

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



ALGORİTMA ANALİZİ
ÖDEV RAPORU

Öğrenci No: 20011037
Öğrenci Adı Soyadı: Mehmet Şadi Özcan
Öğrenci E-Posta: sadi.ozcan@std.yildiz.edu.tr

Ders/Grup: BLM3021 / Grup 2

Video Linki: www.youtube.com/watch?v=phsglj2g2RM

Ders Yürütücüsü
Doç. Dr. Mehmet Amaç GÜVENSAN
Kasım, 2022

İÇİNDEKİLER

Ödevin İçeriği

Problemin Çözümü İçin Kullanılan Yöntemler

a-Brute Force

b-Divide and Conquer

Uygulama

a-Brute Force

b-Divide and Conquer

Kaynakça

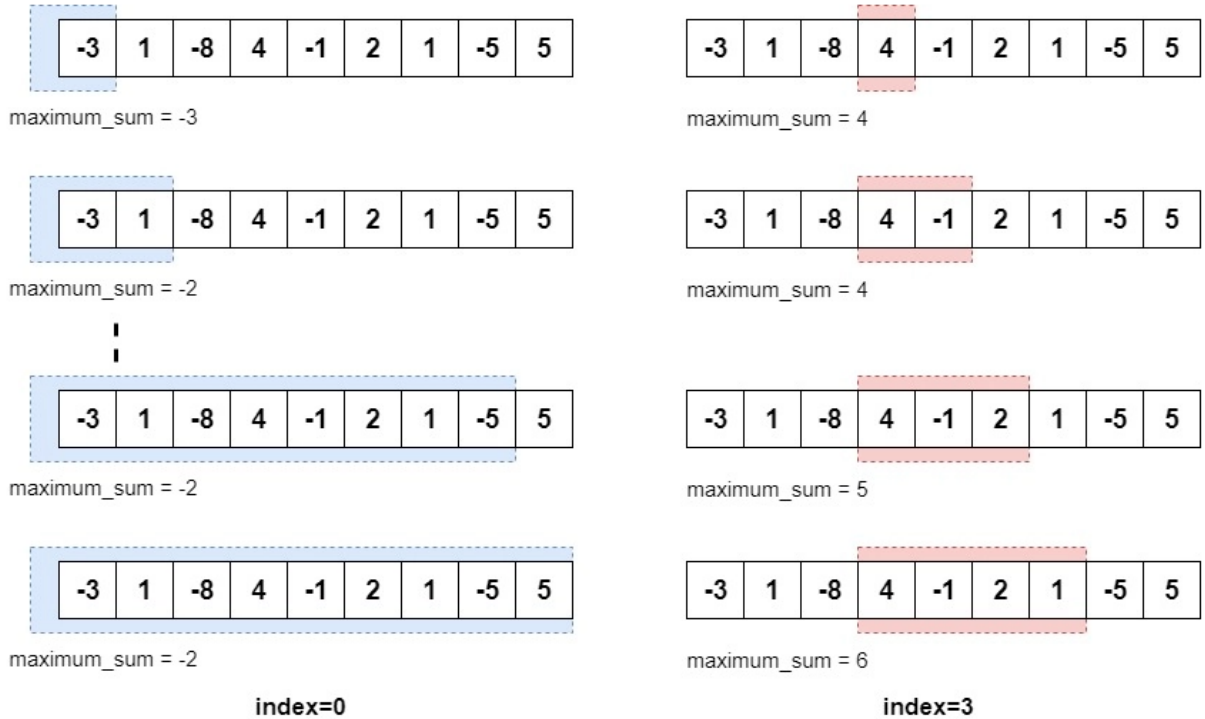
Ödevin İçeriği

Ödevde maksimum alt küme toplamı problemi bir madenci örneği üzerinden kurgulanmıştır. Problemin çözümü için brute force ve divide and conquer algoritmaları kullanılmış ve iki algoritma matematiksel olarak analiz edilmiştir.

Problemin Çözümü İçin Kullanılan Yöntemler

a- Brute Force Çözüm Yöntemi

Brute force çözüm yönteminin uygulanmasında verilen dizinin ilk elemanından başlanarak o elemanın başında bulunduğu bütün sıralı dizilere bakılarak her dizideki toplamlar hesaplanmıştır. Maksimum toplam max_sum değişkeninde saklanmıştır. Maksimum toplamı veren dizinin hangi indeksten hangi indekse kadar olduğunu ekrana yazdırmak üzere kod içerisinde start ve end değişkenleri tutulmuştur.



Brute force algoritmasıyla yapılan çözüm sözde kod olarak aşağıdaki gibi ifade edilebilir:

max_sum $\rightarrow -\infty$

for i from 0 to n-1 do

sum $\rightarrow 0$

for j from i to n do

sum \rightarrow *sum* + *array[j]*

if sum > *max_sum*

max_sum \rightarrow *sum*

Yukarıdan da görüleceği üzere algoritma içerisinde iç içe iki for döngüsü olduğu için karmaşıklığı Big-O Gösterimi ile **$O(n^2)$** olarak ifade edilir.

b- Divide and Conquer Çözüm Yöntemi

Divide and Conquer algoritmasıyla çözüm yapabilmek için öncelikle max adlı, verilen iki sayı arasından büyük olanı bulan bir fonksiyon tanımlanmıştır.

Bu algoritmadaki mantık eldeki diziyi küçük parçalara bölerek rekürsif bir şekilde problemi çözmektir. Rekürsif fonksiyonun base case'i, dizinin tek eleman içeren bir dizi olduğu durumdur. Bu durumda fonksiyon, tek elemanı olan dizideki tam sayıyı döndürecektir.

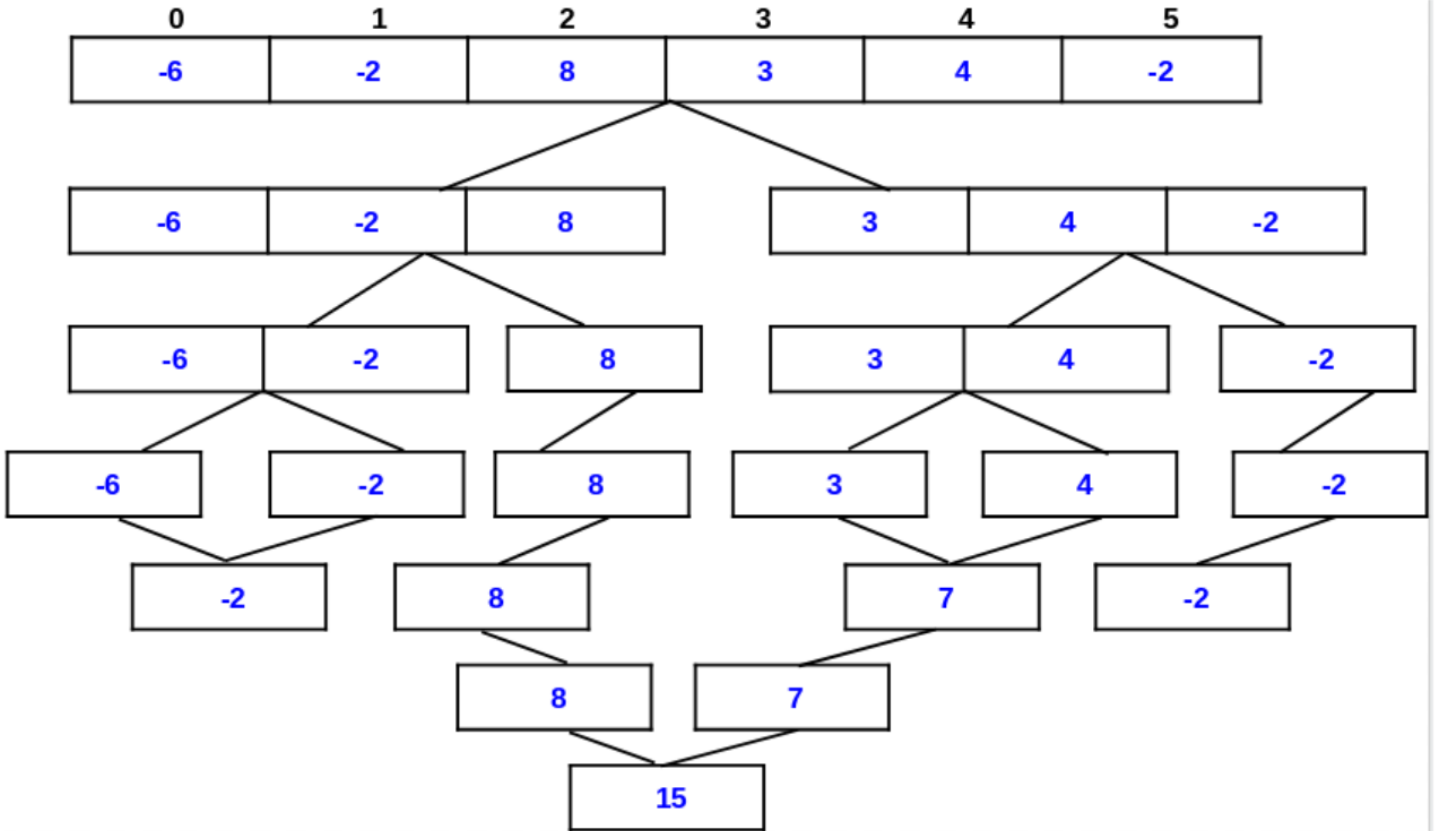
Base case yazıldıktan sonra mid tanımlanmıştır. Mid eldeki diziye ortasından ikiye bölmek üzere kullanılacaktır.

Eldeki dizi ikiye bölündükten sonra önce soldaki dizinin maksimum sıralı toplamına bakılır. Buna bakmak için dizide mid noktasından geriye doğru gidilir. Toplam, maksimum toplamdan büyükse maksimum toplam güncellenir. Başlangıç noktası kaydedilir.

Hemen ardından sağdaki diziye bakılır. Ortanca eleman soldaki dizide olduğu için, sağdaki dizi ortanca elemandan bir büyük elemandan başlar ve son elemana kadar gider. Toplam, maksimum toplamdan büyükse maksimum toplam güncellenir ve bitiş noktası kaydedilir.

Dizideki en büyük toplamı veren kısım tamamen sol dizinin içinde kalmış olabilir, tamamen sağ dizinin içinde kalmış olabilir, ya da ikisinin kesişiminde olabilir. Bunların her birini kontrol etmek üzere öncelikle sağ ve sol dizilerin maksimum toplamları karşılaştırılır ve büyük olan kaydedilir. Daha sonrasında sağ ve sol dizinin maksimum toplamlarının toplamı, bu kaydedilen değerle karşılaştırılır. Bu karşılaştırmalar sonucu en büyük bulunan değer `max_sum` olarak kaydedilir ve fonksiyon tarafından döndürülür.

Örnek bir çözüm:



Divide and Conquer algoritmasıyla yapılan çözüm sözde kod olarak aşağıdaki gibi ifade edilebilir:

function divideandconquer

if $start = end$

return $A[start]$

else

$mid = (start + end) / 2$

$left_sum \rightarrow -\infty$

$sum \rightarrow 0$

for i ***from*** mid ***to*** $start$ ***do***

$sum \rightarrow sum + array[i]$

if $sum > left_sum$

$left_sum \rightarrow sum$

$right_sum \rightarrow -\infty$

$sum \rightarrow 0$

for i ***from*** $mid+1$ ***to*** end ***do***

$sum \rightarrow sum + array[i]$

if $sum > right_sum$

$right_sum \rightarrow sum$

```
max_left_right →  
max(divideandconquer(array,start,mid),  
divideandconquer(array,mid+1,end))  
  
max_sum → max(max_left_right, left_sum + right_sum)  
  
return max_sum  
  
endfunction
```

Bu rekörsif fonksiyonun rekürans bağıntısı

$$T(n) = O(1) , n=1'se$$

$$T(n) = 2T(n/2) + O(n) , n>1'se$$

olarak yazılabilir.

Buradan Master Theorem kullanarak zaman karmaşıklığını

$O(n \log(n))$ olarak buluruz.

UYGULAMA

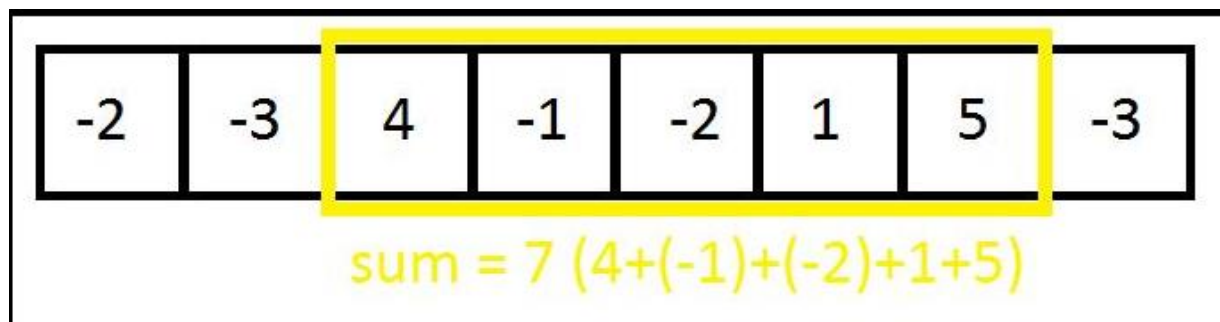
a- Brute Force

Örnek 1:

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

```
C:\Users\mehme\Desktop\algo analiz + dev\main.exe
Blok sayisi: 10
10 tam sayi giriniz.
8
-30
36
2
-6
52
8
-1
-11
10
Brute Force yontemi ile cozmek icin 1'e, Divide and Conquer yontemiyle cozmek icin 2'ye basiniz: 1
Kazilmasi gereken kesintisiz blok butunu 2 - 6
En yuksek kazanc: 92
-----
Process exited after 17.8 seconds with return value 0
Press any key to continue . . .
```

Örnek 2:



```
C:\Users\mehme\Desktop\algo analiz + dev\main.exe
Blok sayisi: 8
8 tam sayi giriniz.
-2
-3
4
-1
-2
1
5
-3
Brute Force yontemi ile cozmek icin 1'e, Divide and Conquer yontemiyle cozmek icin 2'ye basiniz: 1
Kazilmasi gereken kesintisiz blok butunu 2 - 6
En yuksek kazanc: 7
-----
```

Örnek 3:

-3	1	-8	4	-1	2	1	-5	5
----	---	----	---	----	---	---	----	---

```
C:\Users\mehme\Desktop\algo analiz + dev\main.exe
Blok sayisi: 9
9 tam sayi giriniz.
-3
1
-8
4
-1
2
1
-5
5
Brute Force yontemi ile cozmek icin 1'e, Divide and Conquer yontemiyle cozmek icin 2'ye basiniz: 1
Kazilmasi gereken kesintisiz blok butunu 3 - 6
En yuksek kazanc: 6
-----
Process exited after 18.02 seconds with return value 0
Press any key to continue . . .
```

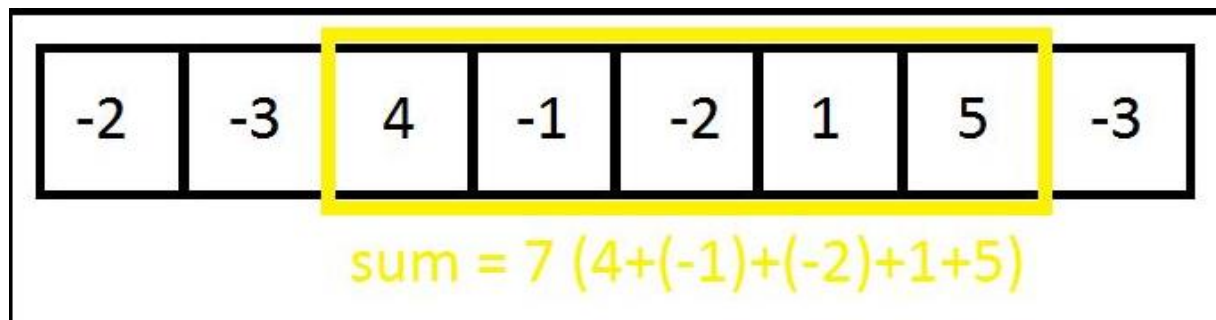
b- Divide and Conquer

Örnek 1:

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

```
C:\Users\mehme\Desktop\algo analiz + dev\main.exe
Blok sayisi: 10
10 tam sayi giriniz.
8
-30
36
2
-6
52
8
-1
-11
10
Brute Force yontemi ile cozmek icin 1'e, Divide and Conquer yontemiyle cozmek icin 2'ye basiniz: 2
Kazilmasi gereken kesintisiz blok butunu 2 - 6
En yuksek kazanc: 92
-----
Process exited after 14.06 seconds with return value 0
Press any key to continue . . .
```

Örnek 2:



```
C:\Users\mehme\Desktop\algo analiz + dev\main.exe
Blok sayisi: 8
8 tam sayi giriniz.
-2
-3
4
-1
-2
1
5
-3
Brute Force yontemi ile cozmek icin 1'e, Divide and Conquer yontemiyle cozmek icin 2'ye basiniz: 2
Kazilmasi gereken kesintisiz blok butunu 2 - 6
En yuksek kazanc: 7
-----
Process exited after 11.05 seconds with return value 0
Press any key to continue . . .
```

Örnek 3:

-3	1	-8	4	-1	2	1	-5	5
----	---	----	---	----	---	---	----	---

```
C:\Users\mehme\Desktop\algo analiz + dev\main.exe
Blok sayisi: 9
9 tam sayi giriniz.
-3
1
-8
4
-1
2
1
-5
5
Brute Force yontemi ile cozmek icin 1'e, Divide and Conquer yontemiyle cozmek icin 2'ye basiniz: 2
Kazilmasi gereken kesintisiz blok butunu 3 - 6
En yuksek kazanc: 6
-----
Process exited after 14.57 seconds with return value 0
Press any key to continue . . .
```

Ödevi Yaparken ve Raporu Hazırlarken Kullandığım Kaynaklar

<https://www.baeldung.com>

<https://scottc130.medium.com>

youtube.com/user/mycodeschool