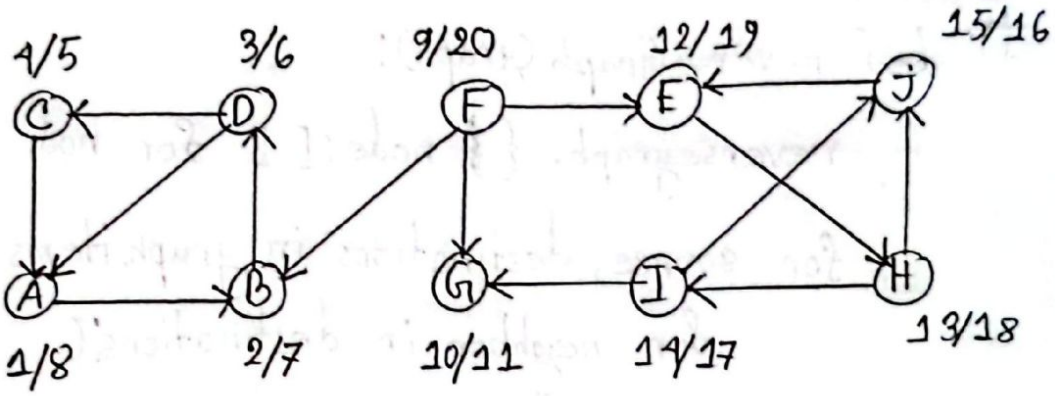
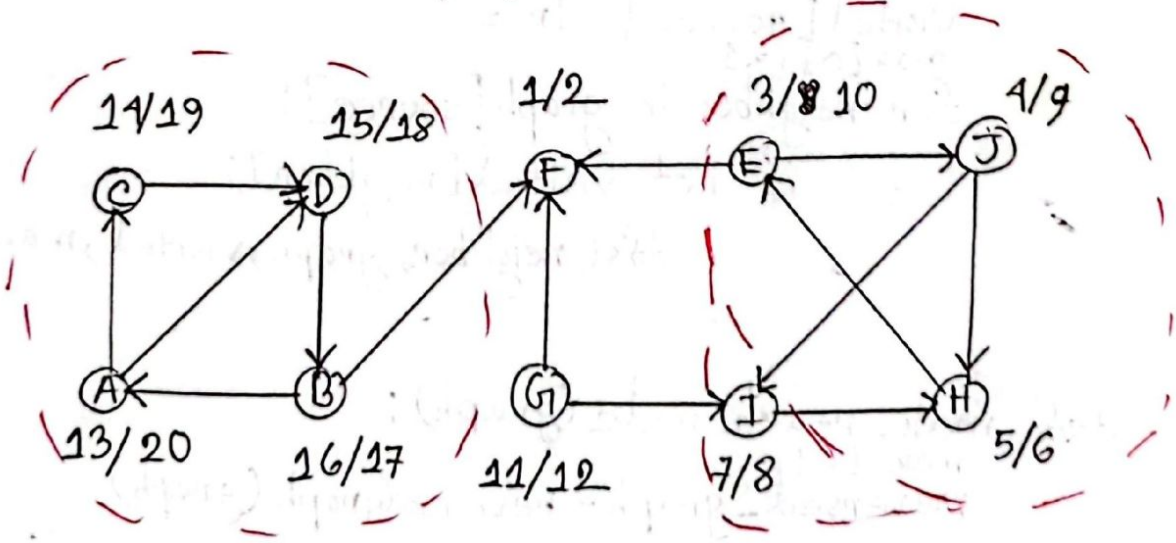
Summer - 23 (A)

1) a) Kosaraju's Strongly Connected Components Algorithm

b)



Decreasing order: F → E → H → I → J → G → A → B → D → C



Two Groups:

1	2
Carol	Ethan
Dan	Jack
Bob	Isaac
Alice	

farhan

farhan

c) Fahim
Glen

d)
```
def reverseGraph(graph):
    reversegraph = {node:[] for node in graph}
    for source, destinations in graph.items():
        for neighbor in destinations:
            reversegraph[neighbor].append(source)
    return reversegraph
def dfs(source, graph, visited, max):
    visited[source] = True
    max[0] += 1
    for neighbor in graph[source]:
        if not visited[neighbor]:
            dfs(neighbor, graph, visited, max)

def max_reach_node(graph):
    max = [0]
    reversed_graph = reverseGraph(graph)
    max_visit = float("-inf")
    max_node = None
    for node in graph:
        if not visited
```

```
visited = {node: False for node in graph}
for node in graph:
    dfs(node, graph, visited.copy(), max)
    if max_visit < max[0]:
        max_visit = max[0]
        max_node = node
    max[0] = 0
return (max_node, max_visit)
```

2)

a)

0000	0100	0100	1000	0100	1000
a	e	e	i	e	i

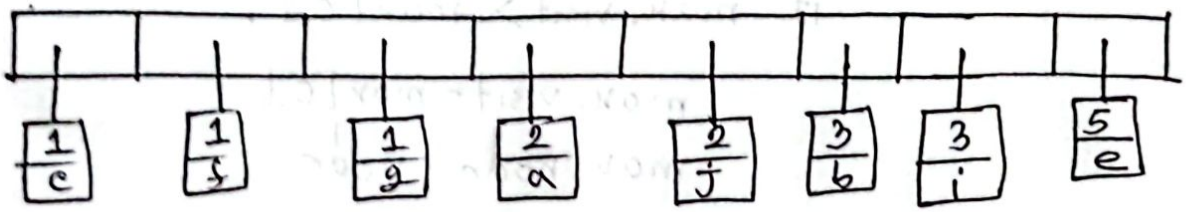
0100	1000	0001	0101	0001	0001
e	i	b	f	b	b

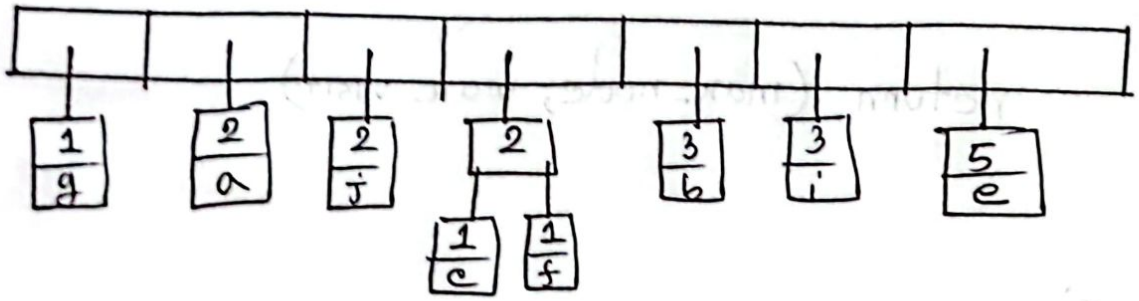
0110	0000	0100	1001	0010	1001
g	a	e	j	c	j

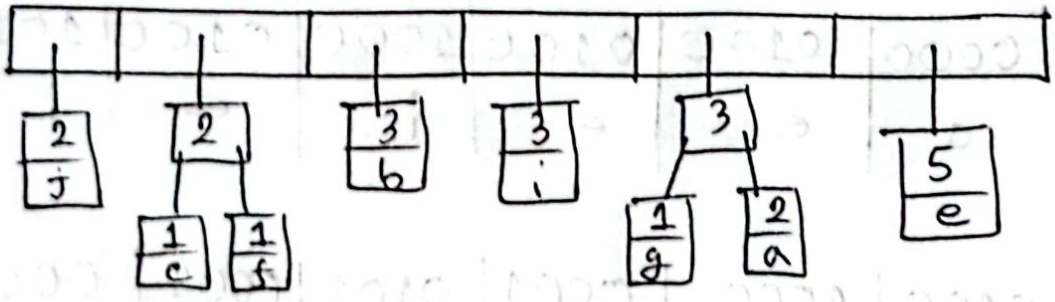
Message = aeeieieibfbbgaejcj

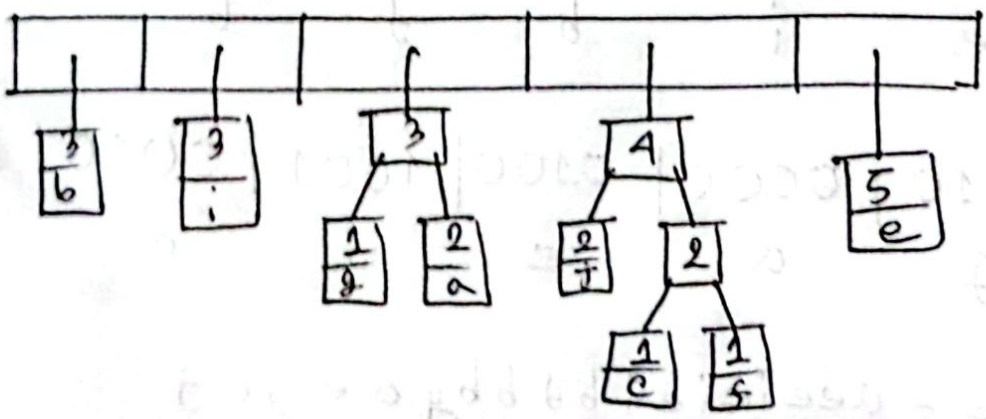
b) Frequency Table

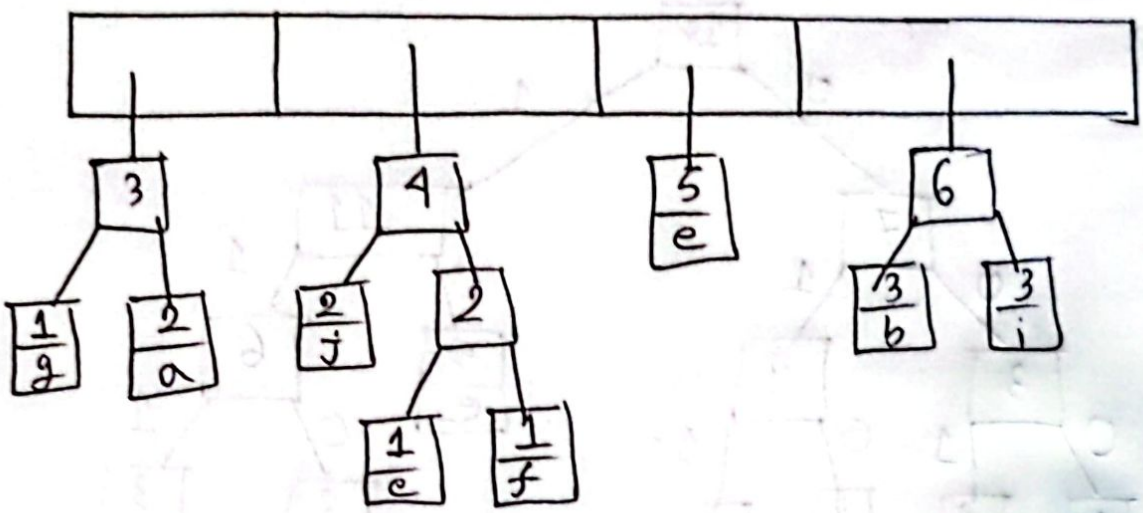
Character	c	f	g	a	j	b	i	e
Frequency	1	1	1	2	2	3	3	5

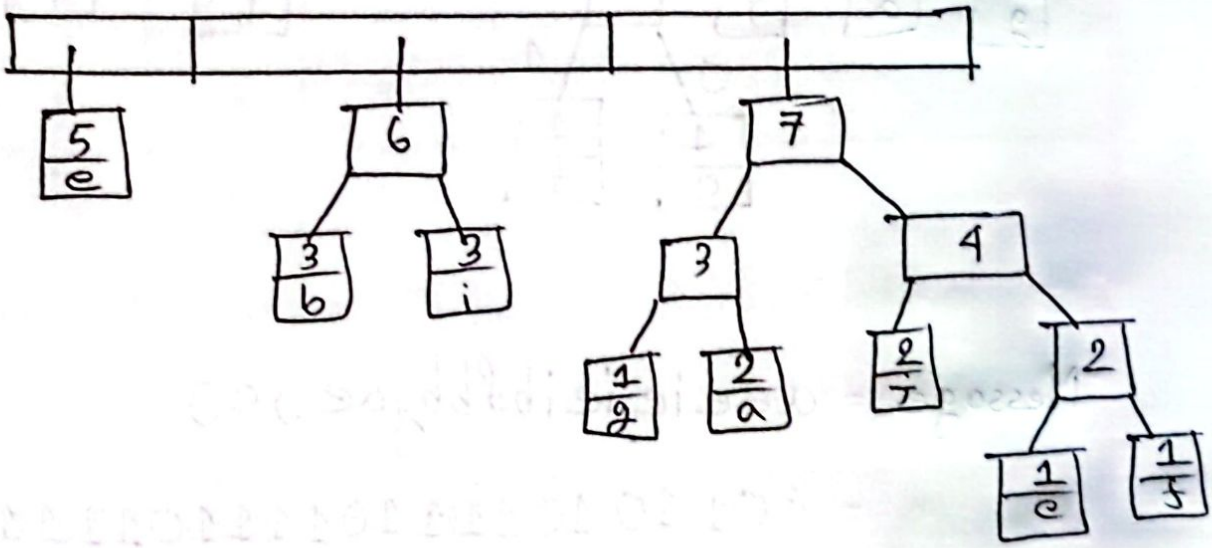


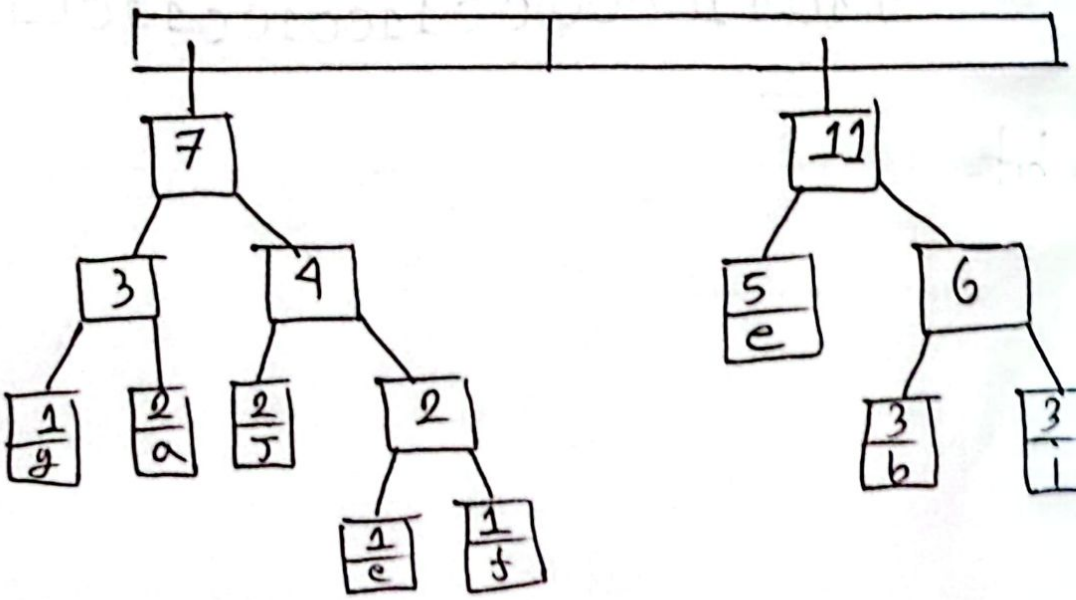


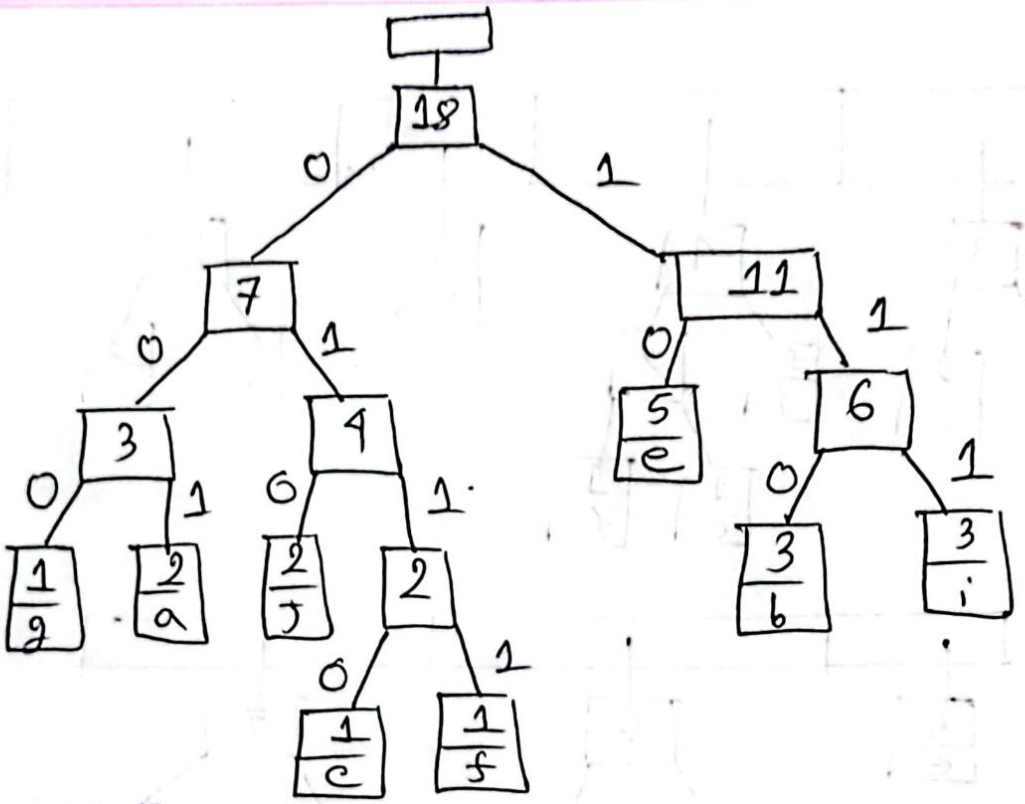





3
4
5
e
6
1
g
2
a
2
j
2
1
c
1
f
3
b
3
i


5
e
6
7
3
b
3
i
3
4
1
g
2
a
2
j
2
1
c
1
f


7
11
3
4
5
e
6
1
g
2
a
2
j
2
3
b
3
i
1
c
1
f



Message = aeeieieibfbbgaejcj

= 001 10 10111101111011111001111

1101100000011001001100 10

c) 51 bits

2) a) Leo's end early

b)

Faculty	Start	End	Picked	Reason
MZU	2	3	✓	
FGZ	3	6	✓	
RIM	5	7	X	
AGD	6	8	✓	
SBD	7	10	X	
MIBA	1	10	X	
MNR	10	11	✓	

Maximum 4 Class — MZU, FGZ, AGD, MNR

c)

Time	Start/End	People Needed	Count
1	+Miba	{Miba}	1
2	+ MZU	{Miba, MZU}	2
3	+FGZ, -MZU	{Miba, FGZ}	2
5	+ RIM	{Miba, FGZ, RIM}	3
6	+AGD, -FGZ	{Miba, AGD, RIM}	3
7	+ SBD, - RIM	{Miba, SBD, AGD}	3
8	- AGD	{Miba, SBD}	2
10	-SBD Miba, +MNR	{ MNR}	1
11	- MNR	0	0

Minimum number of students = 3

3)

a) $knapSack(n,W) = \begin{cases} knapSack(n-1,W) & \text{if } W_j > W \\ \max(V_n + knapsack(n-1, W-W_j), \\ \quad knapsack(n-1,W)) \end{cases}$

b)

Value	Size	Item	Maximum Available Space						
			0	1	2	3	4	5	
		0	0	0	0	0	0	0	
5	1	1	0	(5)	5	5	5	5	✓
7	2	2	0	5	7	(12)	12	12	✓
8	2	3	0	5	8	13	15	(20)	✓
9	3	4	0	5	8	13	15	20	
12	4	5	0	5	8	13	15	20	

$20 - 8 = 12 \rightarrow (8, 2)$ Max profit = 20

$12 - 7 = 5 \rightarrow (7, 2)$

$5 - 5 = 0 \rightarrow (5, 1)$

© $\partial(n, W, k) = \max(v_n + \partial(n-1, W-W_j, k),$
$v_n + \partial(n-1, \frac{W-W_j}{2}, k-1),$
$\partial(n-1, W, k))$

1

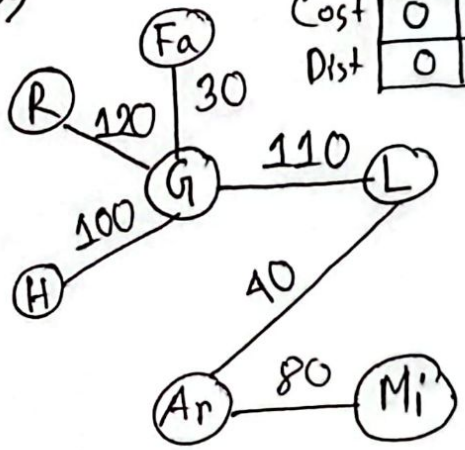
Edge	Weight
✓ (G-L)	110
✓ (Ar-Mi)	80
✓ (Fa-G)	30
✓ (Ar-L)	40
X (G-Mi)	50
Y (Fa-Mi)	60
X (Ar-Fa)	70
Y (Fa-L)	90
✓ (G-H)	100
✓ (G-R)	120
X (H-Mi)	150
X (Mi-R)	180

|V| = 7

|E| = 12

T = ∅ ✓ ✗ ✗ ✗ ✗ 6

d)



	R	G	Fa	H	L	Ar
Cost	0	120	150	240	230	270
Dist	0	40	50	65	85	105

Mi
350
135

b) 480

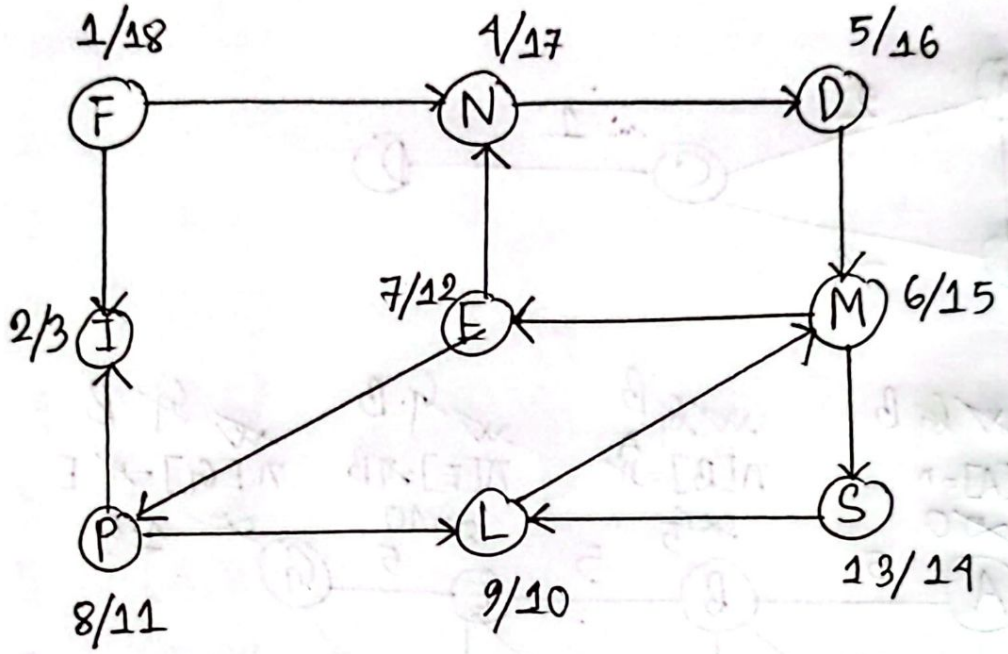
c) 460

~~Vertex Set~~ = ~~{G}, {L}, {Ar}, {Mi},~~
~~{Fa}, {R}, {H}~~
= ~~{G, L}, {Ar, Mi}~~ ~~{Fa, G, L}~~
= ~~{Ar, Mi, Fa, G, L}~~
= ~~{Ar, Mi, Fa, G, H, L}~~
= {Ar, Mi, Fa, G, H, L, R}

a)

Fall 2023 (B)

1) a)

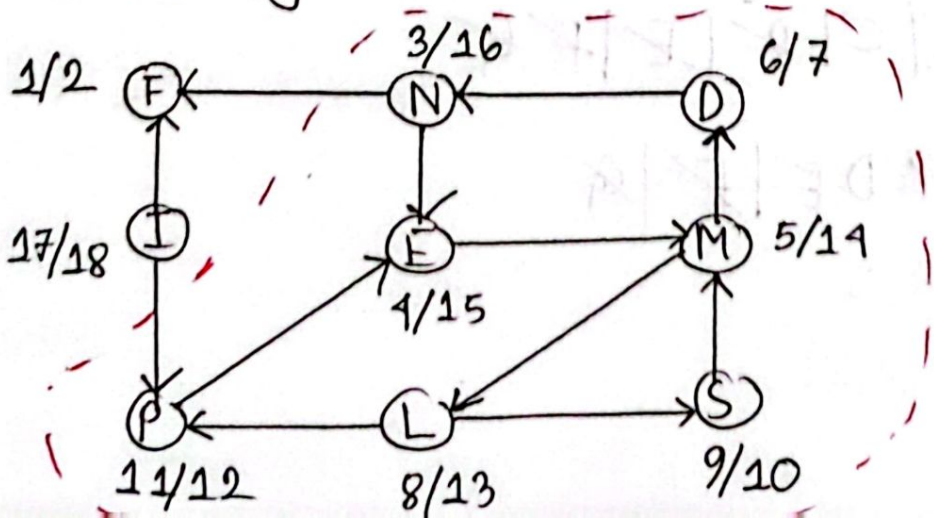


Node	D	E	F	I	M	S	L	P	N
Start	5	7	1	2	6	13	9	8	4
End	16	12	18	3	15	14	10	11	17

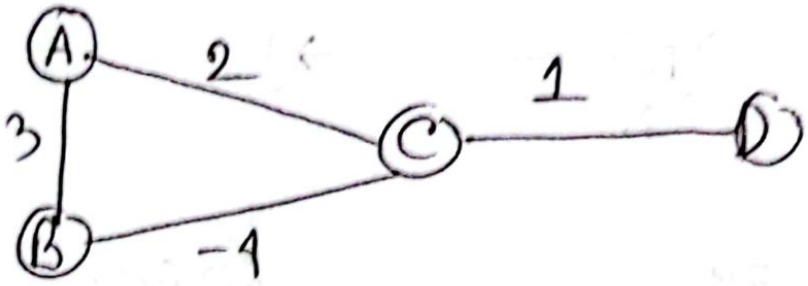
b) i) Korasaju's Strongly Connected Component Algorithm

ii)

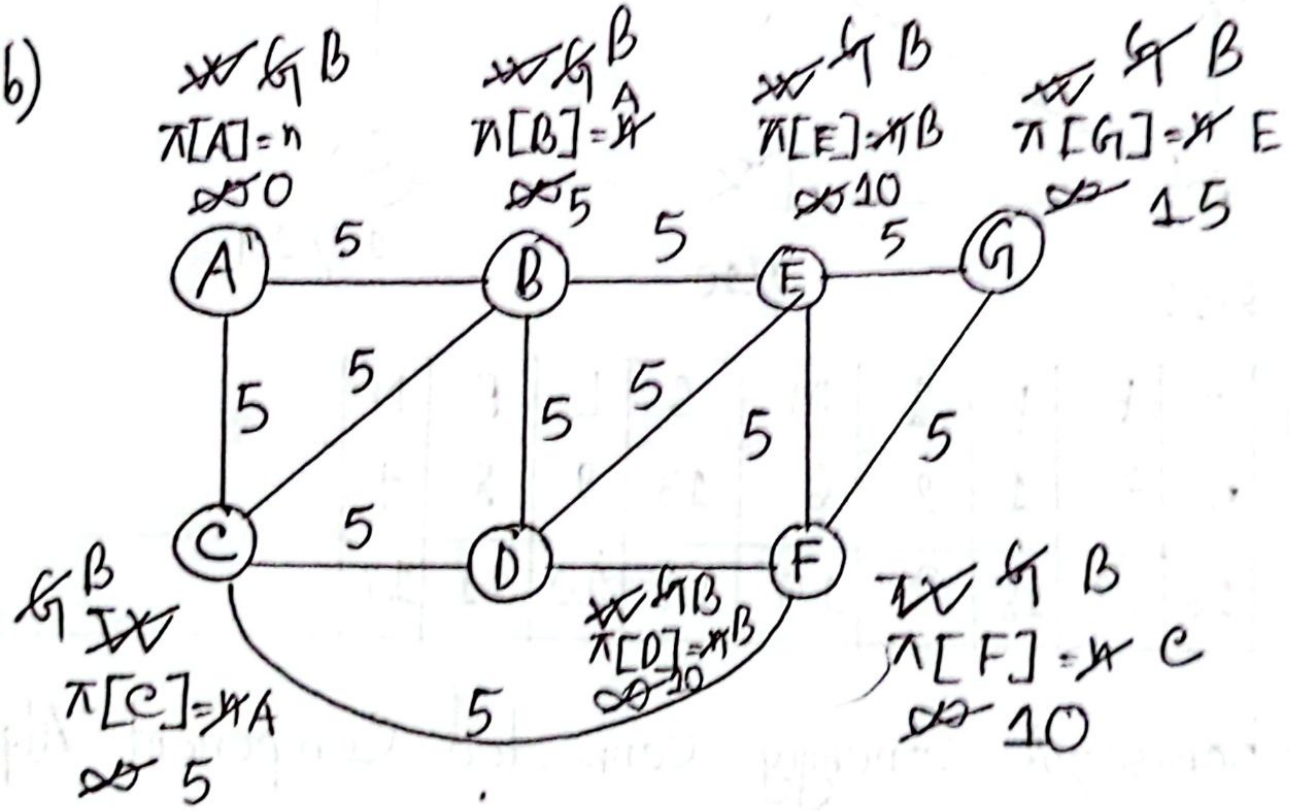
Decreasing order: F → N → D → M → S → E → P → L → I



2) a) False



b)



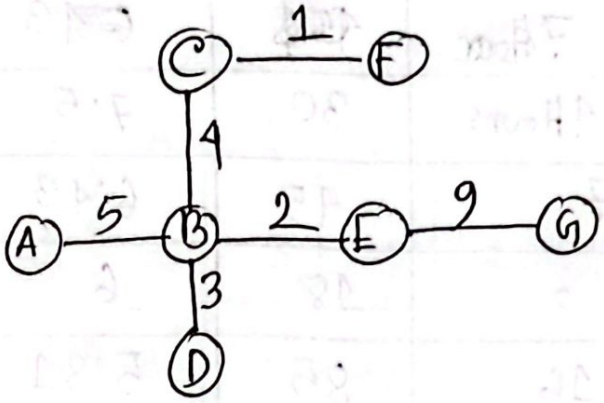
Q = ∅ ~~A B C D E F G~~

U = ~~A~~ | ~~B~~ | ~~C~~ | ~~D~~ | ~~E~~ | ~~F~~ ~~G~~

V = ~~B, C~~ | ~~D E~~ | ~~F~~ | ~~G~~

©

Weight	Edge
✓ 1	(C, F)
✓ 2	(B, E)
✓ 3	(B, D)
✓ 4	(B, C)
x 5	(C, D)
✓ 5	(A, B)
x 6	(D, E)
x 6	(D, F)
✓ 9	(E, G)
x 13	(A, C)



i) Explorer will choose ticked bridges

ii) 24

$|V| = 7$

$|E| = 10$

$T = \cancel{0}\ \cancel{1}\ \cancel{2}\ \cancel{3}\ \cancel{4}\ \cancel{5}\ 6$

Verten Set = ~~{A}~~, ~~{B}~~, ~~{C}~~, ~~{D}~~, ~~{E}~~, ~~{F}~~, ~~{G}~~

= ~~{C, F}~~ ~~{B, E}~~ ~~{B, D, E}~~

= ~~{B, C, D, E, F}~~ ~~{A, B, C, D, E, F}~~

= {A, B, C, D, E, F, G}

3) a) Fractional Knapsack

Time Limit = 20 Hours

Game	Time	Prize	Value Index	Choosen	Prize
~~HOD~~	~~7 Hour~~	~~45$~~	~~6.43~~		
VC	4 Hours	30	7.5	4	30
HOD	7	45	6.43	7	43
FC	3	18	6	3	18
RR	16	85	5.31	6	31.86
GTA	20	95	4.75	0	0
DMC	11	50	4.55	0	0

124.86$

b) i) Encoding table

Character	Encoded	Character	Encoded	Character	Encoded
U	000	r	1001		
t	001	e	101		
B	0100	sp	110		
l	0101	a	1110		
o	0110	c	1111		
h	0111				
s	1000				

ii) 0111|101|110|1001|101|1000|1111|000|101
0100|1001|1110|1111|000|0101|1110|110|001|0110|110|001|
Bracula to the rescue

3) a) seq(6, 4)

b)

			G	A	T	C
		0	1	2	3	4
	0	0	0	0	0	0
C	1	0	-1	-1	-1	1
T	2	0	-1	-2	0	0
G	3	0	0	-1	-1	-2
A	4	0	-1	0	-1	-2
G	5	0	0	-1	-2	-3
C	6	0	-1	-2	-3	-2

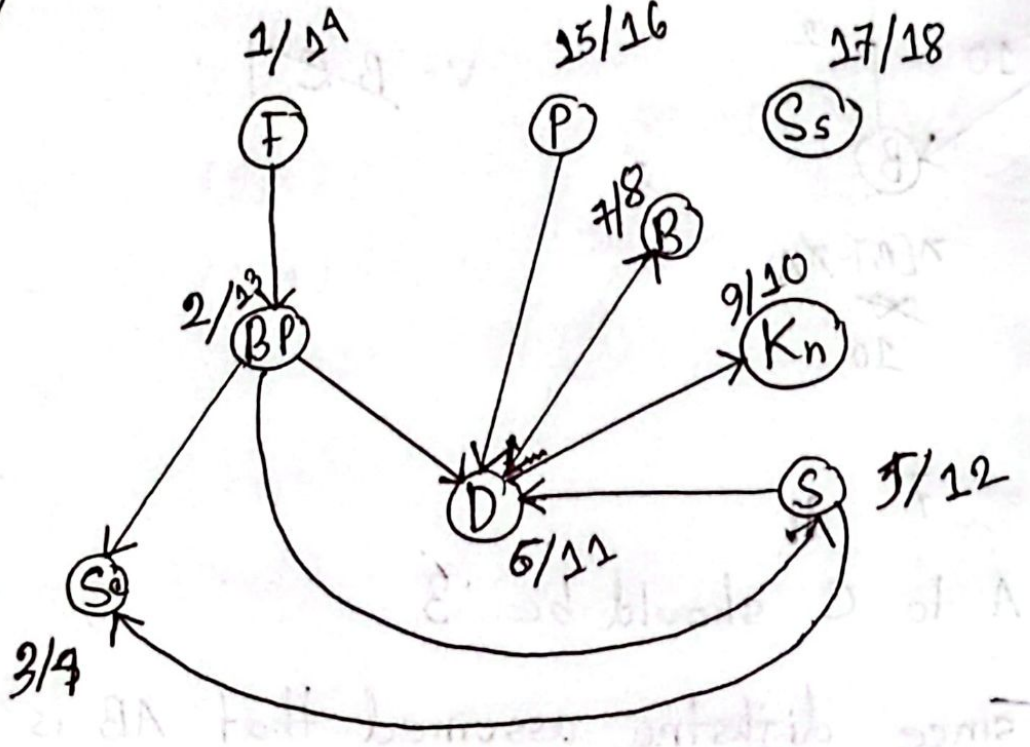
c) O(NM)

d) To reduce time complexity we just need to track the previous row. So, it can be done in O(min(N,M))

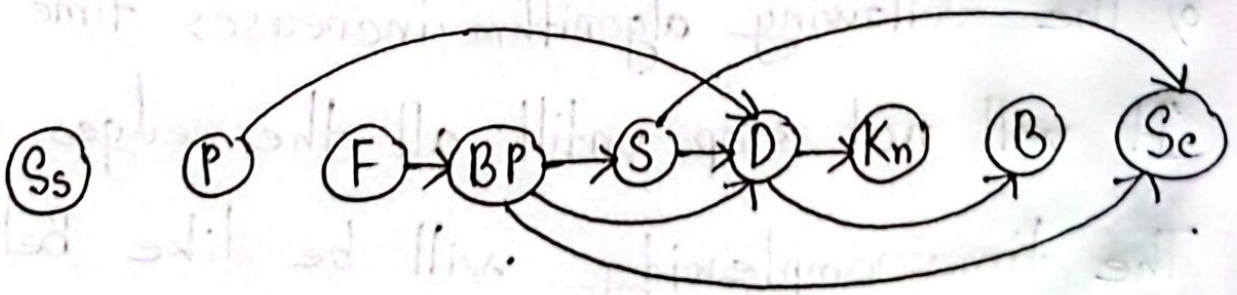
Spring 24 (B)

1) a) Topological Sort Algorithm

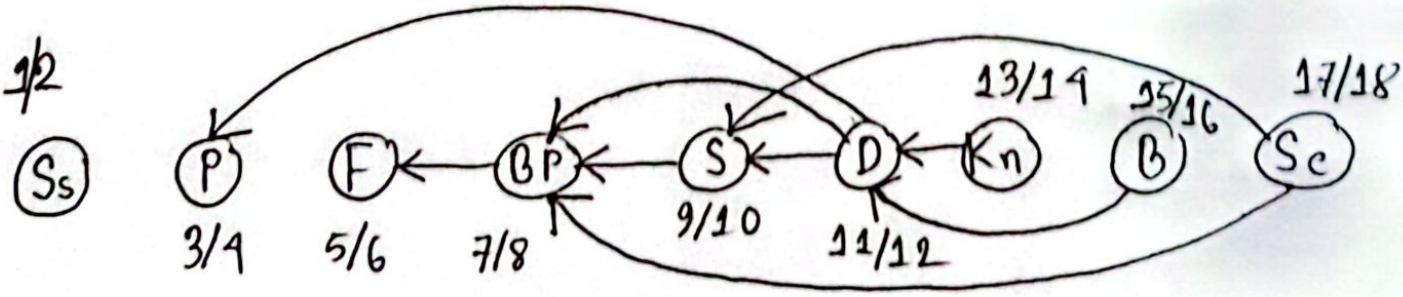
b)



end	18	16	14	13	12	11	10	8	4	
Node	Ss	P	F	BP	S	D	Kn	B	Sc	

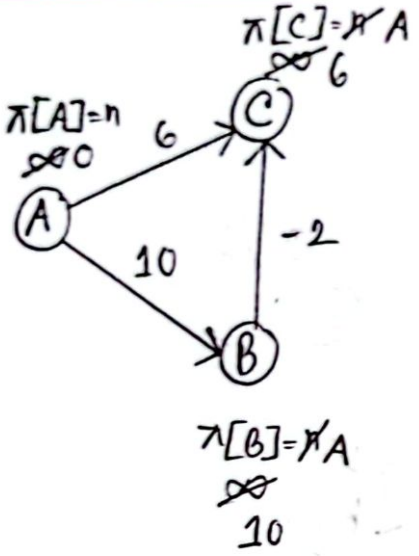


c)



All are SCC

2) a)



PQ = A B C

U = A B C B

V = B C

Let's, BC = -7

So, A to C should be 3

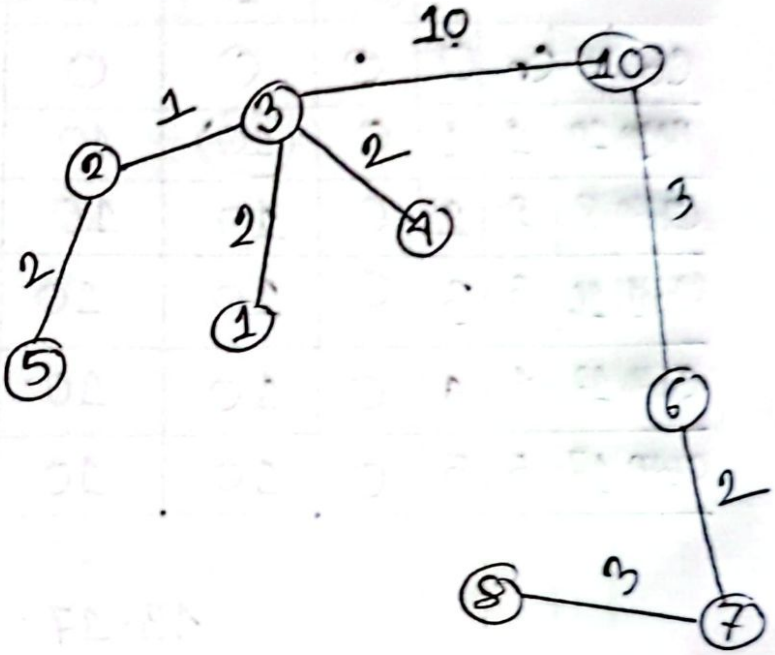
But since dijkstra assumed that AB is 10 and AC is 6. Dijkstra fails.

b) The following algorithm increases time complexity. It will not stop untill all the edges are visited. The timecomplexity will be like Bellman ford

$O(V * E)$

c)

Weight	Edge
✓ 1	(2,3)
✓ 2	(2,5)
✓ 2	(3,1)
✓ 2	(3,4)
✓ 2	(6,7)
× 3	(2,1)
× 3	(4,1)
✓ 3	(10,6)
✓ 3	(8,7)
× 4	(5,1)
× 7	(6,8)
✓ 9	(3,10)



|V| = ~~10~~ 9

|E| = 12

T = ~~∅ 1 2 3 4 5 6 7 8~~

Vertex Set = {1}, ~~{2}, {3}~~, {4}, {5}, ~~{6}, {7}~~, {8}, ~~{9}, {10}~~

= ~~{2,3}~~ ~~{2,3,5}~~ ~~{1,2,3,5}~~ ~~{1,2,3,4,5}~~ ~~{6,7}~~ ~~{6,7,10}~~ ~~{6,7,8,10}~~ {1,2,3,4,5,6,7,8,10}

3) a)

				0	1	2	3	4	5	6	7	8	9
Co	V	Cr	0	Q	0	0	0	0	0	0	0	0	0
PHY112	10	1	1	0	(10)	10	10	10	10	10	10	10	10
CSE330	8	2	2	0	10	10	18	18	18	18	18	18	18
CSE331	16	3	3	0	10	10	18	(26)	26	34	34	34	34
CSE370	13	4	4	0	10	10	18	26	26	34	34	34	34
CSE499	17	5	5	0	10	10	18	26	26	34	34	35	(43)

43-17 = 26 CSE499

26-16 = 10 CSE331

10-10 = 0 PHY112

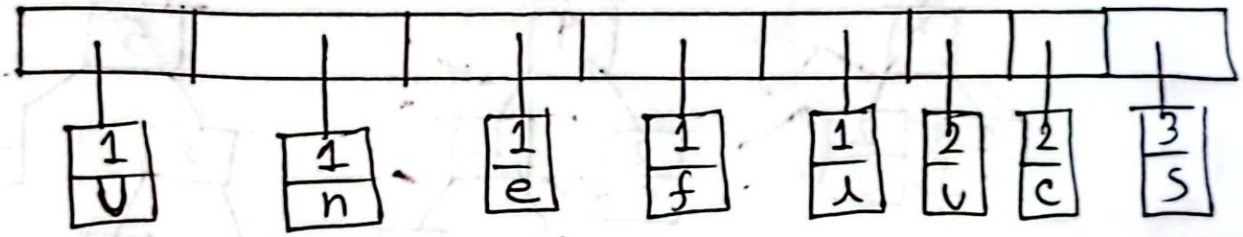
b)

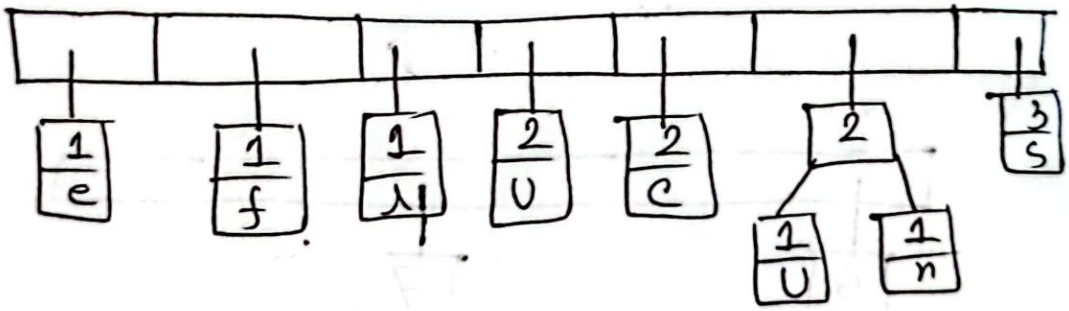
Course	Value	Credit	Value Index	Choosen	
PHY112	10	1	10	1	
CSE331	16	3	5·3	3	
CSE330	8	2	4	2	
CSE399	17	5	3·4	0	
CSE370	13	4	3·25	0	

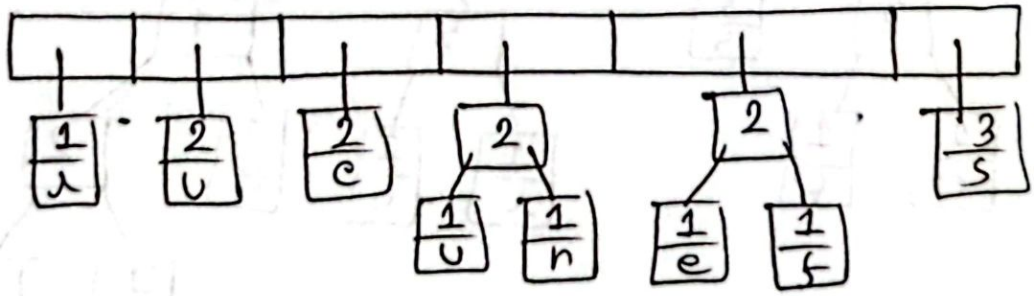
6 < 9 Unsuccessful

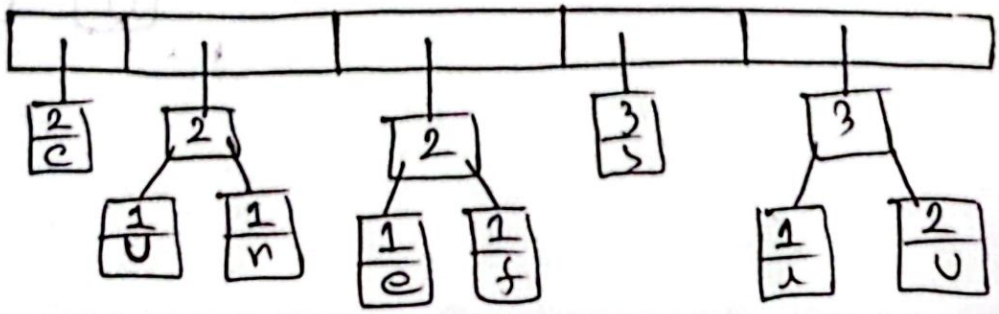
© Frequency Table

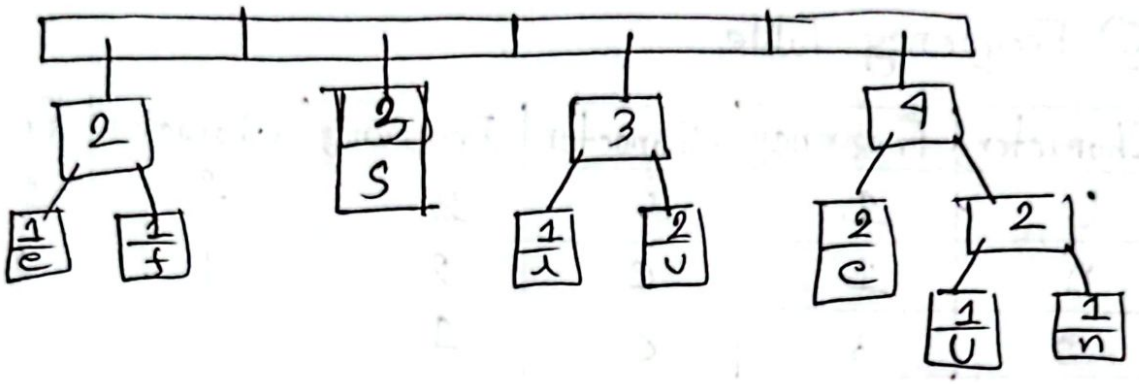
Character	Frequency	Character	Frequency	Character	Frequency
U	1	v	2	f	1
n	1	c	2	l	1
s	3	e	1		

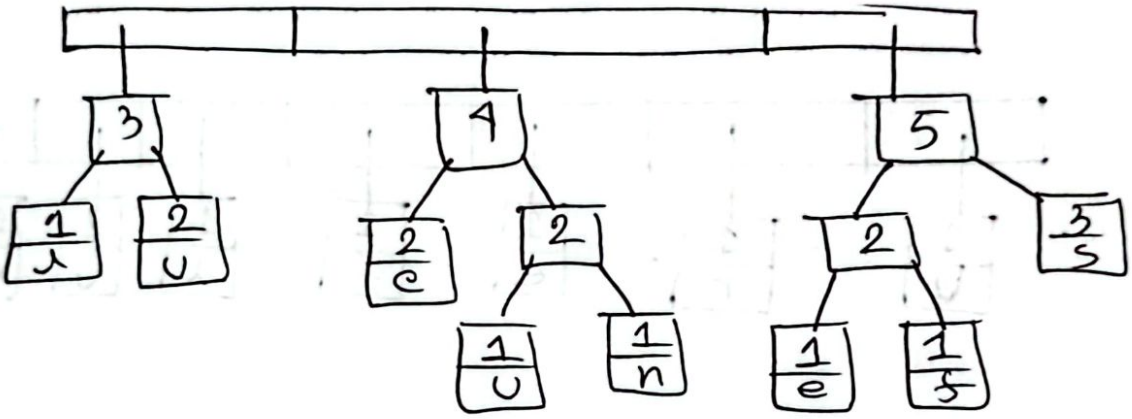


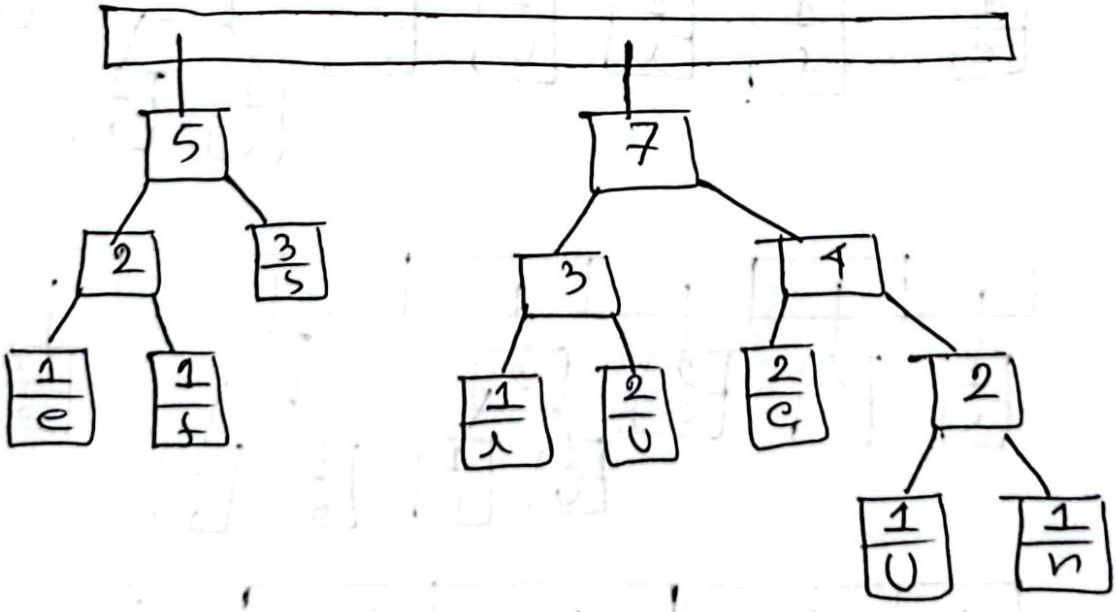


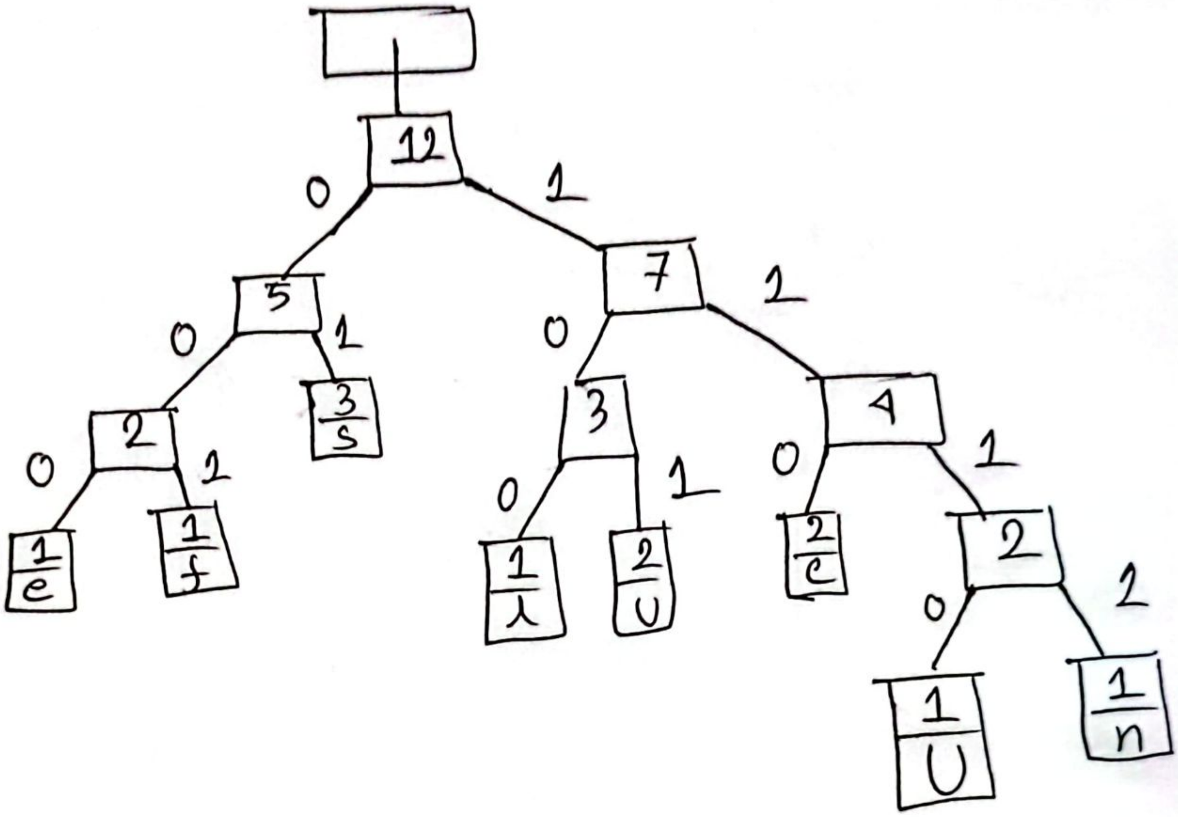





2
3
S
3
4
e
f
λ
U
e
2
U
n


3
4
5
λ
U
e
2
2
S
U
n
e
f


5
7
2
3
S
3
4
e
f
λ
U
e
2
U
n



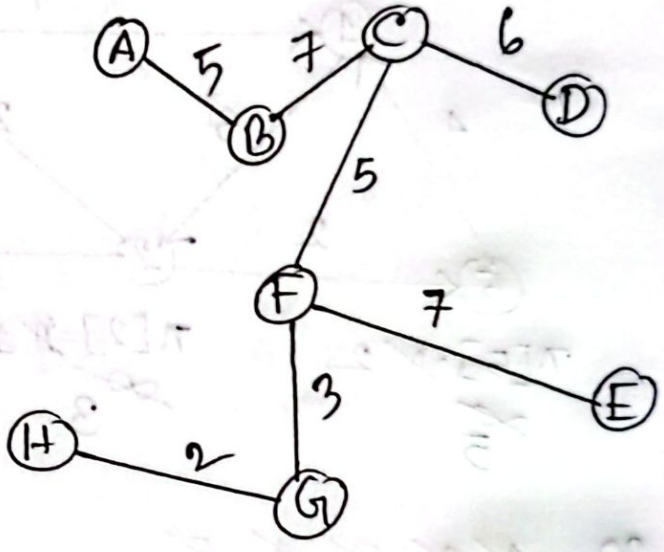
Encoded Message

1 1 10 1 1 1 1 01 101 110 1100000101

001 101 100

a) Kruskal's MST

Weight	Edge
✓ 2	(H,G)
✓ 3	(F,G)
✓ 5	(A,B)
✓ 5	(F,C)
✓ 6	(C,D)
✓ 7	(B,C)
✓ ~~7~~	~~(F,E)~~
x 8	(D,E)
x 9	(A,H)
x 10	(B,H)
x 12	(F,D)



b) Minimal cost = 35. Greedy.

c) It will add (A,H) 9. Total cost will be = 41

|M| = 8

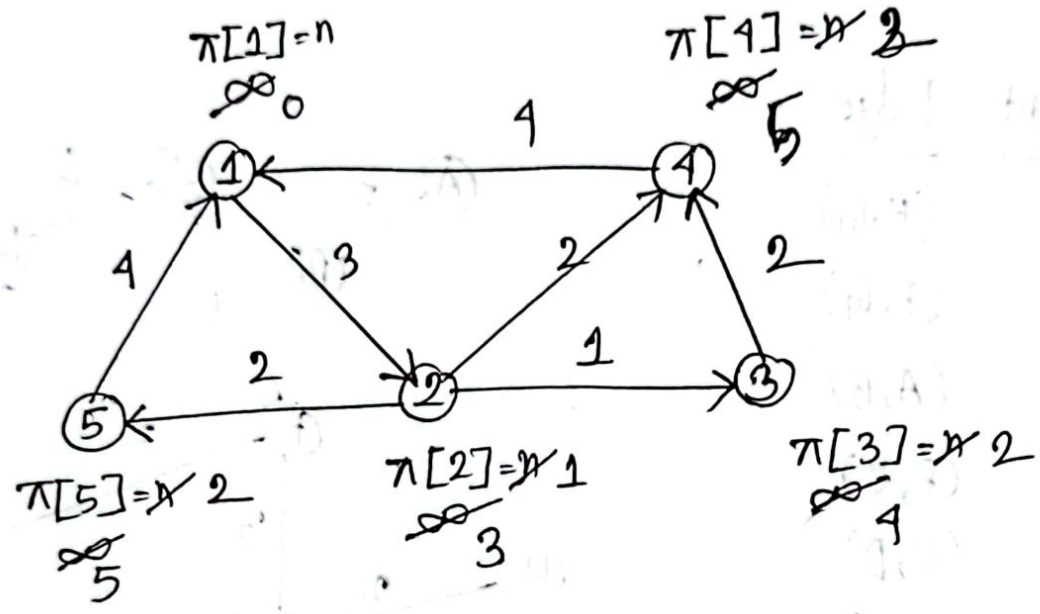
|E| = 11

T = ∅ ✓ ✓ ✓ x x ✓ 7

Vertex Set = {~~A~~}, {~~B~~}, {~~C~~}, {D}, {E}, {~~F~~}, {~~G~~}, {~~H~~},

= {~~H,G~~} {~~F,G,H~~} {~~A,B~~} {~~C,F,G,H~~} {~~C,D,F,G,H~~} {~~A,B,C,D,F,G,H~~} {A,B,C,D,E,F,G,H}

(a)

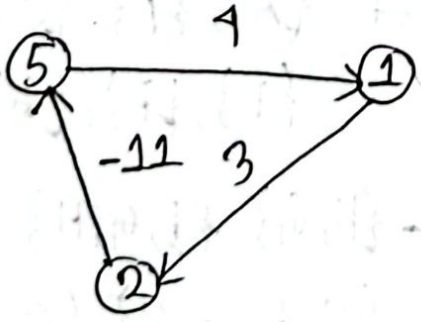


PQ = ~~1 2 3 4 5~~

U = ~~1 2 3 5 4~~

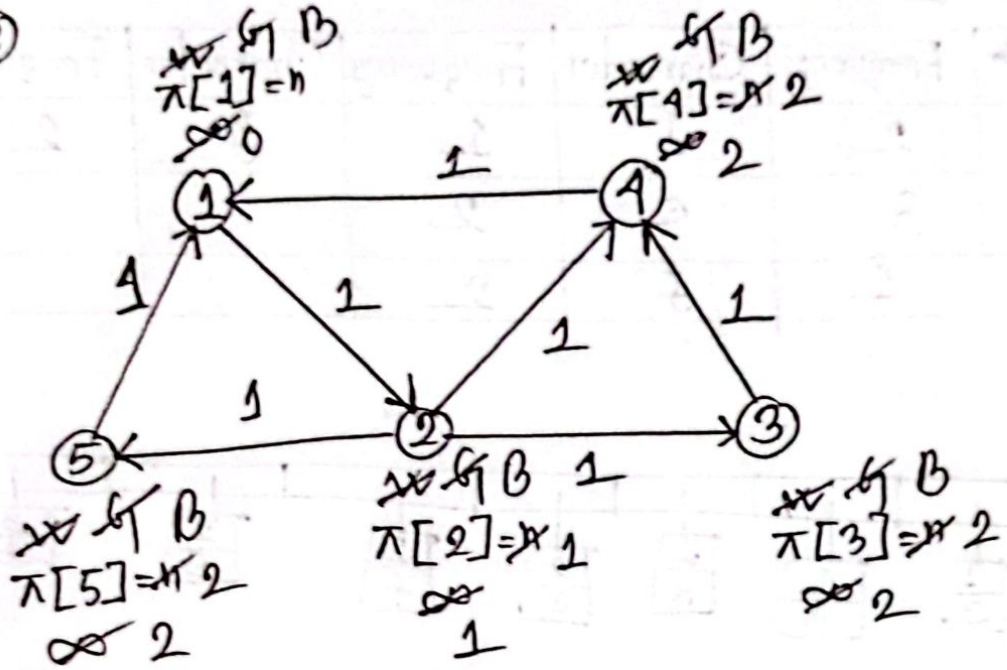
V = ~~2~~ | ~~5 4 3~~ | ~~4~~ 1 |

(b)



Negative weighted cycle
dijkstra won't work

(c)



Q = ∅ 1 2 4 3 5

U = 1 2 4 3 5

V = 2 4 1 3 5

(3)

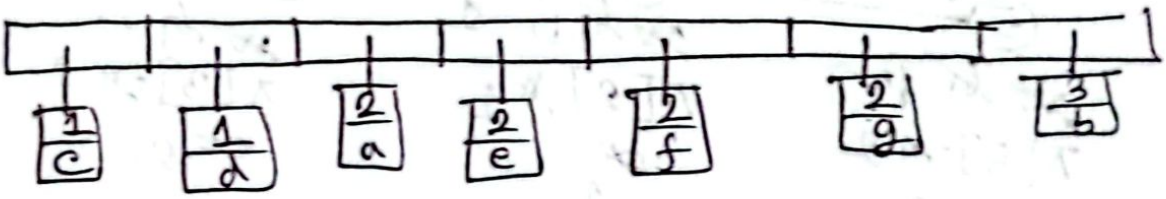
a b c d e f g

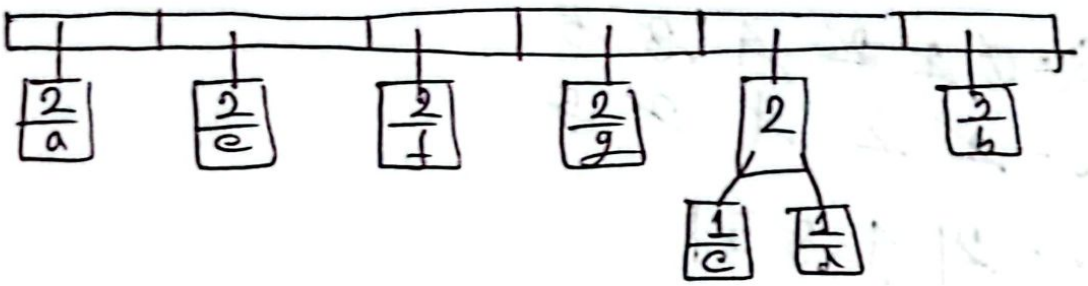
		0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0
b	1	0	0	(1)	1	1	1	1	1
f	2	0	0	1	1	1	1	2	2
e	3	0	0	1	1	1	(1)	2	2
g	4	0	0	1	1	1	2	2	(3)
b	5	0	0	1	1	1	2	2	3
a	6	0	1	1	1	1	2	2	3

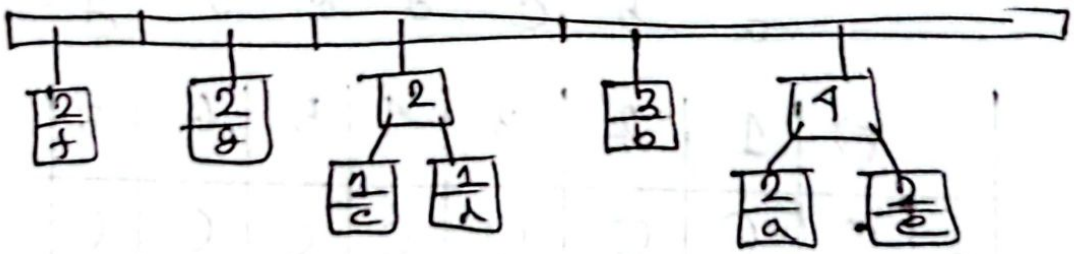
beg

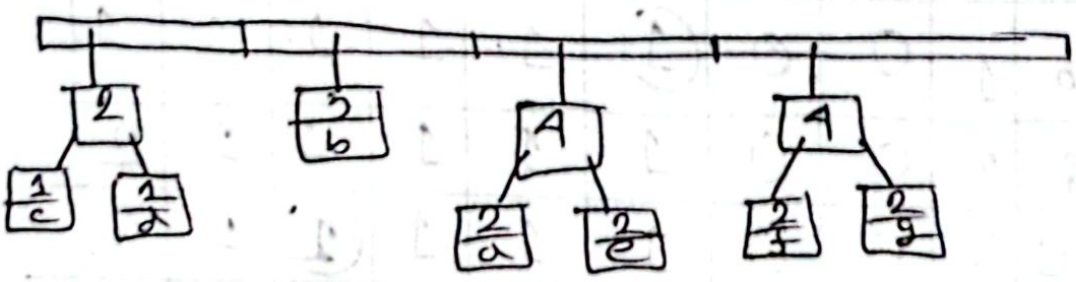
(b) Frequency Table

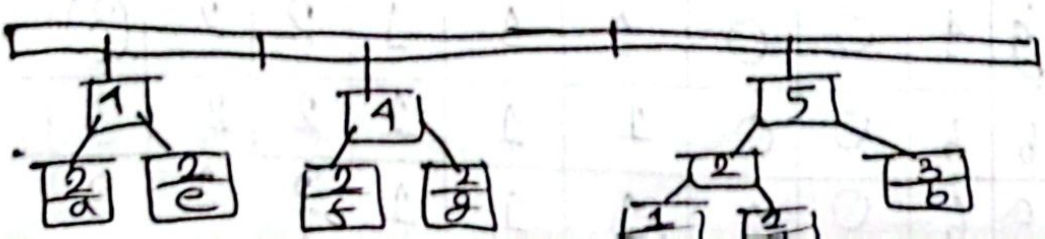
Character	Frequency	Character	Frequency	Character	Frequency
a	2	d	1	g	2
b	3	e	2		
c	1	f	2		

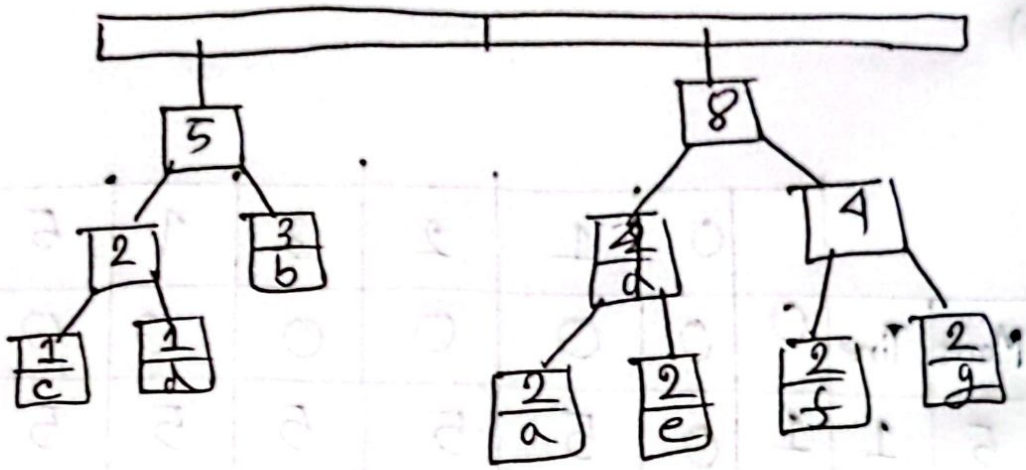


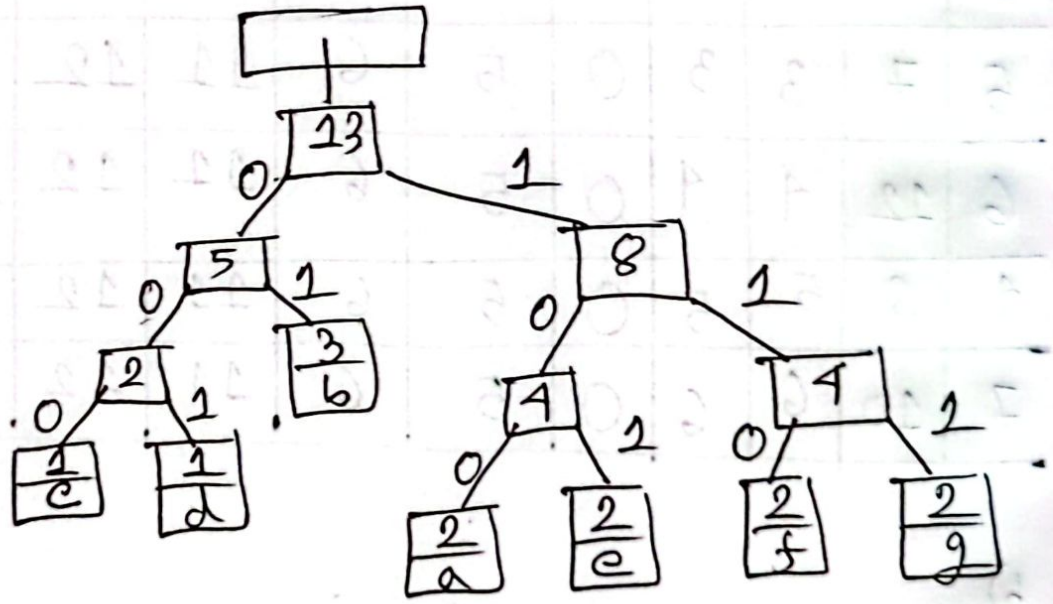












Encoded message

100010000011011011101110101111001100

Total Bits = 36

A) a)

				0	1	2	3	4	5	6
Question	Marks	Time	0	0	0	0	0	0	0	0
9	5	1	1	0	5	5	5	5	5	5
8	6	2	2	0	5	6	11	11	11	11
5	7	3	3	0	5	6	11	12	13	18
6	12	4	4	0	5	6	11	12	17	18
4	8	5	5	0	5	6	11	12	17	18
7	10	6	6	0	5	6	11	12	17	18

b)

Question	Marks	Time	Value Index	Choosen	Profit
9	5	1	5	1	5
8	6	2	3	2	6
6	12	4	3	3	9
5	7	3	2.3	0	
7	10	6	1.67	0	
4	8	5	1.6	0	

20

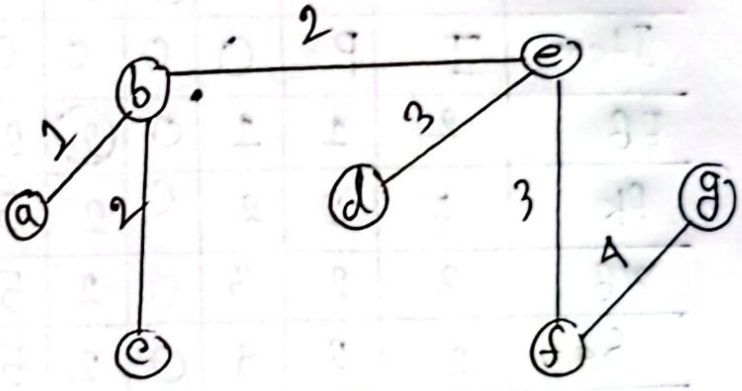
Yes scored more

c) Memoization

Spring-23 (B)

1)a)

Weight	Edge
✓ 1	(a,b)
✓ 2	(b,c)
✓ 2	(b,e)
✗ 3	(a,c)
✓ 3	(d,e)
✓ 3	(e,f)
✓ 4	(f,g)
x 5	(e,g)
x 6	(b,d)
x 6	(d,f)
x 6	(c,f)
x 7	(c,d)



Cost = 15

$|V| = 7$

$|E| = 12$

$T = \emptyset$ ~~1 2 3 4 5~~ 6

Vertex Set = ~~{a}, {b}, {c}, {d}, {e}, {f}, {g}~~

= ~~{a,b}~~ ~~{a,b,c}~~ ~~{a,b,c,e}~~ ~~{a,b,c,d,e}~~ ~~{a,b,c,d,e,f}~~ {a,b,c,d,e,f,g}

2)

a)

				0	1	2	3	4	5	6	7	8	9	
Item	I	P	O	0	0	0	0	0	0	0	0	0	0	
EB	2	1	1	0	(2)	2	2	2	2	2	2	2	2	
PK	5	2	2	0	2	5	(7)	7	7	7	7	7	7	
WB	3	2	3	0	2	5	7	8	(10)	10	10	10	10	
Sg	2	2	4	0	2	5	7	8	10	10	12	12	12	
NV	4	3	5	0	2	5	7	8	10	11	12	(14)	14	✓
GB	1	8	6	0	2	5	7	8	10	12	12	14	14	

14 - 4 = 10 Night Vision

10 - 3 = 7 Waterproof Boots

7 - 5 = 2 Painkiller

2 - 2 = 0 Energy bar

b) Exponential $O(2^N)$

3) a) m

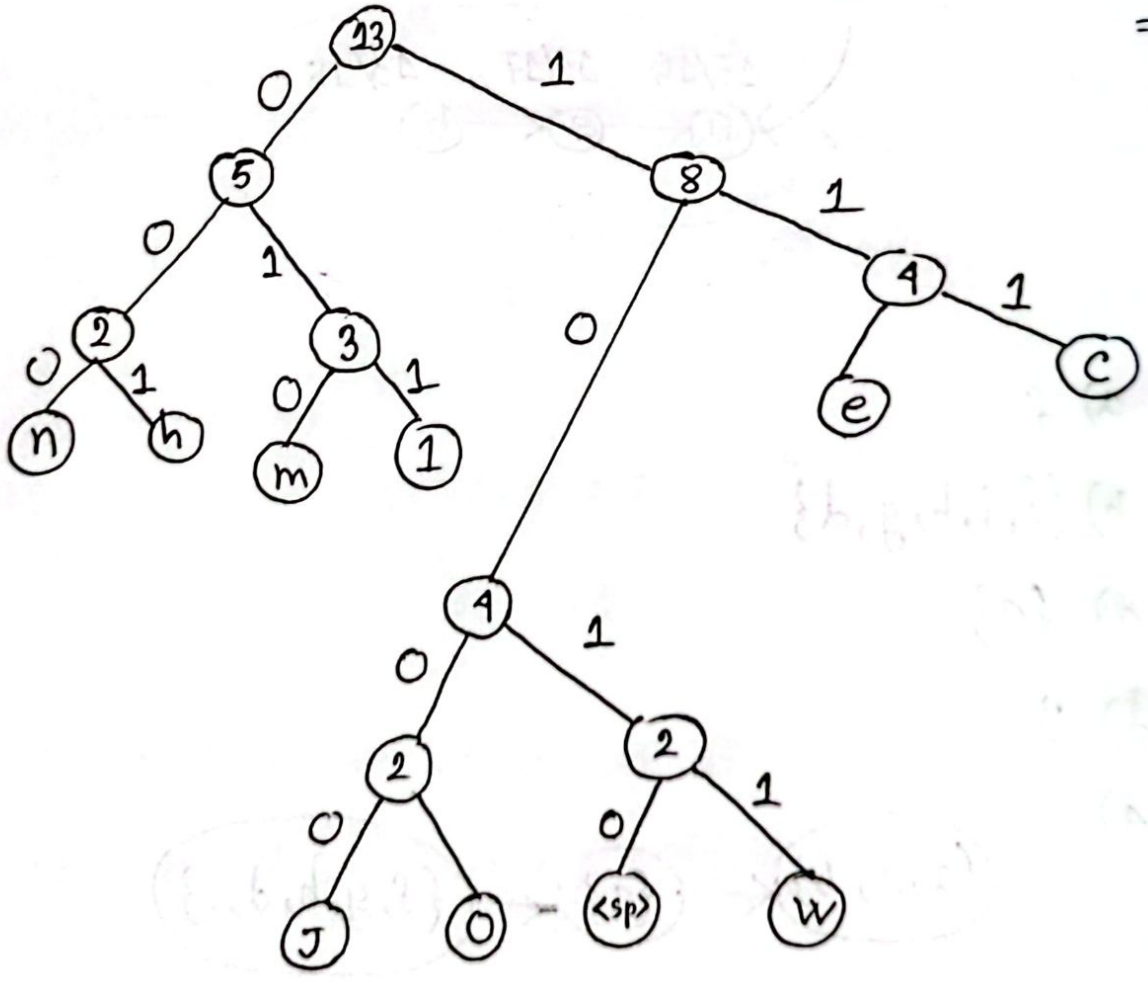
c) e - 2
c - 2

d) Welcome John.

W-1	m-1	<sp> - 1
e-2	J-1	
l-1	h-1	
c-1	n-1	
o-2	.-1	

Tom's required bit = ~~1+1~~
$= \log_2(11)$
$= 4$

b)



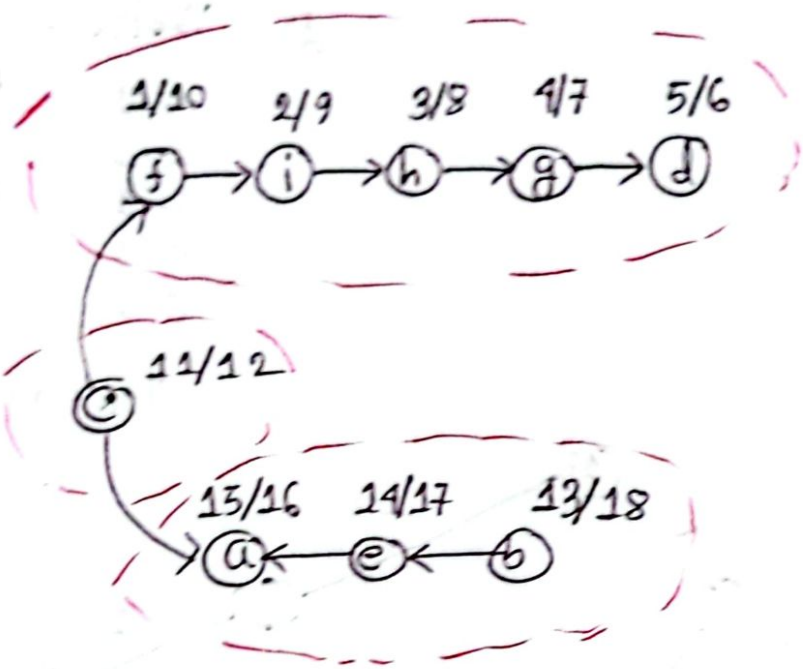
e) 1011100111111001010110101010000100100 1000

13 bits

Mine is better

Tom's = 12 * 4 = 48 bits

4)



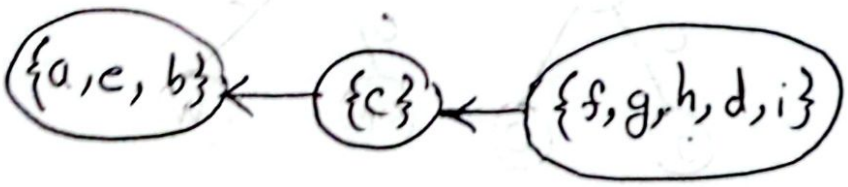
a) f

b) {f, i, h, g, d}

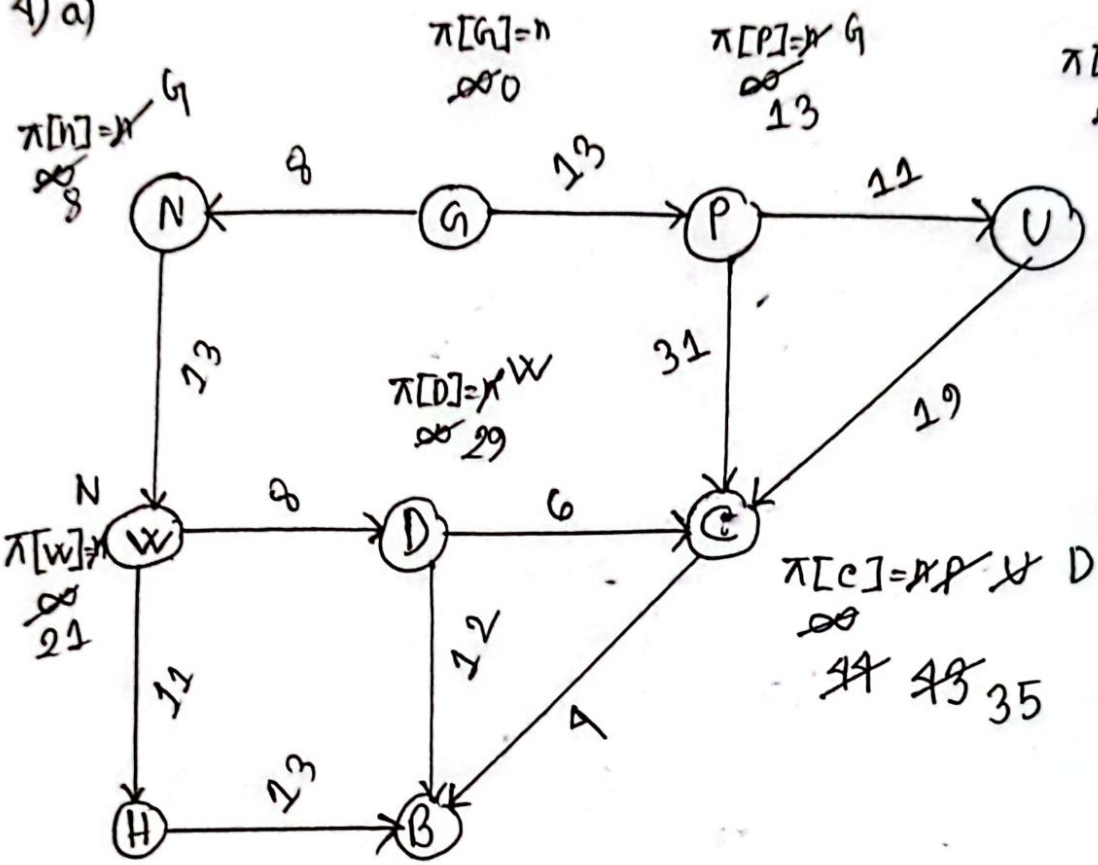
c) {c}

d) 3

4)



4) a)



PQ = ~~G~~ ~~N~~ ~~P~~ ~~U~~ ~~W~~ ~~D~~ ~~C~~ ~~H~~ ~~B~~

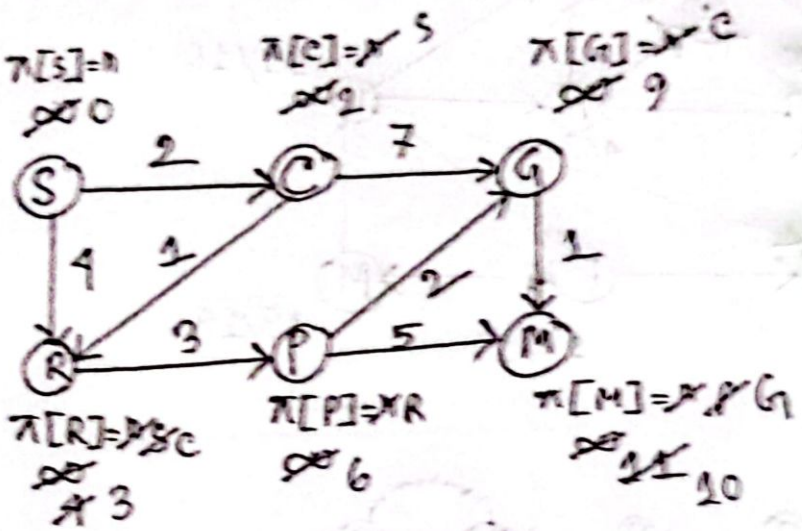
U = ~~G~~ | ~~N~~ | ~~P~~ | ~~W~~ | ~~U~~ | ~~D~~ | ~~H~~ | ~~C~~ | ~~B~~

V = ~~N,P~~ | ~~W~~ | ~~C U~~ | ~~D H~~ | ~~C~~ | ~~B C~~ | ~~B~~ |

b) Run dijkstra from G in graph
Run " " B in reverse graph
Let, w(u,v) is new road wit
$$\min(d\_G(u) + w(u,v) + d\_B(v))$$

Fall 22 (B)

1) a)

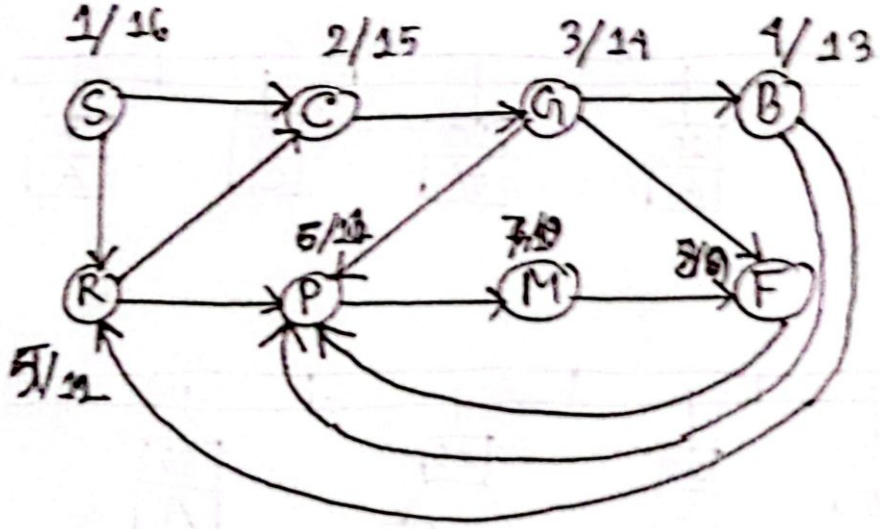


PQ = S R C P G M

U = S C R P G M

V = C R | G R | P | M G |

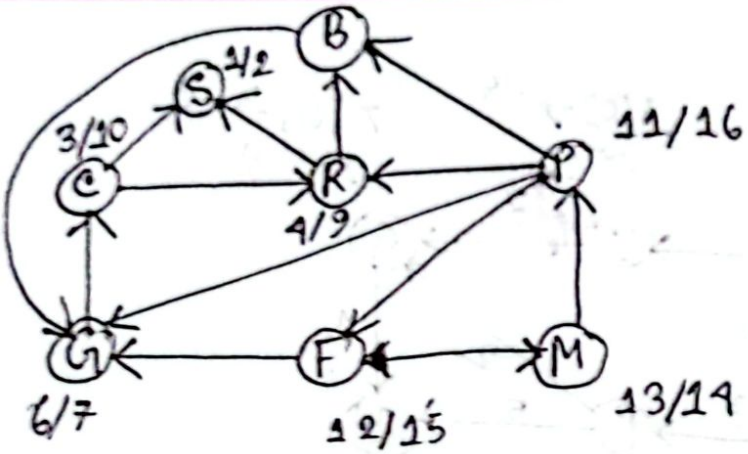
(b)



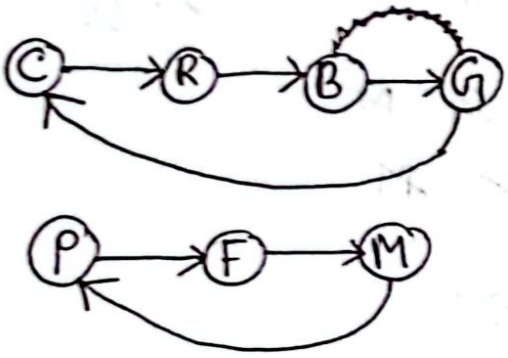
Decreasing order:

S → C → G → B → R → P → M → F

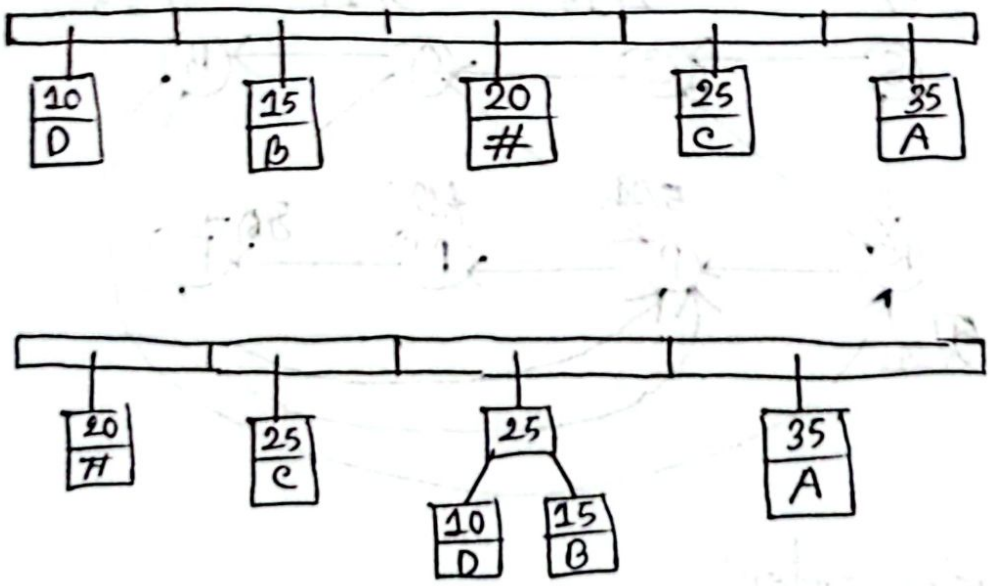
5/8

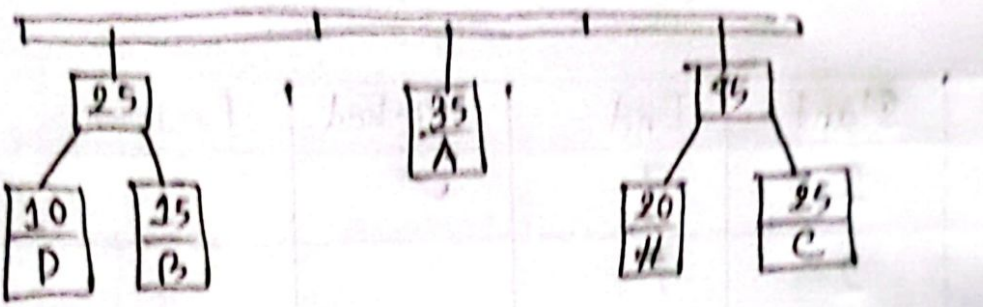

B
S
2/2
3/10
C
R
4/9
P
11/16
G
6/7
F
12/15
M
13/14

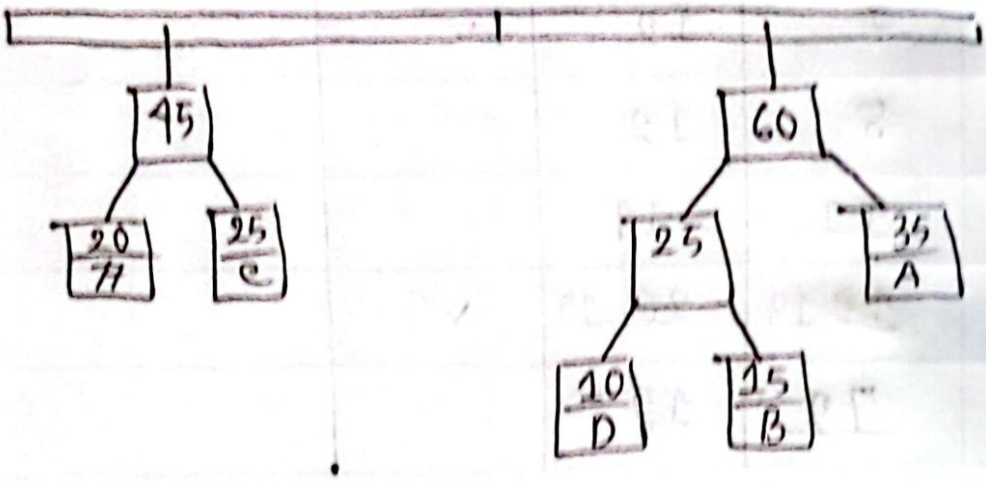
Group:


C
R
B
G
P
F
M

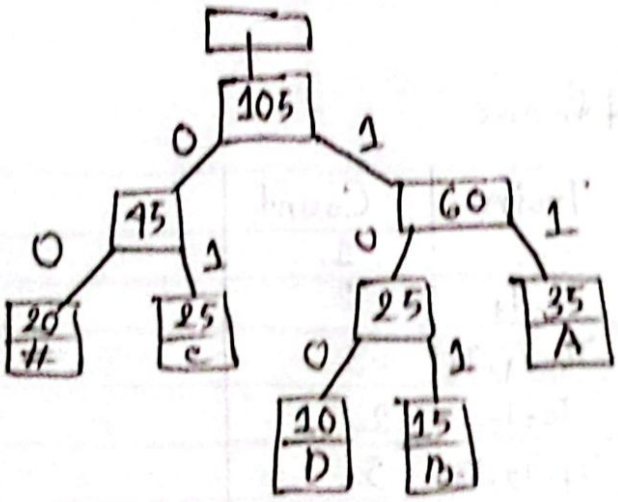
2) a)


10
D
15
B
20
#
25
C
35
A
20
#
25
C
25
10
D
15
B
35
A





1)



Symbol	Codeword
A	11
B	101
C	01
D	100
H	00

100 01 01 11 00 101 ∅

D C C A H B

b) i)

Trains	Start	End	Picked	Reason
T_5	1	7	✓	
T_4	2	7		
T_2	6	9		
T_7	7	12	✓	
T_1	8	12		
T_3	11	14		
T_8	~~12~~ 13	~~20~~ 19	✓	
T_6	~~7~~ 12	~~19~~ 20		

Maximum 3 trains

ii) Minimum 3 Platforms

Time	Strart/End	Train	Count
1	$+T_5$	T_5	1
2	$+T_4$	T_5, T_4	2
6	$+T_2$	T_5, T_4, T_2	3
7	$-T_5; T_4, +T_7$	T_7, T_2	2
8	$+T_1$	T_7, T_2, T_1	3
9	$-T_2$	T_7, T_1	2
11	$+T_3$	T_7, T_1, T_3	3
12	$-T_7 - T_1, +T_6$	T_6, T_3	2
13	$+T_8$	T_6, T_3, T_8	3
14	$-T_3$	T_6, T_8	2
19	$-T_8$	T_6	1
20	$-T_6$	()	0

3) i)

				0	1	2	3	4	5	6	7	8
Items	P	W	0	0	0	0	0	0	0	0	0	0
Book	4	1	1	0	(4)	4	4	4	4	4	4	4
Jewelry	5	3	2	0	4	4	5	(9)	9	9	9	9
Painting	5	4	3	0	4	4	5	9	9	9	10	(14)
Sculpture	9	5	4	0	4	4	5	9	9	13	13	14
Mummy	6	12	5	0	4	4	5	9	9	13	13	14

14 - 5 = 9 Painting

9 - 5 = 4 Jewelry

4 - 4 = 0 Book

ii) No, fractional knapsack fails.

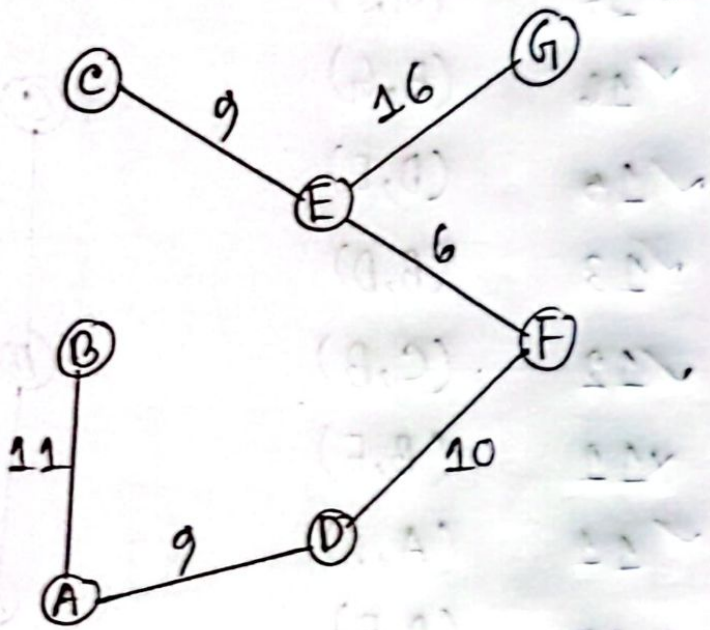
Items	Profit	Weight	Value Index	Choosen	Profit after Picked
Book	4	1	4	1	4
Sculpture	9	5	1.8	5	9
Jewelry	5	3	1.6		
Painting	5	4	1.25		
Mummy	6	12	0.5		
				6<8	

Summer 22 (B)

1)

Weight	Edge
✓6	(E, F)
✓9	(C, E)
✓9	(A, D)
✓10	(D, F)
✓11	(A, B)
x 11	(B, E)
x12	(C, B)
x13	(B, D)
x15	(D, E)
✓16	(E, G)
x 17	(G, F)

a)



b) MCT = 61

129 - 61 = 68

|V| = 7

|E| = 11

T = ~~0 1 2 3 4 5~~ 6

Vertex Set = ~~{A}, {B}, {C}, {D}, {E}, {F}, {G}~~

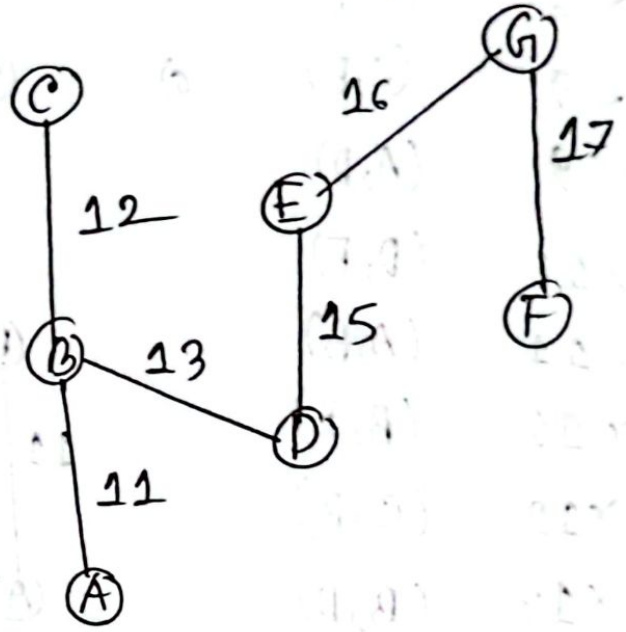
= ~~{E, F}~~ ~~{C, E, F}~~ ~~{A, D}~~

~~{A, C, D, E, F}~~ ~~{A, B, C, D, E, F}~~

{A, B, C, D, E, F, G}

c)

Weight	Edge
✓17	(G, F)
✓16	(E, G)
✓15	(D, E)
✓13	(B, D)
✓12	(C, B)
x11	(B, E)
✓11	(A, B)
x10	(D, F)
x 9	(A, D)
x 9	(C, E)
x 6	(E, F)



|M| = 7

|E| = 11

T = 0

Verten Set = {A}, {B}, {C}, {D}, {E}, {F}, {G}

= ~~{G, F}~~ ~~{E, G, F}~~

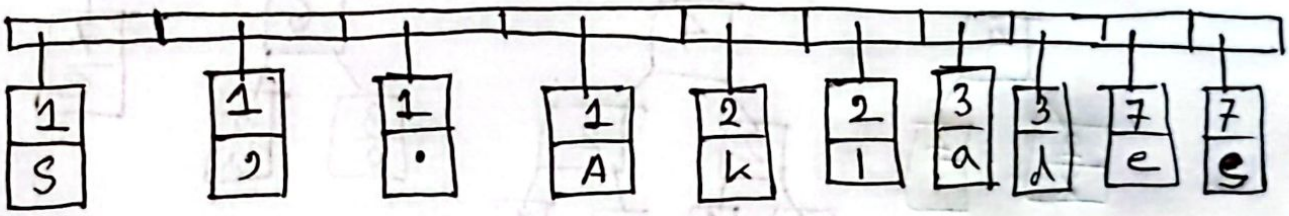
~~{E, G, F, D}~~ ~~{E, G, F, B, D}~~

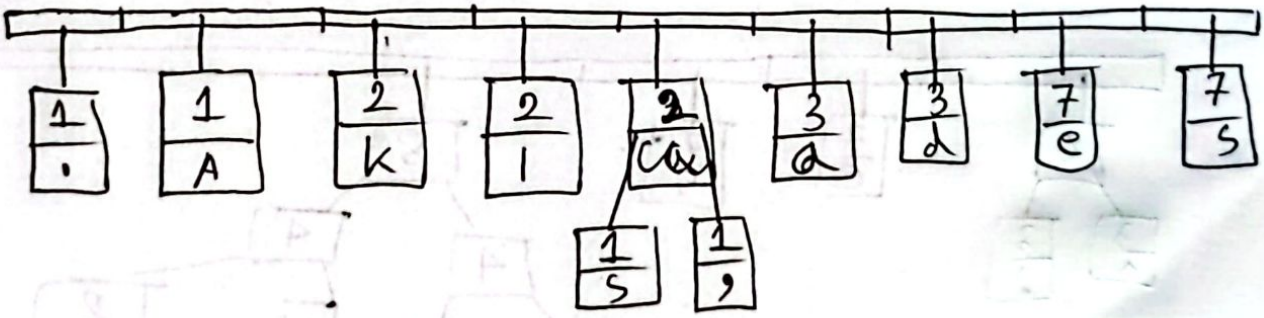
~~{C, E, G, F, B, D}~~

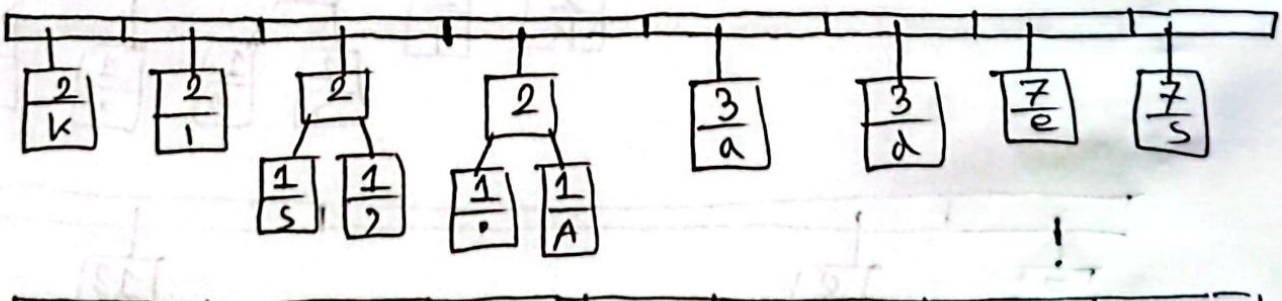
{A, B, C, D, E, F, G}

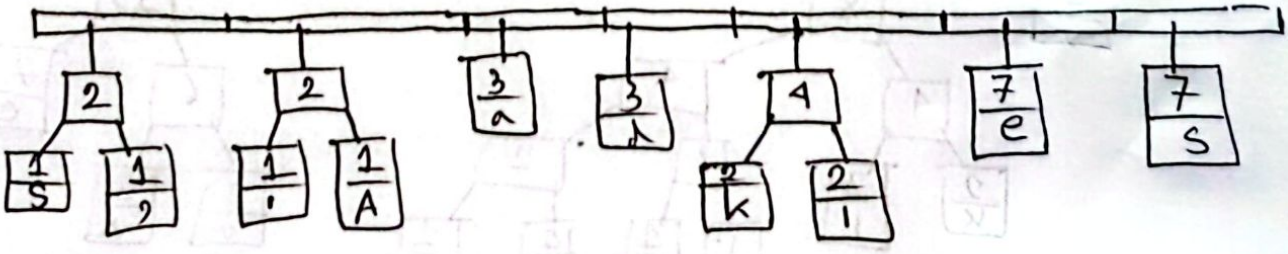
2)

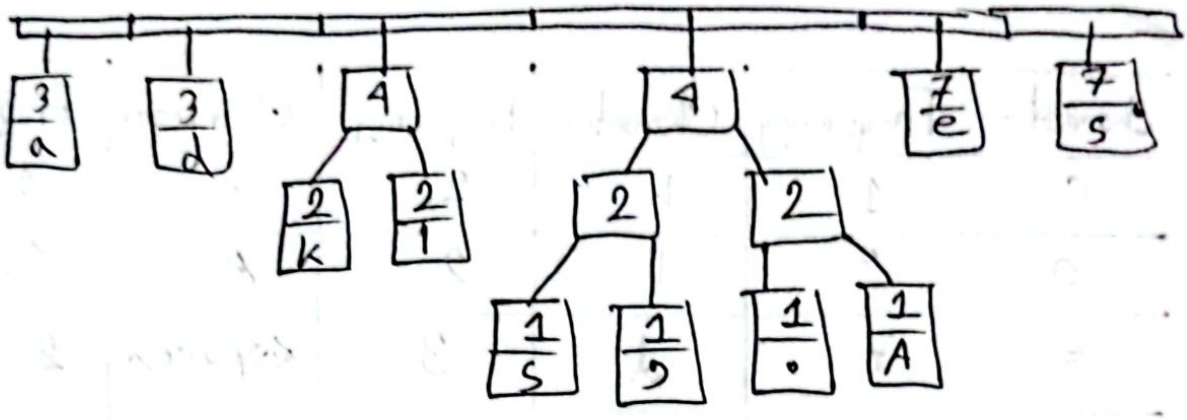
Character	Frequency	Character	Frequency	Character	Frequency
S	1	k	2	.	1
a	3	l	2	A	1
e	7	d	3	<space>	2
s	7	,	1		

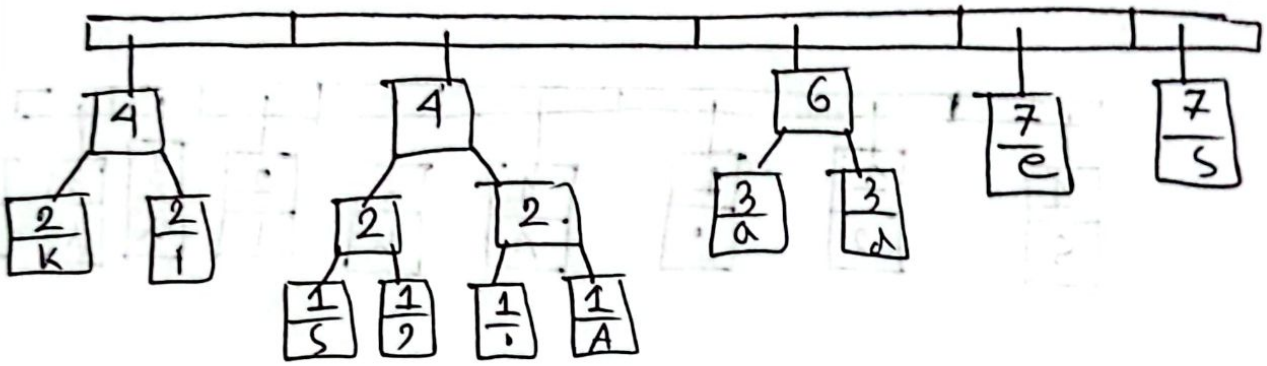


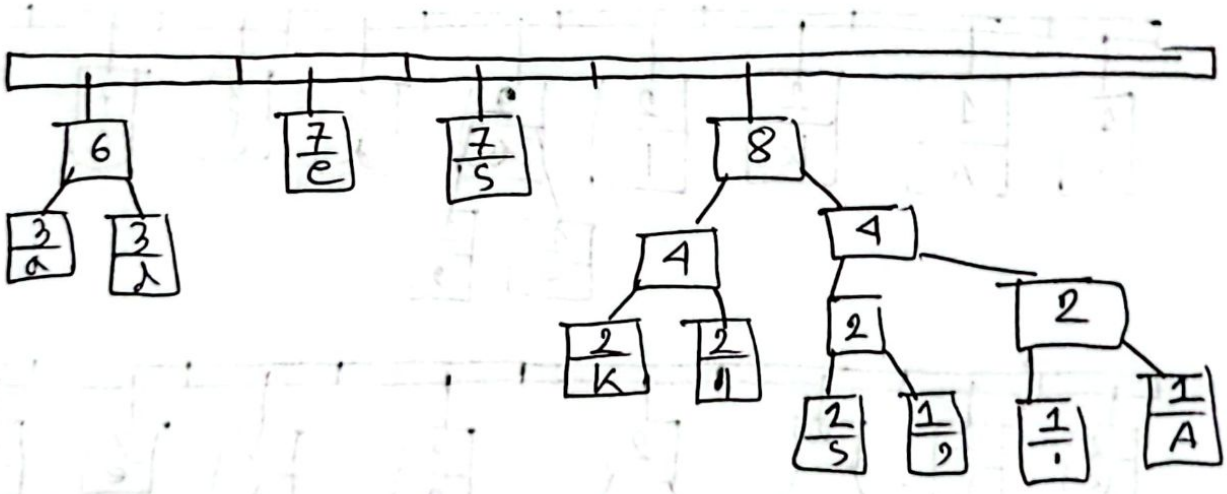


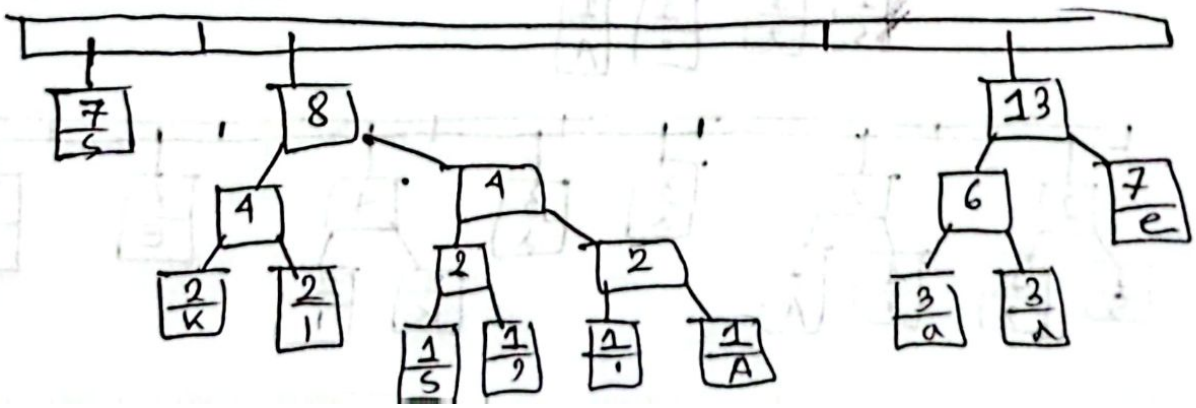


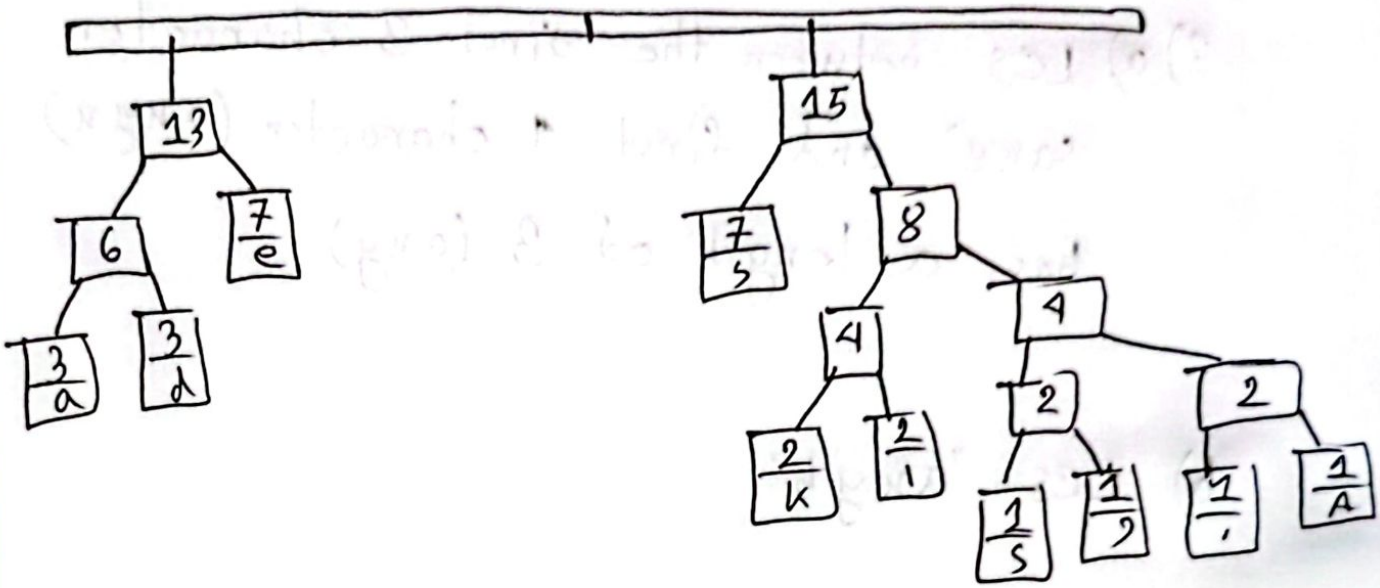


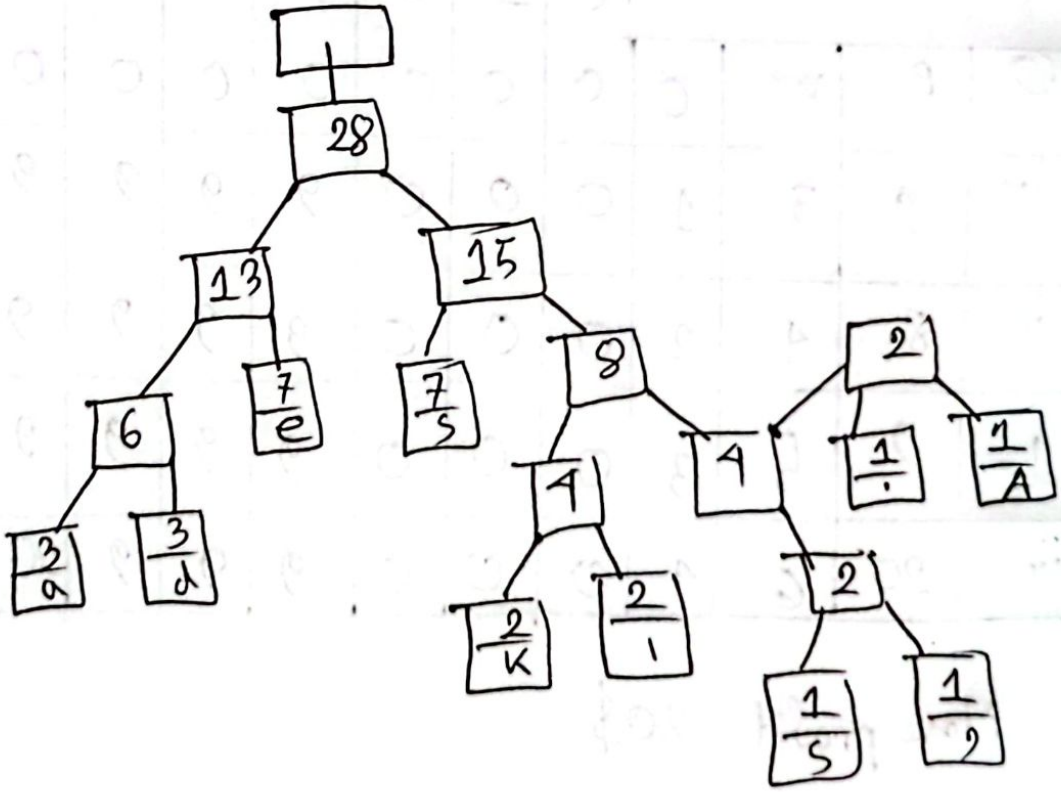

3 a
3 Q
4
2 k
2 1
4
2
2
1 S
1 J
1 .
1 A
7 e
7 S


4
2 k
2 1
4
2
2
1 S
1 J
1 .
1 A
6
3 a
3 A
7 e
7 S


6
3 a
3 A
7 e
7 S
8
4
4
2 k
2 1
2
2
2 S
1 J
1 .
1 A


7 S
8
4
2 k
2 1
4
2
2
1 S
1 J
1 .
1 A
13
6
7 e
3 a
3 A





Couldn't complete missed the space

3) a) LCS between the 'first 3 character 'axy' and first 4 character (axyu) has a length of 3 (axy)

b) LCS = "axyb"

c)

				0	1	2	3	4	5	6	7	8
0	P	w	0	0	0	0	0	0	0	0	0	0
iv	9	3	1	0	0	0	9	9	9	9	9	9
ii	8	4	2	0	0	0	9	9	9	9	17	17
i	9	5	3	0	0	0	9	9	9	9	17	18
iii	20	6	4	0	0	0	9	9	9	20	20	20

Max profit = 20$

Item - III