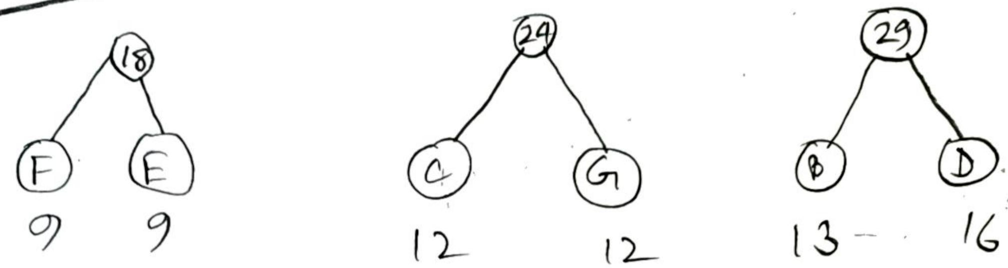
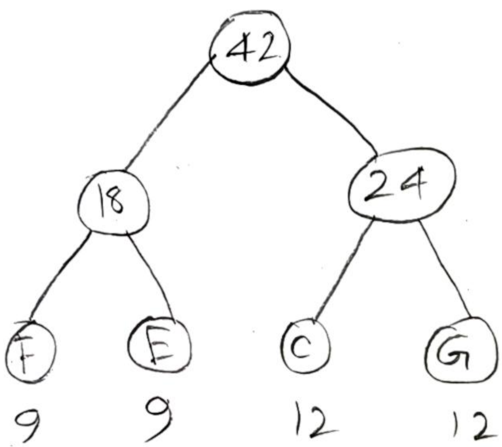
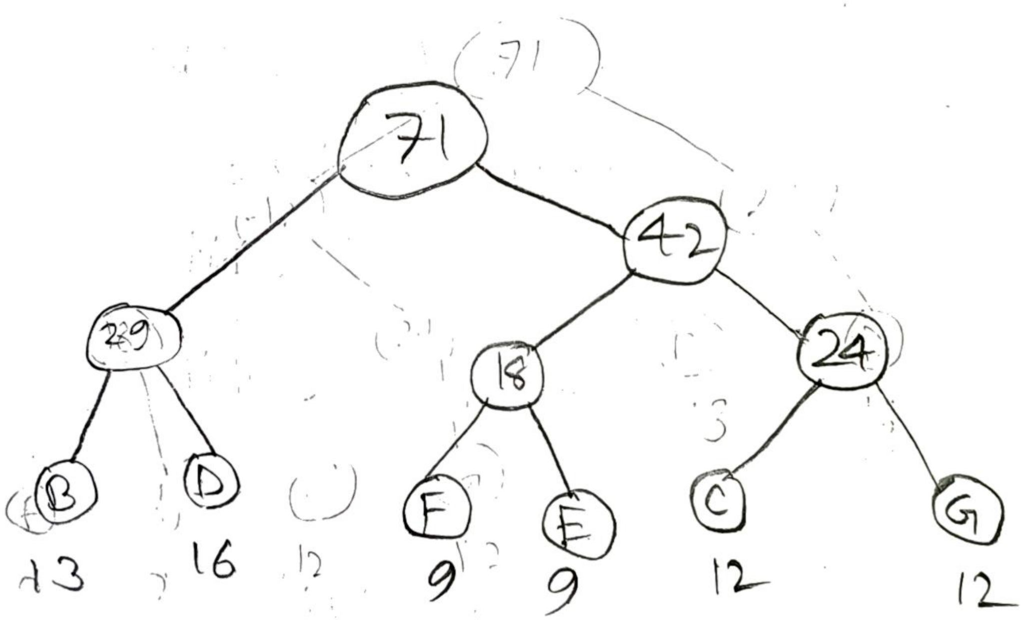
No.1



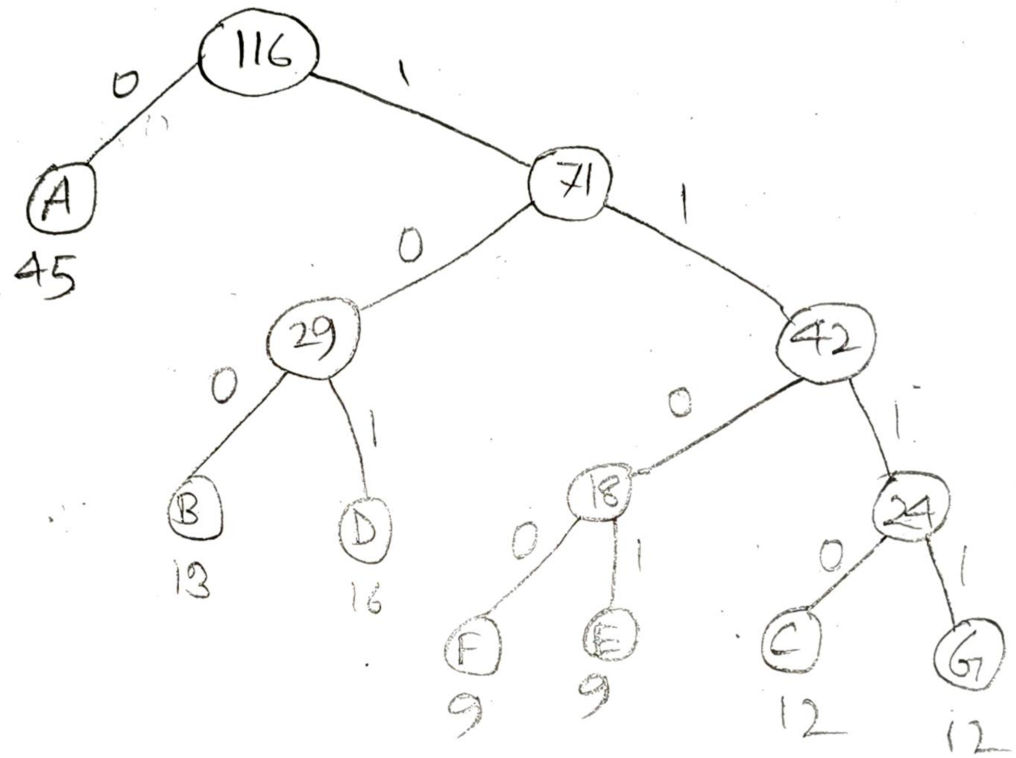
18
F
E
9
9
24
C
G
12
12
29
B
D
13
16



42
18
24
F
E
C
G
9
9
12
12



71
42
29
18
24
B
D
F
E
C
G
13
16
9
9
12
12



G — 1111

A — 0

B — 100

C — 1110

D — 101

E — 1101

F — 1100

If we used fixed 3 bit encoding

then space = 116X3+ 7X8 + 7X3

= 425 bit

encoding space = 300bit

table space = 7X8 + 23

= 79 bit

saved = 42 - (300+79) = 46bit

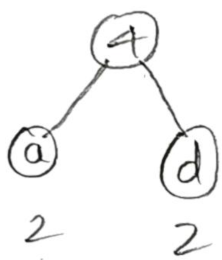
In Huffman coding we pick two smallest friquency symbol and make a new node thus they are placed deeper in the tree. So the symbol that occurs the most has less bit.

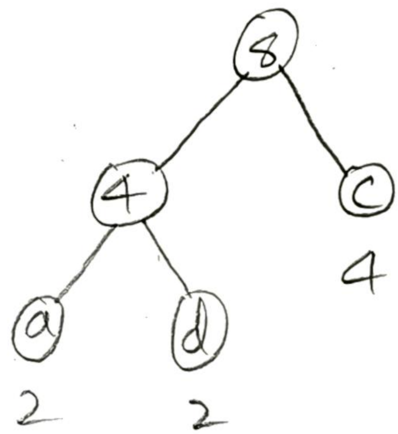
We ensure local optimal decision which results in global optimal prefix.

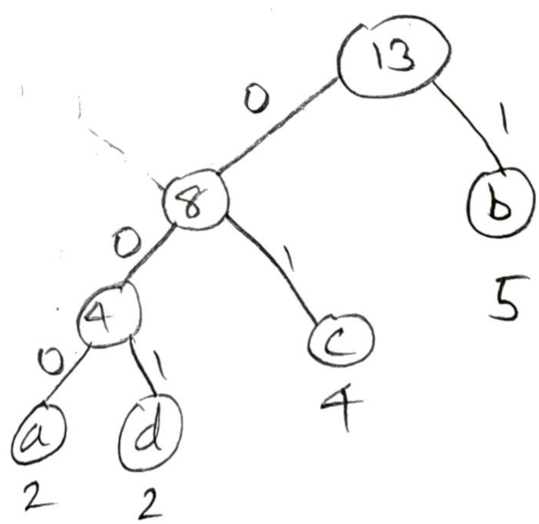
Huffman algorithm only care about the frequency. Placement of same frequency nodes doesn't effect as tree ensures minimum average code length. Both of them will create equal optimal codes.

No.2

a = 2
b = 5
c = 4
d = 2







a - 000

b - 1

c - 01

d - 001

msg = c b a b d c b b a d

01 1 000 1 001 01 1 1 000 001

c b c

01 1 01

bits for encoded msg = $2 \times 3 + 5 \times 1 + 4 \times 2 + 2 \times 3$

$= 25$

table space = $8 \times 4 + 9$

$= 41$