



Algoritma Analizi Ödevi

Muhammet Kayra Bulut

20011901

Dersin Yürütücüsü

Prof.Dr. Mine Elif KARSLIGİL

YÖNTEM

Öncelikle problemi tanımlayacak olursak “N adet tam sayıdan (negatif olabilir) oluşan bir dizide arka arkaya gelen sayılar içerisinde en büyük toplam değeri oluşturan kesintisiz sırayı bulmamız beklenmektedir.” Probleme 2 farklı yaklaşımla yaklaşmamız isteniyor.

Birincisi Brute-Force. Bu yaklaşımda dizinin teker teker tüm komşuluk ilişkilerini kıyaslayarak maksimumu buldum. Buradaki karmaşıklık n^2 oluyordu. Çünkü her türlü komşuluklara bakmam gerekiyordu.

İkinci çözüm olarak Divide-and-Conquer yaklaşımı isteniyordu. Bu yaklaşımla çözmek için diziyi ikiye böle böle gitmenin daha mantıklı olduğunu düşündüm. Bu şekilde önce dizinin ilk yarısının ortadan geriye doğru toplamalarının en büyüğü, sonra diğer yarısının ortadan ileriye doğru toplamalarının en büyüğü bulma şeklinde ilerledim. Bu yaklaşımı sürekli böle böle ilerlettim.

Böylece logaritma iki tabanında N defa çağırma ile, her defasında N kez kıyaslama yaparak çözüme ulaşmış oldum. Burada beni en çok indisleri tutmak zorladı.

UYGULAMA

Dizimin içeriği:

```
int dizi[] = {-2,31, -30,31,3,-1};
```

Aldığım Sonuç:

```
Brute Force Sonucu
En Yuksek Deger ---> 35
Aralik -----> 1 ==== 4
Divide And Conquer Sonucu
En Yuksek Deger ---> 35
Aralik -----> 1 ==== 4
```

Burada dizimin ardışıl en yüksek toplama sahip indisleri bir ile dördüncü indis arası oluyor. Toplamlarına baktığımızda $31-30+31+3 = 35$ sonucuna ulaşıyorum.

Dizimin içeriği:

```
int dizi[] = {1,3,5,7,9,11,13,-1,13};
```

Aldığım Sonuç:

```
Brute Force Sonucu  
En Yuksek Deger ---> 61  
Aralik -----> 0 ==== 8  
Divide And Conquer Sonucu  
En Yuksek Deger ---> 61  
Aralik -----> 0 ==== 8
```

Burada dizimin ardışıl en yüksek toplama sahip indisleri sıfır ile sekizinci indis arası oluyor. Toplamlarına baktığımızda $1+3+5+7+9+11+13-1+13 = 61$ sonucuna ulaşıyorum.

Dizimin içeriği:

```
int dizi[] = {-1,-2,-3,-4,-2,-3,-5,-7};
```

Aldığım Sonuç:

```
Brute Force Sonucu  
En Yuksek Deger ---> -1  
Aralik -----> 0 ==== 0  
Divide And Conquer Sonucu  
En Yuksek Deger ---> -1  
Aralik -----> 0 ==== 0
```

Burada dizimin ardışıl en yüksek toplama sahip indisleri sıfır ile sıfırıncı indis arası oluyor. Toplamlarına baktığımızda $-1 = -1$ sonucuna ulaşıyorum. Burada denediğim durum uç bir durum ve sıfır dışındaki tüm sayılar negatif. Algoritmam bu uç durumda da çalışabiliyor.

Dizimin içeriği:

```
int dizi[] = {1,2,3,4,1,2,3,4,1};
```

Aldığım Sonuç:

```
Brute Force Sonucu  
En Yuksek Deger ---> 21  
Aralik -----> 0 ==== 8  
Divide And Conquer Sonucu  
En Yuksek Deger ---> 21  
Aralik -----> 0 ==== 8
```

Burada dizimin ardışıl en yüksek toplama sahip indisleri sıfır ile sekizinci indis arası oluyor. Toplamlarına baktığımızda $1+2+3+4+1+2+3+4+1 = 21$ sonucuna ulaşıyorum. Burada denediğim durum uç bir durum ve hiçbir negatif sayı yok. Algoritmam bu uç durumda da çalışabiliyor.

DIVIDE AND CONQUER ANALİZİ

Ödev 2

DAC(A[0...n-1], low, high)

if high \leq low then
return A[low]

mid = (low + high) / 2

sum = 0; left_max = min(A);
for i = mid; i \geq low; i--

sum = sum + A[i];
if (sum > left_max)
left_max = sum;

right_max = min(A); sum = 0;

for i = mid + 1; i \leq high; i++

sum = sum + A[i];
if (sum > right_max)
right_max = sum;

left-right = max(DAC(A[i], low, mid), DAC(A[i], mid+1, high))

return max(left-right, left_max + right_max)

DIVIDE
AND
CONQUER

$$T(n) = 2T(n/2) + N \rightarrow \begin{matrix} d=1 \\ a=2 \\ b=2 \end{matrix} \quad 2 = 2^1 \rightarrow n^d \log n \rightarrow n \log n$$

Aer durumda tüm elemanları döşmek zorunda olduğ için

$$C_{best} = \Theta(n \log n) \quad C_{average} = \Theta(n \log n) \quad C_{worst} = \Theta(n \log n)$$

BRUTE FORCE ANALİZİ

DF(N, A[N])

max = A[0]

for i=0; i < N; i=i+1

tmp=0

for j=i; j < N; j=j+1

tmp = tmp + A[j]

if (tmp > max)

max = tmp

left = i;

right = j;

return max

BRUTE
FORCE

N defa dış döngü

N-i defa iç döngü

$$N + (N-1) + (N-2) + \dots + 1 \rightarrow \frac{N * (N+1)}{2} \rightarrow N^2$$

Her durumda tüm elemanları kontrol edeceği için

$$C_{best} = \Theta(n^2) \quad C_{average} = \Theta(n^2) \quad C_{worst} = \Theta(n^2)$$