

# ALGORİTMA ANALİZİ

## ÖDEV 1

Mert Şamil Gül

16011019

Closest-Pair Problem

# Yöntem

1. .txt dosyasında bulunan noktaların sayısı alınır.
2. Noktaların tutulacağı matris için nokta sayısı kadar yer açılır.
3. Noktalar dosyadan satır satır alınarak matrise kaydedilir.
4. Matristeki noktalar x koordinatına göre sıralanır.
5. Noktaların bulunduğu matris, matrisin ilk ve son indisi ve en yakın noktaların koordinatlarının tutulduğu dizinin pointerı rekürsif fonksiyona gönderilir.
6. Rekürsif fonksiyonun durma kontrolü nokta sayısının 3 veya daha az olmasıdır.
7. Eğer nokta sayısı 3'ten fazla ise matris ortadan 2 ye bölünüp sol ve sağ olmak üzere tekrar rekürsif fonksiyona gönderilir.
8. Sol ve sağ'dan dönecek olan dl ve dr değerlerinden küçük olan değer d olarak kaydedilir.
9. En yakın nokta ikilisi sol ve sağ tarafta olabileceğinden medyanın d kadar solunda ve d kadar sağında olan noktalar bulunarak ayrı bir matrise atılır.
10. Yeni matristeki noktalar y koordinatlarına göre küçükten büyüğe sıralanır.
11. Sıralanan noktalar küçükten başlanarak y koordinatı farkı d den küçük olan sonraki noktalarla karşılaştırılır. Uzaklığı d den küçük 2 nokta bulunursa d güncellenir her karşılaştırmada en yakın noktaların bulunduğu diziyle karşılaştırıldığı için noktalar da güncellenir.
12. d değeri son hali ile döndürülür.

Rekürsif fonksiyonda dizi ikiye bölünerek 2 kere çağırıldığı için

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

f(n) içinde sort işlemi ve dikdörtgen durumu kontrolü yapılmaktadır. Bu yüzden  $f(n) = Q(n) + R(n)$  şeklinde yazılabilir.

R(n) dikdörtgen durum kontrolünde soruda belirtildiği bir noktayla en fazla 6 nokta kontrol edildiği için karmaşıklığı  $O(n)$ 'dir.

Q(n) sort fonksiyonun da karmaşıklığı  $O(n \log n)$  dir.

Bu yüzden  $T(n) = 2 * T(n/2) + O(n) + O(n \log n)$

$O(n) \in O(n \log n)$  olduğu için  $O(n)$  silinebilir.

$T(n) = 2 * T(n/2) + O(n \log n)$  durumunda yapılan araştırma üzerine[1]

$$f(n) \in \Theta(n^{\log_b a} \log^k n)$$

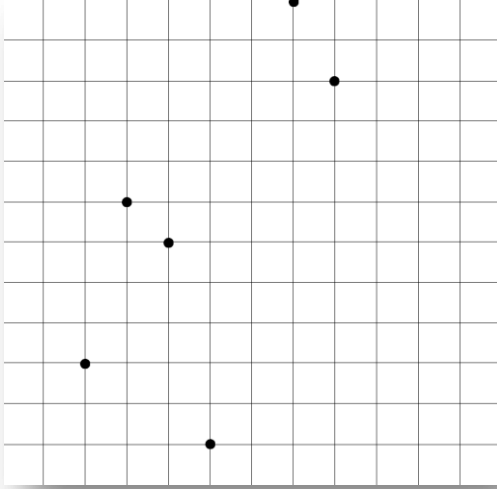
$$T(n) \in \Theta(n^{\log_b a} \log^{k+1} n)$$

Formülüyle  $T(n) = O(n (\log n)^2)$  şeklinde bulunur.

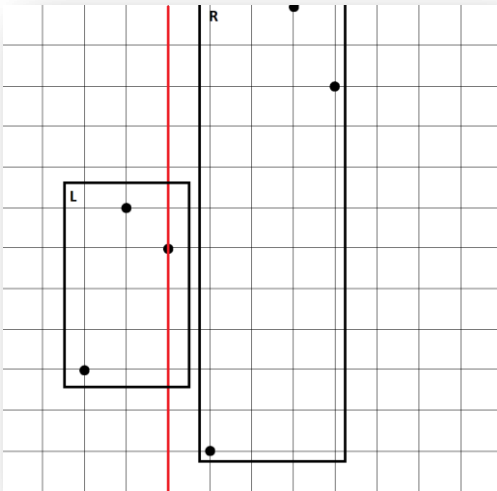
# UYGULAMA

## Örnek 1.

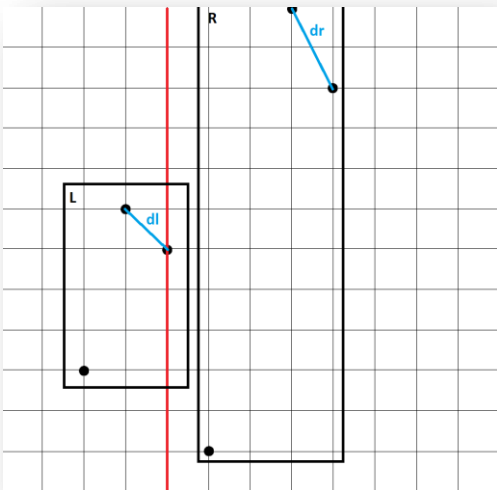
Noktalar: (2,3) , (8,10) , (4,6), (5,1), (3,7), (7,12)



- Verilen noktalar soruda istendiği gibi x koordinatlarına göre sıralanır.
- Matris  $\{(2,3), \{3,7\}, \{4,6\}, \{5,1\}, \{7,12\}, \{8,10\}\}$  şekline getirilir.
- $j(\text{son elemanın indisi}) - i(\text{ilk elemanın indisi}) \leq 2$   
Eğer kontrol sağlanırsa brute force a geçilecektir.
- Kontrol sağlanmadığı için nokta sayısı 3ten fazla olduğu bilindiğinden noktalar medyandan 2 ye bölünüp L ve R olarak rekürsif fonksiyona gönderilir.
- Bunun için noktaların bölüneceği mid noktası  $(i+j)/2$  hesaplanır
- Ardından matrisin i indisinden mid e kadar olan kısmı sol
- mid+1 den j ye kadar olan kısmı sağ bölge olarak rekürsif fonksiyona gönderilir.

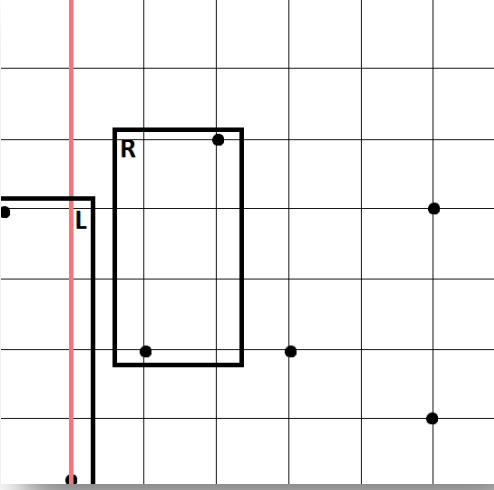


- Rekürsif fonksiyonlardan dönecek olan değerler sol kısımdaki en yakın noktaların uzaklığı  $d_l$  ve sağ kısımdaki noktaların en yakın uzaklığı  $d_r$  olarak saklanır.
- Gönderilen ilk rekürsif fonksiyonda  $j-i \leq 2$  kontrolü doğru olduğundan brute-force a geçilir. Brute-force kısmında  $j-i = 2$  kontrolü yapılarak noktaların 2 mi yoksa 3 mü olduğu anlaşılır. Şu anki durumda 3 olduğu için 3 noktanın birbiri ile uzaklığı , distance fonksiyonu ile hesaplanıp minimum olan uzaklık döndürülür.
- Sağdaki 3 nokta içinde aynı işlem yapılarak sonuç döndürülür.

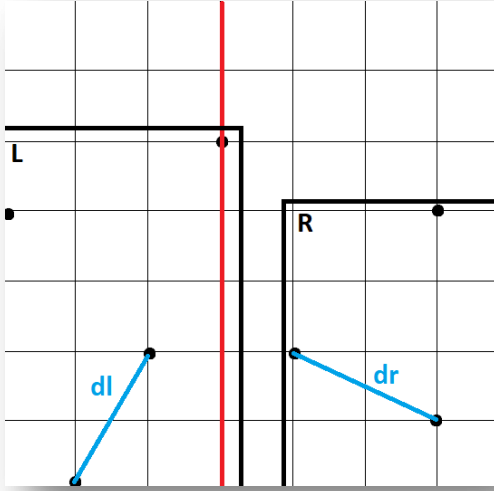


- Dönen sonuçlardan ( $d_l$  ve  $d_r$ ) küçük olanı  $d = \min(d_l, d_r)$  bulunur.
- Fakat en yakın noktaların solda ve sağda olabildiği bilindiği için medyanın d kadar solundaki ve d kadar sağındaki noktaların da kontrol edilmesi gerekir.
- Kontrol edilmesi gereken noktaları bulmak için  $i2 = \text{mid} - 1$  ve  $j2 = \text{mid} + 1$  değerleri atanır.  $i2$  değerinin gösterdiği noktanın mid noktasına x koordinatının uzaklığı d uzaklığından küçük olduğu sürece  $i2$  değeri 1 azaltılarak medyana d uzaklığındaki noktaların başlangıç indisi bulunur. Aynı şekilde  $j2$  noktasını arttırarak noktaların son indisi bulunur.

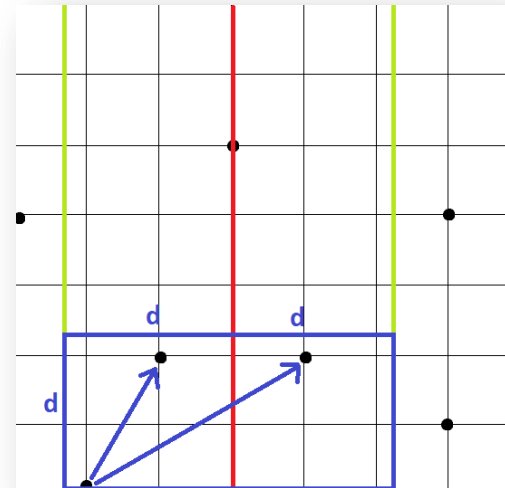




-Sol taraftaki noktalar tekrar sağ ve sola ayırıldıktan sonra brute-force uygulanır fakat en yakın nokta ikilisi sol ve sağ taraftadır. Özel durumdan faydalanarak  $d_l$  bulunur.

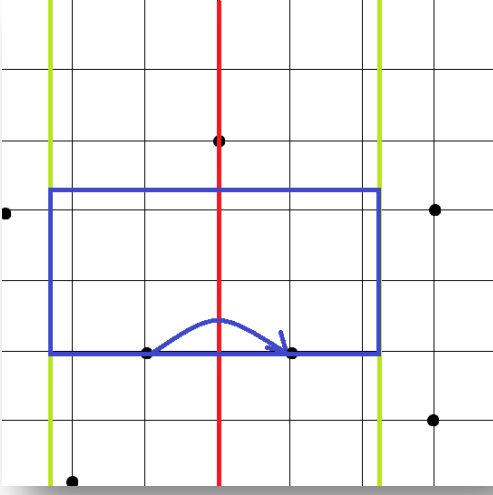


-Sol ve sağ tarafın minimum uzaklıklar eşit olduğundan  $d = d_l = d_r$  bulunur.



-Ardından medyandan  $d$  kadar uzaklaşılarak özel durumu kapsayan noktalar tespit edilir.

-Noktalar y koordinatlarına göre sıralandıktan sonra şekildeki gibi en aşağıdaki noktadan başlanarak o nokta ile dikdörtgenindeki diğer noktalar karşılaştırılır.



- Bu işleme devam edilince 2. Nokta ile aynı dikdörtgende bulunan 2 birim sağındaki noktanın uzaklığının minimum  $d$  uzaklığından daha küçük olduğu anlaşılıp  $d$  ve en yakın noktalar dizisi güncellenir.
- Devamında başka bir güncelleme olmadan  $d$  değeri 2 olarak fonksiyonda döndürülür.