



**2021-2022 Güz Yarıyılı**

**Algoritma Analizi**

**Ödev – 1**

**Ad Soyad: Rayene Bech**

**Numara : 18011115**

## 1. Yöntem Bölümü:

- a. Öncelikle “sample.txt” dosyasında sırasıyla verilen x ve y koordinatları **readFile** fonksiyonu kullanarak okunmaktadır.
- b. Noktalar **Struct Nokta** yapısı kullanarak bir dizide kaydedilmektedir.
- c. Noktalar tutan dizi öncelikle, en kötü durumda **O(nlogn)** karmaşıklığına sahip olan, **Merge Sort** algoritması kullanarak x’e göre sıralanmaktadır.
- d. Sıralanan diziyi recursive olan **divide** fonksiyonuna input olarak verilmektedir.

- i. İlk başta durma koşullu kontrol edilmektedir.

**Durma koşullu** : dizinin eleman sayısı (Bir düzlemde bulunan Noktaların toplam sayısı)

*if(n<=3) return d;*     **M(2) = 0; or M(3) = 0;**

- ii. Her seferinde, medyan değerini referans alarak diziyi iki bölüme bölünüp her alt dizi için **divide** fonksiyonu yine çağırılmaktadır.

**M(n)=2\*M(n/2)** (çünkü divide fonksiyonu iki seferde çağırılmaktadır. Birincisi sağ dizi için, ikincisi sol dizi için)

*dl= divide(arr,start,medyan, a,b);*

*dr= divide(arr, medyan+1, finish,&a2,&b2);*

- iii. Sağ bölge için en kısa mesafe **dl**, sol bölge için de **dr** bulduktan sonra onlar arasında karşılaştırma yapıp en küçük mesafe elde edilmektedir. Bütün bu operasyonların karmaşıklığı **O(1)** ile verilmektedir.

- iv.  $x=medyan$  doğrusuna dik çizilmiş 2d yakınlığında yer alan noktalar **line[n]** bir dizide tutulmaktadır.

- v. **line** dizisindeki noktalar birbirindeki mesafe  $d_{min}$  ‘den daha kısa mesafede bulunabiliyorsa ancak  $|y_1-y_2| < d_{min}$  olması lazım. Ayrıca en fazla 6 nokta bu şartı sağlayabiliyor. Onun için, **Line** dizisi bu sefer, noktaların y koordinatlarına göre sıralanmaktadır.  $|y_1-y_2| < d_{min}$  şartı sağlayan noktalardan sadece ilk 6 noktaya bakmak yeterli olacaktır. Ve bunun karmaşıklığı  $O(n^2)$  yerine  $O(n)$  olacaktır.

- vi. Dikdörtgen içinde daha kısa bir mesafe bulunursa return edilir.

### Algoritmanın Karmaşıklığı:

- Dizi ilk başta x'e göre sıralanması (rekürsif fonksiyon dışında):  $O(n \log n)$
- Brute Case:  **$O(1)$**
- line[n] için yine Merge Sort uygulanması  **$O(n \log n)$**  olacaktır.

Bu Algoritmanın zaman karmaşıklığı  **$T(n)$**  olsa, bir bölge için diziyi iki böldüğümüz ve rekürsif fonksiyona input olarak verdiğimiz için her bir fonksiyon çağırma için  **$T(n/2)$**  olacaktır. O zaman iki bölge için  **$2 * T(n/2)$**  olacaktır.

- Ek olarak : (v) etap için: Dikdörtgen içindeki noktalar bulmak için  **$O(n)$**  olacaktır.

O zaman  **$T(n) = 2 * T(n/2) + O(n)$**

Master Theorem kullanarak  $a = 2, b = 2, d = 1$ :

$A = b^d$   **$T(n) \in O(n \log n)$**

## 2. Uygulama:

- Diziyi xe göre Sıralamak :

```
****This program finds the smallest distance between Points in a plane****  
The set of points in the plane:  
(4,8)  
(6,7)  
(7,14)  
(10,5)  
(12,11)  
(13,10)  
(20,24)  
(24,16)  
(26,29)  
(40,35)
```

Birinci fonksiyon çağrında, ilk 5 nokta arasındaki en kısa mesafeyi bulmaya çalışılacak:

- İlk üç nokta arasındaki en kısa mesafe **dl**:

```
The shortest distance found so far is : 2.236068  
It is between (4,8) and (6,7)
```

- Dördüncü ve beşinci noktalar arasında en kısa mesafe **dr**:

```
The shortest distance found so far is : 6.324555  
It is between (10,5) and (12,11)
```

- Min(dl,dr):

```
The shortest distance found so far is : 2.236068  
It is between (4,8) and (6,7)
```

Şimdi son 5 nokta arasındaki en kısa mesafeyi bulmaya çalışılacak:

- Altıncı, yedinci ve sekizinci noktalar arasında en kısa mesafe **dl**:

```
The shortest distance found so far is : 8.944272  
It is between (20,24) and (24,16)
```

- Son iki nokta arasında en kısa mesafe **dr**:

```
The shortest distance found so far is : 15.231546  
It is between (26,29) and (40,35)
```

- Dikdörtgende bulunan en kısa mesafe:

```
The shortest distance found so far is : 7.810250  
It is between (20,24) and (26,29)
```

- İlk Çağrındaki yapılan bölmeye göre dikdörtgen içinde en kısa mesafe:

```
The shortest distance found so far is : 1.414214  
It is between (13,10) and (12,11)
```

Sonuç olarak en kısa mesafe:

```
The final result of the shortest distance is: 1.414214  
It is between (13,10) and (12,11)  
-----
```