

**TÜRKİYE CUMHURİYETİ**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**ALGORİTMA ANALİZİ İKİNCİ ÖDEV RAPORU**

**Öğrenci No: 19011038**  
**Öğrenci Adı Soyadı: Şevval Bulburu**  
**Öğrenci E-Posta: sevval.bulburu@std.yildiz.edu.tr**

**Ders/Grup: BLM3021 Algoritma Analizi/ 1.Grup**

**Ders Yürütücüsü**  
**PROF DR MİNE ELİF KARSLIGİL**  
**6 Kasım 2022**

## VIDEO LİNKİ:

[https://drive.google.com/drive/folders/10z\\_QdHyvyAw1vz2kewBr8gwgeCNHVGe4?usp=sharing](https://drive.google.com/drive/folders/10z_QdHyvyAw1vz2kewBr8gwgeCNHVGe4?usp=sharing)

## YÖNTEM:

Verilen problemde bir dizideki sıralı elemanlar toplamı en büyük olan alt dizinin bulunması istenmiştir.

Brute force yaklaşımında iç içe iki for döngüsü ile her bir eleman için kendinden sonraki elemanlar ile en büyük toplam kazanç değeri hesaplanarak bulunmuştur.

## BRUTE-FORCE PSEUDO KOD:

BF ( array, start\_index, stop\_index,)

```
    result[3]; // baslangic ,bitis indeksleri ve toplam kazanci tutar.
```

```
    min_gain = -10000;
```

```
    For i <- 0 to i <- n // n: # of elements
```

```
        gain = 0;
```

```
        For j <- i to j <- n {
```

```
            gain += array[j];
```

```
            if( gain > min_gain )
```

```
                min_gain <- gain
```

```
                start_index = i;
```

```
                stop_index = j;
```

```
        }
```

```
    }
```

```
    return result // result dizisi gain,start_index ve stop_index değerlerini sırası ile içerir
```

```
}
```

Divide and Conquer yaklaşımında dizi her seferinde ortadaki elemana göre parçalanmıştır. Parçalanmış dizideki maksimum aralığı bulmak için findMaxWithMiddle() fonksiyonu yazılmıştır. Fonksiyonun yaptığı işlem dizinin sağ ve sol tarafında ortadaki elemanı ve dizinin en fazla kazancını içeren alt dizileri bulmaktır. Sağ ve sol dizilerden elde edilen alt dizilerle ortak bir dizi elde edilmiştir. Daha sonra parçalanmış dizinin sağından, solundan ve iki tarafı da kapsayan diziden elde edilen sonuçlar kıyaslanarak en büyük aralık bulunmuştur.

### **DIVIDE AND CONQUER PSEUDO KOD:**

```
findMaxWithMiddle (array, left, right) {  
    result[3]; // baslangic ,bitis indeksleri ve toplam kazanci tutar.  
  
    left_gain = 0;  
    min_gain = -1000;  
    For j <- middle to j <- left {  
        left_gain += array[j]  
        if( left_gain > min_gain )  
            min_gain <- left_gain  
            start_index <- j;  
    }  
    right_gain <- 0;  
    min_gain <- -1000;  
    For j <- middle to j <- right {  
        right_gain += array[j];  
        if( right_gain > min_gain ){  
            min_gain <- right_gain  
            stop_index <- j;  
        }  
    }  
}
```

```

    if (right_gain > left_gain && sum_gain) {
        gain <- right_gain;
        start_index <- mid;
    }
    else if (left_gain > right_gain && sum_gain){
        gain <- left_gain;
        stop_index <- mid;
    }
    else {
        gain <- sum_gain
    }
    return result; //result dizisi gain,start_index ve stop_index değerlerini sırası ile içerir.
}

```

```

FindMaxWDiv (array, left, right) {
    if ( right - left < 2)
        return findMaxGain(array,left,right)

    left_gain = FindMaxWDiv (array, left, (right - left) / 2) + left))
    right_gain = FindMaxWDiv (array, ((right - left) / 2) + left) + 1, right)
    sum_gain = findMaxWithMiddle (array, left_gain->start_index, right_gain-> stop_index)
    if (left_gain-> gain >= right_gain-> gain && sum_gain-> gain)
        return left_gain;
    else if (right_gain-> gain >= left_gain-> gain && sum_gain-> gain)
        return right_gain;
    else
        return sum_gain;
}

```

Yapılan matematiksel analizler sonucu karmaşıklıklar aşağıdaki gibi bulunmuştur:

Brute Force Yaklaşım Matematiksel Analiz:

$$C(n) = \sum_{i=0}^n \sum_{j=i}^n 5 = 5 \cdot \frac{n \cdot (n+1)}{2} \sim n^2$$

Her durumda iki for döngüsüne de girerek

Kontrolleri yapar. Karmaşıklık  $n^2$  olur.  $\rightarrow O(n^2), \Theta(n^2), \Omega(n^2)$  ile gösterilebilir.

Divide and Conquer Yaklaşım ile Matematiksel Analiz:

$$T(n) = \underbrace{2 T\left(\frac{n}{2}\right)}_{\text{2. result function}} + \underbrace{n}_{\text{find max function}}$$

Master Theorem:

$$a=2 \quad b=2 \quad f(n)=n$$

$$a=2^1 \rightarrow T(n) = n^1 \log_2 n //$$

## UYGULAMA:

Girilen dizi: {-30, 36, 2, -6, 52, 8, -1, -11, 10, 4}

Eleman sayısı: 10

Beklenen sonuç: 1-9 indeksleri arasında olmalı.

Maksimum kazanç 94 olmalı.

```
C:\Users\sewva\OneDrive\Masaüstü\HW2_19011038\HW2_19011038.exe
DIZI: -30 36 2 -6 52 8 -1 -11 10 4
Rekursif Cozum:

Baslangic indeksi:1
Bitis indeksi:9
Maximum Kazanc:94

Brute Force Cozum:

Baslangic indeksi:1
Bitis indeksi:9
Maximum Kazanc:94
```

Girilen dizi: {-30, -40, 2, -6, 12, 8, -1, -11, -10}

Eleman sayısı: 9

Beklenen sonuç: 4-5 indekleri arasında olmalı.

Maksimum kazanç 20 olmalı.

```
C:\Users\sewa\OneDrive\Masaüstü\HW2_19011038\HW2_19011038.exe
DIZI: -30 -40 2 -6 12 8 -1 -11 -10
Rekursif Cozum:

Baslangic indeksi:4
Bitis indeksi:5
Maximum Kazanc:20

Brute Force Cozum:

Baslangic indeksi:4
Bitis indeksi:5
Maximum Kazanc:20
```

Girilen dizi: {-15, -39, -12, -6, -12, -8, -1, -11, -10, 0, -5, -3}

Eleman sayısı: 12

Beklenen sonuç: 9 indeksi başlangıç ve bitiş olmalı.

Maksimum kazanç 0 olmalı.

```
C:\Users\sewa\OneDrive\Masaüstü\HW2_19011038\HW2_19011038.exe
DIZI: -15 -39 -12 -6 -12 -8 -1 -11 -10 0 -5 -3
Rekursif Cozum:

Baslangic indeksi:9
Bitis indeksi:9
Maximum Kazanc:0

Brute Force Cozum:

Baslangic indeksi:9
Bitis indeksi:9
Maximum Kazanc:0
```