



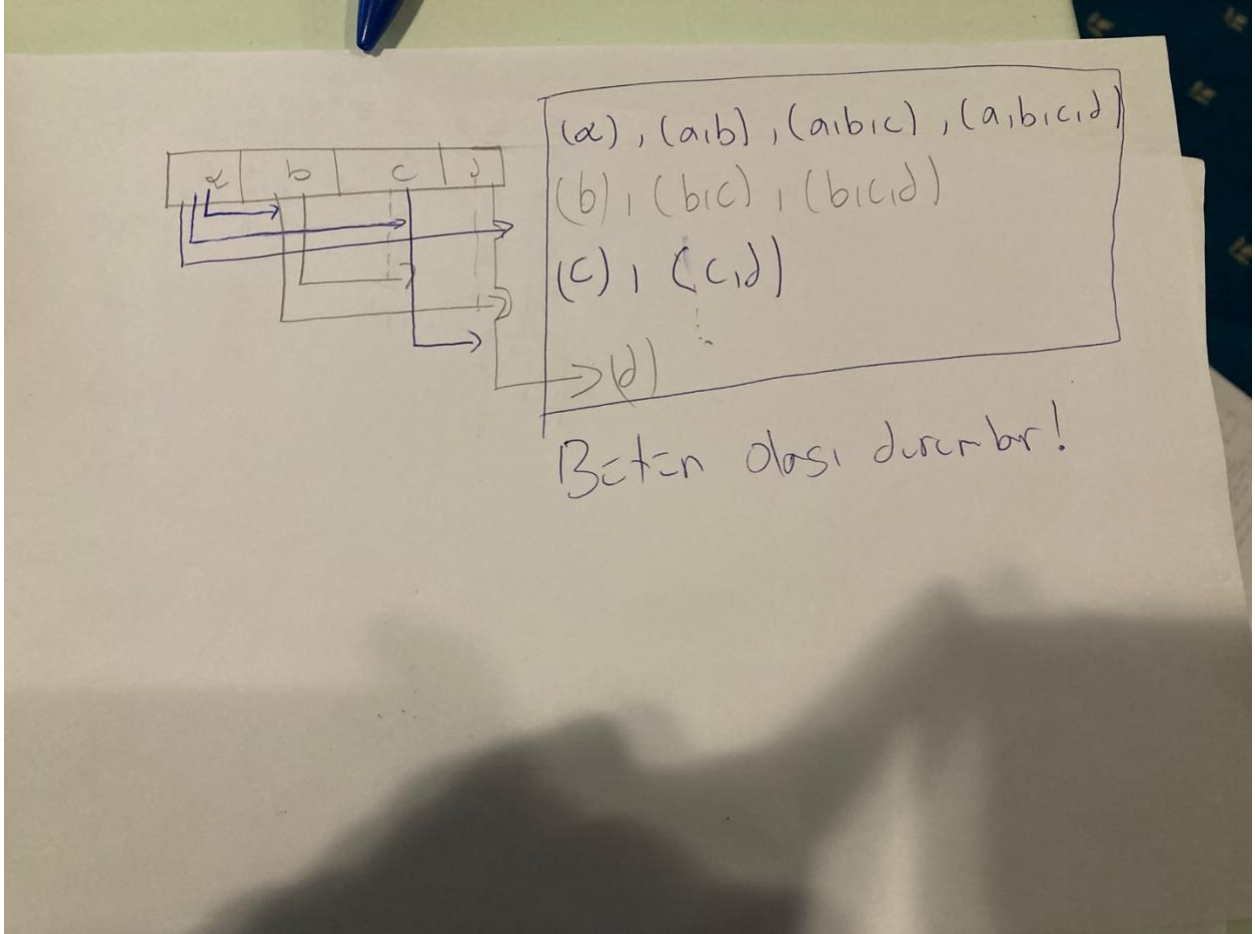
ALGORİTMA ANALİZİ GRUP 2

2. Ödev

Basel Kelziye 20011906

YÖNTEM

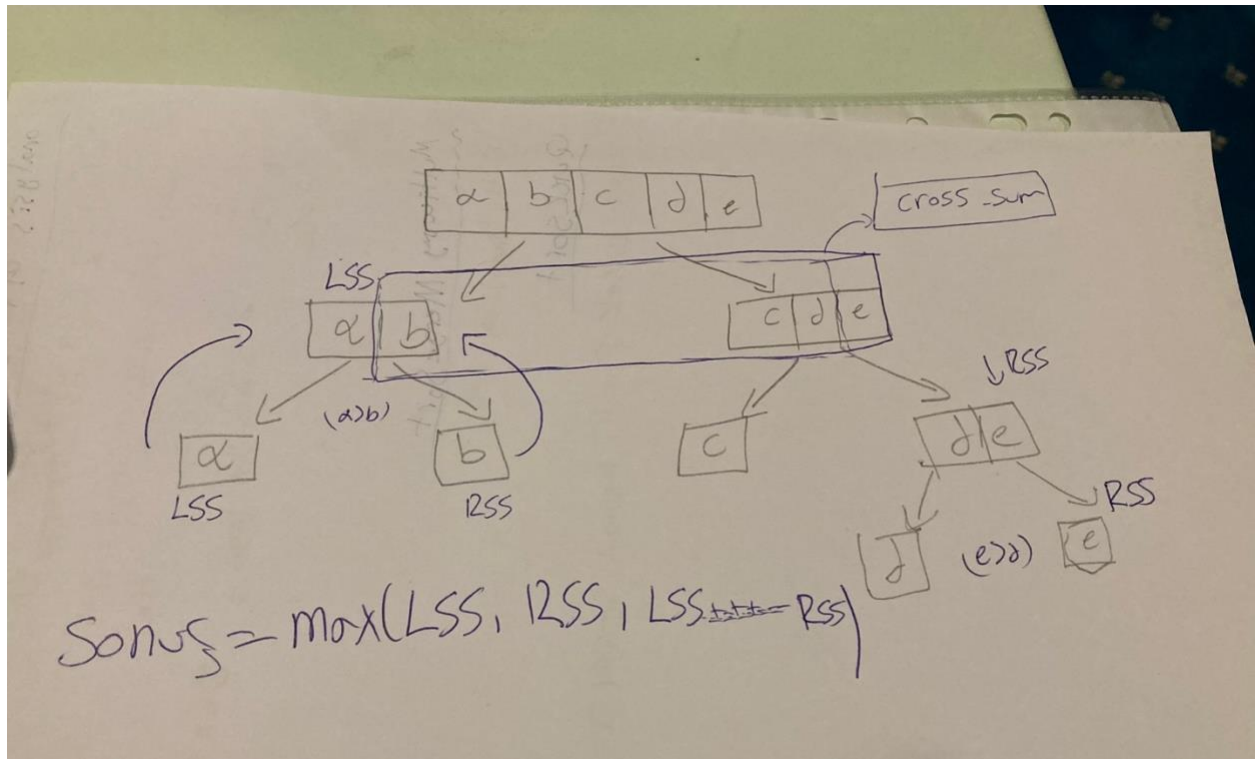
1-) Brute Force:



İç içe 2 For ile fotoğrafta gösterildiği gibi bütün durumları oluşturdum. (a,b) = (b,a) durumu sağladığı için (sırası önemli değil) tersten gelmeye gerek duyulmamıştır.

Anlık max 1 ilk eleman kabul edip oluşturulan al dizilerin toplamını her adımda kontrol eder eğer daha büyük ise max ve sol indisi (dış döngünün iteratörü) 1 sağ indisi ise iç döngünün iteratörüne atadım.

2-) Divide and Conquer



Diziyi 2 e bölerek ilerledim, her adımda oluşan alt dizinin tek elemanı kalana kadar ilerledim sonra tersten toplamaya başladım. Toplarken şunu göze alarak topladım:

Benim soldaki tek eleman mı en büyüğü yoksa sağdaki eleman mı yoksa soldan sağa kadar (cross sum) elemanların toplamı mı.

Eğer her oluşan diziyi bu mantığa inşa edersek oluşturulan ilk diziye gelince maximum toplamı elde etmiş oluruz.

UYGULAMA

Girdi:

```
// pdfteki ornek  
int arr[] = {8,-30,36,2,-6,52,8,-1,-11,10,4};
```

Brute Force Çıktısı:

```
baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main  
max sum is -> 94, left -> 2 right -> 10
```

Divide And Conquer:

```
baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main  
max sum is -> 94, left -> 2 right -> 10
```

Girdi:

```
// en buyuk alt dizi son eleman  
int arr[] = {-3,-2,-5,-3,-10,-1,5};
```

Brute Force Çıktısı:

```
baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main  
max sum is -> 5, left -> 6 right -> 6
```

Divide And Conquer:

```
● baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main  
max sum is -> 5, left -> 6 right -> 6
```

Girdi:

```
// en buyuk alt dizi sag kisimda (5,7)  
int arr[] = {-2,-3,4,-5,-2,1,5,3};
```

Brute Force Çıktısı:

```
● baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main  
max sum is -> 9, left -> 5 right -> 7
```

Divide And Conquer:

```
● baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev  
max sum is -> 9, left -> 5 right -> 7
```

Girdi:

```
// en buyuk alt dizi sol kisimda (1,2) veya(1,3) cunku sifir var  
int arr[] = {-1,3,2,0,-4,1};
```

Brute Force Çıktısı:

```
● baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main  
max sum is -> 5, left -> 1 right -> 3
```

Divide And Conquer:

```
● baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev  
max sum is -> 5, left -> 1 right -> 3
```

Girdi:

```
// en büyük alt dizi ortada (2,6)
int arr[] = {-2,-3,4,-1,-2,1,5,-3};
```

Brute Force Çıktısı:

```
baselkelziye@Base ~/Desktop/3-1/algo
max sum is -> 7, left -> 2 right -> 6
```

Divide And Conquer:

```
baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main
max sum is -> 7, left -> 2 right -> 6
```

Girdi:

```
//en büyük ilk eleman
int arr[] = {5,-2,-1,-6,-35,-39};
```

Brute Force Çıktısı:

```
baselkelziye@Base ~/Desktop/3-1/algo analizi/2.odev ./main
max sum is -> 5, left -> 0 right -> 0
```

Divide And Conquer:

```
baselkelziye@Base ~/Desktop/3-1/algo analizi/2.
max sum is -> 5, left -> 0 right -> 0
```


Brute Force sözde kod & analizi:

```

Max_subset_brute_force(arr[0...n-1])
    max ← arr[0]
    for i ← 0 to n-1 do
        cumulative_sum ← 0
        for j ← i+1 to n-1 do
            cumulative_sum ← cumulative_sum + arr[j]
            if max ≤ cumulative_sum
                max ← cumulative_sum
                left ← i
                right ← j
    return max, left, right

```

$\sum_{i=0}^{n-1} 1 \sum_{j=i+1}^{n-1} 1 + 1 \rightarrow \sum_{i=0}^{n-1} 1 \sum_{j=i+1}^{n-1} 2 \rightarrow \sum_{i=0}^{n-1} (1 + 2n) \rightarrow n + 2n(n-1)$
 $\rightarrow \sum_{i=0}^{n-1} 1 + \sum_{i=0}^{n-1} 2n \rightarrow n + n^2 - n = n^2 \in \Theta(n^2)$
 $\frac{n(n-1)}{2}$

Divide and Conquer sözde kod & analizi:

max-subset-div-conquer(arr[0..n-1], len, left, right)

if (len == 1)
return (A[0])

mid \leftarrow len/2

left-sum $\leftarrow -\infty$

right-sum $\leftarrow -\infty$

sum $\leftarrow 0$

(left-max-subset, left, right) \leftarrow max-subset-div-conquer(arr[0..n-1], mid, left, right)

(right-max-subset, left, right) \leftarrow max-subset-div-conquer(arr[mid..n-1], len-mid, left, right)

for i \leftarrow mid to len-1 do

sum \leftarrow sum + arr[i]

if right-sum < sum

right-sum \leftarrow sum

right \leftarrow i

sum $\leftarrow 0$

for i \leftarrow mid-1 to 0 do

sum \leftarrow sum + arr[i]

if (left-sum < sum)

left-sum \leftarrow sum

left \leftarrow i

if (left-max-subset > right-max-subset)

max-sum \leftarrow left-max-subset

else

max-sum \leftarrow right-max-subset

if (max-sum > left-sum + right-sum)

return (max-sum, left, right)

else

return (left-sum + right-sum, left, right)

Analiz:

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

$\boxed{a=2}$ her sefer 2 defa çağırıyoruz.

$\boxed{b=2}$ her sefer 2 defa bölüyoruz.

$f(n) \in O(n)$ çünkü dizileri toparlarken
1 for ile cross-sum'ı buluyoruz.

$n^d = n$ olduğundan $\boxed{d=1}$

$$a = b^d \rightarrow 2 = 2^1 \checkmark$$

$$T(n) \in O(n^d \log n) \rightarrow \boxed{O(n \log n)}$$

Video linki :

<https://www.youtube.com/watch?v=IhyzRm6UjQ4>