

(Vize Haftası) - 9. Hafta

Vize Soruları

1. LCD
2. Matris Çarpımı
3. Algoritma Analizi
4. Sıralama $O(n)$

???

PRİM ALGORİTMASI**Kapsama Algoritması (Min Kapsama Ağacı)'nın Adımları**

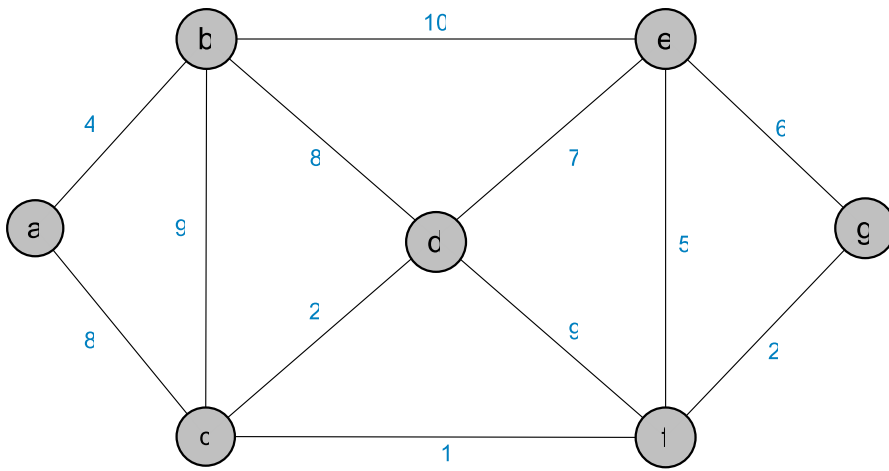
Ağırlıklı yönsüz graf

Adım 0 Herhangi bir eleman seçilir

$$S = \{r\} \quad A = \emptyset \quad r : \text{kök}$$

Adım 1 En kısa kenarı bulKısa kenarın bir ucu S 'de diğer ucu $V - S$ 'de

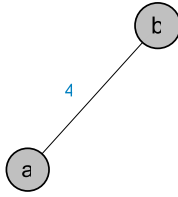
Bu kenarı A kümesine ekle

Adım 2 $\underbrace{V - S = \emptyset}_{V \text{ fark } S} \Rightarrow \text{dur}$ (S, A) ağacını yazdır**Örnek:** Min. Kapsama ağacını bulunuz?

$$S = \{a\} \quad V - S = \{b, c, d, e, f, g\}$$

Hangi düğümden
başladığı
farketmez

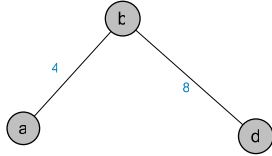
Burada {a}
seçilmiş



$$ek = (a, b)$$

$$A = \{(a, b)\}$$

$$S = \{(a, b)\} \quad V - S = \{c, d, e, f, g\}$$



$$ek = \{(a, c), (b, d)\}$$

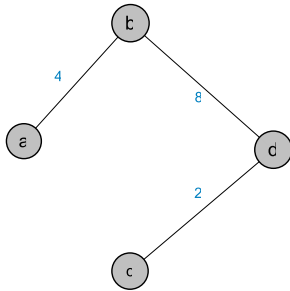
$$A = \{(a, b), (b, d)\}$$

$$S = \{(a, b, d)\} \quad V - S = \{c, e, f, g\}$$

$$(a, c) = 8$$

$$(b, d) = 8$$

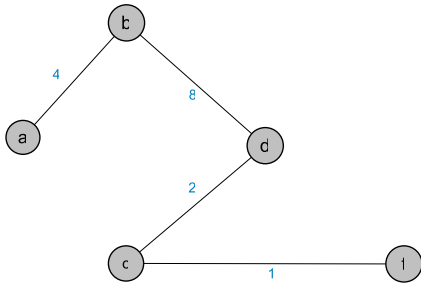
(b, d) yi aldık



$$ek = \{(d, c)\}$$

$$A = \{(a, b), (b, d), (d, c)\}$$

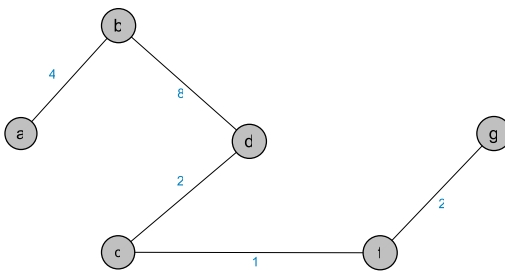
$$S = \{(a, b, d, c)\} \quad V - S = \{e, f, g\}$$



$$ek = \{(c, f)\}$$

$$A = \{(a, b), (b, d), (d, c), (c, f)\}$$

$$S = \{(a, b, d, c, f)\} \quad V - S = \{e, g\}$$



$$ek = \{(f, g)\}$$

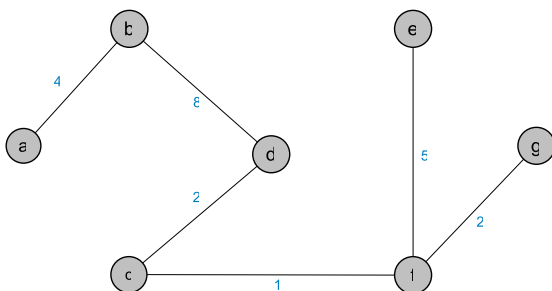
$$A = \{(a, b), (b, d), (d, c), (c, f), (f, g)\}$$

$$S = \{(a, b, d, c, f, g)\} \quad V - S = \{e\}$$

$$(f, e) = 5$$

$$(g, e) = 6$$

(f, e) seçildi
çünkü 5 < 6



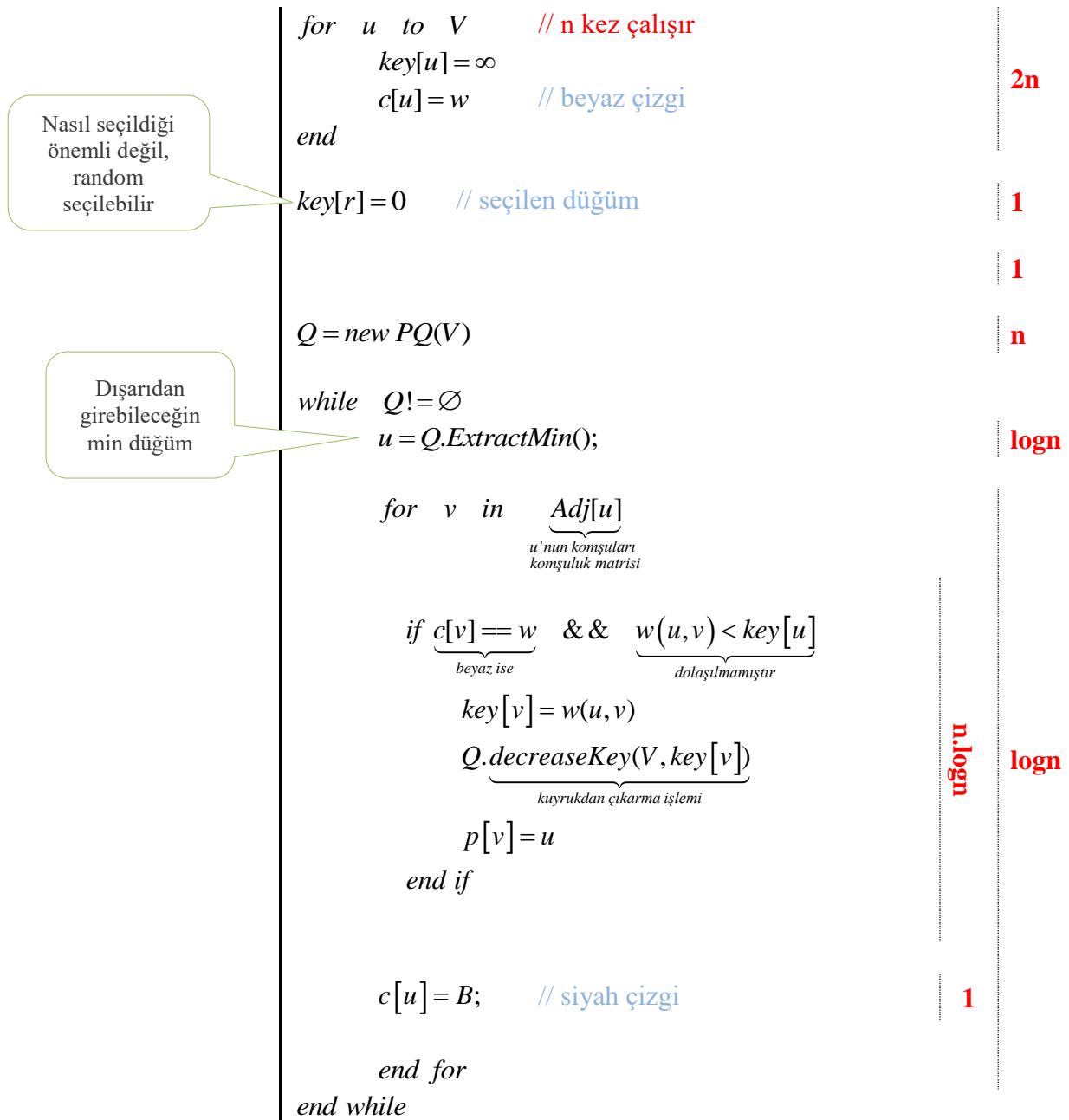
$$ek = \{(f, e)\}$$

$$A = \{(a, b), (b, d), (d, c), (c, f), (f, g), (f, e)\}$$

$$S = \{(a, b, d, c, f, g, e)\} \quad V - S = \emptyset$$

PRİM ALGORİTMASI

Kapsama Algoritması (Min Kapsama Ağacı) 'nın Algoritması



PRİM ALGORİTMASI**Kapsama Algoritması (Min Kapsama Ağacı) 'nın Analizi**

- ✓ En küçük kenar **kuyruk kullanılarak** bulunuyor. (Kullanılan yöntem performansı etkileyecektir)
- ✓ **ExtractMin()**; → Kuyruktan en küçük değere sahip elemanı çıkartıyor

- Kuyruktaki her bir elemanın çıkarılması $O(\log n)$
- Her kenar için n kere yapıldığından $O(n \cdot \log n)$
- **decreaseKey** kısmında $O(n \cdot \log n)$
- Her kenar için en az bir kere yap $O(e \cdot \log n)$

e edge
(kenar)

Toplam Maliyet

$$T(n) = O((e + n) \cdot \log n)$$

$$T(n, e) = 3n + 2 \sum_{u=1}^V [O(\log n) + O(\deg(u) \log n)]$$

$$T(n, e) = 3n + 2 O \left(\log n \sum_{u=1}^V [1 + \deg(u)] \right)$$

$$T(n, e) = 3n + 2 O \left(\log n \left(n + \underset{\text{ihmal edilebilir}}{2e} \right) \right)$$

$$T(n, e) = O[(V + E) \cdot \log n]$$

12.hafta
Quiz

KRUSKAL ALGORİTMASI / Greedy Yaklaşım**Amaç :** Min Kapsama ağacını bulmak (Yönsüz ve Ağırlıksız | Döngü İçermeyen)

Kapsama Algoritması (Min Kapsama Ağacı)'nın Adımları

```

T = ∅
for i = 1 to n
  if T ∪ {i}'de çevrim yoksa
    T'ye ekle
  end if
end for
return T

```

Detaylı Algoritma Analizi**Girdi** $G = (V, E)$ **Çıktı** G 'nin kapsamam ağacı T = Kenarlar kümesi

```

E'yi artan sırada sırala
for v in V
  KümeOluştur({v})
end for

```

```

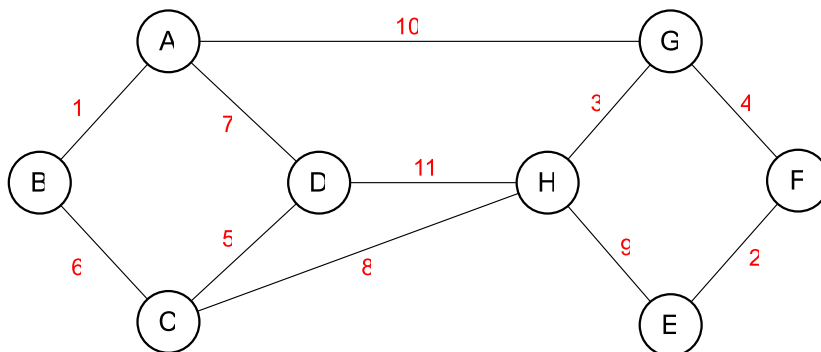
while T < n-1
  (x, y) E'nin bir sonraki kenarı
  if Bul(x) != Bul(y)
    Add(x, y) to T
    Birleşim(x, y)
  end if

```

n-1 adet çalışır
sürekli azalıyor

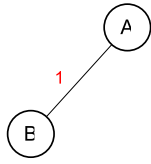
Farklı iki düğümse

T'ye ekle

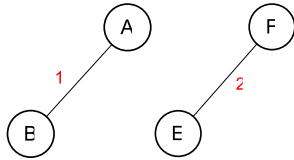
 $O(n \log n) + O(m \log n)$ $n > m$ m+m
2m
defa
çalışır**1.Örnek**

0 ve 1 çocuk sayısını gösteriyor

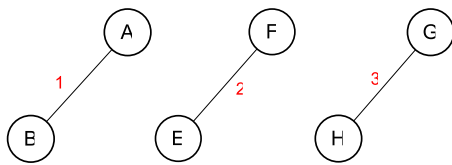
⁰A ⁰B ⁰C ⁰D ⁰E ⁰F ⁰G ⁰H



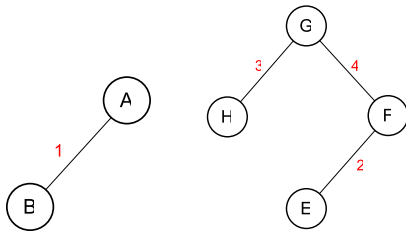
¹A ⁰B ⁰C ⁰D ⁰E ⁰F ⁰G ⁰H
A — B



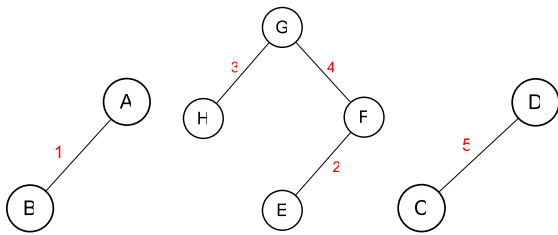
¹A ⁰B ⁰C ⁰D ¹E ⁰F ⁰G ⁰H
A — B
E — F



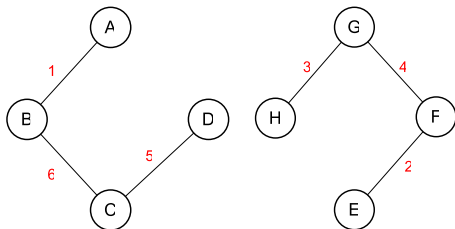
¹A ⁰B ⁰C ⁰D ¹E ⁰F ¹G ⁰H
A — B
E — F
G — H



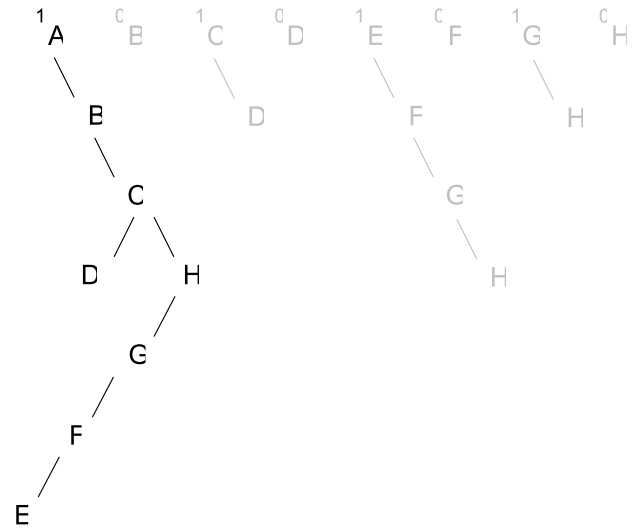
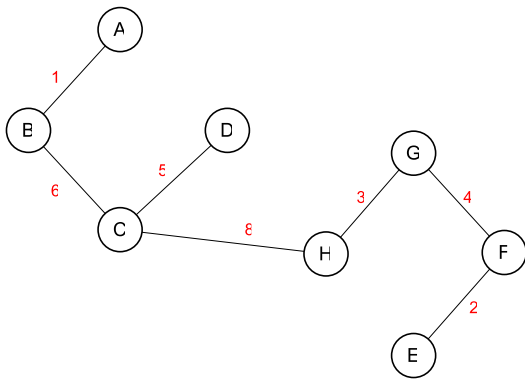
¹A ⁰B ⁰C ⁰D ¹E ⁰F ¹G ⁰H
A — B
E — F — G — H
G — H



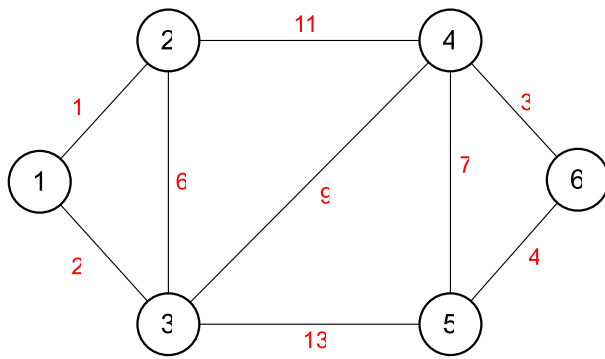
¹A ⁰B ¹C ⁰D ¹E ⁰F ¹G ⁰H
A — B
C — D
E — F — G — H
G — H



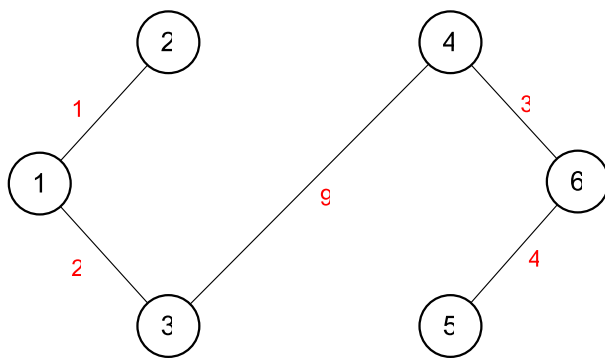
¹A ⁰B ¹C ⁰D ¹E ⁰F ¹G ⁰H
A — B — C — D
E — F — G — H
G — H



2.Örnek:



Çevrim oluşturmamasına dikkat edilerek en küçükten başlanarak çiziliyor.



HUFFMAN ALGORİTMASI

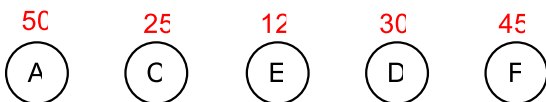
(Veri Sıkıştırma Algoritması)

Örnek:

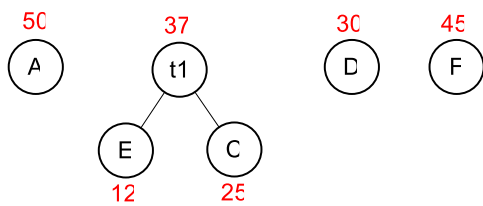
Bir dosya analiz edilmiş ve aşağıdaki sonuçlar çıkmıştır (Rakamlar tekrar sayısını yani frekansı)

A	50	x8bit
C	25	x8bit
E	12	x8bit
D	30	x8bit
F	45	x8bit

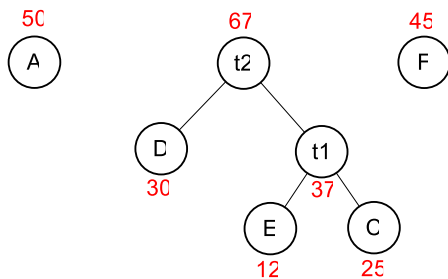
$$162 \times 8 = 1296 \text{bit}$$



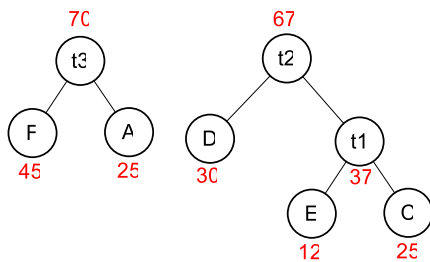
Frekansı küçük olan iki düğüm birleştirilir



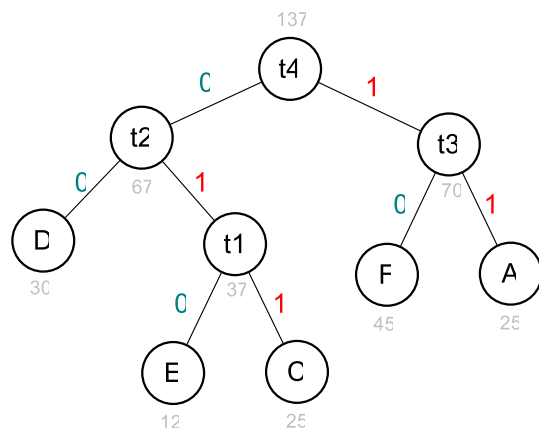
Küçük olan sola büyük olan sağa eklenir



Tüm düğümler bitinceye kadar bu işlemi tekrarlıyoruz



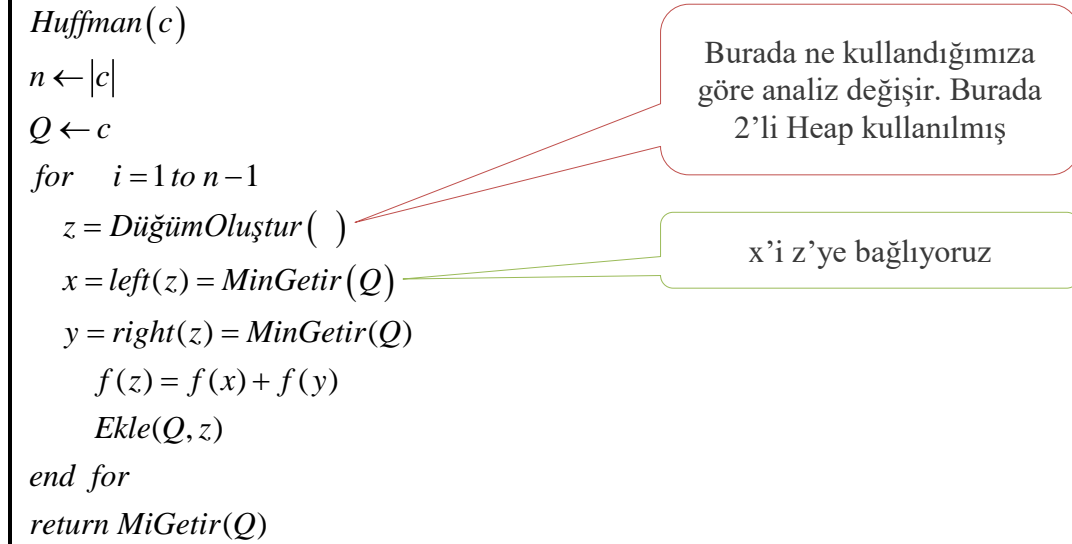
Bilgiler yapraklarda



Soldaki kollara 0, Sağdaki kollara 1 yazıyoruz

A	11	2bit	x50=100
C	110	3bit	x25=75
E	010	3bit	x12=36
D	00	2bit	x30=60
F	01	2bit	x45=90

Toplam : 361bit

HUFFMAN ALGORİTMASI Analizi**Zaman Kaybı :** Sıkıştırma ve açma işlemleri sırasında olacak**Not:** Az sayıda karakter dosya boyutunu küçültme yerine büyütür

İkili (Binary) Heap kullanıldığı için;

Heap Oluşturma $O(n)$ Çıkartma ve ekleme işlemi $O(\log n)$

$$T(n) = \sum_{i=1}^n \log i = O(\log(n!)) = O(n \log n)$$

$$T(n) = O(n \log n)$$

?