

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



Algoritma Analizi 2. Ödev Raporu

Öğrenci No: 21011084

Ad-Soyad: Berkay Gümüşay

E-Posta: berkay.gumusay@std.yildiz.edu.tr

Telefon: 0 535 840 79 78

Video Linki: https://www.youtube.com/watch?v=gx2k5IIAN_A

Rapor İçerikleri

1. Soru

- a. Problemin Tanımı.....3
- b. Problemin Çözümü.....3
- c. 100 Kişi İçin Çözümün Gösterimi.....4
- d. Rekürans Bağıntısı ve Karmaşıklık Hesabı.....5

2. Soru

- a. Rekürans Bağıntısı ve Rekürans Bağıntısının Çözümü.....6
- b. Sözde Kod İle Karmaşıklık Analizi.....8
- c. Örnek Kod Çıktıları9

1.a) Problemin Tanımı

Bu ödevde N adet kişi içerisindeki doğruyu söyleyen en az 1 kişiyi (X,Y) şeklinde ikili sorgular yaparak bulmamız istenmektedir. Doğruyu söyleyen kişilerin sayısı yalan söyleyenlerden fazladır. Bizden N=100 olacak şekilde bu problemi 100 kişi için çözmemizi istemektedir.

1.b) Problemin Çözümü

Böl ve yönet yaklaşımı kullanarak kişiler ikili gruplara bölünür (bu örnekte 50 grup). Daha sonrasında sorgu işlemi başlar. Eğer iki kişi de karşıdakinin doğru söylediğini söylüyorsa bu ikili tutulur. Eğer ki bir kişi bile karşıdakinin yalan söylediğini söylüyorsa bu ikili elenir. Sonrasında kalan ikililer kendi içlerinde T. sıradaki eleman ($T+N/2$). sıradaki eleman ile eşleşecek şekilde tekrardan sorgulama işlemi yapılır (N = 0 adımda dizide bulunan kişi sayısı). Bu işlem 2 adım boyunca eleman elenmeyene kadar devam eder. 2 Adım boyunca eleman elenmediyse geriye Doğruyu Söyleyen elemanlar kalmış demektir.

100 Kişi için 1 Kişiyi Kaç Adımda Bulabiliriz? :

Algoritmanın soruyu kaç adımda çözeceği doğruyu söyleyen ve yalan söyleyen kişilerin sıralanma şekline bağlıdır. Fakat **Average Case**'i baz alırsak her adımda kişi sayısı yarıya ineceği için **ortalama olarak $\log_2 N$** adımda bu algoritma sonuca ulaşacaktır.

1- 100 Kişi için bu sorunun cevabı $\log_2 100 \approx 6,64 \approx 7$ **gruplandırma işlemi** ile sonuca ulaşılır.

2- Eğer bu gruplandırma işlemlerindeki sorgu sayısını da hesaba katarsak $O(N)$ karmaşıklıkta yani Average Case'de 100 adımda çözülecektir.

Sorgu sayısı için cevap = $O(N) \approx 100$ adım

Gruplandırma işlemi sayısı için cevap = $\log_2 N \approx 7$

1.c) 100 Kişi İçin Çözümün Gösterimi

Mevcut kişi listesi: (Kırmızı ile belirtilen kişi doğruyu söyleyen kişidir ve sayıları yalan söyleyen kişi sayısından fazladır)

P1, P2, P3, P4 **P49, P50, P51, P52** **P99, P100**

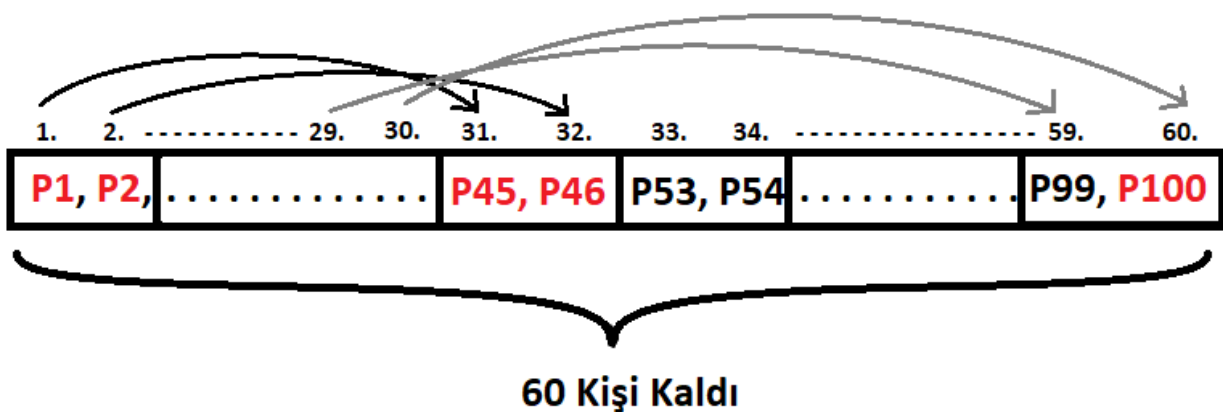
Adım 1) Listeye 'Merge' işlemi uygulanır:

P1, P2, P3, P4 **P49, P50, P51, P52** **P99, P100**

Adım 2) "1.b) Problemin Çözümü kısmında anlatıldığı gibi eleme işlemleri yapılır:

P1, P2, **P45, P46** **P53, P54** **P99, P100**

Adım 3) Kalan ikililer birleştirilir ve T. Eleman ($T+N/2$). Eleman ile ikili grup oluşturarak tekrar sorgulama ardından eleme işlemleri yapılır. **Geriye 60 kişi kalmış şekilde varsayımda bulunulursa 1. Eleman 31. Elemanla eşleşecek şekilde sorgulamalar yapılır.**

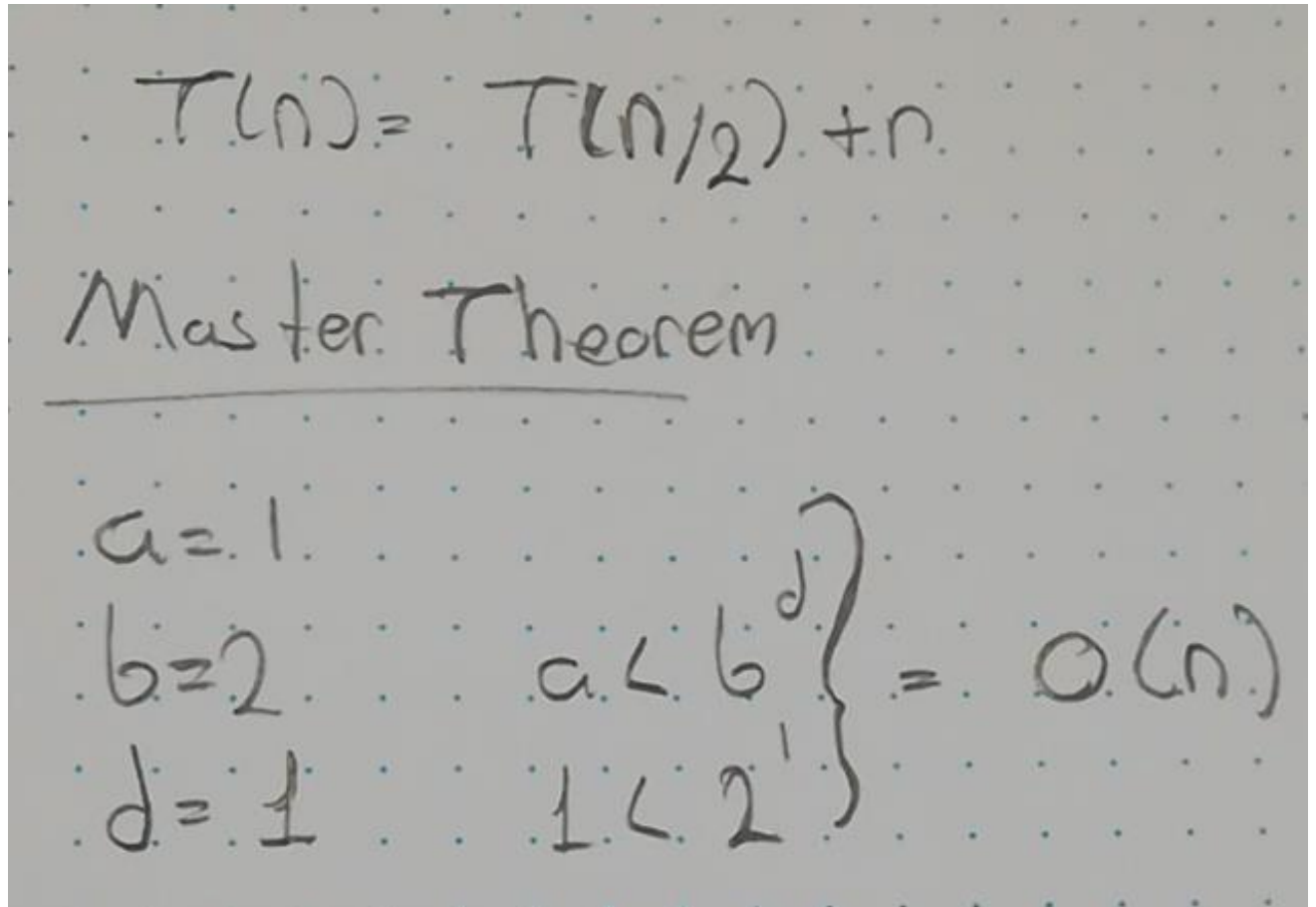


Adım 4) Adım 3 tekrarlanır. Eğer Adım 3 iki kez tekrarlanmasına rağmen eleme işlemi yapılmıyorsa algoritma durur. Kalan elemanlar doğruyu söyleyen kişilerdir.

1.d) Rekürans Bağıntısı ve Karmaşıklık Hesabı

Rekürans Bağıntısı: $T(n) = T(n/2) + n$

Karmaşıklık Hesabı:



$T(n) = T(n/2) + n$

Master Theorem

$a = 1$
 $b = 2$
 $d = 1$

$a < b^d$
 $1 < 2^1$

$\left. \begin{array}{l} a < b^d \\ 1 < 2^1 \end{array} \right\} = O(n)$

2.a) Rekürans Bağintısı ve Rekürans Bağintısı Çözümü

Çözüm Şekli: Farklı boyutlarda anahtarları ve kilitleri birbiriyle eşleştirmek için QuickSort ile anahtar ve kilitler karşılıklı kıyaslanarak $O(n \cdot \log n)$ karmaşıklık ile eşleştirme işlemi gerçekleştirilmiştir.

Worst Case:

Worst Case

$$T(n) = T(n-1) + n$$
$$T(n) = [T(n-2) + n - 1] + n$$
$$T(n) = [T(n-3) + n - 2] + n - 1 + n$$

↓

$$T(n) = T(1) + 2 + 3 + 4 + \dots + (n-1) + n$$
$$T(n) = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2} = O(n^2)$$

Best Case:

Best Case

$$T(n) = 2T(n/2) + n$$

↳ Partition Fonksiyonu

$$\left. \begin{array}{l} a=2 \\ b=2 \\ d=1 \end{array} \right\} n^d \log n = (n \log n)$$

Master Theorem

Average Case:

Average Case

$$C_{\text{AVG}}(n) = \frac{1}{n} \sum_{s=0}^{n-1} [(n+1) + C_{\text{AVG}}(s) + C_{\text{AVG}}(n-1-s)]$$

for $n \geq 1$

$$C_{\text{AVG}}(1) = \emptyset$$

$$C_{\text{AVG}} \approx 2 \ln n \approx 1.39 n \log n$$

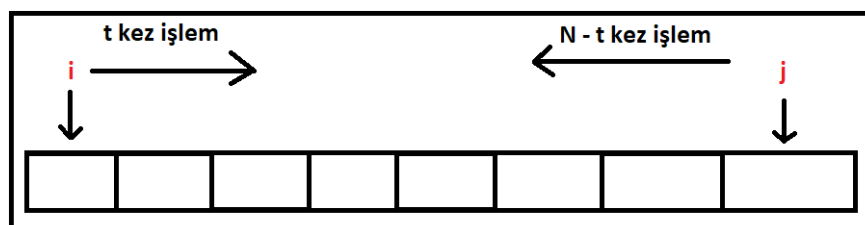
Quick Sort Avg. Case

2.b) Sözde Kod İle Karmaşıklık Analizi

partition() fonksiyonu :

Algoritmada bulunan en karmaşık yer partition() fonksiyonu olduğu için sözde kod analizi bu kısımda yapılmıştır.

```
partition(array[], low, high, pivot)
i ← low; j ← high;
arr ← array[i]
while (i ≤ j) Do:
    while (j ≥ 0 && arr[j] ≥ pivot) Do:
        if (arr[j] = pivot) THEN
            swap(arr[j], arr[high])
        ENDIF
        j ← j - 1
    ENDWHILE
    while (arr[i] < pivot) Do:
        i ← i + 1
    ENDWHILE
    swap(arr[i], arr[j])
ENDWHILE
swap(arr[i], arr[high])
return i
```



Yukarıdaki şekilde gösterildiği gibi **i** dizinin solundan **j** ise dizinin sağından başlar ve **j > i** olana kadar döngü devam eder. Yani bu fonksiyon **O(N)** karmaşıklığa sahiptir.

2.c) Örnek Kod Çıktıları

1. Örnek:

Karışık iki dizi.

```
Keys And Locks Lists :
```

```
Key : 7 37 5 9 21 16 3  
Lock: 5 16 7 21 3 9 37
```

```
Key-Lock Match !!
```

```
Key : 3 5 7 9 16 21 37  
Lock: 3 5 7 9 16 21 37
```

2. Örnek

Birisi sıralı diğeri tersten sıralı iki dizi.

```
Keys And Locks Lists :
```

```
Key : 2 4 6 8 10 12 14  
Lock: 14 12 10 8 6 4 2
```

```
Key-Lock Match !!
```

```
Key : 2 4 6 8 10 12 14  
Lock: 2 4 6 8 10 12 14
```

3. Örnek

İkisi de sıralı iki dizi.

```
Keys And Locks Lists :  
  
Key : 2 4 6 8 10 12 14  
Lock: 2 4 6 8 10 12 14  
-----  
  
Key-Lock Match !!  
  
Key : 2 4 6 8 10 12 14  
Lock: 2 4 6 8 10 12 14
```

4. Örnek

İkisi de tersten sıralı iki dizi.

```
Keys And Locks Lists :  
  
Key : 14 12 10 8 6 4 2  
Lock: 14 12 10 8 6 4 2  
-----  
  
Key-Lock Match !!  
  
Key : 2 4 6 8 10 12 14  
Lock: 2 4 6 8 10 12 14
```

5. Örnek

İki elemanlı iki dizi.

```
Keys And Locks Lists :  
  
Key : 39 12  
Lock: 12 39  
-----  
  
Key-Lock Match !!  
  
Key : 12 39  
Lock: 12 39
```

6. Örnek

On iki elemanlı iki karışık dizi.

```
Keys And Locks Lists :  
  
Key : 71 11 24 4 8 43 91 16 21 1 67 7  
Lock: 11 7 8 4 67 43 71 16 24 21 1 91  
-----  
  
Key-Lock Match !!  
  
Key : 1 4 7 8 11 16 21 24 43 67 71 91  
Lock: 1 4 7 8 11 16 21 24 43 67 71 91
```

Berkay GümüŖay

21011084

berkay.gumusay@std.yildiz.edu.tr