

Huffman Kodu

1

BIL 236 – PROGRAMLAMA LABORATUVARI
5. ÖDEV

Huffman Kodu Nedir?

2

- Huffman Kodu, veri sıkıştırma için kullanılan kayıpsız bir algoritmadır.
- David A. Huffman tarafından 1952 yılında bulunmuştur.
- Pek çok güncel sıkıştırma algoritması tarafından kullanılır. Örn: MP3, JPEG...

8 bit sınır mı?

3

- ASCII, 8 bit
- 29 farklı karakter 5 bit ile kodlanabilir.
- $5/8 = \%62,5$ sıkıştırma
- Sabit sayıda bit ile kodlamaya ihtiyacımız var mı?

Morse Kodu

4

A ● ■
B ■ ● ● ●
C ■ ● ■ ●
D ■ ● ●
E ●
F ● ● ■ ●
G ■ ■ ●
H ● ● ● ●
I ● ●
J ● ■ ■ ■
K ■ ● ■
L ● ■ ● ●
M ■ ■
N ■ ●
O ■ ■ ■
P ● ■ ■ ●
Q ■ ■ ● ■
R ● ■ ●
S ● ● ●
T ■

U ● ● ■
V ● ● ● ■
W ● ■ ■
X ■ ● ● ■
Y ■ ● ■ ■
Z ■ ■ ● ●

1 ● ■ ■ ■ ■
2 ● ● ■ ■ ■
3 ● ● ● ■ ■
4 ● ● ● ● ■
5 ● ● ● ● ●
6 ■ ● ● ● ●
7 ■ ■ ● ● ●
8 ■ ■ ■ ● ●
9 ■ ■ ■ ■ ●
0 ■ ■ ■ ■ ■

Kod kelimelerinin devamlılığı

5

- Sık kullanılan harfler için kısa kod kelimeleri kullanılmıştır.
- Fakat
 - Bağlama göre sıklıklar değişebilmektedir. Örn: Türkçe, İngilizce, ses kütüğü, görüntü kütüğü, vs...
 - Kod kelimeleri aralarında bekleme olmaksızın ayırd edilememektedir. Neden?

Morse Kodu'nda belirsizlik

6



Olası belirsizlikler

- ATT
- ETTT
- WT

A	• —
B	— • • •
C	— • — •
D	— • •
E	•
F	• • — •
G	— — •
H	• • • •
I	• •
J	• — — —
K	— • —
L	• — • •
M	— —
N	— •
O	— — —
P	• — — •
Q	— — • —
R	• — •
S	• • •
T	—

U	• • —
V	• • • —
W	• — —
X	— • • —
Y	— • — —
Z	— — • •

1	• — — —
2	• • — —
3	• • • —
4	• • • • —
5	• • • • •
6	— • • • •
7	— — • • •
8	— — — • •
9	— — — — •
0	— — — — —

Huffman'ın önerileri

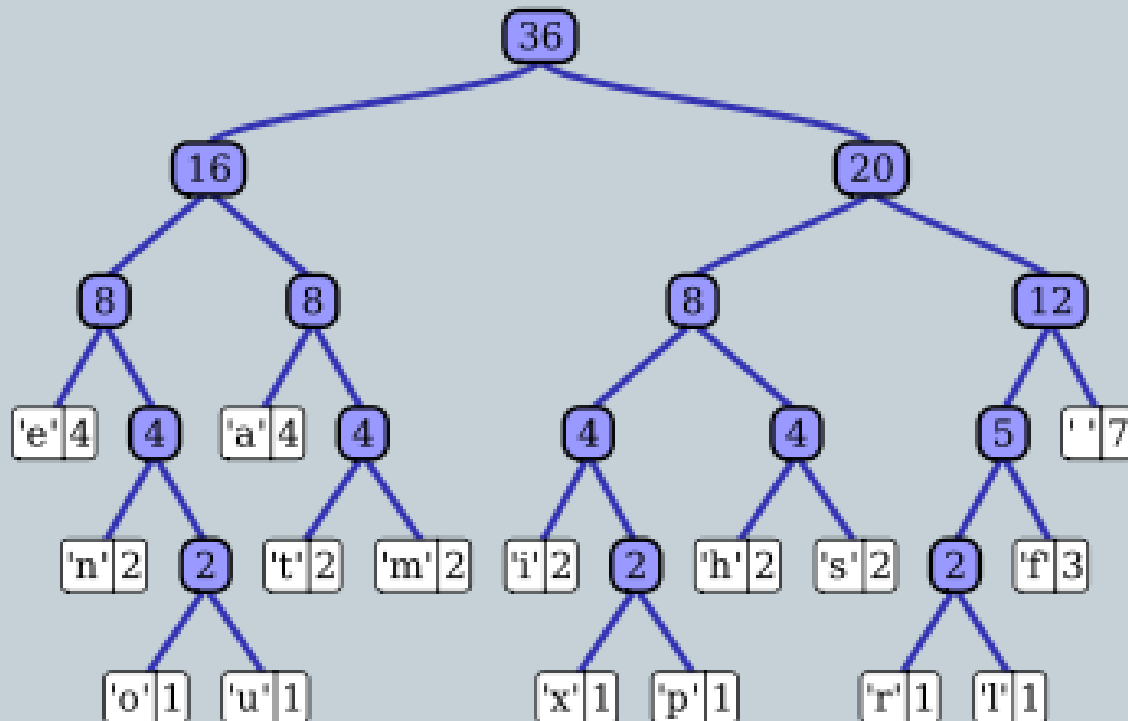
7

- Bağlama göre sıklık belirlenmesi
- Aralarında boşluk olmadan kullanılabilen kod kelimeleri (önek ağacı)
 - Örn: Telefon numaraları
 - ✦ +90 Türkiye
 - 312 Ankara
 - 297 Beytepe
 - 7500 Bilgisayar Mühendisliği Bölümü

Huffman Ağacı

8

- Bu yapıda dallar takip edildikçe kod sözcüğü oluşmaktadır.



Algoritma

9

- Karakterlerin sıklıkları bulunur ve bir öncelik kuyruđuna konulur.
- En düşük sıklıđa sahip iki düđüm yeni bir düđümün çocukları olmak üzere kuyruktan alınır.
- Yeni oluşturulan düđüm kuyruđa eklenir.
- Kuyrukta bir adet düđüm kalana kadar bu işleme devam edilir

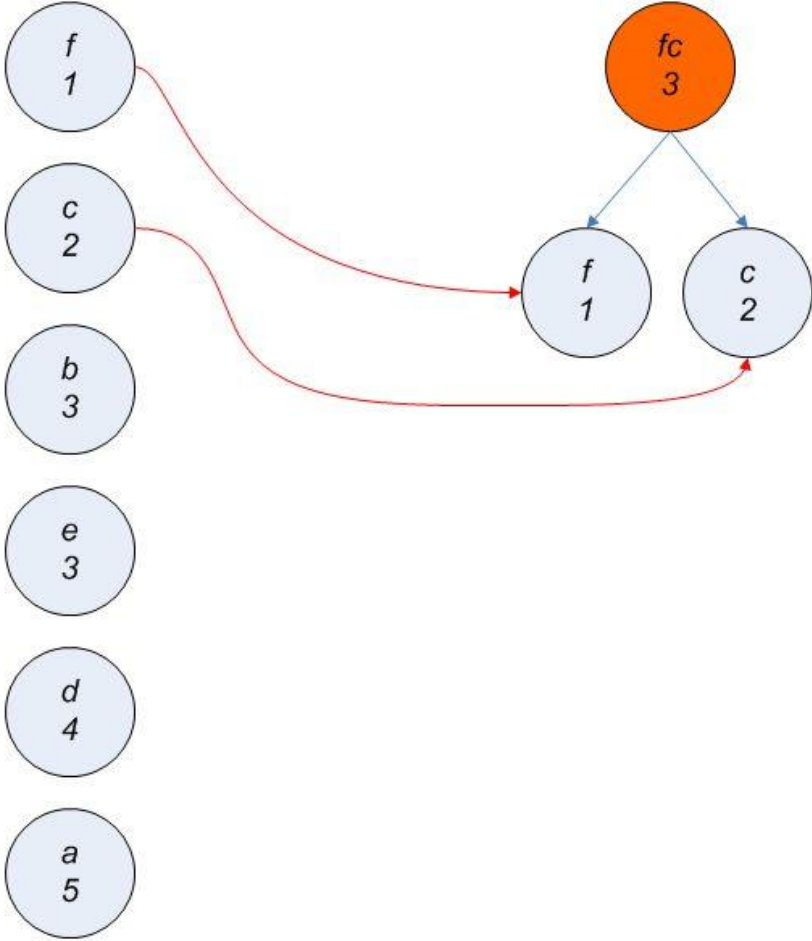
Örnek

10

- Elimizde sıkıştırılması istenen alttaki katarın olduğunu varsayalım;
 - aaaabbbccddddeefa
- Bu katardaki karakter sıklıkları hesaplandığında;
 - $f = 1, c = 2, b = 3, e = 3, d = 4, a = 5$
- ASCII olarak tutulduğunda bu katar toplam $18 * 8 = 144$ bayt yer kaplamaktadır.

Örnek – 1. Adım

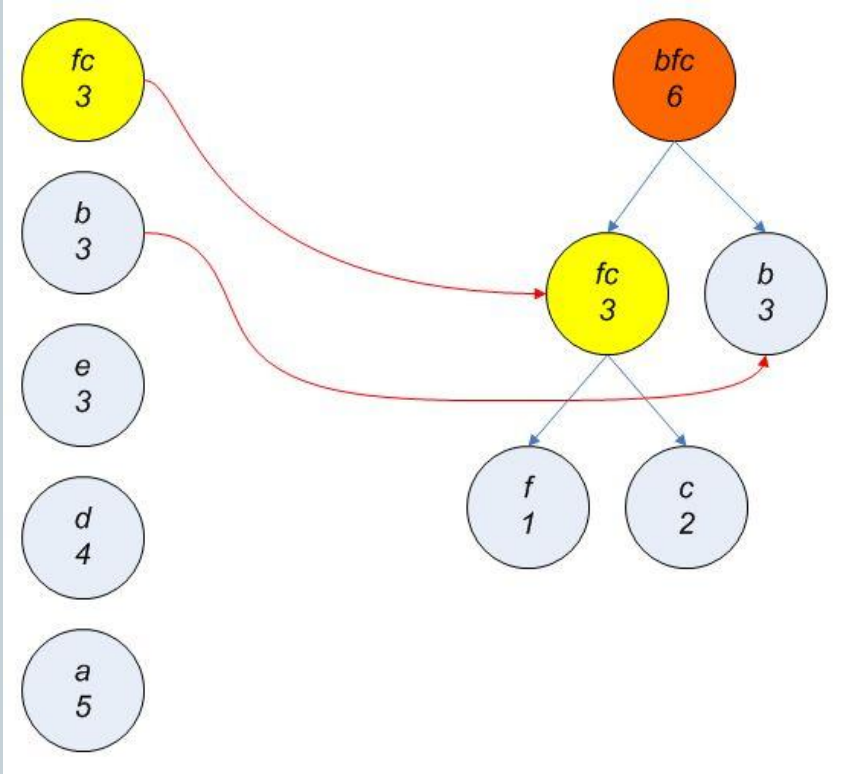
11



- En küçük sıklıklı iki düğüm f ve c tek bir düğüm altında birleştirilir.
- Yeni oluşturulan fc düğümünün sıklığı; f ve c 'nin toplam sıklığı olur.

Örnek – 2. Adım

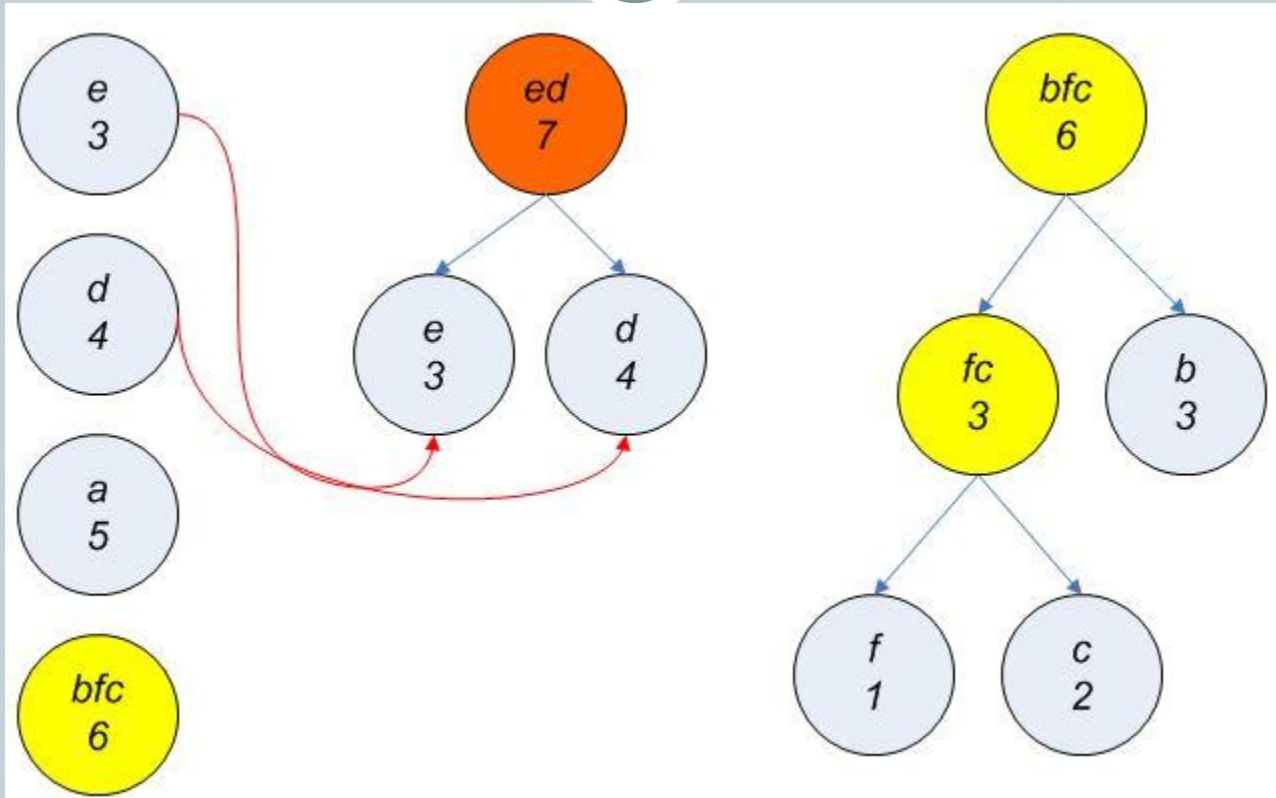
12



- Sıradaki en küçük sıklıklı b ve fc birleştirilir.
- Önceki adımda oluşturulan fc düğümü tekrardan listeye eklenmiştir. Bu düğümlere farklı bir davranış yoktur. Normal düğümlerle aynı şekilde ele alınırlar.

Örnek – 3. Adım

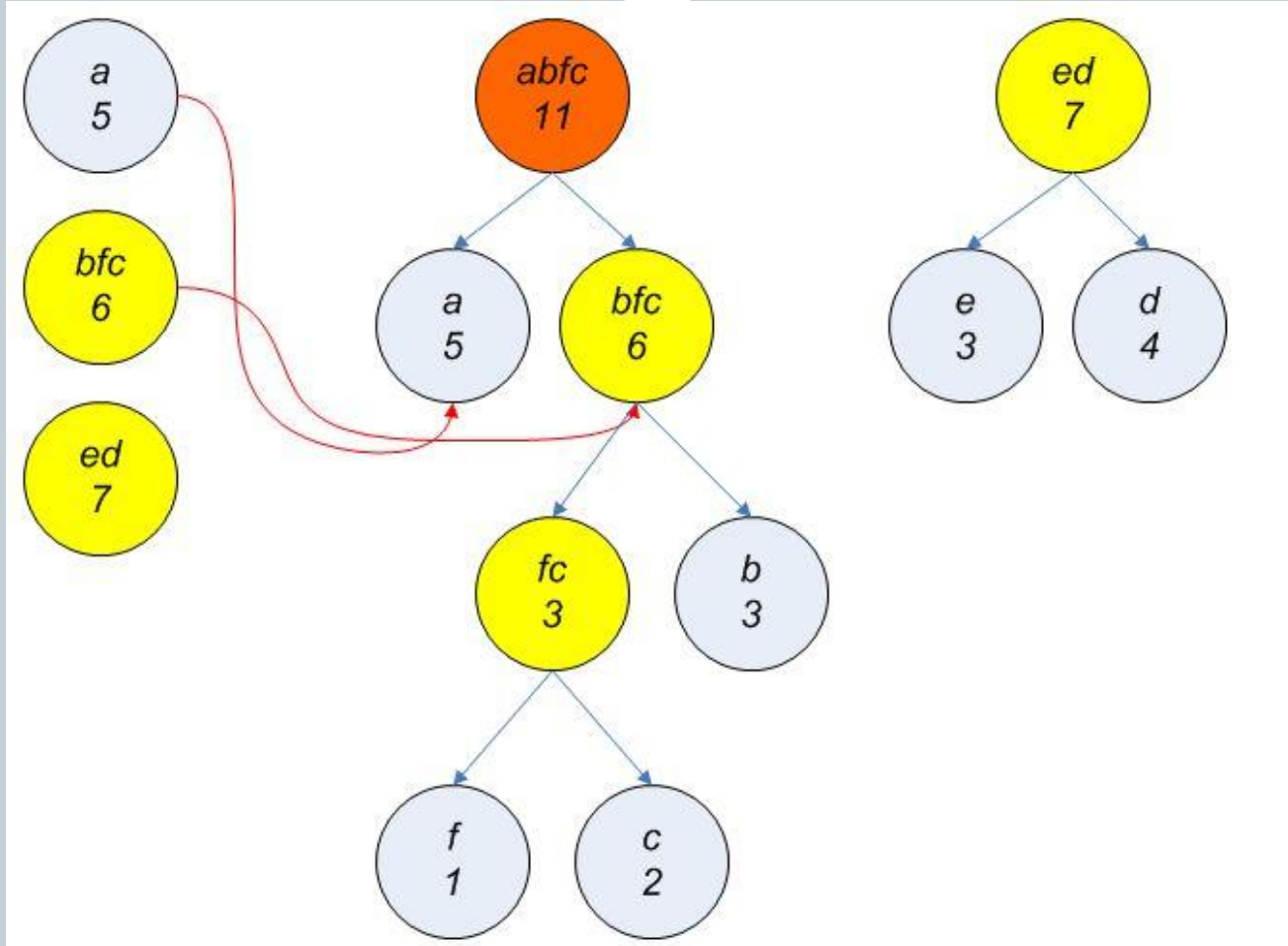
13



En düşük sıklıklı iki düğüm önceden oluşturulmuş ağaçtan bağımsız olabilir. Tekrardan bu iki düğümün birleştirilmesi ile oluşturulan düğüm listeye ekleneceğinden veri kaybı olmaz.

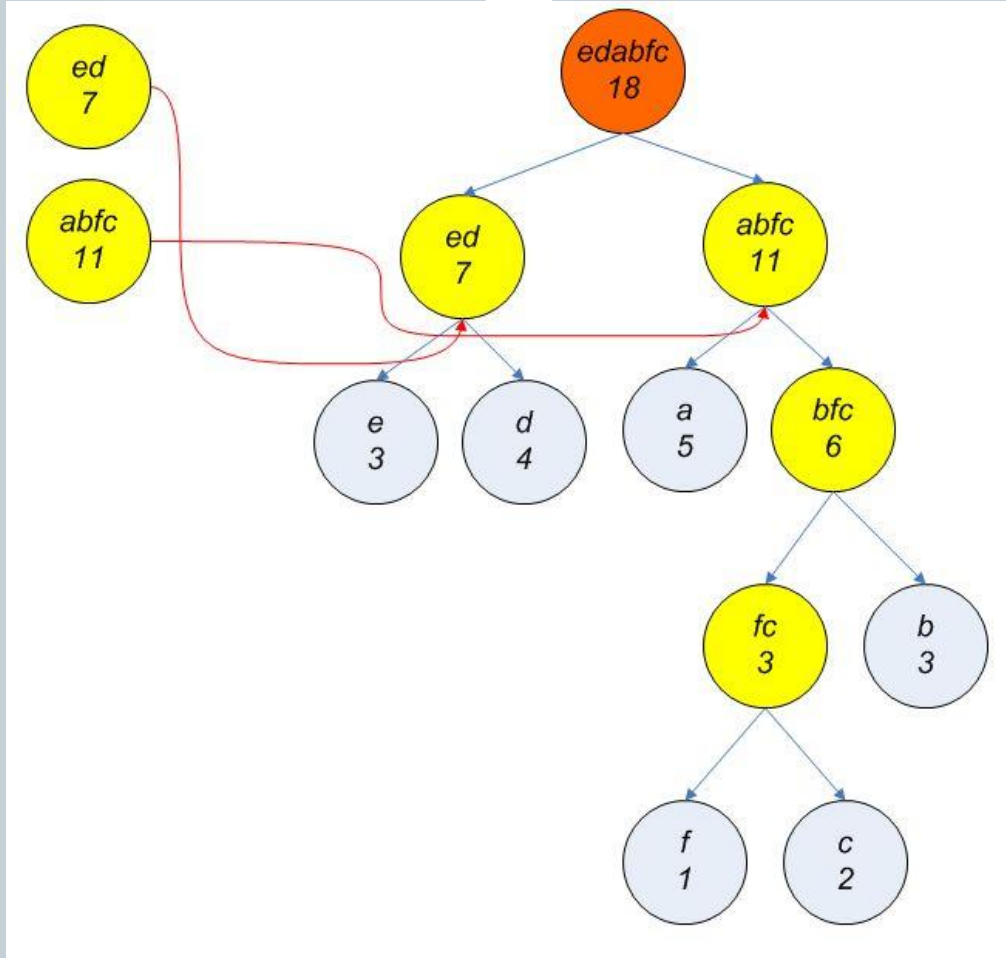
Örnek – 4. Adım

14



Örnek – 5. Adım

15

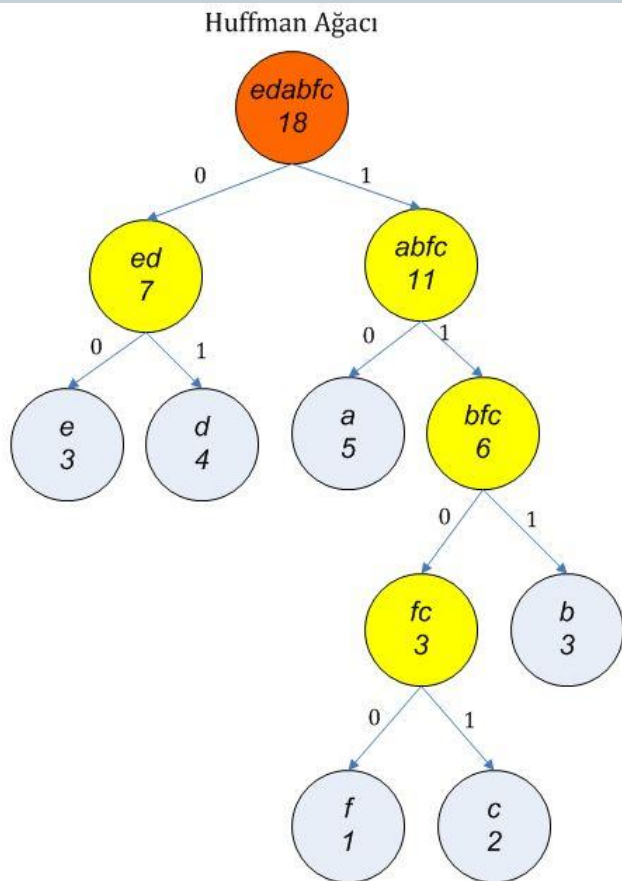


Örnek – 6. Adım

16

Kök Düğüm

edabfc
18



- Son birleştirme de gerçekleştirildiğinde dallara değerleri atandığında artık ağacımızda hangi karakterin kodunun ne olduğunu bulabiliriz.

- Bu örnekte;

- e: 00
- d: 01
- a: 10
- b: 111
- f: 1100
- c: 1101

Sonuç

17

- Sık olan karakterler üst dallarda, seyrek olanlar alt dallarda.
- Sıkıştırmadan önce
 - 144 bit (18 * 8 bit)
- Huffman kodu
 - 45 bit (sıklık * kod sözcüğü uzunluğu) yer
- Sıkıştırma oranı
 - $45 / 144 = \%31$

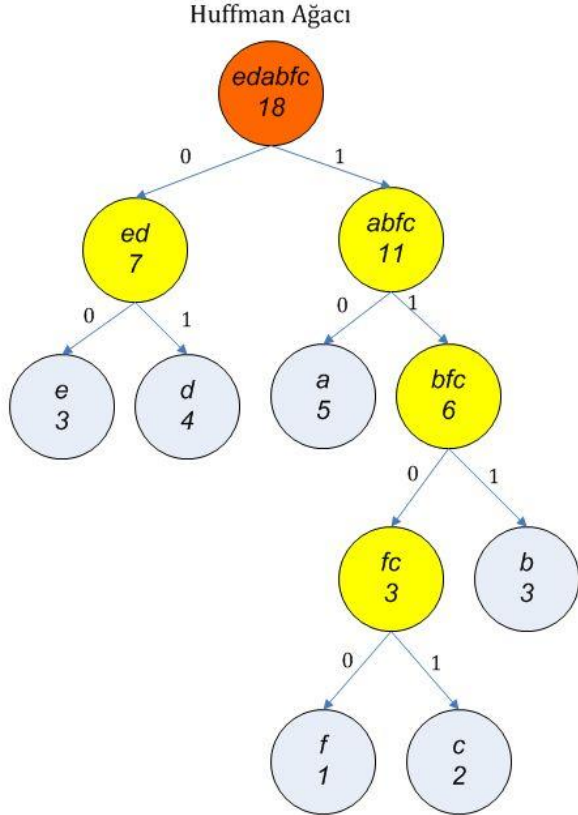
- a: 10 sıklık: 5
- d: 01 sıklık: 4
- e: 00 sıklık: 3
- b: 111 sıklık: 3
- c: 1101 sıklık: 2
- f: 1100 sıklık: 1

Sıkıştırımayı açmak

18

Kök Düğüm

edabfc
18



- Örneğin elimizde girdi katarımızın “**abbcccd**” kısmını çevirdiğimizi düşünelim.
- Bu kısmın Huffman kodu ile ifadesi şöyledir; “**101111111111101110101**”
- İlk karakterden başlayıp yolları takip ettiğimizde;
 - 1 -> abfc
 - 0 -> a (hiç çocuğu yok, köke dön)
 - 1 -> abfc
 - 1 -> bfc
 - 1 -> b (hiç çocuğu yok, köke dön)

Sorularınız

19

TEŞEKKÜRLER...