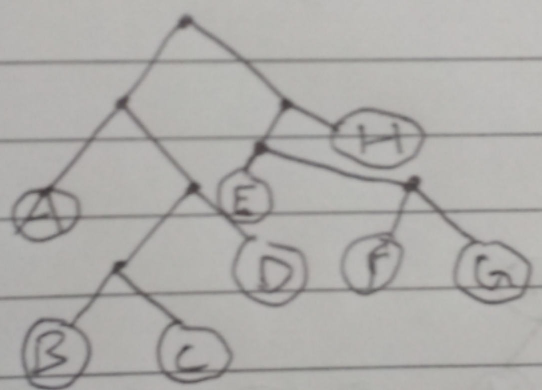
Name: Shreyas Bhat Kera
ID: 2018A7PS1119P

a) A = 0000
B = 0001
C = 001
D = 01
E = 100
F = 101
G = 110
H = 111

b)



c) next page

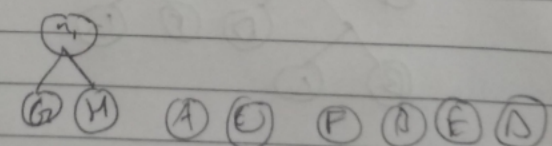
c) We create a tree, where each node is identified by a label, and can be determined as a leaf node based on its frequency, which is the sum of it children's for an internal node. Using a min priority queue to store the characters:

(G, $f_G = 1/40$
H, $f_H = 2/40$
A, $f_A = 3/40$
C, $f_C = 4/40$
F, $f_F = 5/40$
B, $f_B = 6/40$
E, $f_E = 9/40$
D, $f_D = 10/40$)

Algorithm: ~~for / while~~ till the queue is not empty take 2 elements and create a parent whose frequency is the sum of its children

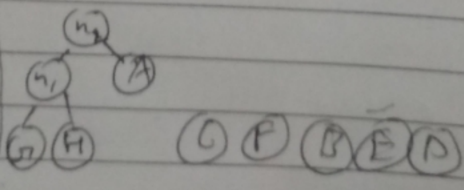
i) Priority Queue | Corresponding Tree

($n_1, f_{n_1} = 3/40$)
($A, f_A = 3/40$)
($C\ f_C = 4/40$)
($F\ f_F = 5/40$)
($B\ f_B = 6/40$)
($E\ f_E = 9/40$)
($D\ f_D = 10/40$)



ii) Queue | Tree

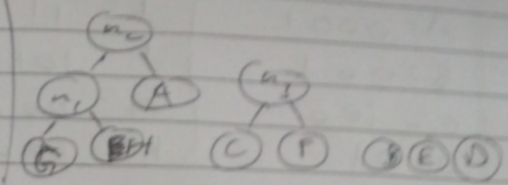
$C\ f_C = 4/40$
$F\ f_F = 5/40$
$n_2, f_{n_2} = 6/40$
$B, f_B = 6/40$
$E, f_E = 9/40$
($D\ f_D = 10/40$)



iii) Queue

(n_2, $f_{n_2} = 6/40$)
(B, $f_B = 6/40$)
(n_3, $f_{n_3} = 9/40$)
(E, $f_E = 9/40$)
(D, $f_D = 10/40$)

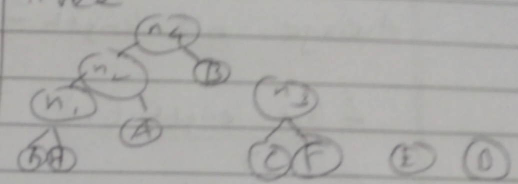
Tree



iv) Queue

(n_3, $f_{n_3} = 9/40$)
(E, $f_E = 9/40$)
(D, $f_D = 10/40$)
(n_4, $f_{n_4} = 12/40$)

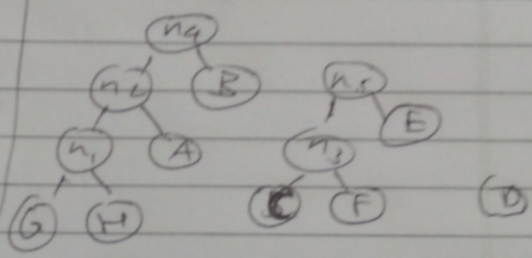
Tree



v) Queue

(D, $f_D = 10/40$)
(n_4, $f_{n_4} = 12/40$)
(n_5, $f_{n_5} = 18/40$)

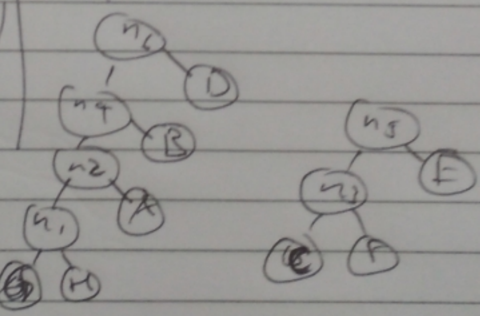
Tree



vi) Queue

(n_5 $f_{n_5} = 18/40$)
(n_6 $f_{n_6} = 22/40$)

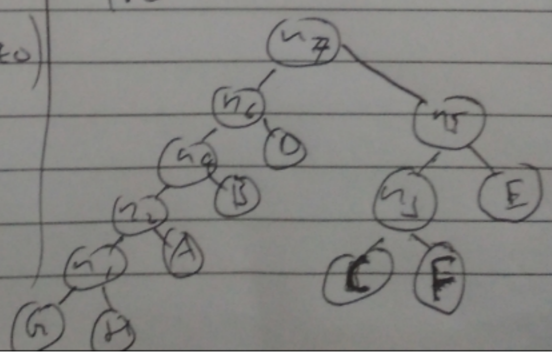
Tree



vii) Queue

(n_7 $f_{n_7} = 40/40$)

Tree



d) Bit lengths:

A = 0001	E = 11
B = 001	F = 101
C = 100	G = 00000
D = 01	H = 00001

Average bit length =

$$\frac{\sum \text{Bit length}(i)}{8} \quad \text{for } i \text{ in } \{A, B, C, D, E, F, G, H\}$$

$= (4 + 3 + 3 + 2 + 2 + 3 + 5 + 5)/8$

$= 3.375$