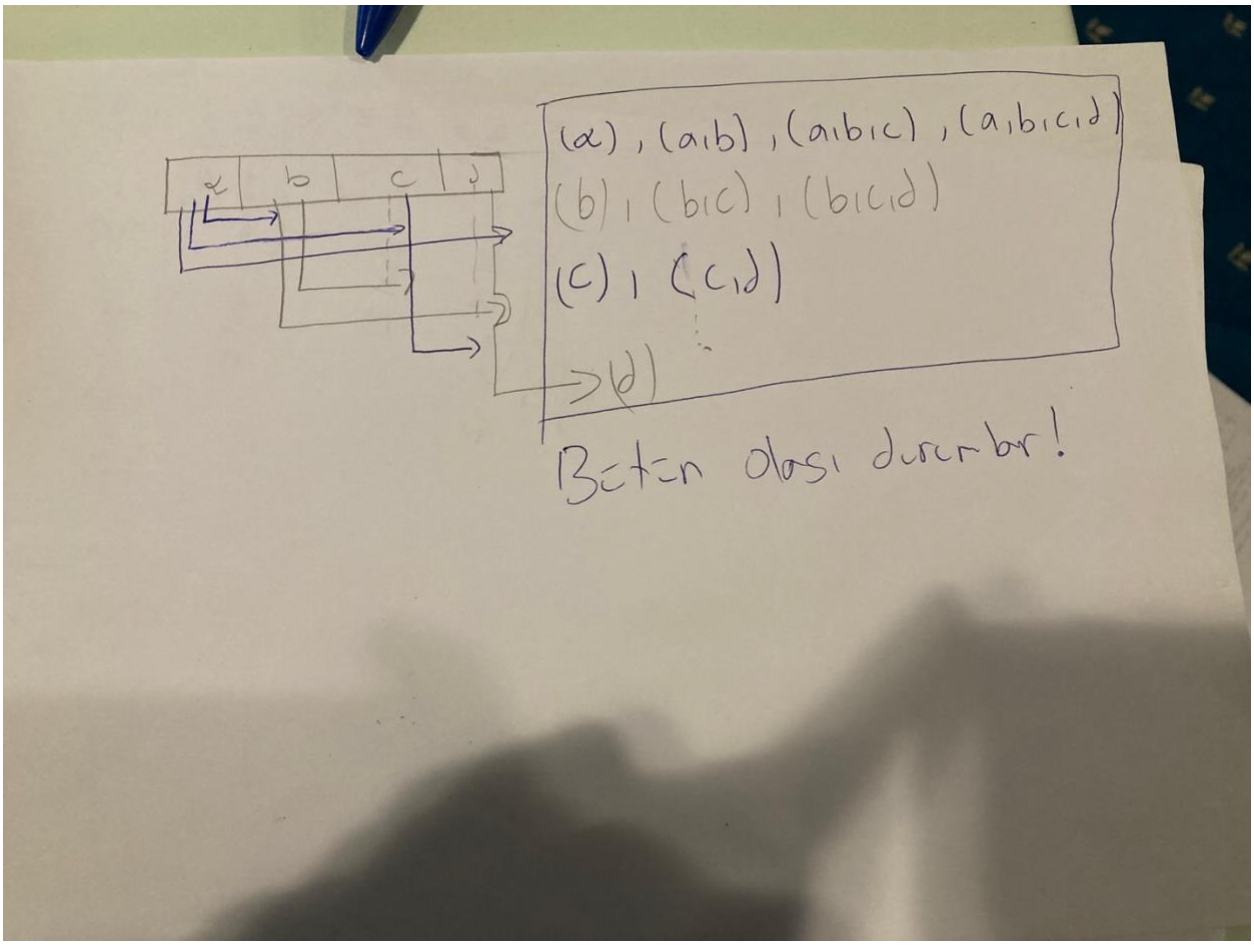
ALGORİTMA ANALİZİ GRUP 2

2. Ödev

Basel Kelziye 20011906

YÖNTEM

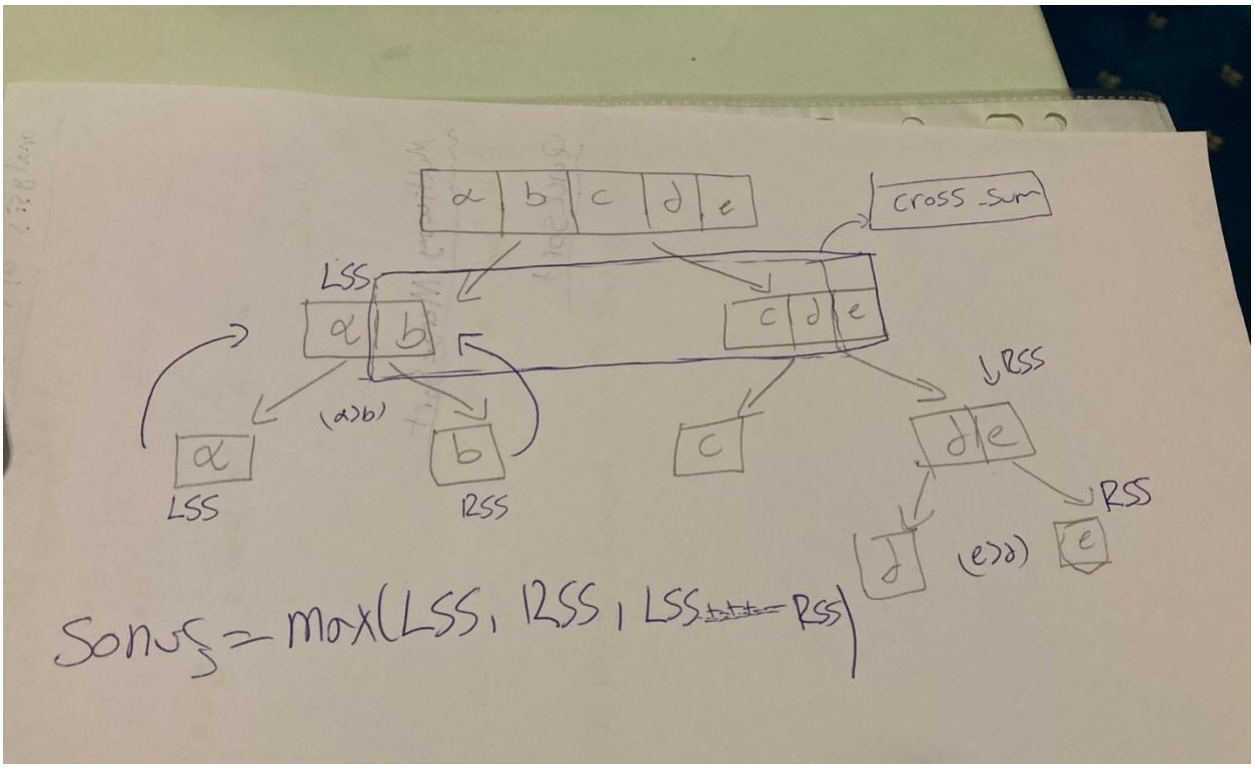
1-) Brute Force:



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 ı ilk eleman kabul edip oluşturulan al dizilerin toplamını her adımda kontrol eder eğer daha büyük ise max ve sol indisi (dış döngünün iteratörü) ı 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 buyuk 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 buyuk ilk eleman
    int arr[] = {5,-2,-1,-6,-35,-39};
```

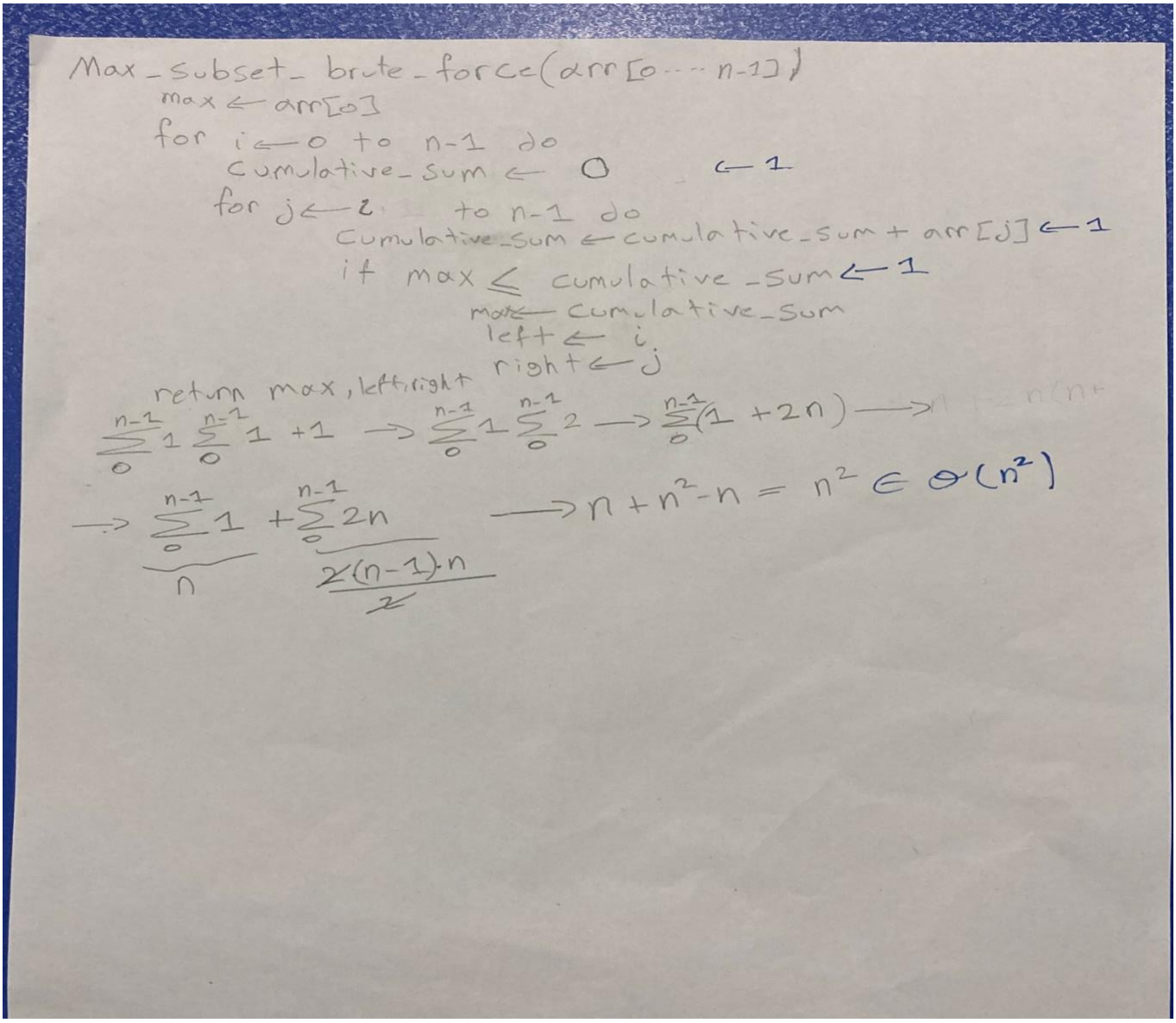
Brute Force Çıktısı:

```
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:



Divide and Conquer sözde kod & analizi:

```
max_subset_div_conquer(arr[0...n-1], len, left, right)
    if (len=1)
        return (A[0])

    mid ← len/2
    left_sum ← -∞
    right_sum ← -∞
    Sum ← 0
(left_max_subset, left, right) ← max_subset_div_conquer(arr[0...n-1], mid, left, right)
(right_max_subset, left, right) ← max_subset_div_conquer(arr[mid...n-1], len-mid, left, right)

    for i ← mid to len-1 do
        Sum ← Sum + arr[i]
        if right_sum < Sum
            right_sum ← Sum
            right ← i

    Sum ← 0
    for i ← mid-1 to 0 do
        Sum ← sum + arr[i]
        if (left_sum < Sum)
            left_sum ← Sum
            left ← i
    if (left_max_subset > right_max_subset)
        max_sum ← left_max_subset
    else
        max_sum ← right_max_subset
    if (max_sum > left_sum + right_sum)
        return (max_sum, left, right)
    else
        return (left_sum + right, left, right)
```

Analiz:

$T(n) = \alpha T(n/b) + f(n)$

$\alpha = 2$ her sefer 2 defa çoğırıyoruz.

$b = 2$ her sefer 2 defa bölüyoruz.

$f(n) \in \Theta(n)$ çünkü dizileri toplarken 1 for ile cross-sum'ı buluruz.

$n^d = n$ olduğundan $d = 1$

$\alpha = b^d \rightarrow 2 = 2^1$ ✓

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

Video linki :

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