

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



ALGORİTMA ANALİZİ 2. ÖDEV

Öğrenci No: 19011010

Öğrenci Adı Soyadı: Beyda Güler

Öğrenci e-posta: beyda.guler@std.yildiz.edu.tr

Ders/Grup: BLM3021 - Algoritma Analizi / Grup:1

Ders Yürütücüsü

Mine Elif Karslıgil

Kasım, 2022

İçindekiler

YÖNTEMLER.....	3
Brute Force Technique	3
Divide and Conquer Technique	3
UYGULAMA.....	5
Örnek-1.....	5
Örnek-2.....	5
Örnek-3.....	6
Örnek-4.....	6

YÖNTEMLER

Brute Force Technique

BRUTE FORCE TECHNIQUE

```
for i ← 0 to N do
  sum ← 0
  for j ← i to N do
    sum ← sum + A[i][j]
    if sum > max
      max ← sum
```

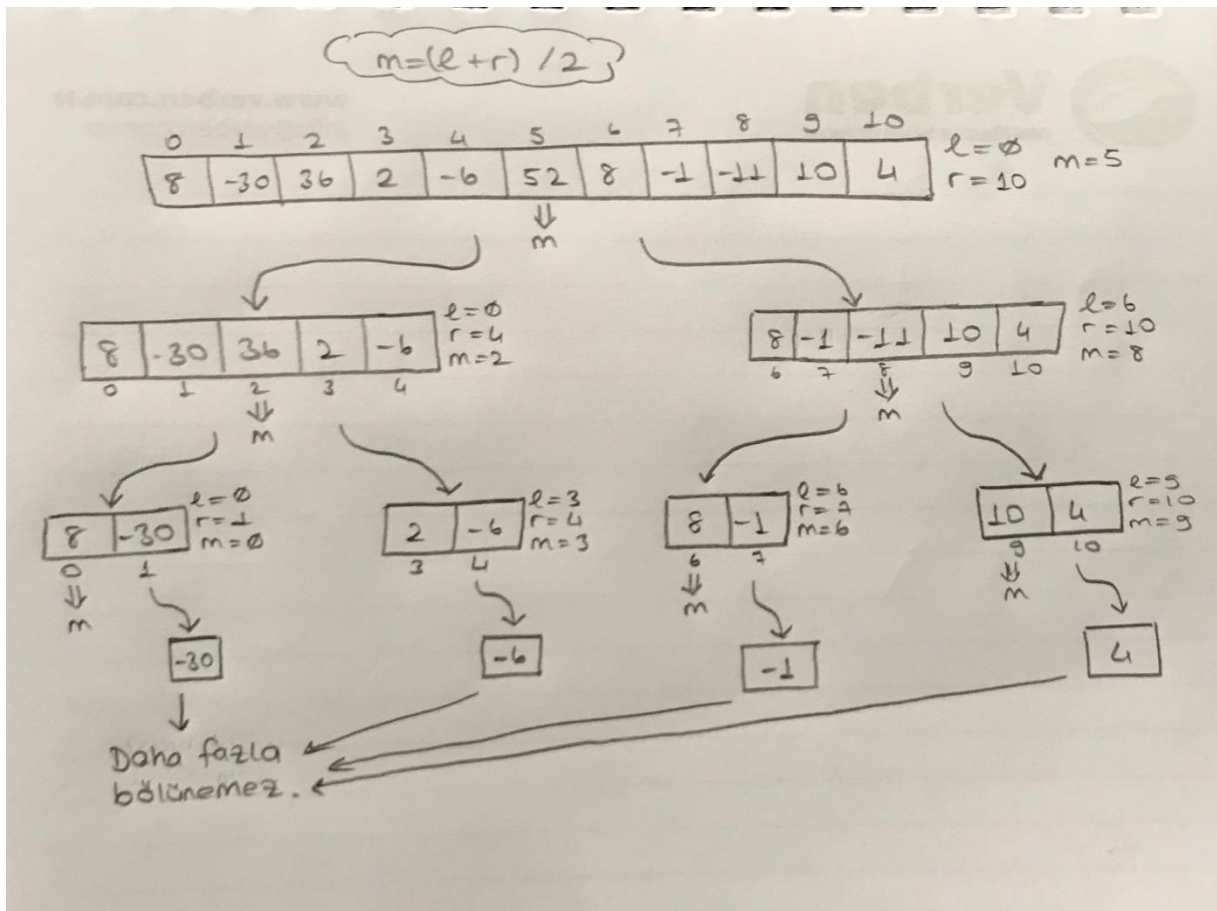
Basic Operation $\rightarrow \sum_{i=0}^N \sum_{j=i}^N 1$

$$= \sum_{i=0}^N (N-i+1) = \frac{(N+1)(N+2)}{2} = \frac{N^2+3N+2}{2} \approx N^2$$

$\left. \begin{matrix} C_{best} \\ C_{avg} \\ C_{worst} \end{matrix} \right\} \theta(N^2)$

Brute Force, en kolay implemente edilen fakat karmaşıklığı en yüksek olan çözümdür. İki for döngüsü kullanarak dizinin 0. indisinden N-1. indise kadar bütün elemanları tek tek kendinden N-1. Elemana kadar topluyor ve her adımda max değişkeni ile karşılaştırıyor. Max değişkeninden daha büyük bir toplam elde edildiğinde max güncelleniyor.

Divide and Conquer Technique



DIVIDE & CONQUER TECHNIQUE

```

DivideConquer(arr, left, right)
    mid ← (left + right) / 2
    if (left == right)
        return arr[left]
    else
        return MAX(DivideConquer(arr, left, mid-1),
                   DivideConquer(arr, mid+1, right),
                   MaxSum(arr, left, mid, right))
    
```

MaxSum(arr, left, mid, right)

```

sum ← 0
left_sum ← -9999
right_sum ← -9999
    
```

```

for i ← mid to left do
    sum ← sum + arr[i]
    if sum > left_sum
        left_sum ← sum
sum ← 0
    
```

```

for i ← mid to right do
    sum ← sum + arr[i]
    if sum > right_sum
        right_sum ← sum
    
```

```

return MAX(left_sum + right_sum - arr[mid],
           left_sum,
           right_sum)
    
```

Basic Operation
↓
 $O(N)$

Recurrence Relation

$$T(N) = \begin{cases} O(1) & , N=1 \\ 2*T(N/2) + N & , N>1 \end{cases}$$

↓
MaxSum

Basic Operation

$$\begin{aligned}
 T(N) &= 2*T(N/2) + N \\
 &= 2*(2*T(N/4) + N/2) + N \\
 &= 4*T(N/4) + 2N \\
 &= 4*(2*T(N/8) + N/4) + 2N \\
 &= 8*T(N/8) + 3N
 \end{aligned}$$

$$\Rightarrow 2^i * T(N/2^i) + i * N$$

$$i = \log_2 N$$

$$\Rightarrow N * \underbrace{T(1)}_1 + N * \log_2 N$$

$$\approx \Theta(N \log N)$$

$\begin{cases} C_{best} \\ C_{avg} \\ C_{worst} \end{cases}$

! Başlangıç ve bitiş indexlerini bulma işlemleri hesaplamaya dahil edilmedi.

Bu teknikte dizi her adımda left ve right olmak üzere iki subarray'e ayrılıyor. Bu subarraylerin içinde de subarrayler recursive şekilde oluşturuluyor (subarray'in boyutu 1 olana kadar). Mid değişkenini kapsamayan left subarray, right subarray ve mid'i de toplama dahil eden diğer dizi her adımda karşılaştırılarak maximum toplama ulaşıyor. Recurrence Relation yukardaki hesaplamalarda da görüldüğü gibi $T(N) = 2*T(N/2) + N$ şeklinde yazılabilir. Recurrence Relation'ı yazdıktan sonra Backward Substitution yaparak total karmaşıklığı $\Theta(N \log N)$ olarak buluyoruz.

UYGULAMA

Örnek-1

```
Please enter the size of your array --> 11

Elements :

0 --> 8
1 --> -30
2 --> 36
3 --> 2
4 --> -6
5 --> 52
6 --> 8
7 --> -1
8 --> -11
9 --> 10
10 --> 4

-----USING BRUTE FORCE SOLUTION-----

From 2'th index to 10'th index , maximum sum is : 94

-----USING DIVIDE AND CONQUER SOLUTION-----

From 2'th index to 10'th index ,maximum sum is : 94
```

Örnek-2

```
BEYDA GULER
19011010

Please enter the size of your array --> 5

Elements :

0 --> 7
1 --> 9
2 --> -66
3 --> 5
4 --> 19

-----USING BRUTE FORCE SOLUTION-----

From 3'th index to 4'th index , maximum sum is : 24

-----USING DIVIDE AND CONQUER SOLUTION-----

From 3'th index to 4'th index ,maximum sum is : 24
```

Örnek-3

```
BEYDA GULER
19011010

Please enter the size of your array --> 3
Elements :
0 --> -9
1 --> -6
2 --> -5

-----USING BRUTE FORCE SOLUTION-----
From 2'th index to 2'th index , maximum sum is : -5

-----USING DIVIDE AND CONQUER SOLUTION-----
From 2'th index to 2'th index ,maximum sum is : -5
```

Örnek-4

```
Please enter the size of your array --> 15
Elements :
0 --> 7
1 --> 9
2 --> 12
3 --> -19
4 --> 6
5 --> 9
6 --> -2
7 --> -30
8 --> 45
9 --> 9
10 --> 1
11 --> -7
12 --> 6
13 --> 4
14 --> 3

-----USING BRUTE FORCE SOLUTION-----
From 8'th index to 14'th index , maximum sum is : 61

-----USING DIVIDE AND CONQUER SOLUTION-----
From 8'th index to 14'th index ,maximum sum is : 61
```