1-)HOCAM PDF’İ SONUNA KADAR TAKİP EDİNİZ.

2-)HOCAM BİLMOODLEDAKİ KODUMUN MAİN METHODUNU DERLEYİCİMDE ÇALIŞITARAMADIM(BİLMOODLEDA SORUNSUZ ÇALIŞIYOR) BUNDAN DOLAYI YENİ BİR TANE DAHA JAVA PROJESİ AÇIP HEM BRUTE FORCE HEM DIVIDE AND CONQUER’I YAZDIM DİVİDE AND CONQUERDA ARRAYLİST KULLANDIM VE BU YAZDIĞIM YENİ PROJEİNİN MAİNİ SORUNSUZ ÇALIŞTIRDIM VE BUARADAKİ BÜTÜN GRAFİKLERİ O KOD ÜZERİNDEN HESAPLAYARAK ÇIKARTTIM.(HER İKİ DOSYADA BENDE MEVCUT BİLMOODLEDAKİ DE,SONRADAN YAZDIĞIMDA)

3-) VERİLER BİLGİSAYAR HER TIKLAMADA FARKLI HESAP YAPTIĞI İÇİN O ANKİ TIKLAMA BAZ ALINARAK OLUŞTURULMUŞTUR.

4-)GRAFİKLERDE K VE C DEĞERLERİ YOKTUR LAKİN A-4 KAĞIDINA ÇÖZÜMÜ YAPILIP PDF DE BULUNMAKTADIR.

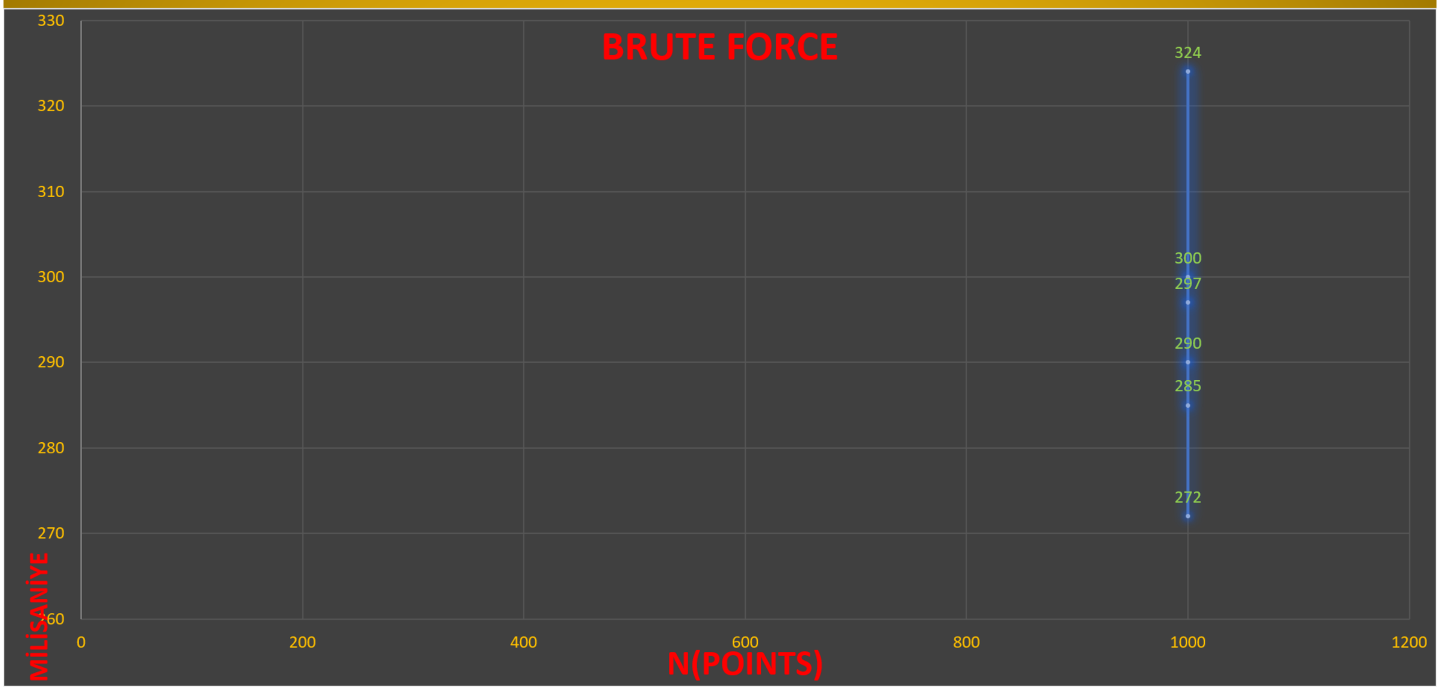
5-) KODLARIN BÜTÜN AÇIKLAMALARI // YORUM SATIRI ŞEKLİNDE .JAVA UZANTILI DOSYALAR İÇİNDE YAPILMIŞTIR İSTENDİĞİ TAKDİRDE SİZE GÖNDERİLECEKTİR.

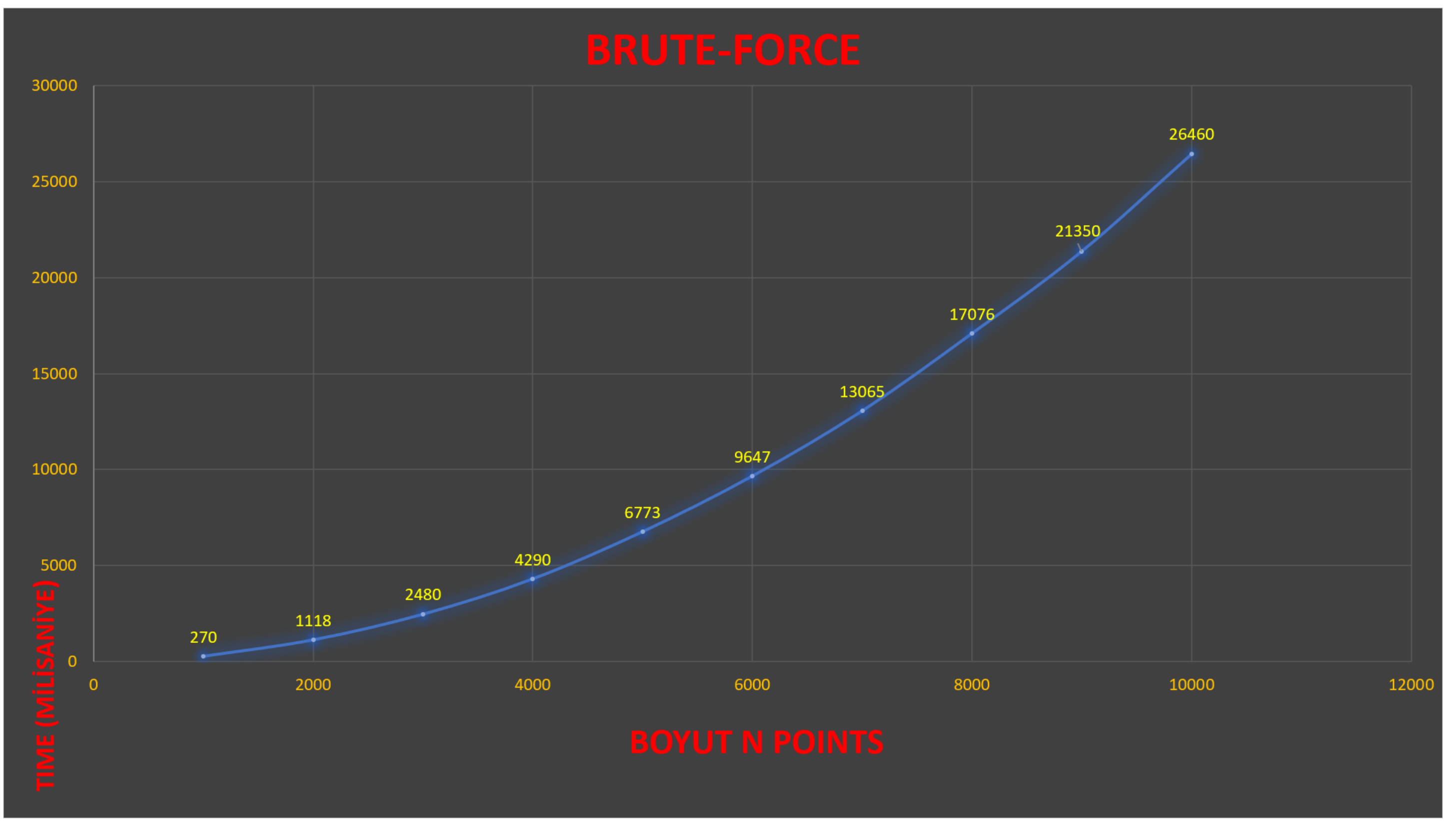
6-)GRAFİKLER EXCEL VE POWERPOINT SUNUSU KULLANILARAK OLUŞTURULMUŞTUR O DOSYALARDA BENDE MEVCUT VE İSTENİLDİĞİ TAKDİRDE SİZE GÖNDERİLECEKTİR.(6-7 TANE GRAFİK OLUŞTURDUM DAHA GÜZEL ANALİZ EDEBİLMEK İÇİN)

7-) MASTER TEOREMİ SORUSU ÇÖZÜMÜ DE GENE PDF DE MEVCUTTUR.

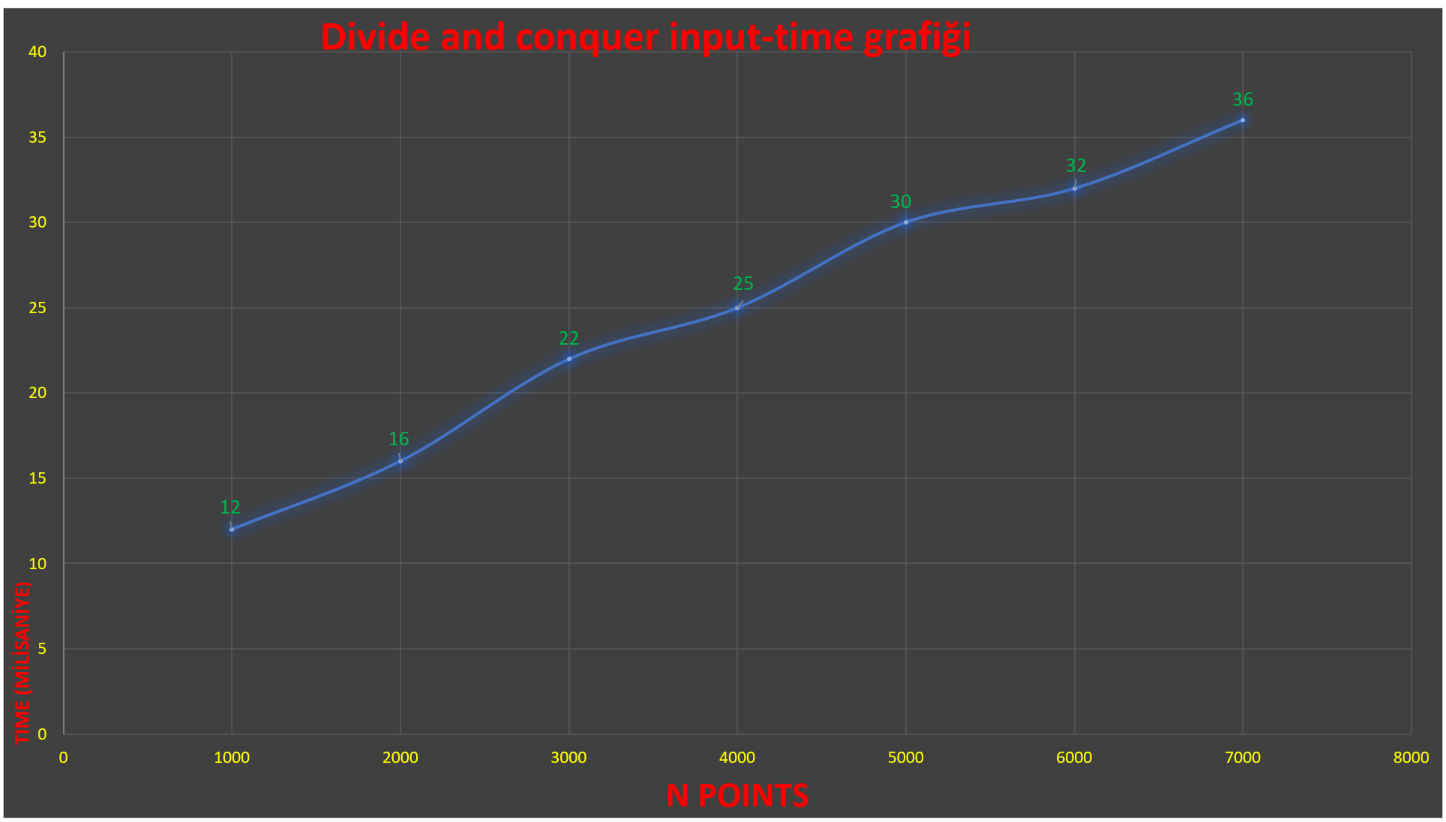
BERK BARIŞ KARA 18253007 ARA-SINAV ÖDEVİ ALGORİTMALAR

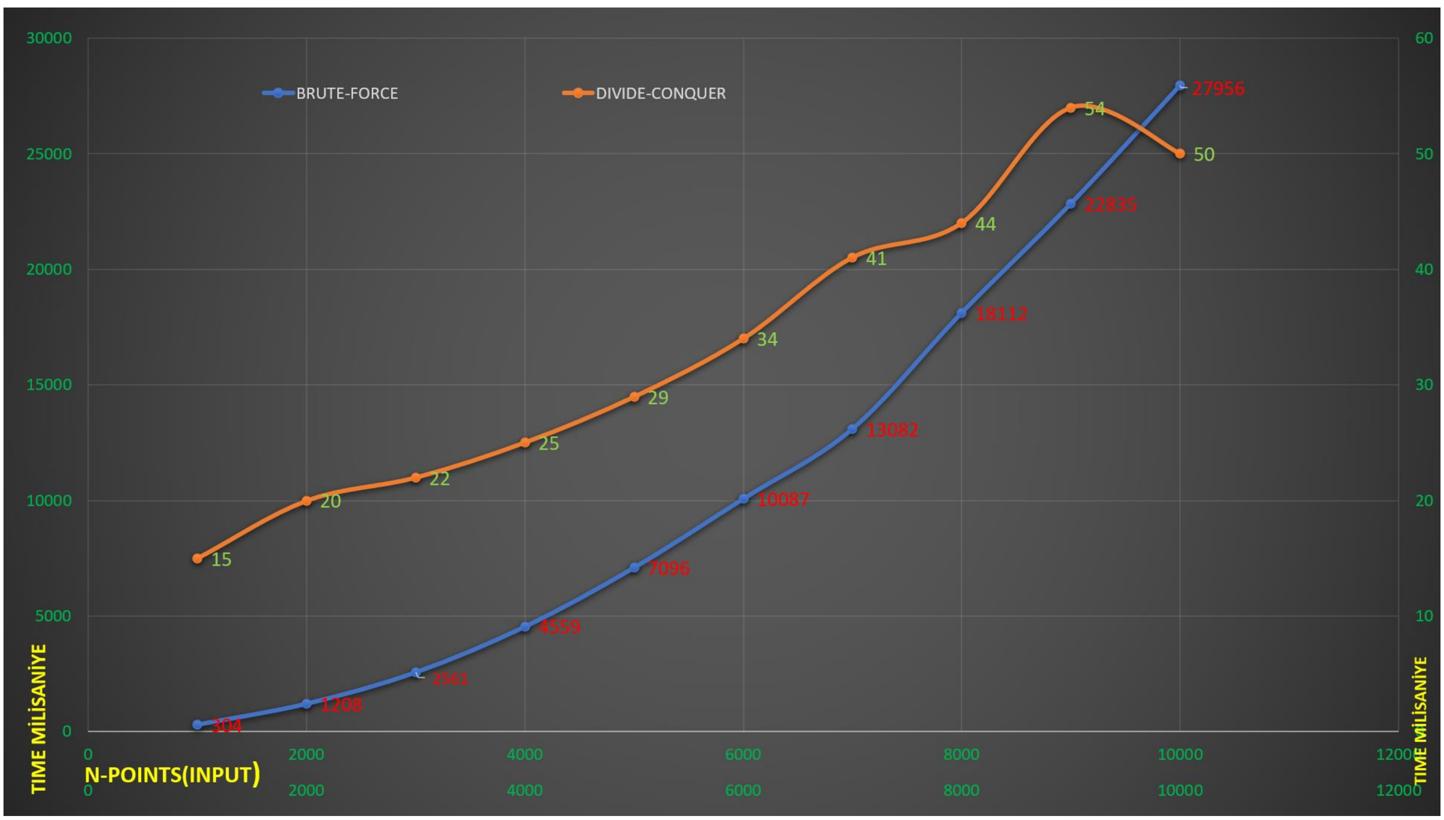
N sabit olduğunda yani sadece 1000 nokta gönderdiğimizdeki grafik,her gelen 1000 nokta için farklı zamanda hesap yapabilmiştir. Buna dayanarak diğer grafiklerde anlık o anki çalışma süresi baz alınarak çizilmiştir

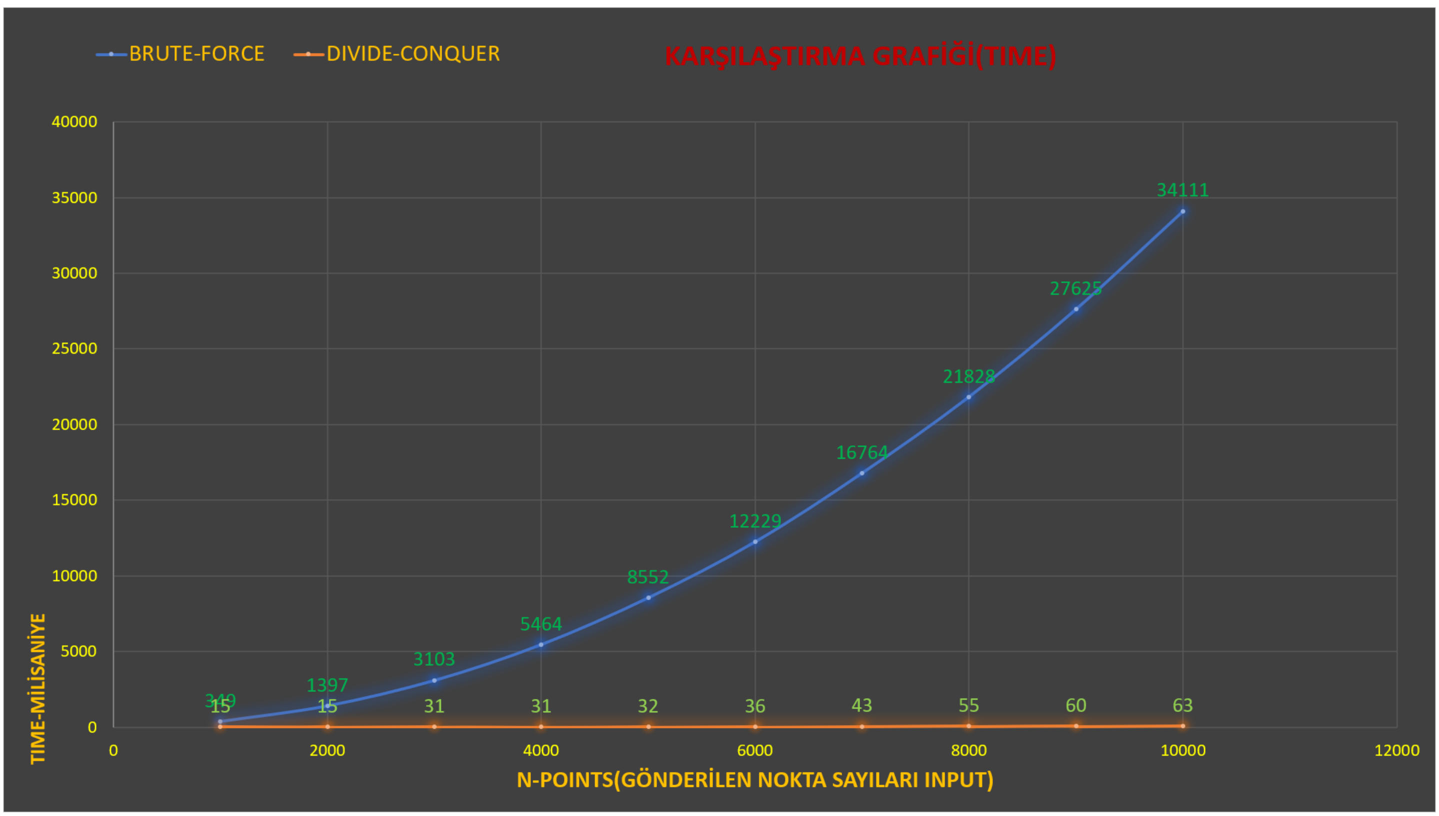




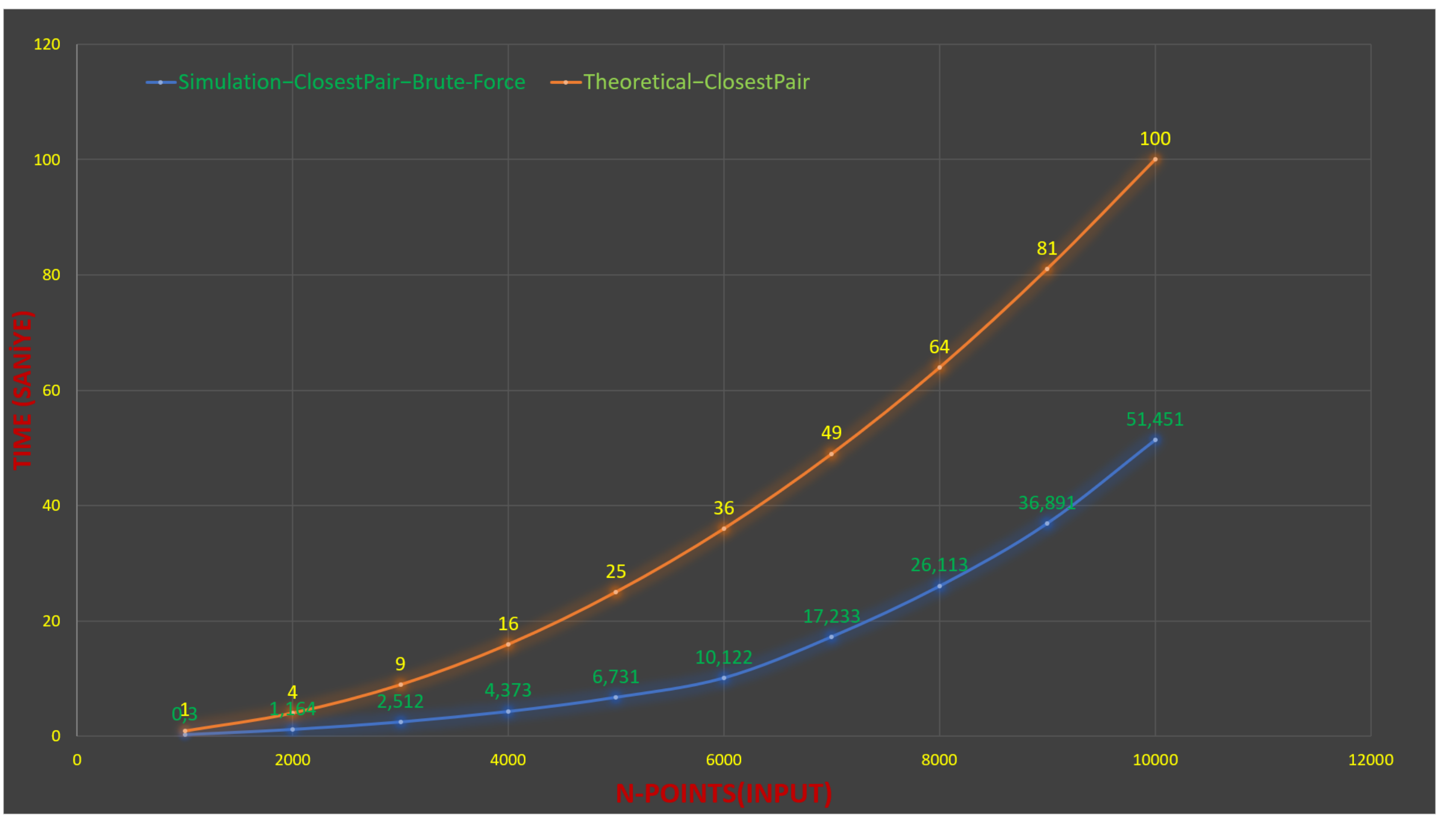
BRUTE-FORCE
30000
25000
20000
15000
10000
5000
0
0
2000
4000
6000
8000
10000
12000
270
1118
2480
4290
6773
9647
13065
17076
21350
26460
TIME (MİLİSANİYE)
BOYUT N POINTS


Divide and conquer input-time grafiği
TIME (MİLİSANİYE)
N POINTS
0
5
10
15
20
25
30
35
40
0
1000
2000
3000
4000
5000
6000
7000
8000
12
16
22
25
30
32
36

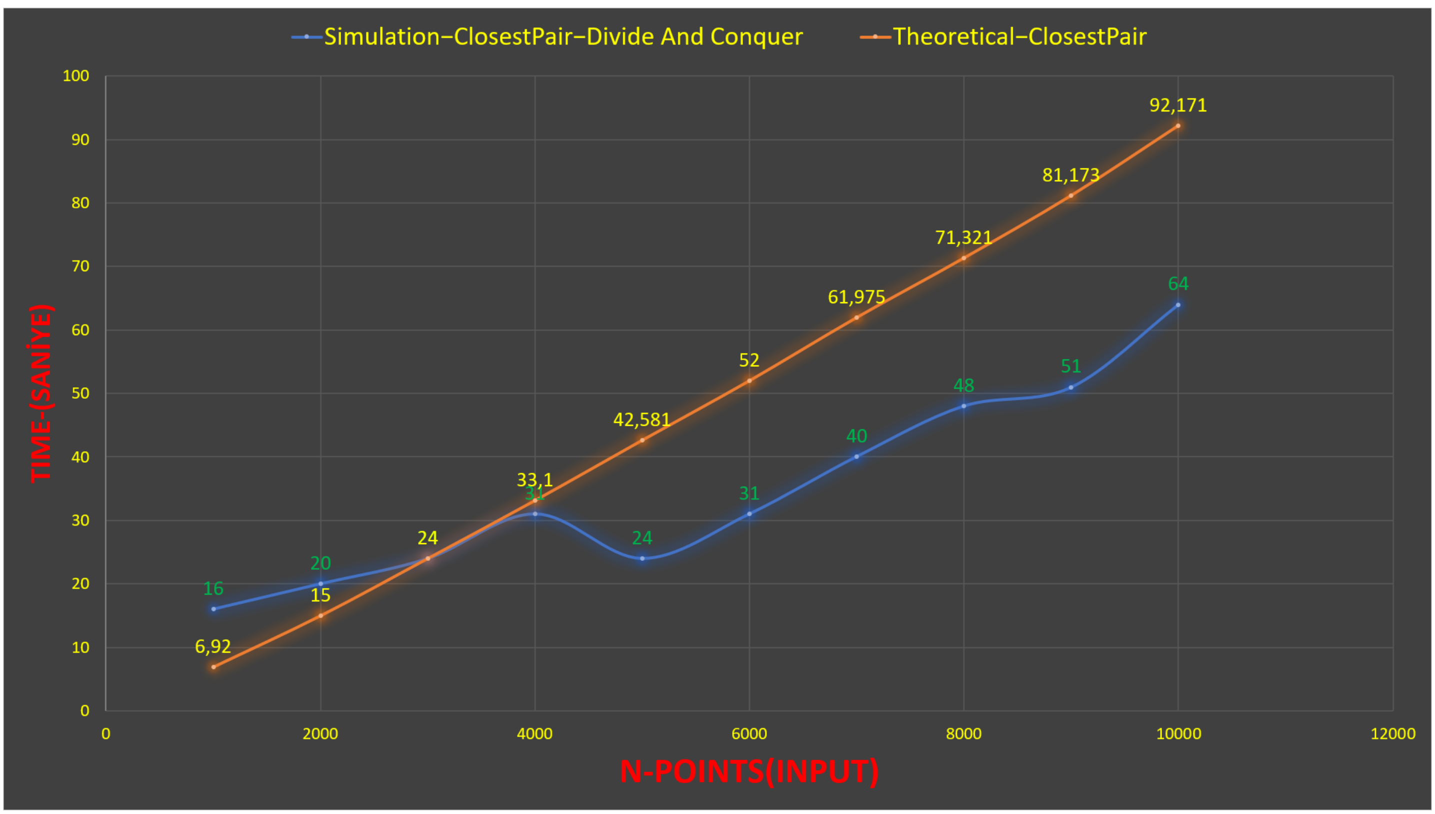

BRUTE-FORCE
DIVIDE-CONQUER
TIME MİLİSANİYE
N-POINTS(INPUT)
TIME MİLİSANİYE
304
1208
2561
4559
7096
10087
13082
18112
22835
27956
15
20
22
25
29
34
41
44
54
50



BRUTE-FORCE
DIVIDE-CONQUER
KARŞILAŞTIRMA GRAFİĞİ(TIME)
40000
35000
30000
25000
20000
15000
10000
5000
0
TIME-MİLİSANİYE
0
2000
4000
6000
8000
10000
12000
N-POINTS(GÖNDERİLEN NOKTA SAYILARI INPUT)
349
1397
3103
5464
8552
12229
16764
21828
27625
34111
15
15
31
31
32
36
43
55
60
63



Simulation–ClosestPair–Brute-Force
Theoretical–ClosestPair
TIME (SANIYE)
N-POINTS(INPUT)
0
20
40
60
80
100
120
0
2000
4000
6000
8000
10000
12000
1
4
9
16
25
36
49
64
81
100
1,164
2,512
4,373
6,731
10,122
17,233
26,113
36,891
51,451


Simulation−ClosestPair−Divide And Conquer
Theoretical−ClosestPair
100
90
80
70
60
50
40
30
20
10
0
TIME-(SANİYE)
0
2000
4000
6000
8000
10000
12000
N-POINTS(INPUT)
92,171
81,173
71,321
61,975
52
42,581
33,1
24
15
6,92
64
51
48
40
31
24
31
20
16

Brute-Force

$$n.c + n^2.c_1 + n.c_2 + n^2.c_3$$

yazdığım koda göre constant (sabit) kısımlar hesaplanaraktan big-oh 'su $\boxed{O(n^2)}$ gelmektedir.

Berk Barış KARA
18253007

$T(n) = 4T(n/5) + n^3$, $T(1) = 1$, tekrar etme ilişkisinde $T(n)$'in zaman karmaşıklığını master teoremi ile bulunuz.

$$T(n) = \begin{cases} O(n^{\log_b a}) & f(n) = O(n^{\log_b a - \varepsilon}) \\ O(n^{\log_b a} \log n) & f(n) = O(n^{\log_b a}) \\ O(f(n)) & f(n) = \Omega(n^{\log_b a + \varepsilon}) \text{ ve } af(n/b) < c f(n) \text{ büyük } n \end{cases} \quad \varepsilon > 0,\ c < 1$$

$T(n) = 4T(n/5) + n^3$

$a = 4,\ b = 5,\ f(n) = n^3$

$n^{\log_b a} = n^{(\log_5 4)}$

$n^{3.2\log_5 2} \rightarrow n^{3.2\log_5 2}$

$2\log_5 2 = 0.861 \cong 1$

$n^{3.1} = O(n^3)$

$\rightarrow n^{\log_b a + \varepsilon}$ → Epsilon (0.139) → 3. Kural

$T(n) = \Theta(f(n))$

$\boxed{T(n) = \Theta(n^3)}$ bulunur.

Divide and Conquer

$$T(n) = T(n/2) + T(n/2) + f(n)$$

$$= 2T(n/2) + f(n)$$

$$T(n) = aT(n/b) + f(n)$$

$T(n) = 2T(n/2) + n$ örneğini divide and conquer yaklaşımı ile bulup time-complexty'sini hesaplaydım.

$$T(n/2) = 2.T(n/2 \cdot 2) + n/2$$

$$= 2T(n/4) + n/2$$

$$T(n) = 2.(2T(n/4) + n/2) + n$$

$$= 4T(n/4) + 2n$$

$$= 8T(n/8) + 3n$$

$$\vdots$$

$$nT(n/n) + ?n$$

$\boxed{n = 2^k}$ $\boxed{k = \log_2 n}$

$$T(n) = n.T(1) + k.n$$

$$T(n) = \log n . n$$

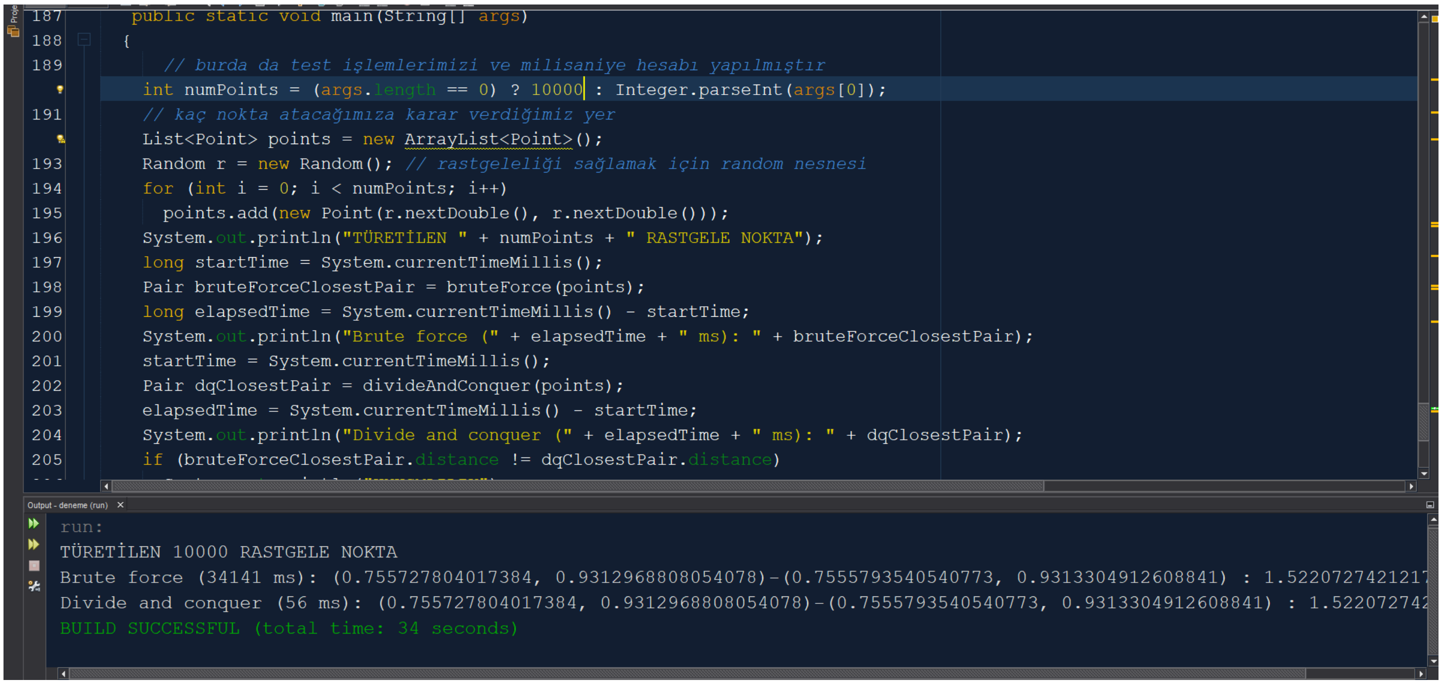
$$= n.\log n$$

$\boxed{O(n.\log n)}$ elde ederiz.

Berk Barış KARA

18253007

DERLEYİCİDEN BİR KARE:



```
public static void main(String[] args)
{
    // burda da test işlemlerimizi ve milisaniye hesabı yapılmıştır
    int numPoints = (args.length == 0) ? 10000 : Integer.parseInt(args[0]);
    // kaç nokta atacağımıza karar verdiğimiz yer
    List<Point> points = new ArrayList<Point>();
    Random r = new Random(); // rastgeleliği sağlamak için random nesnesi
    for (int i = 0; i < numPoints; i++)
      points.add(new Point(r.nextDouble(), r.nextDouble()));
    System.out.println("TÜRETİLEN " + numPoints + " RASTGELE NOKTA");
    long startTime = System.currentTimeMillis();
    Pair bruteForceClosestPair = bruteForce(points);
    long elapsedTime = System.currentTimeMillis() - startTime;
    System.out.println("Brute force (" + elapsedTime + " ms): " + bruteForceClosestPair);
    startTime = System.currentTimeMillis();
    Pair dqClosestPair = divideAndConquer(points);
    elapsedTime = System.currentTimeMillis() - startTime;
    System.out.println("Divide and conquer (" + elapsedTime + " ms): " + dqClosestPair);
    if (bruteForceClosestPair.distance != dqClosestPair.distance)
```

```
run:
TÜRETİLEN 10000 RASTGELE NOKTA
Brute force (34141 ms): (0.755727804017384, 0.9312968808054078)-(0.7555793540540773, 0.9313304912608841) : 1.5220727421217
Divide and conquer (56 ms): (0.755727804017384, 0.9312968808054078)-(0.7555793540540773, 0.9313304912608841) : 1.522072742
BUILD SUCCESSFUL (total time: 34 seconds)
```

ÖĞRETİM GÖREVLİSİ: ELİF HAYTAOĞLU

BERK BARIŞ KARA (18253007) ARA-SINAV ÖDEVİ ALGORİTMALAR