	$a11 = 10^8$	$a16 = 5\times10^9$
$a2 = 10^5$	$a7 = 5\times10^6$	$a12 = 2\times10^8$	$a17 = 10^{10}$
$a3 = 2\times10^5$	$a8 = 10^7$	$a13 = 5\times10^8$	
$a4 = 5\times10^5$	$a9 = 2\times10^7$	$a14 = 10^9$	
$a5 = 10^6$	$a10 = 5\times10^7$	$a15 = 2\times10^9$	

k' nın küçük değerleri için *y2* diğer yöntemlerden daha hızlıdır. k = 50 (Şekil 5.1) değeri için bütün aralıklarda, k = 100 (Şekil 5.2) için ise $[2\times10^6, 2\times10^6+10^4]$ aralığına kadar *y2*'nin çalışma zamanı diğer yöntemlere göre daha iyidir. Fakat k değeri arttırıldıkça *y2*'nin hızı *y1*'in gerisine düşmektedir. *mid* ise bütün k değerleri için her zaman *y1*'in gerisinde kalırken aralık başlangıcı büyüdükçe, k = 250 ve 500 (Şekil 5.3 ve 5.4) değerleri için, *y2*'yi geçmektedir. Çizelge 5.2'de $[10^6, 10^6+10^4]$ aralığından $[10^7, 10^7+10^4]$ aralığına kadar sadece k = 500 değerinde, $[2\times10^7, 2\times10^7+10^4]$ aralığından $[10^{10}, 10^{10}+10^4]$ aralığına kadar da k = 250 ve 500 değerlerinde *mid*, *y2*'den daha hızlıdır. k = 500 için elde edilen zamanlama ölçümleri karşılaştırıldığında *mid/y2* oranının gittikçe düştüğü görülür. Örneğin Çizelge 5.2'de k = 500 için $[10^4, 2\times10^4]$ aralığında *mid/y2* oranı 46060/29200 = 1,57, $[10^5, 10^5+10^4]$ aralığında bu oran 1,27 ve $[10^{10}, 10^{10}+10^4]$ aralığında 0,76'dır.

y2, test genelinde verilen aralıkta Goldbach Çifti bulunamayan sayı miktarı açısından en verimli yöntemdir. *y2*'den sonra en az bulunamayan sayı miktarına sahip olan *mid* yöntemidir. *y1* diğer yöntemlere göre daha az zaman harcamasına rağmen bu yöntemde verilen bir aralıkta Goldbach Çifti bulunamayan sayıların adedi diğer yöntemlere göre daha fazladır. Değişen *k* değerleri için, *y2* ve *mid*'in aksine *y1* için ile elde edilen Goldbach Çifti bulunamayan çift sayıların miktarında herhangi bir değişiklik görülmemektedir. Bu değişmezlik *y1*'in algoritmasındaki (Algoritma 5.1) 2. adımdan kaynaklanmaktadır. *k* değeri ne kadar büyültülürse büyültülsün, *a*'dan önceki *k* tane asal sayı 3. adımdaki asal sayılarla toplandığında elde edilen çift sayılar listesine eklenen yeni elamanlar *[a, b]* aralığında yer alan çift sayılardan küçüktür. Çünkü algoritmanın 3. adımındaki asal sayı listesi *k* değerine değil *[a, b]* aralığına bağlıdır ve bu nedenle *k* değeri değişsede bu liste sabit kalır. Bu sebepten dolayı *y1*'in algoritması düzenlenmiş ve *y2*'nin algoritması (Algoritma 5.2) elde edilmiştir. Bu algoritmada *a*'dan önceki ve sonraki *k/2* asal sayı bulunarak elde edilecek çift sayıların *[a, b]* aralığındaki çift sayılara denk gelmesi sağlanmıştır.

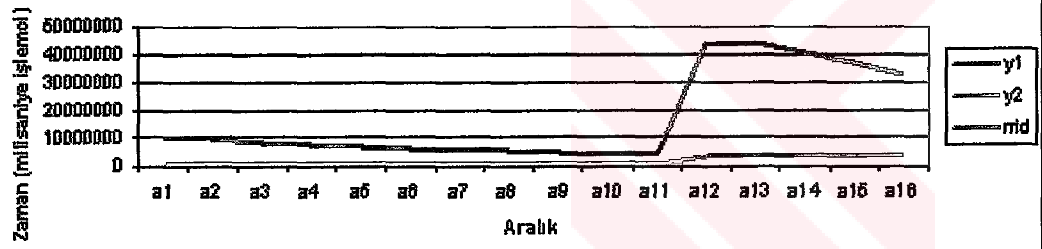
y1 için Goldbach Çifti bulunamayan çift sayıların miktarında değişiklik olmaması nedeniyle aralık uzunluğu 100000 olarak büyültülmüş ve bu değişmezliğin aralık büyüdüğünde de devam edip etmediğine bakmak için 10^5, 2×10^5, 5×10^5, 10^6, 2×10^6, 5×10^6, 10^7, 2×10^7, 5×10^7, 10^8, 2×10^8, 5×10^8, 10^9, 2×10^9, 5×10^9 ve 10^{10} aralık başlangıçları ile k = 50, 100, 250 ve 500 değerleri alınarak yapılan test tekrarlanmıştır. Bu değerler ile elde edilen sonuçlar Çizelge 5.3 ve 5.4'te görülmektedir.

Çizelge 5.3 10^5 aralık uzunluğunda farklı algoritma ve *k* değerleri için ilk aşamada Goldbach Çifti bulunamayan sayıların adedi

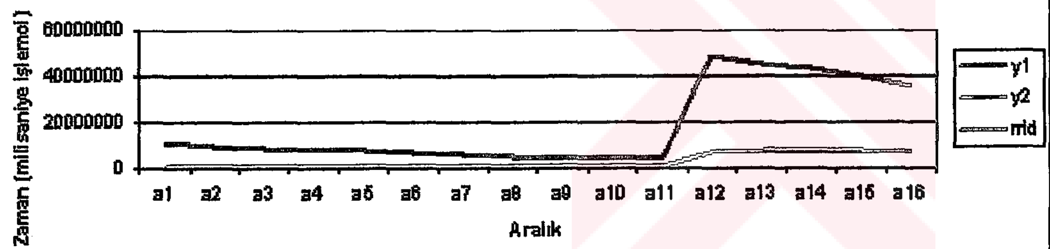
		k = 50	k = 100	k = 250	k = 500			k = 50	k = 100	k = 250	k = 500
$[10^5, 10^5+10^5]$	y1	69	59	59	59	$[5.10^7, 5.10^7+10^5]$	y1	67	61	61	61
	y2	16	0	0	0		y2	17	0	0	0
	mid	2	0	0	0		mid	19	1	0	0
$[2.10^5, 3.10^5]$	y1	53	41	41	41	$[10^8, 10^8+10^5]$	y1	63	70	70	70
	y2	10	0	0	0		y2	13	0	0	0
	mid	6	0	0	0		mid	7	1	0	0
$[5.10^5, 6.10^5]$	y1	51	43	43	43	$[2.10^8, 2.10^8+10^5]$	y1	96	51	51	51
	y2	18	0	0	0		y2	20	0	0	0
	mid	12	2	0	0		mid	36	4	0	0
$[10^6, 10^6+10^5]$	y1	64	40	40	40	$[5.10^8, 5.10^8+10^5]$	y1	67	80	80	80
	y2	21	0	0	0		y2	12	0	0	0
	mid	1	0	0	0		mid	31	2	0	0
$[2.10^6, 2.10^6+10^5]$	y1	53	45	45	45	$[10^9, 10^9+10^5]$	y1	100	89	89	89
	y2	13	0	0	0		y2	13	0	0	0
	mid	3	0	0	0		mid	40	0	0	0
$[5.10^6, 5.10^6+10^5]$	y1	54	48	48	48	$[2.10^9, 2.10^9+10^5]$	y1	101	73	73	73
	y2	16	0	0	0		y2	24	0	0	0
	mid	8	1	0	0		mid	45	7	0	0
$[10^7, 10^7+10^5]$	y1	63	53	53	53	$[5.10^9, 5.10^9+10^5]$	y1	92	81	81	81
	y2	12	0	0	0		y2	14	0	0	0
	mid	12	2	0	0		mid	75	15	0	0
$[2.10^7, 2.10^7+10^5]$	y1	165	85	85	85	$[10^{10}, 10^{10}+10^5]$	y1	141	85	85	85
	y2	10	0	0	0		y2	10	0	0	0
	mid	15	0	0	0		mid	69	6	0	0

Çizelge 5.4 10^5 aralık uzunluğunda farklı algoritma ve *k* değerleri için elde edilen çalışma zamanları

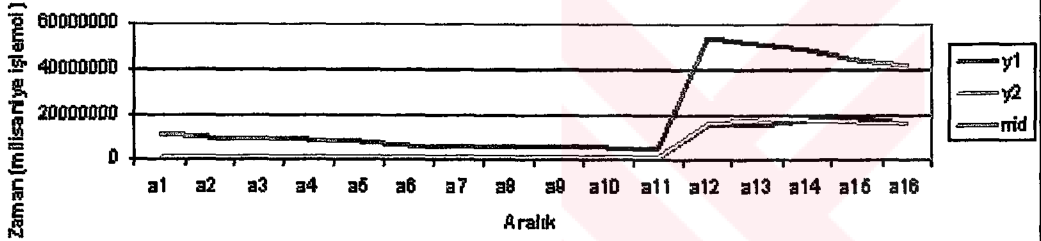
		k = 50	k = 100	k = 250	k = 500			k = 50	k = 100	k = 250	k = 500
$[10^5, 10^5+10^5]$	y1	422990	673690	1419380	2563580	$[5.10^7, 5.10^7+10^5]$	y1	452340	678060	1376230	2510[illegible]
	y2	430100	685240	1462770	2710280		y2	429220	687600	1470420	2712530
	mid	10507730	10786720	11547000	12851060		mid	4561040	4702190	5195690	6011950
$[2.10^5, 3.10^5]$	y1	415480	671290	1408350	2577730	$[10^8, 10^8+10^5]$	y1	421060	672110	1386630	2514560
	y2	429210	696510	1454810	2685540		y2	423530	677760	1451340	2558660
	mid	9631760	9856570	10566290	11876360		mid	4276960	4400830	4919550	5748060
$[5.10^5, 6.10^5]$	y1	421260	674620	1400110	2566320	$[2.10^8, 2.10^8+10^5]$	y1	416840	670820	1390950	2439520
	y2	431160	690180	1438810	2687880		y2	423000	681610	1450480	2708770
	mid	9313920	8525000	9247270	10301590		mid	4118200	4196820	4667960	5439840
$[10^6, 10^6+10^5]$	y1	414560	678900	1402260	2573390	$[5.10^8, 5.10^8+10^5]$	y1	3420100	6716580	15496810	3561[illegible]10
	y2	426170	682670	1459850	2734430		y2	3343400	6661870	16339160	3126[illegible]60
	mid	7682200	7878890	8525850	9638330		mid	43168710	46627940	53891920	6311[illegible]10
$[2.10^6, 2.10^6+10^5]$	y1	422720	673560	1406410	2514970	$[10^9, 10^9+10^5]$	y1	3647020	7427950	15272020	34729290
	y2	432660	684920	1445070	2726630		y2	3615730	7211690	17634400	35558610
	mid	7036990	7189860	7833270	8928760		mid	43870290	45478960	50874830	60206000
$[5.10^6, 5.10^6+10^5]$	y1	422200	675840	1385430	2510610	$[2.10^9, 2.10^9+10^5]$	y1	3145380	6770170	17857380	33773350
	y2	428040	678040	1456750	2699330		y2	3513210	7427750	17977650	36893470
	mid	6142060	6346240	6943230	7889470		mid	40989580	43075560	49025900	56842540
$[10^7, 10^7+10^5]$	y1	426460	674630	1409170	2508430	$[5.10^9, 5.10^9+10^5]$	y1	3649900	7354250	18456300	33973160
	y2	431780	679540	1439900	2632070		y2	3912100	7724890	16795930	36654360
	mid	5532780	5692120	6254840	7126830		mid	37641000	39480000	44584330	52802970
$[2.10^7, 2.10^7+10^5]$	y1	414770	661710	1367480	2483150	$[10^{10}, 10^{10}+10^5]$	y1	3122490	6291900	15991150	31423980
	y2	415770	650040	1373850	2593140		y2	3264020	6446770	16018440	37353620
	mid	5289100	5457130	6019650	6893940		mid	32721440	36155520	41237260	50545530



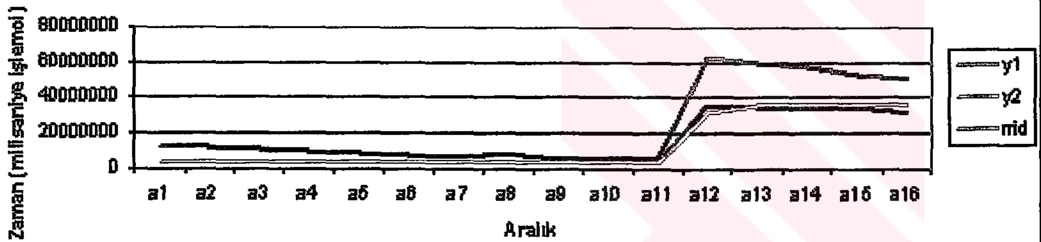
Şekil 5.5 k=50 için algoritmaların çalışma zamanları



Şekil 5.6 k=100 için algoritmaların çalışma zamanları



Şekil 5.7 k=250 için algoritmaların çalışma zamanları



Şekil 5.8 k=500 için algoritmaların çalışma zamanları

Şekil 5.5,..,5.8'de algoritmaların farklı k değerlerindeki çalışma zamanları ile ilgili grafikleri görülmektedir. Bu grafiklerdeki aralık başlangıç değerleri şöyledir;

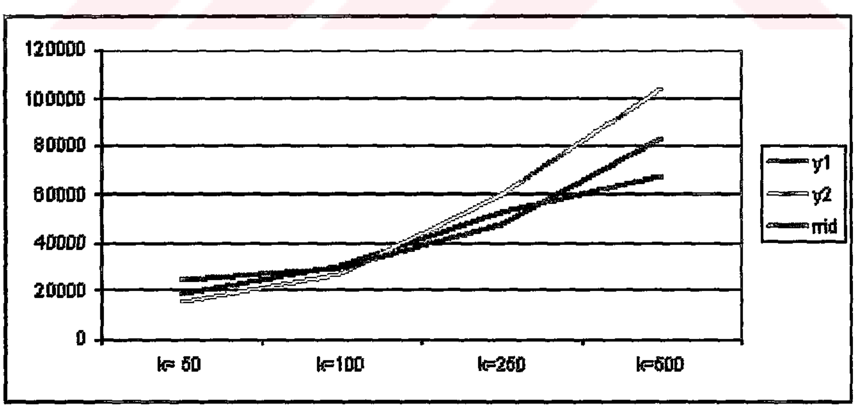
$a1 = 10^5$	$a6 = 5\times10^6$	$a11 = 2\times10^8$	$a16 = 10^{10}$
$a2 = 2\times10^5$	$a7 = 10^7$	$a12 = 5\times10^8$	
$a3 = 5\times10^5$	$a8 = 2\times10^7$	$a13 = 10^9$	
$a4 = 10^6$	$a9 = 5\times10^7$	$a14 = 2\times10^9$	
$a5 = 2\times10^6$	$a10 = 10^8$	$a15 = 5\times10^9$	

Aralık uzunluğu arttırıldığında elde edilen bu yeni sonuçlarda *y1*'in verilen aralıkta Goldbach Çifti bulunamayan çift sayılar adedi bütün aralıklarda k = 100 değerinde azalma göstermiş fakat k = 250 ve 500 değerlerinde yine aynı kalmıştır. Bulunamayan çift sayıların adedi açısından *y2* en verimli yöntemdir. Bütün aralıklar için k = 100 değerinde *y2* ile Goldbach Çifti bulunamayan çift sayı kalmamaktadır. k = 50 için küçük aralık başlangıçlarına da *mid*, *y2*'den daha iyidir, fakat k aralık başlangıçları büyüdükçe *y2*, *mid*'den daha verimli olmaktadır. Mid yöntemi en fazla k = 100'e kadar bulunamayan sayı içerir, k = 250'den itibaren *mid* ile bulunamayan çift sayı kalmamaktadır.

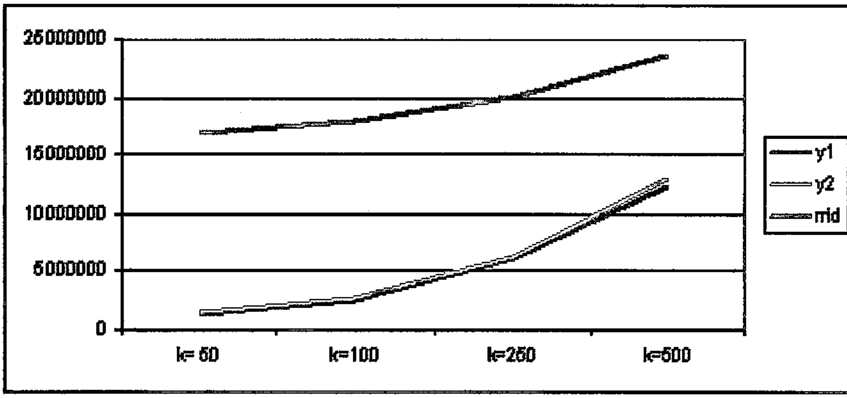
Hız bakımından *y1* bu test sonuçlarında da liderliğini sürdürmüştür. *y2*'nin çalışma zamanı olarak *y1* ile başa baş gitmektedir. Bir önceki test sonuçlarında *mid*'in hızı *y2*'yi k = 250 ve 500 için aralık başlangıçları büyüdüğünde geçmişti fakat bu testte *mid*, *y2*'yi hiçbir aralıkta ve k değerinde geçememiştir. Bu test sonuçlarına göre bir hız sıralaması yapıldığında $y1$, $y2$ ve *mid* şeklinde bir sıralama elde edilir.

Sonuç olarak hız açısından bakıldığında $y1$, Goldbach Çifti bulunamayan çift sayıların adedinin azlığı bakımından $y2$ verimlidir. *Mid* ise hız açısından kötü olmasına rağmen $y1$'den daha az bulunamayan sayı içermesinden dolayı tercih edilebilir.

Aralık uzunluğu 10^4'ten 10^5'e arttırıldığında, üç algoritma için de Goldbach Çifti bulunamayan çift sayıların adedinde bir artma olduğu görülmektedir. Sadece $y1$ için sabit kalan bulunamayan sayıların adedi $k = 100$ değerinde bir azalma göstermekte fakat $k = 250$ ve 500 değerleri için yine sabit kalmaktadır. Aralık uzunluğu arttırıldığında, üç algoritmanın da çalışma zamanları artmıştır. Şekil 5.9 ve 5.10 yöntemlerin farklı k değerleri için ortalama çalışma zamanlarını göstermektedir.



Şekil 5.9 10^4 aralık uzunluğu için yöntemlerin ortalama çalışma zamanları



Şekil 5.10 10^5 aralık uzunluğu için yöntemlerin ortalama çalışma zamanları

5.5 Sonuçların İrdelenmesi

Bu bölümde Goldbach Sanısı'nın verilen bir aralıkta doğrulanması için üç yönteme bakılmış ve çeşitli aralıklarda her bir yöntem için sonuçlar alınmıştır.

- ✓ Deshouillers ve te Riele tarafından geliştirilen Yöntem 1 hız açısından en iyi yöntem olmasına rağmen verilen bir aralıkta *k* değeri artmasına rağmen bulunamayan sayılar adedinin azalmamasından dolayı verimli değildir.

- ✓ Yöntem 2 hız açısından Yöntem 1 kadar iyi olmasada verilen aralıkta *k* değeri arttıkça Goldbach Çifti bulunamayan sayıların adedinin azalması açısından en iyi yöntemdir..

bulunamayan sayıların adedinin azalması açısından en iyi yöntemdir..

✓ Mid hız açısından en yavaş çalışan yöntemdir. Verilen aralıkta *k* değeri arttıkça bulunamayan sayılar azalmaktadır, bu açıdan Yöntem 1'den daha verimlidir fakat Yöntem 2 kadar da iyi değildir.

6. VERİLEN BİR ARALIKTA TÜM GOLDBACH ÇİFTLERİNİN BULUNMASI

Bu bölümde verilen bir aralıkta tüm Goldbach Çiftlerinin bulunması işlemi üzerinde durulmuştur. Bir çift sayının tüm Goldbach çiftlerinin sayısı $f(n)$ fonksiyonu olarak adlandırılmıştır.

Goldbach Çifti ***(Goldbach Partition)*****:** Dörtten büyük bir çift sayı, n, iki asal sayının toplamı şeklinde ifade edilebiliyorsa bu asal sayı çiftine n'nin bir *Goldbach Çifti* denir.

$$n = p_1 + p_2 \text{ , } p_1 \le p_2 \text{ , } p_1 \text{ ve } p_2 \text{ asal sayı} \quad \Rightarrow \quad (p_1, p_2) \text{ bir Goldbach Çiftidir}$$

Örnek 6.1:

n = 10 için;

10 = 3 + 7 = 5 + 5 → f(10) = 2

n = 100 için;

100 = 3 + 97 = 11 + 89 = 17 + 83

= 29 + 71 = 41 + 59 = 47 + 53 → f(100) = 6

n = 1000 → f(1000) = 28

n = 10000 → f(10000) = 127

n = 100000 → f(100000) = 810

n = 1000000 → f(1000000) = 5402

$f(n)$ fonksiyonu hesaplanırken elemanları aynı olan çiftler bir kez dikkate alınmaktadır. Örneğin (3,7) çifti dikkate alındı ise (7,3) çift alınmamıştır. $f(n)$ fonksiyonu, en kötü durum için, $O(n^2\sqrt{n})$ zamar karmaşıklığına sahiptir.

6.1 f(n) Fonksiyonu ile İlgili Uygulamalar

6.1.1 f(n) Değerleri için Artış-Azalışlara Bakılması

Elde edilen $f(n)$ değerleri üzerinde yapılan çalışmalardan bir değerlerin artış-azalış karşılaştırmasıdır.

Eğer $f(n) > f(n+2)$ ise bir azalma vardır ve $f(n+2)$ değeı işaretlenir. Çizelge 6.1'de 1000'e kadar olan çift sayılar için azalmanı olduğu $f(n)$ değerleri görülmektedir. Azalış değerleri kırmızı il belirtilmiştir.

Eğer $f(n) < f(n+2)$ ise bir artış vardır ve $f(n+2)$ değe işaretlenir. Çizelge 6.2'de 1000'e kadar olan çift sayılar için artışları olduğu $f(n)$ değerleri kırmızı renk ile belirtilmiştir.Her iki çizelgede c kırmızı renkli rakamlar izlendiğinde bunların bir diagonal çizgi üzerir düştüğü gözlenmektedir. Çizelge 6.2'de ise yeşil renk ile belirtile f(912)=31'de bu desenin bozulduğu görülmektedir. Çizelge 6.2'de dikk çeken başka bir noktada diagonal çizgilerin altında ya da üstünc birbirine eşit iki değerin bulunmasıdır. Bu durumdaki rakamlar çizelgeı mavi renk ile belirtilmiştir.

Çizelge 6.1 Azalışların olduğu $f(n)$ değerleri tablosu

ARALIK f(n) - [2-1000]										
2-20	0	1	1	1	2	1	2	2	2	2
22-40	3	3	3	2	3	2	4	4	2	3
42-60	4	3	4	5	4	3	5	3	4	6
62-80	3	5	6	2	5	6	5	5	7	4
82-100	5	8	5	4	9	4	5	7	3	6
102-120	8	5	6	8	6	7	10	6	6	12
122-140	4	5	10	3	7	9	6	5	8	7
142-160	8	11	6	5	12	4	8	11	5	8
162-180	10	5	6	13	9	6	11	7	7	14
182-200	6	8	13	5	8	11	7	9	13	8
202-220	9	14	7	7	19	6	8	13	7	9
222-240	11	7	7	12	9	7	15	9	9	18
242-260	8	9	16	6	9	16	9	8	14	10
262-280	9	16	8	9	19	7	11	16	7	14
282-300	16	8	12	17	10	8	19	8	11	21
302-320	9	10	15	8	12	17	9	10	15	11
322-340	11	20	7	10	24	6	11	19	9	13
342-360	17	10	9	16	13	10	20	9	10	22
362-380	8	14	18	8	14	18	10	11	22	13
382-400	10	19	12	9	27	11	11	21	7	14
402-420	17	11	13	20	13	11	21	10	11	30
422-440	11	12	21	9	14	19	13	11	21	14
442-460	13	21	12	13	27	12	12	24	9	16
462-480	28	12	13	24	15	13	23	14	11	29
482-500	11	14	23	9	19	22	13	13	23	13
502-520	15	27	15	14	32	11	14	23	11	17
522-540	24	11	15	25	14	17	22	13	14	30
542-560	10	13	30	11	19	23	11	11	23	18
562-580	14	24	13	13	31	11	16	26	12	19
582-600	25	12	13	29	16	15	27	12	15	32
602-620	12	14	27	13	20	26	15	19	26	18
622-640	17	31	12	16	41	10	14	28	15	18
642-660	25	17	16	27	21	15	29	13	19	41
662-680	14	16	31	11	21	33	15	17	28	21
682-700	16	30	16	16	39	11	19	30	14	24
702-720	31	18	19	24	16	17	37	14	15	39
722-740	14	15	31	15	21	31	15	19	29	18
742-760	19	31	18	19	39	14	17	35	15	21
762-780	30	17	17	31	26	18	32	16	15	44
782-800	14	18	30	15	22	34	17	14	38	21
802-820	16	32	16	14	39	18	20	34	17	20
822-840	29	16	21	34	22	22	33	18	17	51
842-860	18	17	32	15	25	31	20	19	39	18
862-880	17	33	17	21	46	18	19	36	14	25
882-900	39	21	18	37	23	19	34	20	19	48
902-920	15	17	34	15	31	31	20	18	35	23
922-940	20	47	18	18	43	17	20	36	18	24
942-960	34	18	20	33	25	23	37	19	22	45
962-980	16	18	45	17	27	32	17	19	35	26
982-1000	17	39	20	23	52	13	25	37	17	28

Çizelge 6.2 Artışların olduğu $f(n)$ değerleri tablosu

ARALIK f(n) - [2-1000]										
2-20	**0**	1	1	1	2	**1**	2	2	2	2
22-40	3	3	3	**2**	3	**2**	4	**4**	**2**	3
42-60	4	**3**	4	5	**4**	**3**	5	**3**	4	6
62-80	**3**	5	6	**2**	5	6	5	5	7	**4**
82-100	5	8	**5**	**4**	9	**4**	5	7	**3**	6
102-120	8	**5**	6	8	**6**	7	10	6	6	12
122-140	**4**	5	10	**3**	7	9	**6**	**5**	8	**7**
142-160	8	11	**6**	**5**	12	**4**	8	11	**5**	8
162-180	10	**5**	**6**	13	**9**	**6**	11	7	7	14
182-200	**6**	8	13	**5**	8	11	**7**	9	13	**8**
202-220	9	14	7	7	19	**6**	8	13	**7**	9
222-240	11	7	7	12	**9**	**7**	15	9	9	18
242-260	**8**	9	16	**6**	9	16	**9**	**8**	14	**10**
262-280	**9**	16	**8**	9	19	**7**	11	16	**7**	14
282-300	16	**8**	12	17	**10**	**8**	19	**8**	11	21
302-320	**9**	10	15	**8**	12	17	**9**	10	15	**11**
322-340	**11**	20	**7**	10	24	**6**	11	19	**9**	13
342-360	17	**10**	**9**	16	**13**	**10**	20	**9**	10	22
362-380	**8**	14	18	**8**	14	18	**10**	11	22	**13**
382-400	**10**	19	**12**	**9**	27	11	11	21	**7**	14
402-420	17	**11**	13	20	**13**	**11**	21	**10**	11	30
422-440	**11**	12	21	**9**	14	19	**13**	**11**	21	**14**
442-460	**13**	21	**12**	13	27	12	12	24	**9**	16
462-480	28	**12**	13	24	**15**	**13**	23	**14**	**11**	29
482-500	**11**	14	23	**9**	19	22	13	13	23	**13**
502-520	15	27	**15**	**14**	32	**11**	14	23	**11**	17
522-540	24	**11**	15	25	**14**	17	22	**13**	14	30
542-560	**10**	13	30	**11**	19	23	11	11	23	**18**
562-580	**14**	24	13	13	31	**11**	16	26	**12**	19
582-600	25	**12**	13	29	**16**	**15**	27	**12**	15	32
602-620	**12**	14	27	**13**	20	26	**15**	19	26	**18**
622-640	**17**	31	**12**	16	41	**10**	14	28	**15**	18
642-660	25	**17**	**16**	27	**21**	**15**	29	**13**	19	41
662-680	**14**	16	31	**11**	21	33	**15**	17	28	**21**
682-700	**16**	30	16	16	39	**11**	19	30	**14**	24
702-720	31	**18**	19	24	**16**	17	37	**14**	15	39
722-740	**14**	15	31	**15**	21	31	**15**	19	29	**18**
742-760	19	31	**18**	19	39	**14**	17	35	**15**	21
762-780	30	17	17	31	**26**	**18**	32	**16**	**15**	44
782-800	**14**	18	30	**15**	22	34	**17**	**14**	38	**21**
802-820	**16**	32	**16**	**14**	39	**18**	20	34	**17**	20
822-840	29	**16**	21	34	22	22	33	**18**	**17**	51
842-860	**18**	**17**	32	**15**	25	31	**20**	**19**	39	**18**
862-880	**17**	33	**17**	21	46	**18**	**19**	36	**14**	25
882-900	39	**21**	**18**	37	**23**	**19**	34	**20**	**19**	48
902-920	**15**	17	34	**15**	31	31	**20**	**18**	35	**23**
922-940	**20**	47	18	18	43	**17**	**20**	36	**18**	24
942-960	34	**18**	20	33	**25**	**23**	37	**19**	22	45
962-980	**16**	18	45	**17**	27	32	**17**	19	35	**26**
982-1000	**17**	39	**20**	23	52	**13**	25	37	**17**	28

6.1.2 f(n) Değerlerini Kodlama

$f(n)$ fonksiyonu değerleri için kodlama yapılarak dosyaların sıkıştırılma oranlarına bakılmıştır. Sıkıştırma işlemine sadece depolama ya da yerden kazanma olarak bakılmamalıdır. Rastgele dizilerin sıkıştırılma oranları düşük, tekrarlamaların olduğu dizilerin sıkıştırılma oranları yüksektir. Veri ne kadar iyi sıkışıyorsa o kadar yapısaldır.

Yapılan işlem şöyledir:

$f(n) < f(n+2)$ ise $f(n)$ dosyada büyük harf ile

$f(n) = f(n+2)$ ise $f(n)$ dosyada sıfır ile

$f(n) > f(n+2)$ ise $f(n)$ dosyada küçük harf ile kodlanmıştır.

$$\Delta(n) = \frac{f(n+2) - f(n)}{f(n)}$$

Fark fonksiyonu Δ aşağıdaki değerler kullanılarak kodlanmıştır.

$0 < \Delta(n) < 0,1$	→ A, a	$1,6 \leq \Delta(n) < 1,8$	→ N,n
$0,1 \leq \Delta(n) < 0,2$	→ B, b	$1,8 \leq \Delta(n) < 2$	→ O,o
$0,2 \leq \Delta(n) < 0,3$	→ C,c	$2 \leq \Delta(n) < 2,5$	→ Q,q
$0,3 \leq \Delta(n) < 0,4$	→ D,d	$2,5 \leq \Delta(n) < 3$	→ P,p
$0,4 \leq \Delta(n) < 0,5$	→ E,e	$3 \leq \Delta(n) < 3,5$	→ R,r
$0,5 \leq \Delta(n) < 0,6$	→ F,f	$3,5 \leq \Delta(n) < 4$	→ S,s
$0,6 \leq \Delta(n) < 0,7$	→ G,g	$4 \leq \Delta(n) < 5$	→ T,t

$0,7 \le \Delta(n) < 0,8$ → H,h	$5 \le \Delta(n) < 6$ → U,u
$0,8 \le \Delta(n) < 0,9$ → I,i	$6 \le \Delta(n) < 7$ → V,v
$0,9 \le \Delta(n) < 1$ → J,j	$7 \le \Delta(n) < 8$ → W,w
$1 \le \Delta(n) < 1,2$ → K,k	$8 \le \Delta(n) < 9$ → X,x
$1,2 \le \Delta(n) < 1,4$ → L,l	$9 \le \Delta(n) < 10$ → Y,y
$1,4 \le \Delta(n) < 1,6$ → M,m	$10 \le \Delta(n)$ → Z,z

Örnek 6.2:

f(80078) = 503

f(80080) = 1006

f(80082) = 1005

f(80084) = 495

$$\Delta(80078) = \frac{1006 - 503}{503} = 1 \text{ ve } f(80078) < f(80080) \rightarrow K$$

$$\Delta(80080) = \frac{1005 - 1006}{1006} \cong -0,001 \text{ ve } f(80080) > f(80082) \rightarrow a$$

$$\Delta(80082) = \frac{495 - 1005}{1005} \cong -0,5 \text{ ve } f(80082) > f(80084) \rightarrow f$$

Bu durumda $f(80078)$, $f(80080)$ ve $f(80082)$ değerleri için kodlaına *K, a, f* şeklinde olur.

[6, 10^6] aralığındaki $f(n)$ değerleri için bu kodlama yapıldığında elde edilen dosyanın sıkıştırılma oranı % 76'dır. Dosya, herhangi bir desen olup olmadığını incelemek için öncelikle üç daha sonra da altı sütuna ayrılmış ve sıkıştırılma oranlarına tekrar bakılmıştır. Üç sütun için bakıldığında; 1.sütun % 75, 2.sütun % 79 ve 3.sütun % 87 oranında sıkıştılmıştır. Altı sütun için bakıldığında ise; 1.sütun % 75, 2.sütun % 73, 3.sütun % 83, 4.sütun % 75, 5.sütun % 73 ve 6.sütun % 83 oranında sıkıştılmıştır.

Rastgele bir dosyanın sıkıştırılabilirliği ile yukarıda belirtilen tek dosyanın sıkıştırılabilme oranları karşılaştırıldığında;
Olası 53 karakter (26 küçük harf + 26 büyük harf + sıfır) için $\log_2 53 \cong 5.7$ 'dir. Bu da rastgele bir dosyanın sıkıştırılabilirliğinin %28.75 olduğunu gösterir. Yukarıda bahsedilen dosyanın sıkıştırılabilirliği ise %76 olup rastgele dağılımlı bir dosyadan çok daha iyi bir değerdir. Bu da bize $f(n)$ değerlerinin önemli ölçüde bir yapısallığa sahip olduğunu, dolayısı ile bir desen taşıma olasılığının yüksek olduğunu göstermektedir.

Çizelge 6.3, bu uygulamanın [6, 10^6] aralığındaki $f(n)$ değerleri için elde edilen sonuçlarını göstermektedir.

Çizelge 6.3 $[6, 10^6]$ aralığındaki $f(n)$ değerleri için kodlama sonuçları

	6-10^5	10^5-2.10^5	2.10^5-3.10^5	3.10^5-4.10^5	4.10^5-5.10^5	5.10^5-6.10^5	6.10^5-7.10^5	7.10^5-8.10^5	8.10^5-9.10^5	9.10^5-10^6		
	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	TOPLAM	DEĞERLER
0	205	71	61	47	38	30	28	20	26	26	552	
a	2849	3180	3195	3143	3260	3339	3167	3140	3353	3396	32022	[0,0.1]
A	3508	3551	3529	3601	3513	3496	3620	3679	3534	3433	35464	[0,0.1]
b	2199	2160	2151	2158	2145	2126	2119	2108	2089	2188	21443	[0.1,0.2]
B	1847	1669	1643	1678	1662	1640	1623	1653	1632	1627	16674	[0.1,0.2]
c	2511	2674	2681	2665	2709	2683	2722	2725	2708	2693	26771	[0.2,0.3]
C	1610	1540	1528	1517	1465	1515	1567	1539	1493	1489	15263	[0.2,0.3]
d	3062	3129	3158	3191	3167	3251	3182	3140	3188	3213	31661	[0.3,0.4]
D	1750	1785	1799	1826	1871	1809	1818	1834	1841	1811	18144	[0.3,0.4]
e	4544	4680	4720	4705	4806	4808	4712	4765	4806	4787	47333	[0.4,0.5]
E	1746	1672	1671	1518	1494	1502	1512	1519	1477	1478	15389	[0.4,0.5]
f	5753	5711	5737	5718	5666	5593	5728	5677	5643	5635	56861	[0.5,0.6]
F	1342	1266	1250	1244	1244	1237	1244	1212	1258	1259	12556	[0.5,0.6]
g	2861	2790	2754	2771	2733	2749	2756	2757	2731	2744	27646	[0.6,.07]
G	1257	1290	1266	1275	1282	1287	1300	1337	1304	1303	12901	[0.6,.07]
h	179	139	126	117	124	119	126	128	125	118	1301	[0.7,0.8]
H	1223	1064	1055	1032	1022	969	999	987	957	935	10233	[0.7,0.8]
i	0	0	0	0	0	0	0	0	0	0	0	[0.8,0.9]
I	1507	1269	1240	1244	1208	1282	1268	1243	1323	1298	12882	[0.8,0.9]
j	0	0	0	0	0	0	0	0	0	0	0	[0.9,1]
J	2309	2481	2402	2464	2410	2428	2452	2544	2464	2469	24423	[0.9,1]
k	0	0	0	0	0	0	0	0	0	0	0	[1,1.2]
K	2888	3056	3176	3133	3147	3173	3080	3014	3055	3085	30807	[1,1.2]
l	0	0	0	0	0	0	0	0	0	0	0	[1.2,1.4]
L	1520	1507	1541	1530	1587	1496	1546	1562	1534	1530	15353	[1.2,1.4]
m	0	0	0	0	0	0	0	0	0	0	0	[1.4,1.6]
M	1305	1192	1183	1162	1173	1163	1124	1071	1090	1107	11570	[1.4,1.6]
n	0	0	0	0	0	0	0	0	0	0	0	[1.6,1.8]
N	1124	1256	1310	1309	1345	1378	1375	1383	1383	1374	13237	[1.6,1.8]
o	0	0	0	0	0	0	0	0	0	0	0	[1.8,2]
O	411	470	434	465	435	432	456	468	481	491	4543	[1.8,2]
p	0	0	0	0	0	0	0	0	0	0	0	[2,2.5]
P	28	46	41	53	51	50	54	44	48	52	467	[2,2.5]
q	0	0	0	0	0	0	0	0	0	0	0	[2.5,3]
Q	460	452	449	434	443	454	442	451	456	457	4498	[2.5,3]
r	0	0	0	0	0	0	0	0	0	0	0	[3,3.5]
R	0	0	0	0	0	1	0	0	1	1	3	[3,3.5]
s	0	0	0	0	0	0	0	0	0	0	0	[3.5,4]
S	0	0	0	0	0	0	0	0	0	0	0	[3.5,4]
t	0	0	0	0	0	0	0	0	0	0	0	[4,5]
T	0	0	0	0	0	0	0	0	0	0	0	[4,5]
u	0	0	0	0	0	0	0	0	0	0	0	[5,6]
U	0	0	0	0	0	0	0	0	0	0	0	[5,6]
v	0	0	0	0	0	0	0	0	0	0	0	[6,7]
V	0	0	0	0	0	0	0	0	0	0	0	[6,7]
w	0	0	0	0	0	0	0	0	0	0	0	[7,8]
W	0	0	0	0	0	0	0	0	0	0	0	[7,8]
x	0	0	0	0	0	0	0	0	0	0	0	[8,9]
X	0	0	0	0	0	0	0	0	0	0	0	[8,9]
y	0	0	0	0	0	0	0	0	0	0	0	[9,10]
Y	0	0	0	0	0	0	0	0	0	0	0	[9,10]
z	0	0	0	0	0	0	0	0	0	0	0	[10,10+]
Z	0	0	0	0	0	0	0	0	0	0	0	[10,10+]

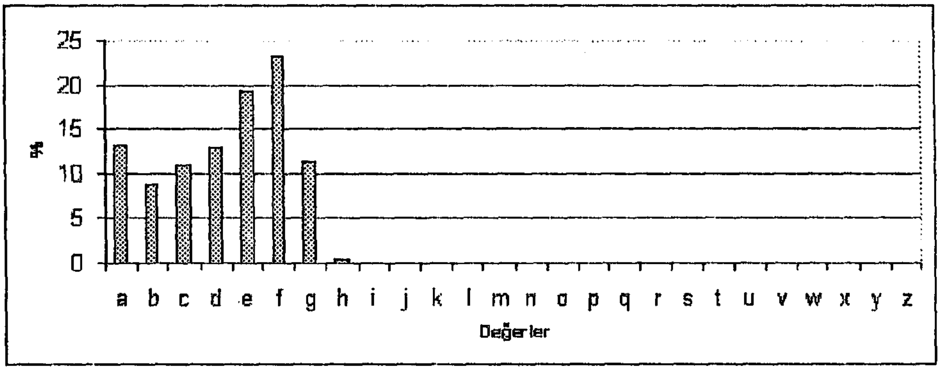
Çizelge 6.3'te görüldüğü gibi bu kodlama işleminin sonucunda *h*'den sonraki küçük harflere ve *R*'den sonraki büyük harflere rastlanmamaktadır. Kodlamada en çok karşılaşılan küçük harf *f*, büyük harf *A*; en az karşılaşılan ise küçük harflerde *h*, büyük harflerde *P*'dir.

Harflerin $[6, 10^6]$ aralığındaki yüzde olarak dağılımları ve grafikleri Çizelge 6.4, Şekil 6.1 ve 6.2'de görülmektedir.

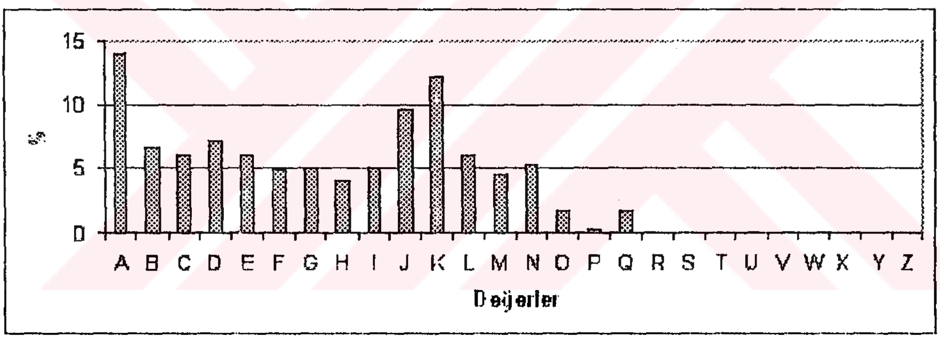
Çizelge 6.4 Harflerin [6, 10^6] aralığındaki yüzde olarak dağılımları

	%
a	13,07
b	8,75
c	10,93
d	12,92
e	19,32
f	23,20
g	11,28
h	0,53
i	0,00
j	0,00
k	0,00
l	0,00
m	0,00
n	0,00
o	0,00
p	0,00
q	0,00
r	0,00
s	0,00
t	0,00
u	0,00
v	0,00
w	0,00
x	0,00
y	0,00
z	0,00

	%
A	13,94
B	6,55
C	6,00
D	7,13
E	6,05
F	4,94
G	5,07
H	4,02
I	5,06
J	9,60
K	12,11
L	6,03
M	4,55
N	5,20
O	1,79
P	0,18
Q	1,77
R	0,00
S	0,00
T	0,00
U	0,00
V	0,00
W	0,00
X	0,00
Y	0,00
Z	0,00



Şekil 6.1 [6, 10^6] aralığında küçük harfler ile kodlama (% olarak)



Şekil 6.2 [6, 10^6] aralığında büyük harfler ile kodlama (% olarak)

Aralıklara göre harflerin toplam dağılımı Çizelge 6.5'de, yüzde olarak dağılım da Çizelge 6.6'da görülmektedir.

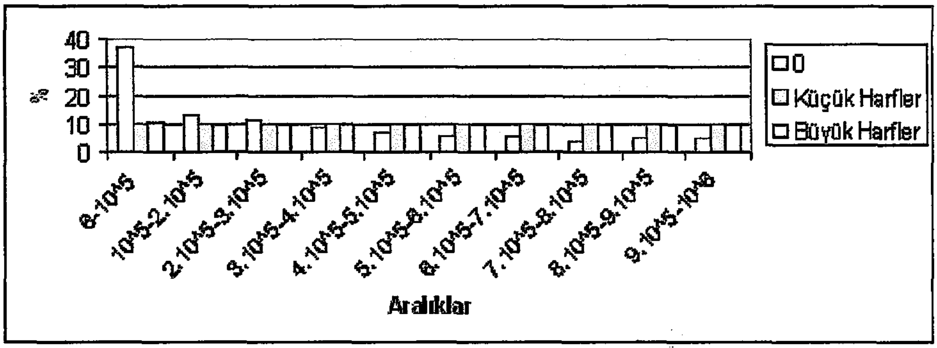
Çizelge 6.5 Aralıklara göre harflerin toplam dağılımı

	6-10^5	10^5-2.10^5	2.10^5-3.10^5	3.10^5-4.10^5	4.10^5-5.10^5	5.10^5-6.10^5	6.10^5-7.10^5	7.10^5-8.10^5	8.10^5-9.10^5	9.10^5-10^6	
	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	TOPLAM
0	205	71	61	47	38	30	28	20	26	26	552
Küçük Harfler	23958	24463	24522	24468	24610	24668	24492	24440	24643	24774	245038
Büyük Harfler	25835	25466	25417	25485	25352	25302	25480	25540	25531	25199	254607

Çizelge 6.6 Aralıklara göre harflerin toplamının yüzde dağılımı

	6-10^5	10^5-2.10^5	2.10^5-3.10^5	3.10^5-4.10^5	4.10^5-5.10^5	5.10^5-6.10^5	6.10^5-7.10^5	7.10^5-8.10^5	8.10^5-9.10^5	9.10^5-10^6	
	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	ADET	TOPLAM
0	37,14	12,86	11,05	8,51	6,88	5,43	5,07	3,62	4,71	4,71	100
Küçük Harfler	9,78	9,98	10,01	9,93	10,04	10,07	10,00	9,97	10,06	10,11	100
Büyük Harfler	10,15	10,01	9,99	10,02	9,97	9,95	10,02	10,04	10,04	9,90	100

Tüm aralıklarda küçük ve büyük harf dağılımının yaklaşık olarak birbirine eşit olduğu görülmektedir. Küçük harfler [9.78, 10.11], büyük harfler [9.90, 10.15] aralıklarında değişen değerler almaktadır. Aralıklara göre yüzde değerlerine ait grafik Şekil 6.3'te gösterilmiştir.



Şekil 6.3 Aralıklara göre değerlerin yüzde olarak dağılımı

$[10^6,\ 2\times10^6]$ aralığındaki $f(n)$ değerleri için elde edilen toplam sonuçları aşağıdaki gibidir:

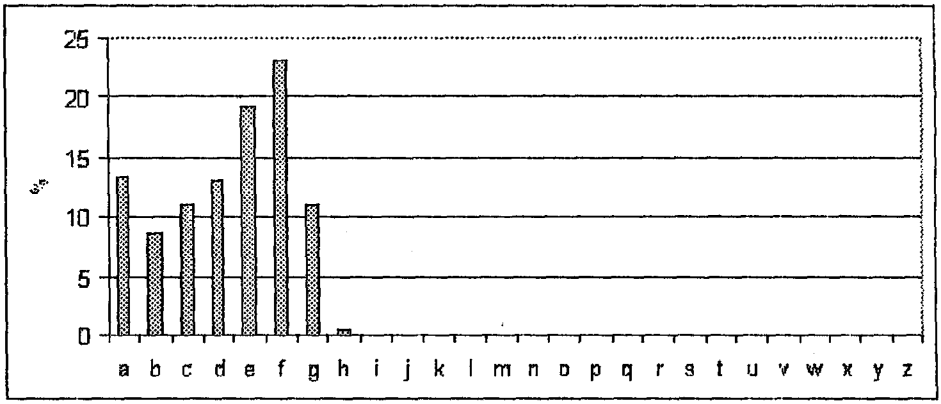
Toplam küçük harf sayısı	: 246.685
Toplam büyük harf sayısı	: 253.134
Toplam sıfır sayısı	: 181

$[10^6,\ 2\times10^6]$ aralığında da $[6,\ 10^6]$ aralığında olduğu gibi kodlama işleminin sonucunda *h*'den sonraki küçük harflere ve *R*'den sonraki büyük harflere rastlanmamaktadır. Bu aralıktaki kodlamada da en çok karşılaşılan küçük harf *f*, büyük harf *A*; en az karşılaşılan ise küçük harflerde *h*, büyük harflerde *P*'dir. Değerlerin $[10^6,\ 2\times10^6]$ aralığında yüzde olarak dağılımı $[6,\ 10^6]$ aralığındaki dağılıma benzerdir. Bu dağılım ve grafikleri Çizelge 6.7, Şekil 6.4 ve 6.5'te gösterilmiştir.

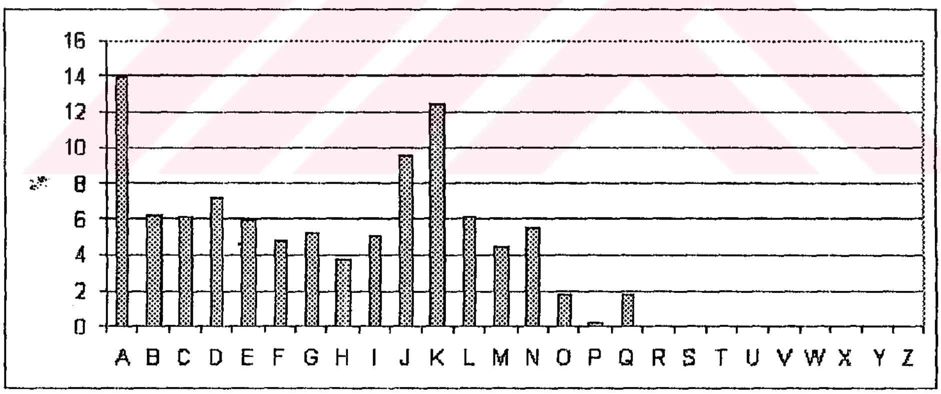
Çizelge 6.7 Harflerin $[10^6, 2\times10^6]$ aralığındaki dağılımları

	ADET	%
a	33.214	13,46
b	21.295	8,63
c	27.085	10,98
d	32.283	13,09
e	47.496	19,25
f	56.792	23,02
g	27.354	11,09
h	1.166	0,47
i	0	0,00
j	0	0,00
k	0	0,00
l	0	0,00
m	0	0,00
n	0	0,00
o	0	0,00
p	0	0,00
q	0	0,00
r	0	0,00
s	0	0,00
t	0	0,00
u	0	0,00
v	0	0,00
w	0	0,00
x	0	0,00
y	0	0,00
z	0	0,00
TOPLAM	246.685	100

	ADET	%
A	35.281	13,94
B	15.846	6,26
C	15.413	6,09
D	18.253	7,21
E	14.979	5,92
F	12.216	4,83
G	13.237	5,23
H	9.325	3,68
I	12.781	5,05
J	24.117	9,53
K	31.459	12,43
L	15.418	6,09
M	11.356	4,49
N	13.891	5,49
O	4.614	1,82
P	474	0,19
Q	4.469	1,77
R	5	0,002
S	0	0,00
T	0	0,00
U	0	0,00
V	0	0,00
W	0	0,00
X	0	0,00
Y	0	0,00
Z	0	0,00
TOPLAM	253.134	100



Şekil 6.4 $[10^6, 2\times10^6]$ aralığında küçük harfler ile kodlama (% olarak)



Şekil 6.5 $[10^6, 2\times10^6]$ aralığında büyük harfler ile kodlama (% olarak)

Farklı aralıklarda elde edilen $f(n)$ değerleri için korelasyon analizi yapılmıştır. Korelasyon analizinde, seçilmiş en az iki veya daha fazla örnek grup alınarak, bu gruplar arasındaki etkileşime bir katsayı yardımıyla bakılır. Bu katsayı korelasyon katsayısıdır. Korelasyon katsayısı, değişkenlerin yönü, etkileşimlerin nasıl olduğu hakkında bilgi verir. İlişkinin yönünü, derecesini ve istatistiki açıdan anlamlılığını tespit eder. "Değişkenler arasında etkileşim var mı, varsa bu etkileşim kuvvetli mi, gözlem gruplarından birinin gözlem değerleri artarken diğerinin azalıyor mu, yoksa aynı yönde mi bir değişim var", bu gibi sorular korelasyon katsayısına bakılarak cevaplanabilir.

Korelasyon katsayısı -1 ile +1 arasında değişen değerler alır. Korelasyon katsayısı, etkileşimin olmadığı durumda 0, tam ve kuvvetli bir etkileşim varsa 1, ters yönlü ve tam bir etkileşim varsa -1 değerini alır.

Çizelge 6.8 ve 6.9 korelasyon analizlerinin sonuçlarını göstermektedir. Bu çizelgelerdeki korelasyon analizlerinde, her bir sembolün $[6, 10^6]$ aralığında ki yüzde dağılımı ile farklı aralıklar için olan yüzde dağılımı dikkate alınmaktadır. Elde edilen sonuçlar ≈0,99 gibi 1'e yakın değerlerdir. Bu da değişkenler arasında kuvvetli bir etkileşim olduğunu göstermektedir.

Çizelge 6.8 Farklı aralıklarda elde edilen küçük harfler için korelasyon analizi

	6-10^5	10^5-2.10^5	2.10^5-3.10^5	3.10^5-4.10^5	4.10^5-5.10^5	5.10^5-6.10^5	6.10^5-7.10^5	7.10^5-8.10^5	8.10^5-9.10^5	9.10^5-10^6	6-10^6
a	11,8916	12,9992	13,0291	12,8453	13,2466	13,5358	12,9308	12,8478	13,6063	13,7079	13,0682
b	9,1786	8,8297	8,7717	8,8197	8,716	8,6185	8,6518	8,6252	8,4771	8,8318	8,7509
c	10,4808	10,9308	10,933	10,8918	11,0077	10,8764	11,1138	11,1498	10,9889	10,8703	10,9252
d	12,7807	12,7907	12,8782	13,0415	12,8688	13,179	12,9103	12,8478	12,9367	12,9692	12,9209
e	18,9665	19,1309	19,248	19,2292	19,5286	19,4908	19,2389	19,4967	19,5025	19,3227	19,3166
f	24,0129	23,3455	23,3953	23,3693	23,0232	22,6731	23,3872	23,2283	22,899	22,7456	23,205
g	11,9417	11,405	11,2307	11,325	11,1052	11,144	11,2527	11,2807	11,0823	11,0761	11,2823
h	0,7471	0,5682	0,5138	0,4782	0,5039	0,4824	0,5145	0,5237	0,5072	0,4763	0,5309
Korelasyon	0,9954	0,9998	0,9999	0,9998	0,9998	0,999	0,9999	0,9998	0,9992	0,999	

Çizelge 6.9 Farklı aralıklarda elde edilen büyük harfler için korelasyon analizi

	6-10^5	10^5-2.10^5	2.10^5-3.10^5	3.10^5-4.10^5	4.10^5-5.10^5	5.10^5-6.10^5	6.10^5-7.10^5	7.10^5-8.10^5	8.10^5-9.10^5	9.10^5-10^6	6-10^6
A	13,5785	13,9441	13,8844	14,1299	13,8569	13,8171	14,2072	14,4049	13,9513	13,6236	13,9399
B	7,1492	6,5538	6,4642	6,5843	6,5557	6,4817	6,3697	6,4722	6,4427	6,4566	6,5541
C	6,2319	6,0473	6,0117	5,9525	5,7786	5,9877	6,1499	6,0258	5,894	5,909	5,9994
D	6,7738	7,0093	7,0779	7,165	7,3801	7,1496	7,135	7,1809	7,2678	7,1868	7,1319
E	6,7583	6,1729	6,1809	5,9564	5,893	5,9363	5,9341	5,9475	5,8308	5,8653	6,049
F	5,1945	4,9713	4,918	4,8813	4,9069	4,8889	4,8823	4,7455	4,9662	4,9962	4,9354
G	4,8655	5,0656	4,9809	5,0029	5,0568	5,0866	5,102	5,2349	5,1478	5,1708	5,071
H	4,7339	4,1781	4,1508	4,0494	4,0312	3,7902	3,9207	3,8645	3,778	3,7105	4,0223
I	5,8332	4,9831	4,8786	4,8813	4,7649	5,0668	4,9765	4,8669	5,2228	5,151	5,0635
J	8,9375	9,7424	9,4504	9,6684	9,5062	9,5961	9,6232	9,9608	9,7272	9,798	9,6
K	11,1786	12,0003	12,4956	12,2935	12,4132	12,5405	12,0879	11,8011	12,0603	12,2425	12,1093
L	5,8835	5,9177	6,0629	6,0035	6,2599	5,9126	6,0675	6,1159	6,0558	6,0717	6,0348
M	5,0513	4,6808	4,6544	4,5595	4,6269	4,5965	4,4113	4,1934	4,303	4,393	4,5478
N	4,3507	4,9321	5,154	5,1364	5,3053	5,4462	5,3964	5,415	5,4597	5,4526	5,2031
O	1,5909	1,8456	1,7075	1,8246	1,7158	1,7074	1,7896	1,8324	1,8989	1,9485	1,7857
P	0,1084	0,1806	0,1613	0,208	0,2012	0,1976	0,2119	0,1723	0,1895	0,2064	0,1836
Q	1,7805	1,7749	1,7665	1,703	1,7474	1,7943	1,7347	1,7659	1,8002	1,8136	1,768
R	0	0	0	0	0	0,004	0	0	0,0039	0,004	0,0012
Korelasyon	0,9901	0,9995	0,9994	0,9998	0,9991	0,9993	0,9996	0,9985	0,9992	0,999	

6.1.3 Özel Sayılar

Elde edilen $f(n)$ değerleri incelendiğinde tekrar eden bazı diziler olduğu görülmüştür. Dizide $6k-2, 6k, 6k+2$ formundaki 6'ya bölünebilen ($6k$) sayıların $f(n)$ değerleri, sol ve sağ taraflarındaki sayıların $f(n)$ değerlerinden daha büyük bir değere sahiptir.

$f(6k) \geq f(6k-2)$ ve $f(6k) \geq f(6k+2)$ olmaktadır.

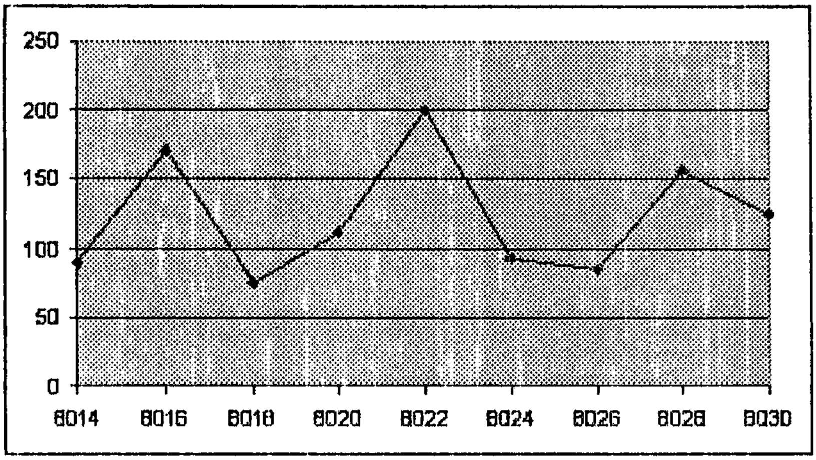
Örnek 6.3:

$8016 = 6 \times 1336$	$8022 = 6 \times 1337$	$8028 = 6 \times 1338$
f(8014) = 90	f(8020) = 112	f(8026) = 85
f(8016) = 171	f(8022) = 200	f(8028) = 157
f(8018) = 75	f(8024) = 94	f(8030) = 125

f(8016) > f(8014) ve f(8016) > f(8018)

f(8022) > f(8020) ve f(8022) > f(8024)

f(8028) > f(8026) ve f(8028) > f(8030)



Şekil 6.6 [8014, 8030] aralığındaki değerlerinin grafiği

Üçlü gruplar halinde bu sayılara baktığımızda, Şekil 6.6'da görüldüğü gibi 6'nın katı olan değerlerde (örneğin 8016, 8022 ve 8028 değerlerinde) sol ve sağ taraflarındaki komşularına göre $f(n)$ değerlerinde yükseliş olmaktadır.

Fakat bu $f(6k) \geq f(6k-2)$ ve $f(6k) \geq f(6k+2)$ özelliğini bozan bazı sayılar mevcuttur. 10^7'ye kadar bakılan $f(n)$ değerleri için bu özelliği bozan 124 adet sayı tesbit edilmiştir. Çizelge 6.10 bu özel sayıları göstermektedir. Bu sayıların 66 tanesi (2 ile biten rakamlar) $f(6k) \leq f(6k-2)$ ve $f(6k) \geq f(6k+2)$, geriye kalan 58 tanesi ise (8 ile biten rakamlar) $f(6k) \geq f(6k-2)$ ve $f(6k) \leq f(6k+2)$ formundadır.

Çizelge 6.10 [6, 10^7] aralığındaki özel sayılar

12	f[10]	:=	2	f[12]	:=	1	f[14]	:=	2
80082	f[80080]	:=	1006	f[80082]	:=	1005	f[80084]	:=	495
190188	f[190186]	:=	1028	f[190188]	:=	2069	f[190190]	:=	2095
340338	f[340336]	:=	1712	f[340338]	:=	3335	f[340340]	:=	3396
380382	f[380380]	:=	3638	f[380382]	:=	3610	f[380384]	:=	1809
460458	f[460456]	:=	2142	f[460458]	:=	4201	f[460460]	:=	4265
497418	f[497416]	:=	2287	f[497418]	:=	4463	f[497420]	:=	4503
580578	f[580576]	:=	2550	f[580578]	:=	5121	f[580580]	:=	5145
620622	f[620620]	:=	5482	f[620622]	:=	5408	f[620624]	:=	2733
680682	f[680680]	:=	6086	f[680682]	:=	5954	f[680684]	:=	2923
760758	f[760756]	:=	3287	f[760758]	:=	6466	f[760760]	:=	6680
850848	f[850846]	:=	3559	f[850848]	:=	7056	f[850850]	:=	7287
860862	f[860860]	:=	7114	f[860862]	:=	7101	f[860864]	:=	3551
920922	f[920920]	:=	7734	f[920922]	:=	7550	f[920924]	:=	4229
950952	f[950950]	:=	7993	f[950952]	:=	7725	f[950954]	:=	4030
1151148	f[1151146]	:=	4523	f[1151148]	:=	9107	f[1151150]	:=	9287
1161162	f[1161160]	:=	9271	f[1161162]	:=	9155	f[1161164]	:=	4662
1191192	f[1191190]	:=	9734	f[1191192]	:=	9402	f[1191194]	:=	4756
1243548	f[1243546]	:=	5008	f[1243548]	:=	9729	f[1243550]	:=	9745
1331328	f[1331326]	:=	5426	f[1331328]	:=	10316	f[1331330]	:=	10595
1361358	f[1361356]	:=	5270	f[1361358]	:=	10661	f[1361360]	:=	10873
1505352	f[1505350]	:=	11430	f[1505352]	:=	11420	f[1505354]	:=	5756
1521522	f[1521520]	:=	11886	f[1521522]	:=	11655	f[1521524]	:=	5874
1551552	f[1551550]	:=	11786	f[1551552]	:=	11647	f[1551554]	:=	5901
1611612	f[1611610]	:=	12324	f[1611612]	:=	12306	f[1611614]	:=	6068
1701702	f[1701700]	:=	13165	f[1701702]	:=	12557	f[1701704]	:=	6565
1841838	f[1841836]	:=	7016	f[1841838]	:=	13609	f[1841840]	:=	13731
1851852	f[1851850]	:=	13641	f[1851852]	:=	13624	f[1851854]	:=	7211
1871868	f[1871866]	:=	6998	f[1871868]	:=	13932	f[1871870]	:=	14268
1901898	f[1901896]	:=	7003	f[1901898]	:=	14100	f[1901900]	:=	14324
1989678	f[1989676]	:=	7959	f[1989678]	:=	14347	f[1989680]	:=	14608
2092092	f[2092090]	:=	15594	f[2092092]	:=	15137	f[2092094]	:=	7562
2172168	f[2172166]	:=	8243	f[2172168]	:=	15527	f[2172170]	:=	15665
2212212	f[2212210]	:=	16532	f[2212212]	:=	15842	f[2212214]	:=	7932
2302302	f[2302300]	:=	16652	f[2302302]	:=	16464	f[2302304]	:=	8264
2322318	f[2322316]	:=	8255	f[2322318]	:=	16572	f[2322320]	:=	16613
2382378	f[2382376]	:=	8555	f[2382378]	:=	16965	f[2382380]	:=	17499
2472468	f[2472466]	:=	8719	f[2472468]	:=	17334	f[2472470]	:=	17862
2662662	f[2662660]	:=	19110	f[2662662]	:=	18554	f[2662664]	:=	10100
2722722	f[2722720]	:=	19631	f[2722722]	:=	19134	f[2722724]	:=	9462
2735808	f[2735806]	:=	9499	f[2735808]	:=	18960	f[2735810]	:=	19082
2892888	f[2892886]	:=	10028	f[2892888]	:=	20335	f[2892890]	:=	20620
2992992	f[2992990]	:=	20804	f[2992992]	:=	20526	f[2992994]	:=	11142
3103098	f[3103096]	:=	10567	f[3103098]	:=	21243	f[3103100]	:=	21347
3193188	f[3193186]	:=	11201	f[3193188]	:=	21657	f[3193190]	:=	21956
3233232	f[3233230]	:=	24275	f[3233232]	:=	21827	f[3233234]	:=	10983
3403398	f[3403396]	:=	11520	f[3403398]	:=	23142	f[3403400]	:=	23625
3481938	f[3481936]	:=	11820	f[3481938]	:=	23364	f[3481940]	:=	23559
3743742	f[3743740]	:=	25973	f[3743742]	:=	24934	f[3743744]	:=	12417
3773772	f[3773770]	:=	25238	f[3773772]	:=	25142	f[3773774]	:=	12572
3803802	f[3803800]	:=	25945	f[3803802]	:=	25337	f[3803804]	:=	12866
3913908	f[3913906]	:=	12993	f[3913908]	:=	25923	f[3913910]	:=	27992
3979362	f[3979360]	:=	26374	f[3979362]	:=	26352	f[3979364]	:=	13092
4064058	f[4064056]	:=	13666	f[4064058]	:=	26657	f[4064060]	:=	26824
4074072	f[4074070]	:=	26720	f[4074072]	:=	26713	f[4074074]	:=	13370
4344342	f[4344340]	:=	28413	f[4344342]	:=	28353	f[4344344]	:=	14260
4604598	f[4604596]	:=	15442	f[4604598]	:=	29856	f[4604600]	:=	30193
4644642	f[4644640]	:=	30167	f[4644642]	:=	30045	f[4644644]	:=	15667
4754748	f[4754746]	:=	15532	f[4754748]	:=	30759	f[4754750]	:=	31266
4764762	f[4764760]	:=	31730	f[4764762]	:=	31644	f[4764764]	:=	15208
4934928	f[4934926]	:=	16866	f[1934928]	:=	31597	f[4934930]	:=	33880
4944942	f[4944940]	:=	32680	f[4944942]	:=	31580	f[4944944]	:=	15839

5065062	f[5065060]	=	32748	f[5065062]	=	32346	f[5065064]	=	16181
5275272	f[5275270]	=	35882	f[5275272]	=	33915	f[5275274]	=	18273
5295288	f[5295286]	=	16824	f[5295288]	=	34166	f[5295290]	=	34249
5515512	f[5515510]	=	37084	f[5515512]	=	34842	f[5515514]	=	18524
5720328	f[5720326]	=	18217	f[5720328]	=	36529	f[5720330]	=	37745
5755752	f[5755750]	=	36620	f[5755752]	=	36093	f[5755754]	=	17984
5785782	f[5785780]	=	37540	f[5785782]	=	36139	f[5785784]	=	18183
5895888	f[5895886]	=	18488	f[5895888]	=	37220	f[5895890]	=	39199
5955948	f[5955946]	=	18557	f[5955948]	=	37336	f[5955950]	=	38299
5985978	f[5985976]	=	19312	f[5985978]	=	37244	f[5985980]	=	37999
6086082	f[6086080]	=	38988	f[6086082]	=	37844	f[6086084]	=	19124
6206202	f[6206200]	=	38682	f[6206202]	=	38679	f[6206204]	=	19239
6296292	f[6296290]	=	41464	f[6296292]	=	39211	f[6296294]	=	19901
6386382	f[6386380]	=	39802	f[6386382]	=	39351	f[6386384]	=	19767
6446442	f[6446440]	=	40224	f[6446442]	=	39827	f[6446444]	=	20052
6466458	f[6466456]	=	19934	f[6466458]	=	39959	f[6466460]	=	43610
6656652	f[6656650]	=	41826	f[6656652]	=	41047	f[6656654]	=	20553
6676668	f[6676666]	=	20625	f[6676668]	=	40965	f[6676670]	=	43148
6760392	f[6760390]	=	42882	f[6760392]	=	41335	f[6760394]	=	20814
6806802	f[6806800]	=	43223	f[6806802]	=	41656	f[6806804]	=	21861
6826818	f[6826816]	=	20996	f[6826818]	=	41820	f[6826820]	=	41941
6976968	f[6976966]	=	21521	f[6976968]	=	42802	f[6976970]	=	44917
7037028	f[7037026]	=	21535	f[7037028]	=	43842	f[7037030]	=	45498
7137132	f[7137130]	=	45766	f[7137132]	=	44896	f[7137134]	=	21575
7212588	f[7212586]	=	21883	f[7212588]	=	43789	f[7212590]	=	45679
7227222	f[7227220]	=	45293	f[7227222]	=	44009	f[7227224]	=	21887
7257252	f[7257250]	=	44468	f[7257252]	=	44039	f[7257254]	=	22947
7317312	f[7317310]	=	47082	f[7317312]	=	46547	f[7317314]	=	22254
7487478	f[7487476]	=	23131	f[7487478]	=	45332	f[7487480]	=	47029
7547538	f[7547536]	=	22885	f[7547538]	=	45602	f[7547540]	=	45616
7607598	f[7607596]	=	22900	f[7607598]	=	46079	f[7607600]	=	46977
7710012	f[7710010]	=	48328	f[7710012]	=	46337	f[7710014]	=	26546
7797792	f[7797790]	=	49349	f[7797792]	=	47903	f[7797794]	=	23837
7827822	f[7827820]	=	50948	f[7827822]	=	47209	f[7827824]	=	23526
7997988	f[7997986]	=	24084	f[7997988]	=	48030	f[7997990]	=	50814
8058048	f[8058046]	=	24250	f[8058048]	=	48190	f[8058050]	=	48955
8128122	f[8128120]	=	49099	f[8128122]	=	48822	f[8128124]	=	25738
8178168	f[8178166]	=	24488	f[8178168]	=	48976	f[8178170]	=	51317
8338332	f[8338330]	=	51410	f[8338332]	=	50026	f[8338334]	=	25939
8368362	f[8368360]	=	51255	f[8368362]	=	49886	f[8368364]	=	25611
8418408	f[8418406]	=	26467	f[8418408]	=	50023	f[8418410]	=	50093
8456142	f[8456140]	=	50459	f[8456142]	=	50028	f[8456144]	=	25032
8508498	f[8508496]	=	26337	f[8508498]	=	51454	f[8508500]	=	52200
8518512	f[8518510]	=	53136	f[8518512]	=	51041	f[8518514]	=	25915
8523972	f[8523970]	=	51882	f[8523972]	=	51241	f[8523974]	=	25226
8748738	f[8748736]	=	26219	f[8748738]	=	51572	f[8748740]	=	55684
8848842	f[8848840]	=	54018	f[8848842]	=	52566	f[8848844]	=	26325
8938932	f[8938930]	=	55359	f[8938932]	=	52486	f[8938934]	=	26587
8998992	f[8998990]	=	55221	f[8998992]	=	54723	f[8998994]	=	26594
9019008	f[9019006]	=	26608	f[9019008]	=	53160	f[9019010]	=	56079
9202272	f[9202270]	=	55997	f[9202272]	=	54112	f[9202274]	=	27044
9209202	f[9209200]	=	55088	f[9209202]	=	54480	f[9209204]	=	27049
9289278	f[9289276]	=	29008	f[9289278]	=	54862	f[9289280]	=	54900
9319308	f[9319306]	=	27592	f[9319308]	=	55820	f[9319310]	=	56245
9359352	f[9359350]	=	56679	f[9359352]	=	55804	f[9359354]	=	27326
9450978	f[9450976]	=	29260	f[9450978]	=	55580	f[9450980]	=	55662
9509502	f[9509500]	=	57414	f[9509502]	=	56606	f[9509504]	=	27818
9529518	f[9529516]	=	29190	f[9529518]	=	55772	f[9529520]	=	57515
9619608	f[9619606]	=	29112	f[9619608]	=	56434	f[9619610]	=	56461
9758208	f[9758206]	=	28682	f[9758208]	=	56849	f[9758210]	=	58190
9869862	f[9869860]	=	61713	f[9869862]	=	57951	f[9869864]	=	28831
9899892	f[9899890]	=	60072	f[9899892]	=	58914	f[9899894]	=	28928

Eğer bu özel sayılar olmasa idi yani $f(6k) \geq f(6k-2)$ ve $f(6k) \geq f(6k+2)$ özelliği bozulmasaydı, o zaman $f(n)$ Goldbach Çiftleri üzerinde bir desen bulunmuş olacaktı. Bu desen sayesinde 6'nın katı olan sayılar Goldbach Sanısı'nın testi sırasında rahatlıkla elenebilecekti. Böylece Goldbach Sanısı'nı test etmekte önemli bir avantaj sağlanmış olacaktı.

6.2 Sonuçların İrdelenmesi

$f(n)$ fonksiyonu ile verilen bir aralıktaki çift sayıların Goldbach Çiftleri bulunmaktadır. $f(n)$ fonksiyonundan yararlanılarak yapılan çalışmalarda, elde edilen değerler üzerinde $f(n)$ değerlerinin artış-azalışlarına bakılmış, $\Delta(n)$ fark fonksiyonu ile $f(n)$ değerleri kodlanmış ve kodlanan dosyaların sıkıştırılma oranlarına bakılmıştır. Bu çalışmalarda herhangi bir desen ile karşılaşılmamıştır. Son olarak $f(n)$ değerlerinin $f(6k) \geq f(6k-2)$ ve $f(6k) \geq f(6k+2)$ özelliğine bakılmış, fakat bu formu bozan özel sayılar tespit edilmiştir. Eğer bu özel sayılar olmasaydı yani $f(6k) \geq f(6k-2)$ ve $f(6k) \geq f(6k+2)$ özelliği korunmuş olsaydı Goldbach Sanısı'nın testinde önemli bir kolaylık elde edilmiş olacaktı.

7. SONUÇLAR

Tez çalışmasında Goldbach Sanısı ve asal sayı örüntüleri ile ilgili çalışmalar yapılmıştır. Bu çalışmalar üç başlıkta toplanabilir:

- **g(n) ve e(n) fonksiyonları:** Bu fonksiyonlar çeşitli aralıklarda çalıştırılmış ve elde edilen sonuçlar karşılaştırılmıştır. Buna göre *g(n)* fonksiyonunun *e(n)* fonksiyonundan daha hızlı olduğu anlaşılmıştır. Fonksiyonların grafikleri incelendiğinde bazı değerlerde sıçramalar olduğu görülmüştür. İki fonksiyonun grafikleri örtüştürüldüğünde de bazen fonksiyonların birinin tepe yaptığı noktada diğerinin dibe vurduğu, bazen de ikisinin birden dibe vurduğu ya da tepe yaptıkları görülmüştür. Her iki fonksiyondan elde edilen sonuçlarda da herhangi bir asal sayı örüntüsüne rastlanmamıştır.

- **Goldbach Sanısı'nın verilen bir aralıkta doğrulanması:** Verilen bir aralıkta Goldbach Sanısı'nın doğruluğu için 3 yöntem incelenmiştir.

  - Deshouillers ve te Riele tarafından geliştirilen Yöntem 1

  - 1. Yöntemin modifiye edilmiş şekli olan Yöntem 2

- Bu tez çalışması sırasında geliştirilen MID Yöntemi

Bu yöntemler çeşitli aralıklarda test edilmiş, elde edilen sonuçlar karşılaştırılmıştır. Bu sonuçlara göre 1. Yöntem en hızlı, 2. Yöntem'de en verimli sonuçları vermiştir. MID yöntemi fazla zaman almış, sonuçlarının verimliliği ise Yöntem 2 kadar iyi olmasa da Yöntem 1 kadar da kötü değildir.

➢ **Verilen bir aralıktaki tüm Goldbach Çiftlerinin bulunması:** Bu işlem kısaca $f(n)$ olarak adlandırılmıştır. Elde edilen $f(n)$ değerleri üzerinde bazı çalışmalar yapılmıştır. Bu çalışmalarda, elde edilen değerler üzerinde $f(n)$ değerlerinin artış-azalışlarına bakılmış, $\Delta(n)$ fark fonksiyonu ile $f(n)$ değerleri kodlanmış ve kodlanan dosyaların sıkıştırılma oranlarına bakılmıştır. Bu çalışmalar sonucunda herhangi bir desen ile karşılaşılmamıştır. Son olarak $f(n)$ değerlerinin $f(6k) \geq f(6k-2)$ ve $f(6k) \geq f(6k+2)$ özelliğine bakılmış, fakat bu formu bozan özel sayılar tespit edilmiştir. Eğer bu özel sayılar olmasaydı Goldbach Sanısı'nın testinde önemli bir kolaylık elde edilmiş olacaktı. Çünkü bu sayılar testlerde hemen elenebilecekti.

Bu çalışmanın ardından yapılabilecek diğer bazı çalışmalar şunlardır:

- Test aralıklarının büyültülmesi,
- Daha büyük aralıklarda ki özel sayıların tespit edilmesi ve aralarında bir bağ olup olmadığına bakılması,
- Verilen bir aralıkta Goldbach Sanısı'nın doğrulanması ile ilgili yöntemlerin iyileştirilmesi, yeni yöntemlerin geliştirilmesidir.

KAYNAKLAR DİZİNİ

Anderson, James A. and Bell, James M., 1997, Number Theory With Applications, Prentice Hall Inc., New Jersey.

As Ultimas Do Mundo Da Matematica, 2001, Goldbach's Conjecture: $1000000 Challenge, http://www.mat.uc.pt/~jaimecs/ult/ult.html.

Brooke Weston City Technology College, 1999, Goldbach's Conjecture, Mathematics Masterclasses, http://www.bwctc.northants.sch.uk/html/master/maths/autumn99/primech-a2.htm.

Caldwell, Chris K., 2002, The University of Tennesse at Martin, Practical Applications of Prime Numbers, http://www.utm.edu/research/primes/notes/1257787.html.

Caldwell, Chris K., 2002, The University of Tennesse at Martin, The Riemann Hypothesis, http://www.utm.edu/research/primes/notes/rh.html.

Caldwell, Chris K., 2002, The University of Tennesse at Martin, How Many Primes Are There?, http://www.utm.edu/research/primes/howmany.shtml.

KAYNAKLAR DİZİNİ (devam ediyor)

Deshouillers J.-M., Effinger G., Te Riele H. and Zinoviev D., 1997, A Complete Vinogradov 3-Primes Theorem Under The Riemann Hypothesis, Electronic Research Announcements Of The American Mathematical Society, Volume 3, p 99-104.

Deshouillers J.-M., Te Riele H. and Saouter Y., 1998, New Experimental Results Concerning the Goldbach Conjecture, Algorithmic Number Theory Symposium III, USA.

Doxiadis, Apostolos, 2000, Uncle Petros and Goldbach's Conjecture, http://www.apostolosdoxiadis.com/petros-features.htm.

Forbes, Tony, 2000, Prime k-Tuplets, http://www.ltkz.demon.co.uk/ktuplets.htm.

Gagnon, William A., 2001, Christian Goldbach, http://niagara.rivier.edu/students/mneville/history/goldbach.htm.

Herkommer, Mark, 2002, Goldbach Conjecture Research, http://home.flash.net/~mherk/goldbach.htm.

Lucile and Gallot,Yves, 2001, The Chronology of Prime Number Records, http://perso.wanadoo.fr/yves.gallot/primes/chrrcds.html.

KAYNAKLAR DİZİNİ (devam ediyor)

O'Connor, J.J. and Robertson, E.F., 2001, Prime Numbers, http://www-history.mcs.st-andrews.ac.uk/history/HistTopics/Prime_numbers.html.

Peterson, Ivars, 2000, Goldbach's Prime Pairs, Science News Online, Vol.158, No.8, http://www.sciencenews.org/20000819/mathtrek.asp.

Richstein, Jörg, 1999, Verifying Goldbach's Conjecture up to 4×10^{14}, CNTA'99, Winnipeg/Canada, http://www.informatik.uni-giessen.de/staff/richstein/ca/Goldbach.htm.

Richstein, Jörg, 2000, Computing the Number of Goldbach Partitions up to 5×10^{18}, Berlin, No. 236, 1745--1749.

Rivest, Ronald L., 1991, Finding Four Million Large Random Primes, Crypto90, 625-626.

Silva, Jaime Carvalho , 2002, Departamento de Matemática, Universidade de Coimbra, http://www.mat.uc.pt/~jaimecs/ult/ult.html.

Stepney, Susan, Perfect Number, http://www-users.cs.york.ac.uk/~susan/cyc/p/perfect.htm.

KAYNAKLAR DİZİNİ (devam ediyor)

The University of Sheffield, Department of Pure Mathematics, 1999, The Prime Number Theorem, http://wwvv.shef.ac.uk/~puremath/ theorems/pnt.html.

Weisstein, Eric W., 1999, Goldbach Conjecture, Wolfram Research, http://mathworld.wolfram.com/GoldbachConjecture.html.

Woon, Max See Chin, 2000, On Partitions of Goldbach's Conjecture, USA, http://www.primepuzzles.net/puzzles/GoldbachPartitions.pdf.

WWW Virtual School Pre-Algebra Homepage, 2000, Goldbach's Conjecture, http:// library.thinkquest.org/3273/math/palgebra.html.

EKLER

Ek 1 Sözlük (Türkçe'den İngilizce'ye)

Ek 2 Sözlük (İngilizce'den Türkçe'ye)

Ek 3 *g(n)* Fonksiyonunun Bulunmasında Kullanılan GAP Programı

Ek 4 *e(n)* Fonksiyonunun Bulunmasında Kullanılan GAP Programı

Ek 5 Özel Sayıların Bulunmasında Kullanılan C Programı

Ek 1

SÖZLÜK (Türkçe'den İngilizce'ye)

Türkçe	İngilizce
Aritmetiğin Ana Teoremi	The Fundamental Theorem of
Asal	Prime
Asal Çarpanlara Ayırma	Prime Factorization
Boşluk	Gap
Çok İşlemli Sistemler	Multitasking Systems
Dördüz Asallar	Prime Quadruplets
Eratosthenes Kalburu	Sieve of Eratosthenes
Fermat'ın Küçük Teoremi	Fermat's Little Theorem
Goldbach Çifti	Goldbach Partition
Goldbach Sanısı	Goldbach Conjecture
İkiz Asallar	Twin Primes
İş İstasyonu	Workstation
Karmaşıklık	Complexity
Kesin	Deterministic
Mükemmel Sayılar	Perfect Numbers
Olası	Probabilistic
Olası Asal	Pseudoprime
Örüntü	Pattern
Paylaşım Paketi	Share Package
Sayı Kuramı	Number Theory
Üçüz Asallar	Prime Triplets

Ek 2

SÖZLÜK (İngilizce'den Türkçe'ye)

İngilizce	**Türkçe**
Complexity	Karmaşıklık
Deterministic	Kesin
Fermat's Little Theorem	Fermat'ın Küçük Teoremi
Gap	Boşluk
Goldbach Conjecture	Goldbach Sanısı
Goldbach Partition	Goldbach Çifti
Multitasking Systems	Çok İşlemli Sistemler
Number Theory	Sayı Kuramı
Pattern	Örüntü
Perfect Numbers	Mükemmel Sayılar
Prime	Asal
Prime Factorization	Asal Çarpanlara Ayırma
Prime Quadruplets	Dördüz Asallar
Prime Triplets	Üçüz Asallar
Probabilistic	Olası
Pseudoprime	Olası Asal
Share Package	Paylaşım Paketi
Sieve of Eratosthenes	Eratosthenes Kalburu
The Fundamental Theorem of Arithmetic	Aritmetiğin Ana Teoremi
Twin Primes	İkiz Asallar
Workstation	İş İstasyonu

Ek 3

g(n) FONKSİYONUNUN BULUNMASINDA KULLANILAN GAP PROGRAMI

```
gn:=function(sayi,sayi2)

     local p,m,t1,t2,t,i,count;
     i:=sayi;
     count:=1;
     Print("0"," ");
     repeat
          m:=0;
          p:=2;
     if(IsEvenInt(i)=false) then
          Print("Sayi cift degildir!\n");
     fi;
     t1:=Runtime();
     if(IsEvenInt(i)=true) then
          for p in [p..i/2] do
               if IsPrimeInt(p) and IsPrimeInt(i-p) then
                    if m<>1 then
                    Print(i," = ",p,"+",i-p," ");
                    Print(p," ");
                    count:=count+1;
                    if count=10 then
                     Print("\n");
                     count:=0;
                    fi;
                    m:=m+1;
                    fi;
               fi;
          od;
     t2:=Runtime();
     t:=t2-t1;
     Print(m," Distinct representations\n");
     Print("Time elapsed ==> ",t," milisecond cpu
time.\n");
     fi;
     i:=i+2;
     until(i=sayi2+2);
     Print("\n");
end;
```

Ek 4

e(n) FONKSİYONUNUN BULUNMASINDA KULLANILAN GAP PROGRAMI

```
en:=function(sayi,sayi2)

     local p,m,t1,t2,t,x,y,i,count,en;
     count:=0;
     en:=count;
     i:=sayi;
     if(IsEvenInt(sayi)=false) then
              Print("Sayi cift degildir!\n");
     fi;
     t1:=Runtime();
     repeat
     m:=i/2;
         if(IsEvenInt(m)=true) then
           count:=count+1;
     else count:=0;
     fi;
           x:=m-count;
           y:=m+count;
           if IsPrimeInt(x) and IsPrimeInt(y) then
                en:=count;
           else
                repeat
                     count:=count+2;
                     x:=x-2;
                     y:=y+2;
                     en:=count;
                until(IsPrimeInt(x) and IsPrimeInt(y));
           fi;
     Print(en," ");
     count:=0;
     i:=i+2;
     until(i=sayi2+2);
       t2:=Runtime();
       t:=t2-t1;
     Print("Time elapsed ==> ",t," milisecond cpu
time.\n");
     Print("\n");
end;
```

Ek 5

ÖZEL SAYILARIN BULUNMASINDA KULLANILAN C PROGRAMI

```
#include <stdio.h>
#include <math.h>
void displayArray(int [], int);
void checkArray(int [], int);

void main( ) {

    FILE *filePtr;
    int i,j,k,N;

    if((filePtr = fopen("Gresult.txt","w")) == NULL) {
      printf("File could not be opened. Exiting ...\n");
      exit(1);
    }
    printf("Please enter N ");
    scanf("%d",&N);

    bool *sieve = new bool[N];
    for(i = 0; i < N; i++) sieve[i] = true;

    // sieveing
    for(i=2;i<=sqrt(N);i++) {
        if(sieve[i])
           for(j=2;j<=(N/i);j++)
              sieve[i*j] = false;
     }

    int primeCnt = 0;
      for(i=2;i<N;i++) if(sieve[i]) primeCnt++;

    if(N<1000) {
     for(i = 2; i < N; i++)
           if(sieve[i]) printf("%d  ",i);
    }
    printf("\nthere are %d primes less than
%d\n",primeCnt,N);
     int *primeArray = new int[primeCnt];
     k = 0;
```

```
     for(i=2;i<N;i++)
        if(sieve[i]) primeArray[k++] = i;

    delete [] sieve;

    int GpairCount[N/2];

    for(k = 0; k < N/2; k++) {
      GpairCount[k]=0;
    };

    if(N<100) {
printf("\nf : "); displayArray(GpairCount,N/2);}

    if(N<1) {
printf("\nprimeArray : ");
displayArray(primeArray,primeCnt);
    }

    for(i = 1; i < primeCnt; i++) {
     //i=1 to ignore even prime 2

      int p1 = primeArray[i];

      for(j = i; j < primeCnt; j++) {

         int p2 = primeArray[j];

         int index = p1 + p2;

         if( index < N ) {

            GpairCount[(index/2)]++;
          }
         else break;

       } // for(j

     }  // for(i

    printf("\n\nAfter proccess completed ");
    if(N<100) {printf("\nf : ");
    displayArray(GpairCount,N/2);}
    checkArray(GpairCount,N/2);
    fclose(filePtr);
 }
```

```
void displayArray(int array[], int size) {

     for(int i = 1; i < size; i++) {
        printf("\nf[%d] = %d ",2*i,array[i]);
     }
}
void checkArray(int array[], int size) {

    for(int i = 1; i < size; i++) {
      if(array[i] < 1) {
        printf("\nError at f[%d] = %d ",2*i,array[i]);
      }
      if(i % 3 == 0 && i+1 < size) {
        if(array[i] < array[i-1] || array[i] < array[i+1])
{
        printf("\nPeculiar number: %d mod 3 = 0 and ",2*i);
        printf("f[%d] = %d, f[%d] = %d, f[%d] = %d ",
              2*(i-1),array[i-1],2*i,array[i],2*(i+1),
              array[i+1]);
        }
      }
    }

}
```

ÖZGEÇMİŞ

Soyadı	: Can
Adı	: Özgü
Doğum Tarihi	: 10.07.1978
Doğum Yeri	: Manisa
Uyruk	: T.C.
Ev Adresi	: Adakale Mah. İzmir Cad. No:54/5 MANİSA
İş Adresi	: Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, Kampus Network Yönetim Grubu Bornova/İZMİR
Email	: ozgucan@bornova.ege.edu.tr
Ev Telefonu	: 0-236-2318527
İş Telefonu	: 0-232-3423232 / 223
Yabancı Dil	: İngilizce, Almanca
Eğitim	:
1999 –	Yüksek Lisans, Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, Ağ Teknolojileri Bilim Dalı
1995 – 1999	Lisans, Selçuk Üniversitesi, Bilgisayar Mühendisliği Bölümü
1988 – 1995	Lise, Manisa Fatih Anadolu Lisesi