ADA

181240116001
Vishwas Acharya

Assignment-5

Q.1) Explain characteristics of Greedy algorithms

Ans Greedy algorithm can be characterized by the following feature:

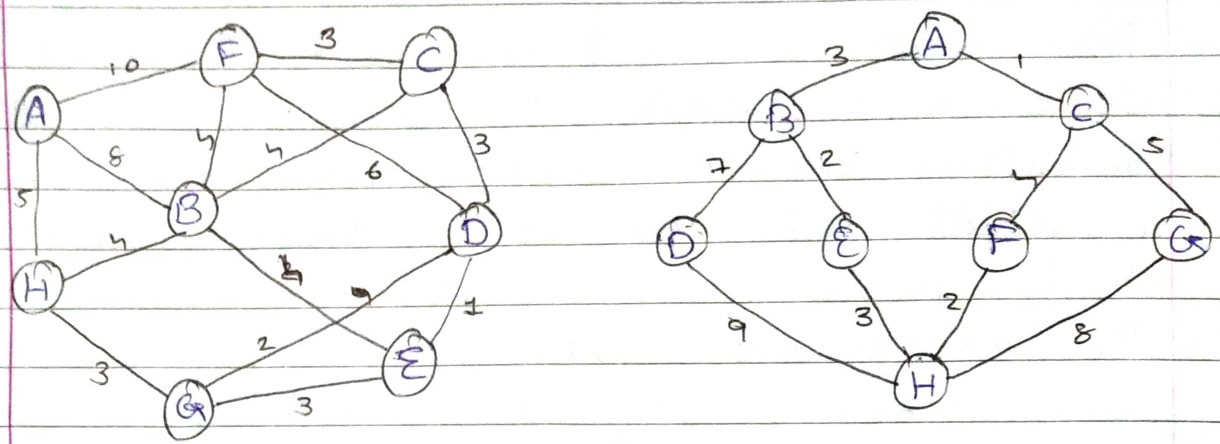
- It forms a set or list of candidates.
- Once a candidate is selected in the solution, it is there forever, once a candidate is executed from the solution, it is never reconsidered.
- To constructs the solution in an optimal way, Greedy algorithm maintainer two sets.
- One set contains candidates that have already been considered & chosen, while another set contains candidates who were considered but rejected.
- The Greedy algorithm consists of four function.

(i) Solution Function: A function that checks whether choosen set of items provide a solution.

(ii) Feasible Function: It tells which of the candidate is the most promising.

(iii) Objective Function: It does not appear explicitly, but gives the values of a solution.

Q.2) Write the Prim's algorithm to find out minimum spanning Tree. Apply the same & find MST for the graph given below.



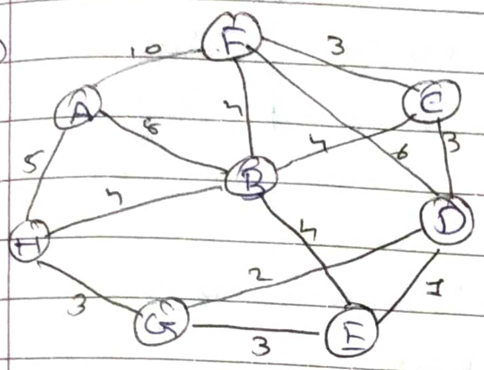
(i) (ii)

Ans - In Prim's algorithm, the minimum spanning tree grows in a natural way, starting from an arbitrary root.
- At each stage we add a new branch to the tree already constructed, the algorithm stops when all the nodes have been reached.
- The complexity for the Prim's algorithm is $O(n^2)$ where n is the total numbers of nodes in the graph G.

Function Prim (G = (N, A) : graph;
length : A - R+) : set of edges
T <- ϕ
B <- { an arbitrary member of N }
while B ≠ N do
find e = {u, v} of minimum length ∋
u ∈ B & v ∈ N \ B

$T \leftarrow T \cup \{e\}$
$B \leftarrow B \cup \{v\}$
return T

(i)



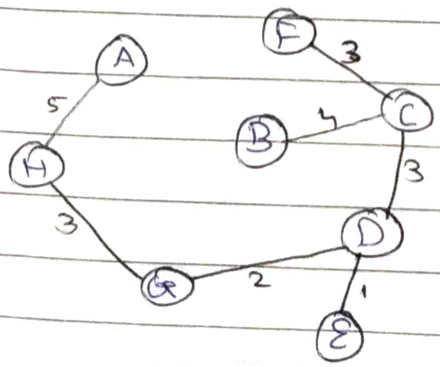
Step 1 select an arbitary node

Node	Edges
A	

Step 2 Find an edge with minimum weight.

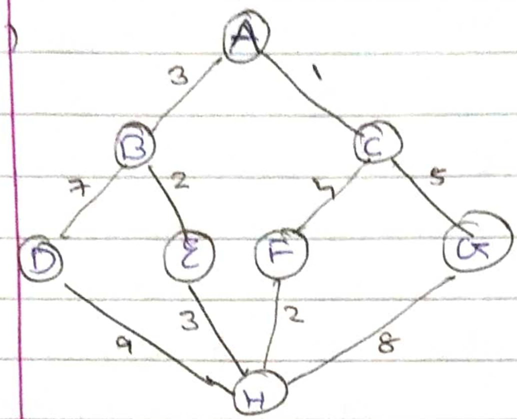
Node	Edges
A	{A, F}, {A, H}, {A, B}
A, H	{H, G}, {H, B}
A, H, G	{G, H}, {G, D}, {H, B}, {A, F}
A, H, G, D	{G, E}, {D, E}, {H, B}, {A, F}
A, H, G, D, E	{H, B}, {D, C}, {D, F}, {A, F}
A, H, G, D, E, C	{A, F}, {H, B}, {C, F}, {C, B}
A, H, G, D, E, C, F	{A, F} {C, B} {D, F}, {A, B}, {E, B}, {F, B}, {H, B}
A, H, G, D, E, C, F, B	

Step 3 Minimum Spanning Tree



Total cost of spanning tree $= 5+3+2+1+3+4+3$
$= \boxed{21}$

(ii)



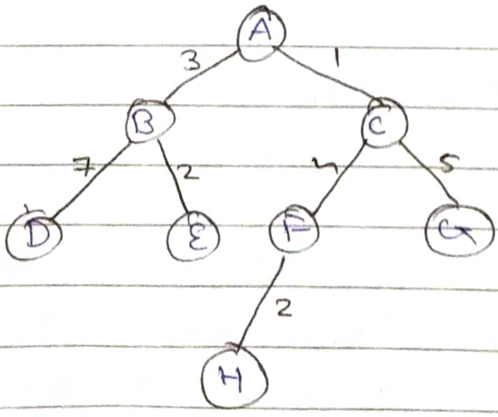
Step: I select an arbitrary node

Node	Edges
A	

Step: 2 Find an edge with minimum weight.

Node	Edges
A	{A, B}, {A, C}
A, C	{A, B}, {C, F}, {C, G}
A, B, C	{B, D}, {B, E}, {C, F}, {C, G}
A, B, C, E	{B, D}, {E, H}, {C, F}, {C, G}
A, B, C, E, F	{B, D}, {F, H}, {E, H}, {C, G}
A, B, C, E, F, H	{B, D}, {H, D}, {C, G}, {H, G}
A, B, C, E, F, H, G	{B, D}, {H, D}
$\boxed{\text{A, B, C, E, F, H, D, G}}$	

Step: 3 Minimum Spanning Tree

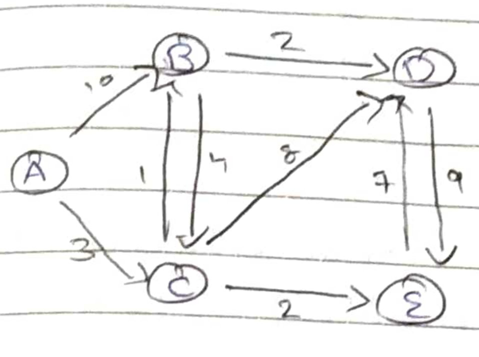


∴ Total cost of spanning Tree

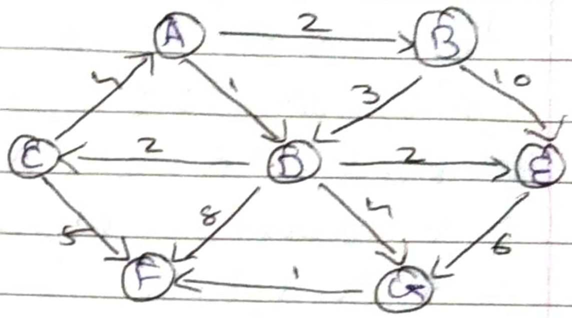
$= 1 + 3 + 7 + 2 + 4 + 5 + 2$

$= \boxed{24}$

Q.3) Write Dijkstra's Algorithm for shortest path. Use the algorithm to find the shortest path from the following graph.



(i)



(ii)

→ Consider now a directed graph ? = (?, A). ? is the set of nodes & A is the set of directed edges of graph ?

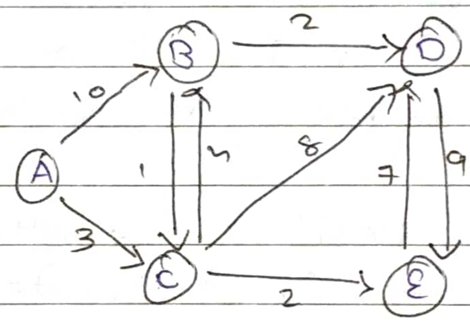
- Each edge has positive length.
- One of the nodes is designated as the source node.
- It is to determine the length of the shortest path from the source to each of the other nodes of the graph.
- Dijkstra's algorithm is for finding the shortest paths between the nodes in a graph.
- The algorith maintains a matrix ? which gives the length of each directed edge

?(?, ?) ≥ 0 if the edge (?, ?) ∈ A &
?[?, ?] = ∞ otherwise.

- Function Dijkstra (L[1...n, 1...n]);
array [2...n] array D[2...n]
C ← {2, 3, ..., n}
{S = N, C exist only implicity}

for i ← 2 to n do
D[i] ← L[1, i]
repeat n-2 times
v ← some element of (minimizing D[v])
C ← C \ {v} { and implicity S ← S ∪ {v}}
for each w ∈ C do
D[w] ← min (D[w], D[v] + L[v+w])
return D

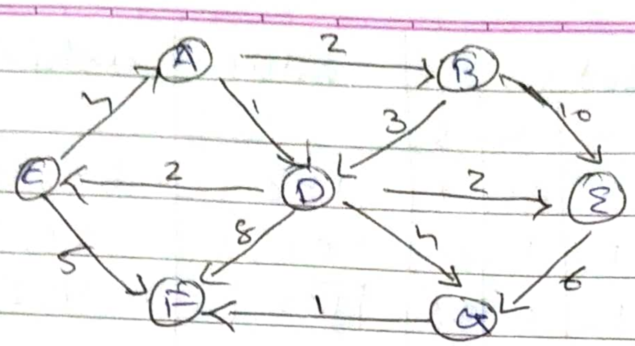
(i)



step	Node v	C	C	B	D	E
inital	–	{B, C, D, E}	3	10	–	–
1	C	{B, D, E}	3	4	11	5
2	B	{D, E}	3	4	6	5
3	D	{E}	3	4	6	16

Shortest path = ACE or ACBDE

(ii)



Node	A	D	B	C	E	F	G
A	0	∞	∞	∞	∞	∞	∞
D	0	1	2	∞	∞	∞	∞
C	0	1	2	3	3	9	5
G	0	1	2	3	3	8	5
F	0	1	2	3	3	6	5

Shortest Path = ADGF

Q.4) We are given 5 objects & the weight carrying capacity of knapsack is $W=100$. for each object, weight w_i & value v_i are given in the following table.

object	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Fill the knapsack with given objects such that the total value of knapsack is maximized.

Ans

object	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
v_i/w_i	2	1.5	2.2	1	1.2

selection	objects					Value	weight capacity
	1	2	3	4	5		100
Max v_i	0	0	1	0.5	1	146	30, 50, 20
Min w_i	1	1	1	1	0	156	10, 20, 30, 40
Max v_i/w_i	1	1	1	0	0.8	$\boxed{164}$	30, 10, 20, 40

Maximized total value is = 164

Q.5) Solve the example with Activity selection algorithm.

- Given arrival & departure times of all trains that reach a railway station, Find the minimum number of platforms required for the railway station so that no train waits. We are given two arrays which represent arrival & departure times of trains that stop.

Arr [] = [9:00, 9:40, 9:50, 11:00, 15:00, 18:00]
dep[] = [9:10, 12:00, 11:20, 11:30, 19:00, 20:00]

Ans

platform	1	2	3	4	5	6
Arr[]	9:00	9:40	9:50	11:00	15:00	18:00
dep[]	9:10	12:00	11:20	11:30	19:00	20:00

Now, arrange departure time in ascending order.

step1

Platform	(arr[i], dep[i])
1	(9:00, 9:10)
3	(9:50, 11:20)
4	(11:00, 11:30)
2	(9:40, 12:00)
5	(15:00, 19:00)
6	(18:00, 20:00)

step 2 $P = \{1\}$

$P = \{1, 3\}$

$P = \{1, 3, 5\}$

$\boxed{P = \{1, 3, 5\}}$

∴ we have required 3 platform.

Q.6) Solve the example with job scheduling with deadline - example.

- Using greedy algorithm find an optimal schedule for following jobs with $n=4$
profit = (20, 10, 40, 30)
Deadline = (4, 1, 1, 1)

Ans

Job i	1	2	3	4
profit	40	30	20	10
Deadline(i)	1	1	4	1

step1 Position $P = \min(n, \max(d_i))$

$= \min(4, 4)$

$= 4$

P	1	2	3	4
Job selected	0	0	0	0

step:2 $d_1 = 1$ assign job 1 to position 1

P	1	2	3	4
Job selected	J1	0	0	0

step:3 $d_2 = 1$: assign job 2 to position 1 but its already occupied, reject job 2

step:4 $d_3 = 4$: assign job 3 to position 4

P	1	2	3	4
Job selected	J1	0	0	J3

step:5 $d_4 = 1$: assign J_4 to positon 1 but its already occupied, reject J_4.

$\boxed{\text{max - profit} = 60}$

- Using greedy algorithm find an optimal schedule for following jobs with n=5
profit = (100, 19, 27, 25, 75)
Deadline = (2, 1, 2, 1, 3)

Ans

Job (j)	1	2	3	4	5
profit	100	27	25	19	15
Deadline (i)	2	2	1	1	3

step:2 Position $P = \min(n, \max(d_i))$
$= \min(5, 3)$
$\boxed{P = 3}$

P	1	2	3
Job selected	0	0	0

step:3 $d_1 = 2$: assign job 1 to position 2

P	1	2	3
Job selected	0	J1	0

step:4 $d_2 = 2$: assign job 2 to position 1

P	1	2	3
Job selected	J2	J1	0

step:5 $d_3 = 1$: asign Job 3 to position 1 but it already occupied, reject J3

step:6 $d_4 = 1$: assign Job 4 to position 1 but it already occupied, reject J_4

step:7 $d_5 = 3$: assign Job 5 to positon 3

P	1	2	3
Job selected	J_2	J_1	J_5

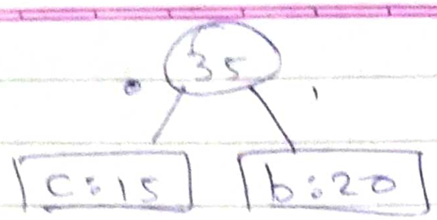
Q.7 Find an optimal Huffman code for the following example.

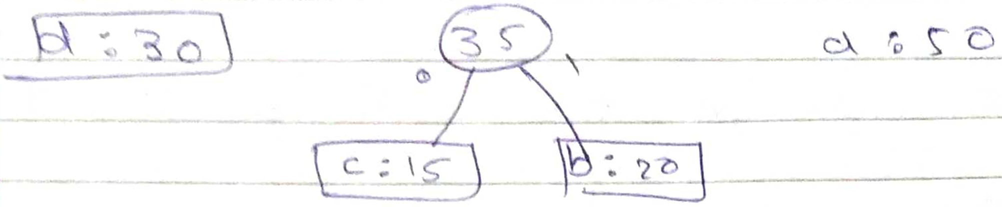
- a = 50 b = 20 c = 15 d = 30

Ans step:1 Arrange the characters in the ascending order of their frequency

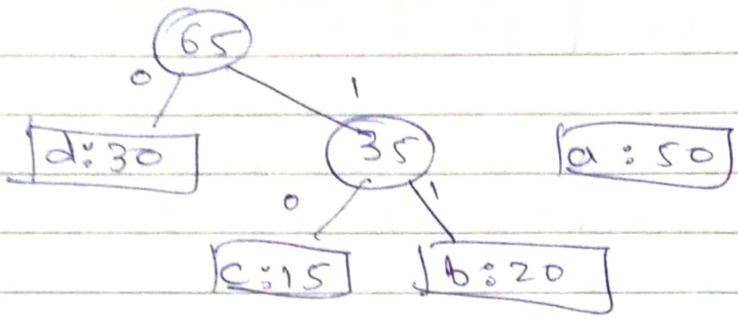
$\boxed{c=15}$ $\boxed{b=20}$ $\boxed{d=30}$ $\boxed{a=50}$

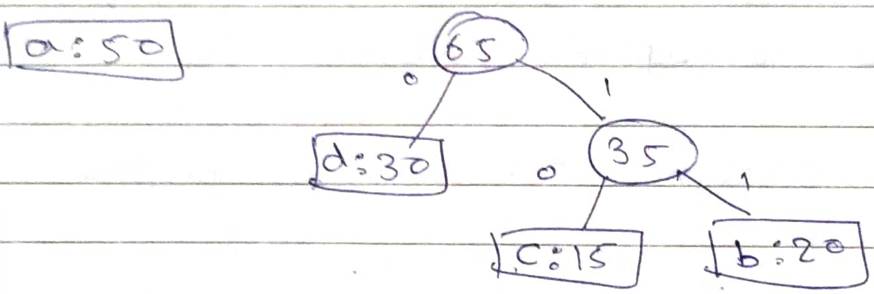
step:2



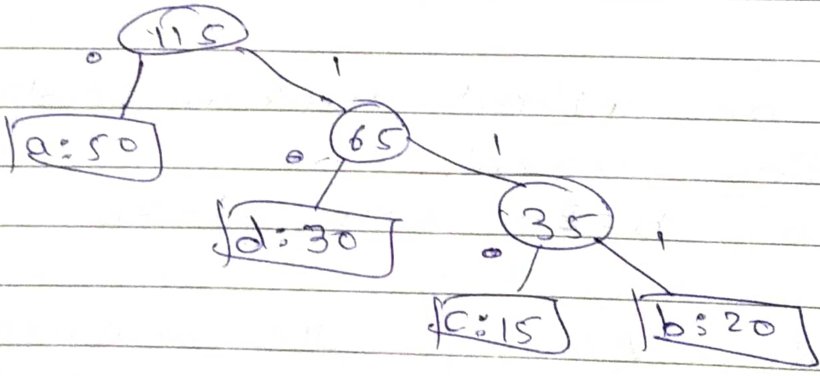


step:3





step:4



charaters	a	b	c	d
frequency	50	20	15	30
	0	111	110	10

- Frequency

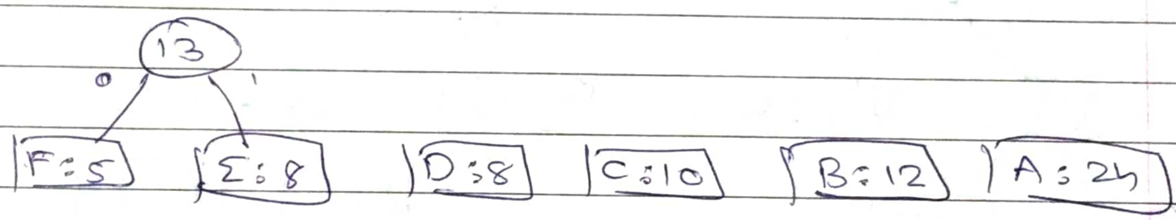
characters	A	B	C	D	E	F
Frequency (in thousand)	24	12	10	8	8	5

step:1 Arrange the charactes in the ascending order with their frequency

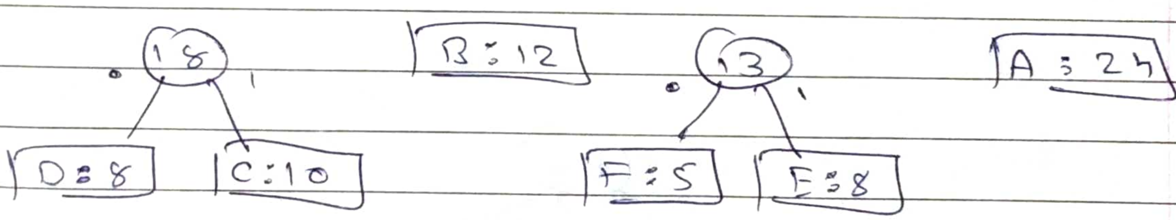
character	F	E	D	C	B	A
frequency	5	8	8	10	12	24

F : 5 E : 8 D : 8 C : 10 B : 12 A : 24

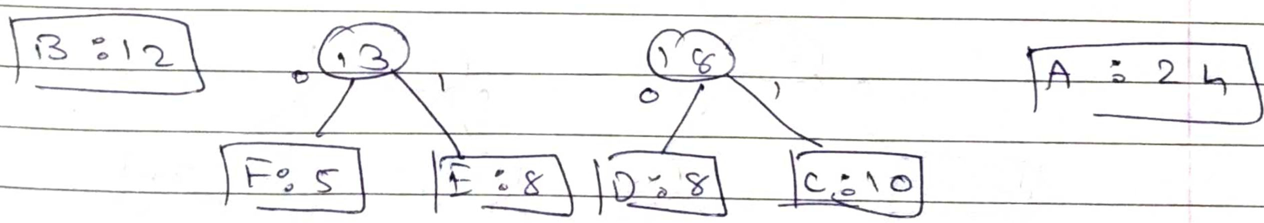
step:2 Extract two nodes with the minimum frequency,



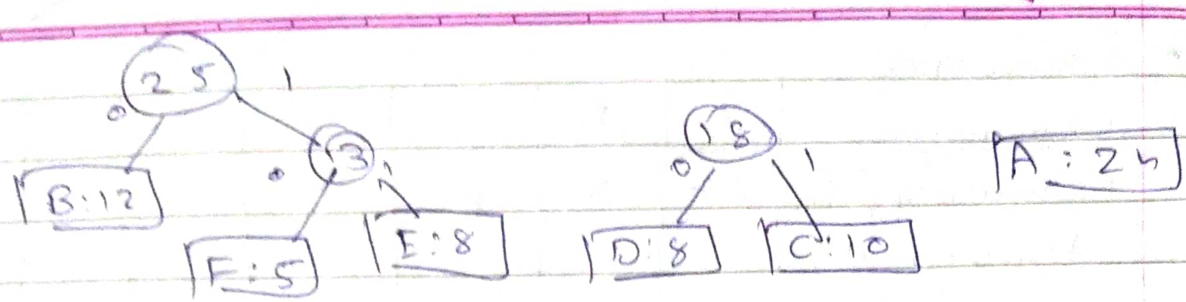
step:3

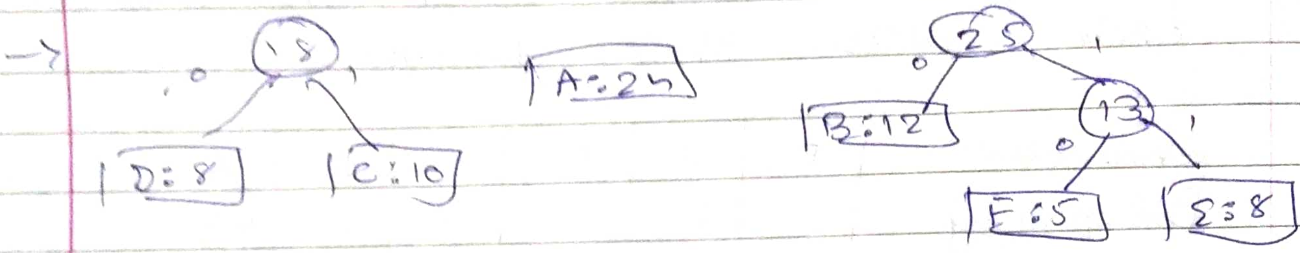


→

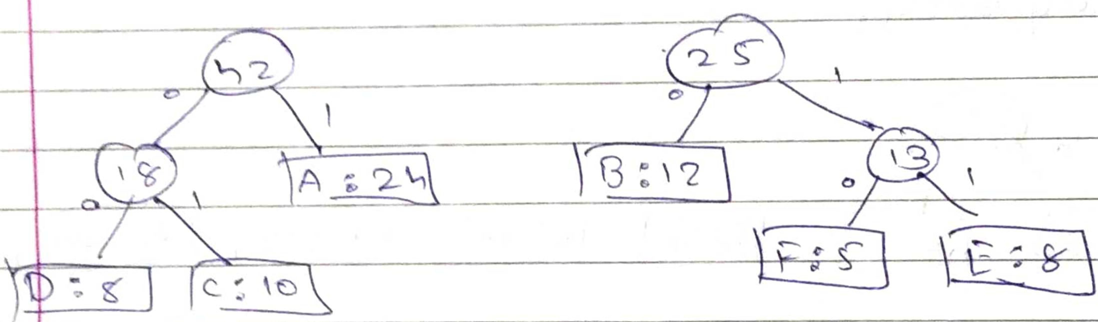


step:4

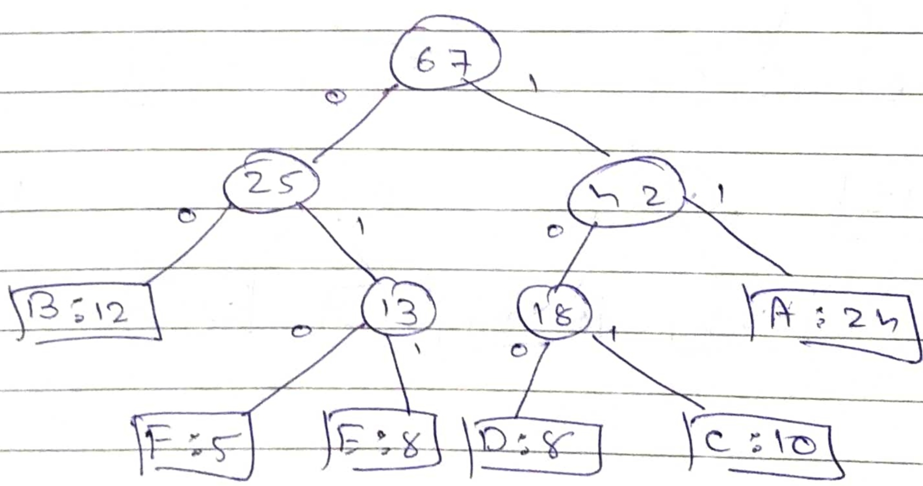




step:5 Extract two nodes with minimum frequency.



step:6



Character	A	B	C	D	E	F
Frequency	24	12	10	8	8	5
	11	00	101	100	011	010

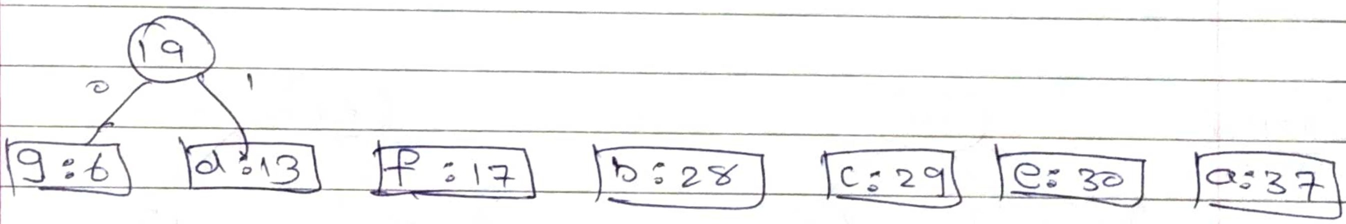
- Frequency

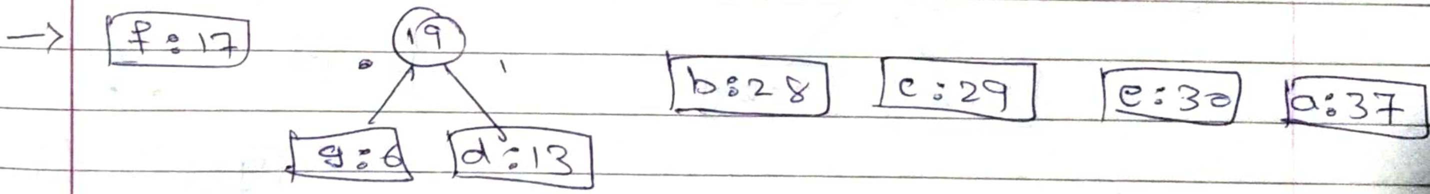
character	a	b	c	d	e	f	g
frequency (in thousand)	37	28	29	13	30	17	6

Ans **step 1** Arrange the characters in ascending order with their frequency

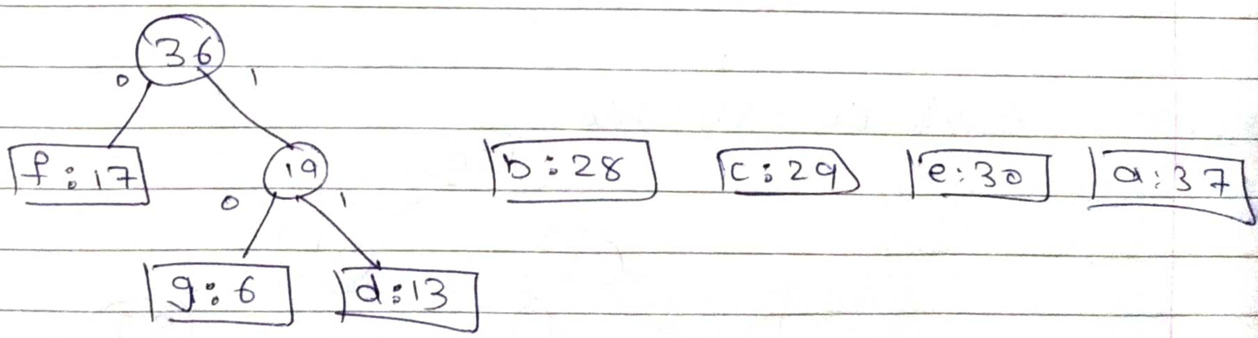
g : 6 | d : 13 | f : 17 | b : 28 | c : 29 | e : 30 | a : 37

step 2 Extract two nodes with minimum frequency

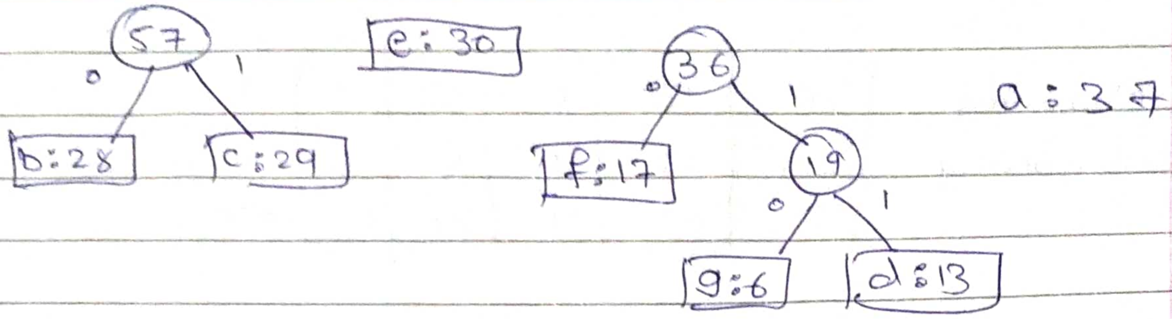




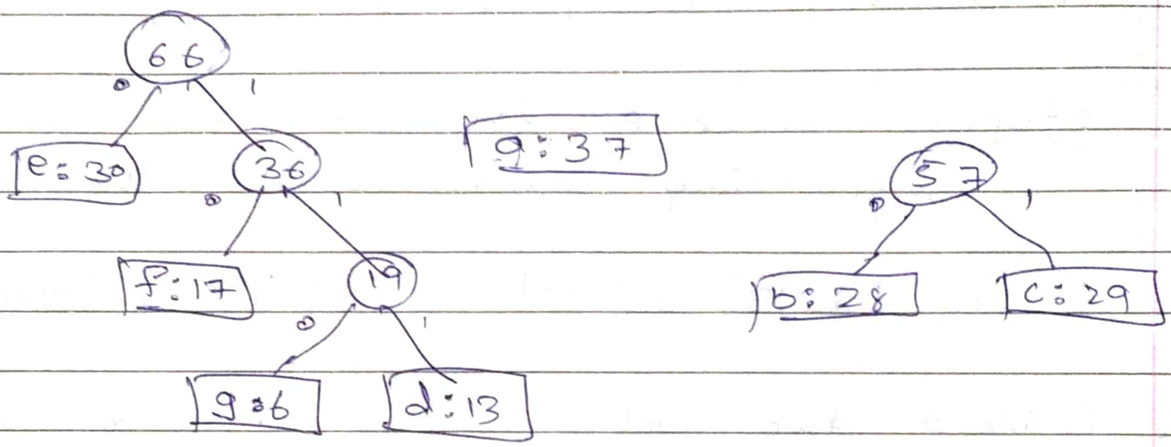
step 3



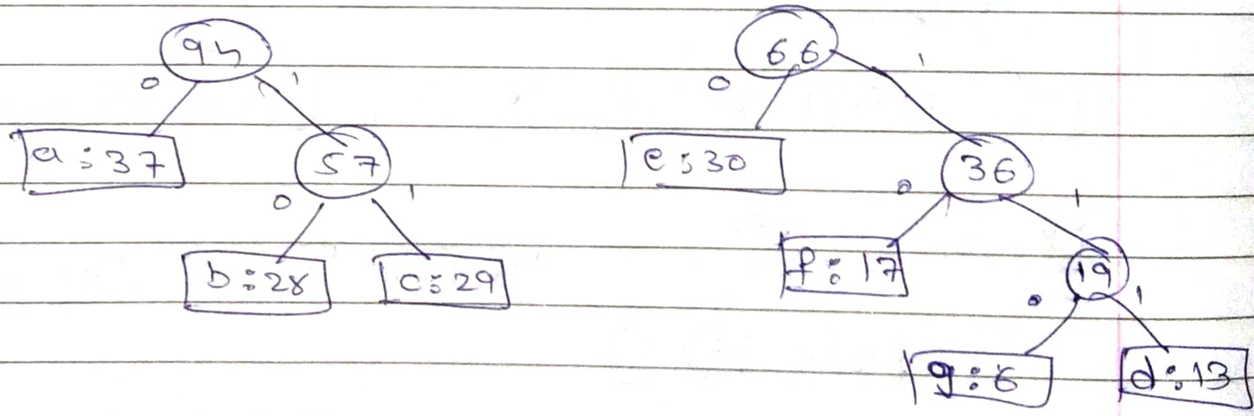
step:4



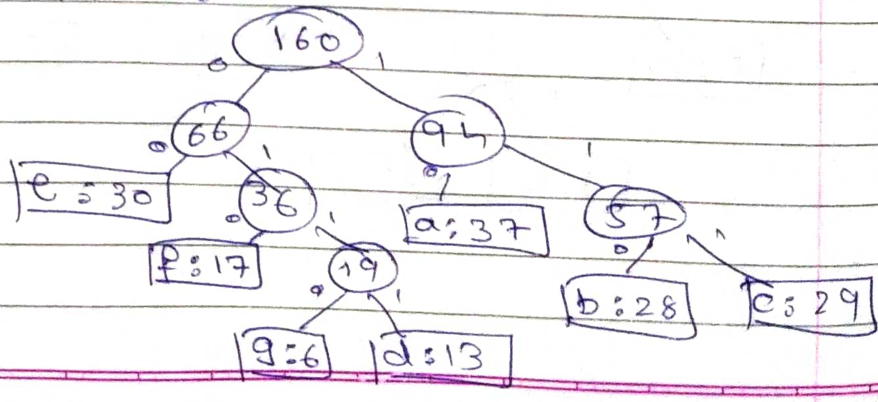
step:5 Extract two nodes with minimum frequency



step:6



step:7 Add 66+94=160



character	frequency	Huffman code
a	37	10
b	28	110
c	29	111
d	13	0111
e	30	00
f	17	010
g	6	0110