Course Name: M.Sc. in Computer Science
Semester – I, Session: 2019-2021
Department of Computer Science
Name of Faculty: Gautam Mahapatra, Associate Professor
Subject: Advanced Data Structure – Huffman Tree and its application for data compression and encoding

Class Taken:
Date: 5th April 2020, Time: 11.30AM – 1.00PM

Number of Students Attended: 25 / 25

Software used: Zoom
Internet Service: Jio-Fi

Details of the subject taught:

Huffman Tree

Data Compressing:
File = ssgsgssssbssAbssssAssssAAzbfgsss

Length(File) = 32
Let a1 = 's'
Count(a1) = 20, Probability(a1) = $\frac{20}{32} = \frac{5}{8}$
Let a2 = 'b'
Count(a2) = 3, Probability(a2) = $\frac{3}{32}$
Let a3 = 'g'
Count(a3) = 3, Probability(a3) = $\frac{3}{32}$
Let a4 = 'f'
Count(a4) = 1, Probability(a4) = $\frac{1}{32}$
Let a5 = 'A'
Count(a5) = 4, Probability(a5) = $\frac{4}{32} = \frac{1}{8}$
Let a6 = 'z'
Count(a6) = 1, Probability(a6) = $\frac{1}{32}$

Probability(a1)+ Probability(a2) + Probability(a3)+ Probability(a4)+ Probability(a5)+ Probability(a6)=1

How many different symbols?
What will be the file size in bytes and in bits?
What is encoding and decoding?

Decimal to Binary, BCD, EBCDIC, ASCII, Unicode

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010

k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

ASCII('s') = 01110011, ASCII('b')= 01100010, ASCII('g')= 01100111, ASCII('f')= 01100110, ASCII('A')= 01000001, ASCII('z')= 01111010

File in memory:
File = ssgsgssssbssAbssssAssssAAzbfgsss
File =
01110011011100110110011101110011011001110111001101110011011100110110001001110011011100110100000101110011011100110111001101110001001110011011100110100000101110011011100110111001101110011010000010111001101110011011100110111001101000001011100110111001101110011010000010100000101111010011000100110011001100111011100110111001101110011

If one 'a' is there?

How to decode?

Input to decode:
File1 =
0111001101110011011001110111001101110011011100110111001101110001001110011011100110100000101110011011100110111001101110011010000010111001101110011011100110111001101000001011100110111001101110011010000010100000101111010011000100110011001100111011100110111001101110011

Output:
ss…

Minimum no of bits to encode all these different symbols used in this file?

$$\lceil bits = log_2(N) \rceil$$

$$Entropy\ (h) = -log_R(p) = -log_R\left(\frac{1}{N}\right) = log_R(N)$$

$$Average\ Entropy\ (H) = \sum_{i=1}^{s} p_i h_i = -\sum_{i=1}^{s} p_i log_R(p_i)$$

What is the entropy for the given file?

Possible Encoding:
a1('s') = 000, a2('b')=001, a3('g')=010, a4('f')=011, a5('a')=100, a6('z')=101

File = ssgsgssssbssAbssssAssssAAzbfgsss

Minimum Encoded form:

File1 = 000000010000000000000001…

All cases length of code is fixed.

Huffman Coding: One optimal code-word design method that is simple to use and which is uniquely decodable and which result in lowest average bit rate is Huffman coding

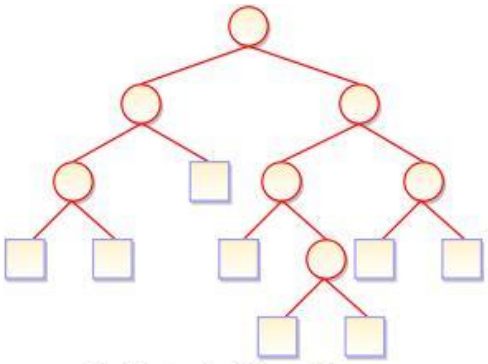
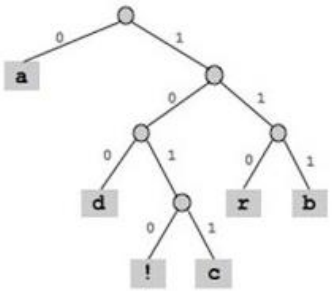
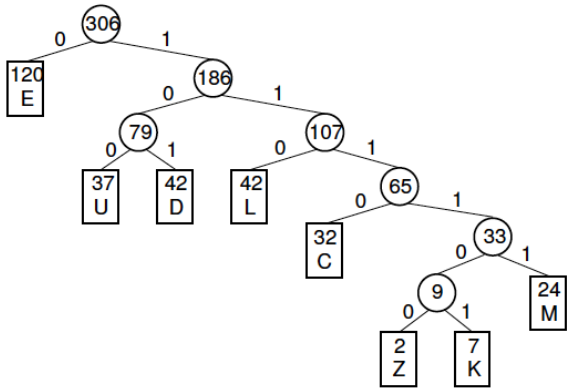


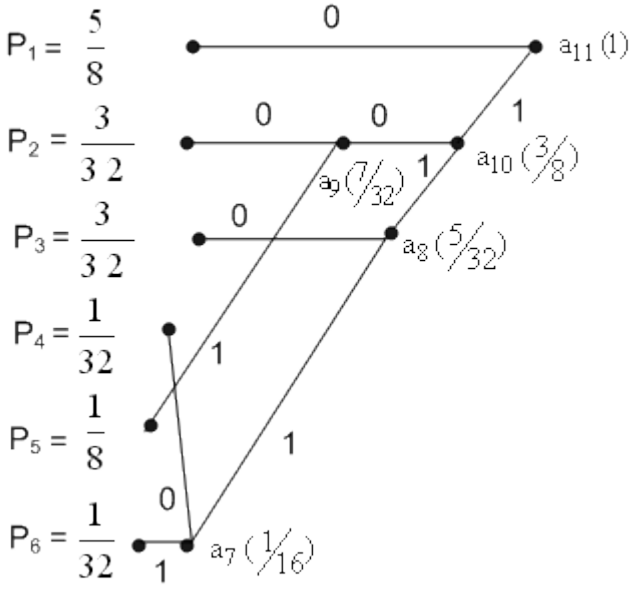
Fig. Extended Binary Tree

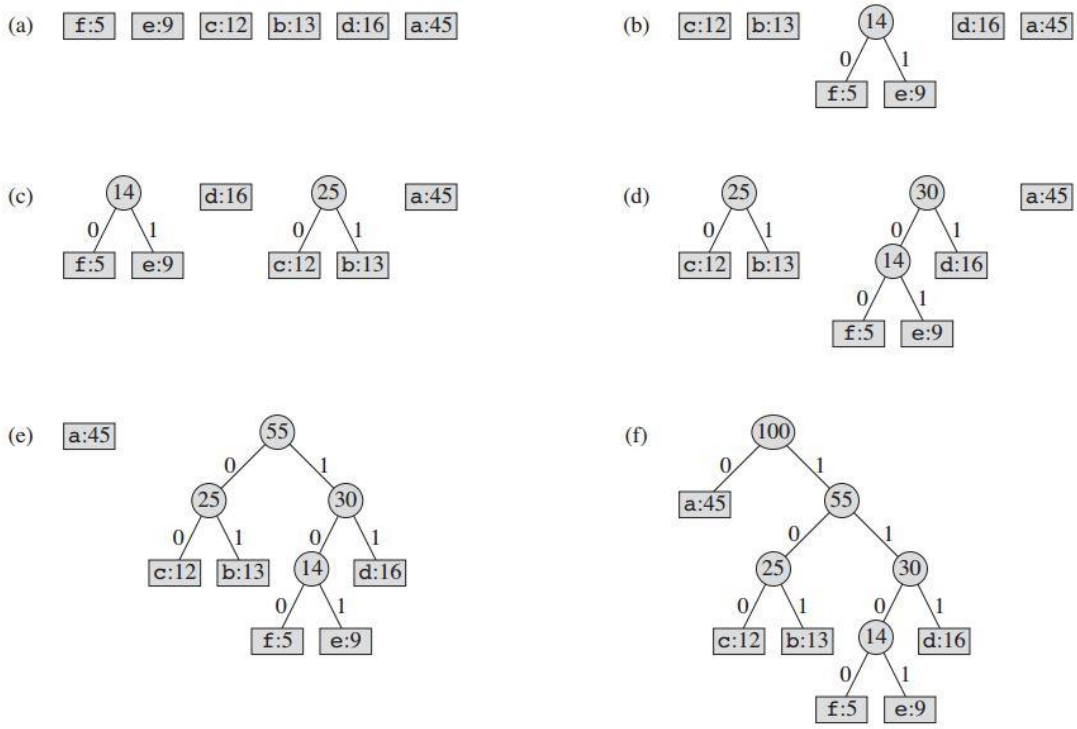


char	encoding
a	0
b	111
c	1011
d	100
r	110
!	1010



Message	Code	Probability
a_1	0	$P_1 = \frac{5}{8}$
a_2	100	$P_2 = \frac{3}{32}$
a_3	110	$P_3 = \frac{3}{32}$
a_4	1110	$P_4 = \frac{1}{32}$
a_5	101	$P_5 = \frac{1}{8}$
a_6	1111	$P_6 = \frac{1}{32}$





Message	Codeword	Probability
a1 (‘s’)	0	$P_1 = 5/8$
a2(‘b’)	100	$P_2 = 3/32$
a3 (‘g’)	110	$P_3 = 3/32$
a4 (‘f’)	1110	$P_4 = 1/32$
a5 (‘A’)	101	$P_5 = 1/8$
a6 (‘z’)	1111	$P_6 = 1/32$

Now the Encoded File:
a1(‘s’) = 0, a2(‘b’)=100, a3(‘g’)=110, a4(‘f’)=1110, a5(‘A’)=101, a6(‘z’)=1111

File = ssgsgssssbssAbssssAssssAAzbfgsss

File1 = 00110011000001000010110000001010000101101111111001110110000

Compressed?

What is the gain?

What is the compression ratio?

Entropy = ?

How to decode / decompressed the file?

Input:
File1 = 00110011000001000010110000001010000101101111111001110110000

Output:
File = ssgsgssssbssAbssssAssssAAzbfgsss