Unit-3)

Q-1) Find optimal knapsack using greedy strategy

$n = 7$, $m = 15$

$O = \{1, 2, 3, 4, 5, 6, 7\}$

Weight $= \{2, 3, 5, 7, 1, 4, 1\}$

Profit $= \{10, 5, 15, 7, 6, 18, 3\}$

Step-1): Compute per unit weight cost

$p(i) = \frac{p(i)}{w(i)}$

O	O_1	O_2	O_3	O_4	O_5	O_6	O_7
w	2	3	5	7	1	4	1
p	10	5	15	7	6	18	3
$p(i) = \frac{p(i)}{w(i)}$	5	1.6	3	1	6	4.5	3

Step-2) Sort or arrange object in descending order with ratio

O	O_5	O_1	O_6	O_3	O_7	O_2	O_4
$p(i) = \frac{p(i)}{w(i)}$	6	5	4.5	3	3	1.6	1
$p(i)$	6	10	18	15	3	5	7
$w(i)$	1	2	4	5	1	3	7

Step-3: Pack item/objects into greedy strategy

$i \leftarrow 1$

~~while (w ≠ 0)~~ while ($w > 0$)

do amt $\leftarrow$ min (w, $w(i)$)

solution (1) $\leftarrow$ amt

$i \leftarrow i{+}{+}$

return solution

↑	O_2	2	15
	O_7	1	13
	O_3	5	12
m=15	O_6	4	7
	O_1	2	3
↓	O_5	1	1

Total Profit $= (6\times1)+(5\times2)+(4.5\times4)+(3\times5)+(3\times1)+(1.6\times2)$

$$\boxed{\text{Total Profit} = 55.2}$$

Q-2) Solve following using partial knapsack.

$p = \{10, 15, 20, 16, 9\}$

$w = \{2, 8, 6, 5, 3\}$ $n = 5$, $M = 15$

Step-1) Compute per unit weight cost

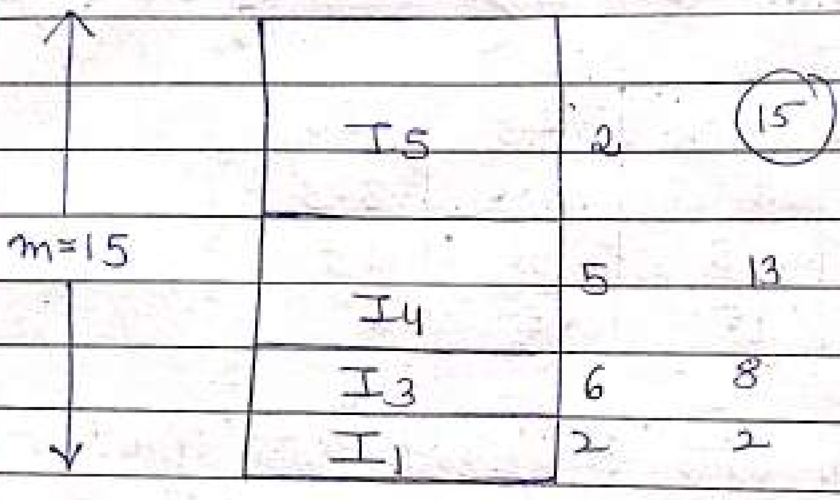
I	I_1	I_2	I_3	I_4	I_5
w	2	8	6	5	3
p	10	15	20	16	9
p/w	5	1.8	3.3	3.2	3

Step-2) Sort Items in descending order with ratio

I	I_1	I_3	I_4	I_5	I_2
p/w	5	3.3	3.2	3	1.8
p	10	20	16	9	15
w	2	6	5	3	8

Step-3) Pack items into Knapsack into greedy strategy

```
i ← 1
while (w > 0)
do amt ← min (w, w(i))
solution(i) ← amt
i++
return solution
```



Total Profit = $(5\times2)+(3.3\times6)+(3.2\times5)+(3\times2)$

$\boxed{\text{Total Profit} = 51.8}$

Q-4) Find the optimal solution to the fractional Knapsack if the Knapsack capacity $W = 60$ kg.

Item	I_1	I_2	I_3	I_4	I_5
weight	5	10	15	22	25
Cost	30	40	45	77	90

Step-1) Compute pere unit weight cost

Item	I_1	I_2	I_3	I_4	I_5
w	5	10	15	22	25
p	30	40	45	77	90
p/w	6	4	3	3.5	3.6

Step-2) Sort them in Descending order of ratio (p/w)

Item	I_1	I_2	I_3	I_5	I_4
p/w	6	4	3	3.6	3.5
p	30	40	45	90	77
w	5	10	15	25	22

Step-3) Pack Item into Knapsnack into Greedy strategy

```
i ← 1
while (w>0)
do amt ← min (w, w(i))
solution[i] ← amt
i++
return solution
```

M=15

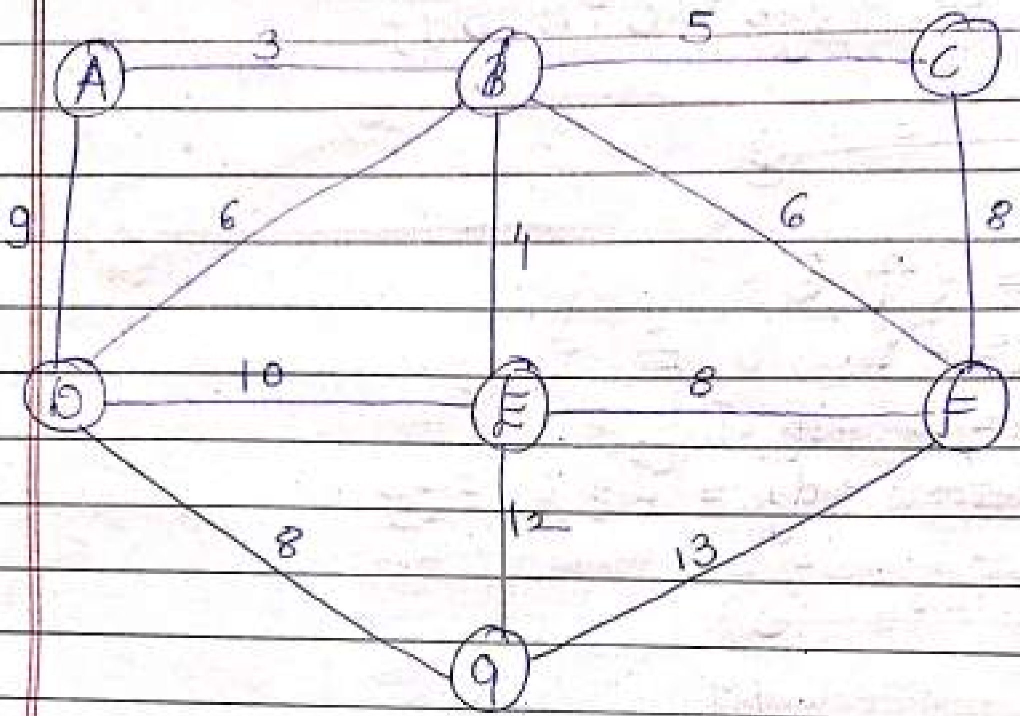
I_4	5	
I_5	25	60
		55
I_3	15	
I_2	10	30
I_1	5	15
		5

Total Profit = $(6 \times 5) + (4 \times 10) + (3 \times 15) + (3.6 \times 25) + (3.5 \times 5)$

$\boxed{\text{Total Profit} = 222.5}$

Page No.
Date :

Q-5) what is Minimum Cost Spanning Tree? Find Minimum Cost spanning Tree for following using Prim's algorithm?



Ans: A M Cost spanning Tree is like the most efficient way to connect a bunch of places together with roads.

A Minimum Cost Spanning Tree (MST) or a minimum weight spanning tree for a weighted, connected, undirected graph is a spanning tree with a less weight of every other spanning tree.

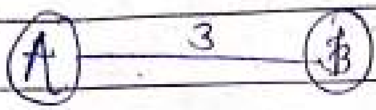
Step 1:

$S = \{A\}$

$R = \{B, C, D, E, F, G\}$

$A = \{\ \}$

Lightest edge $= \{\{A, B\}\}$



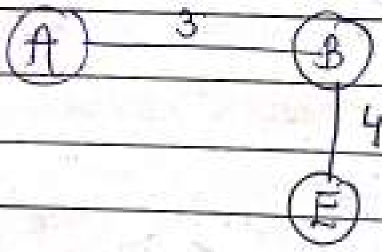
Step-2:

$S = \{A, B\}$

$R = \{C, D, E, F, G\}$

$A = \{\{A, B\},$

Lightest edge $= \{\{B, E\}\}$



Step-3:

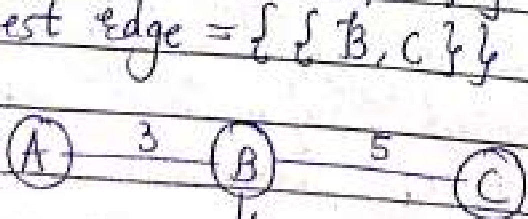
$S = \{A, B, E\}$

$R = \{C, D, F, G\}$

$A = \{\{A, B\}, \{B, E\}\}$

Lightest edge $= \{\{B, C\}\}$

Step-4)



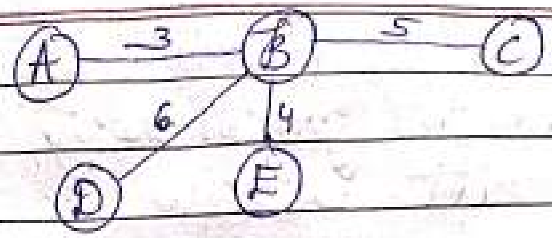
$S = \{A, B, E, C\}$

$R = \{D, F, G\}$

$A = \{\{A, B\}, \{B, E\}, \{B, C\}\}$

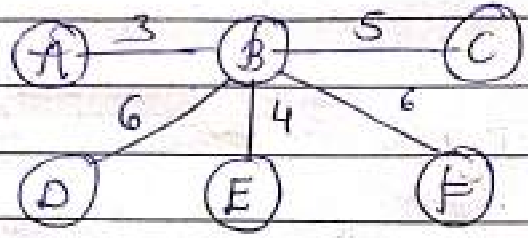
Lightest edge $= \{\{B, D\}\}$

Step-5)



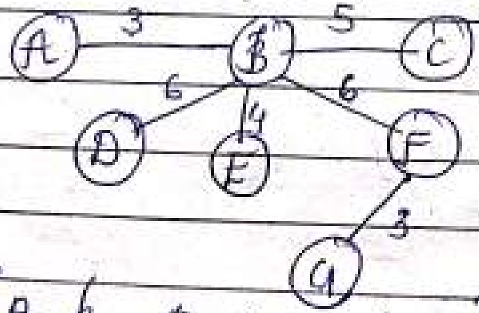
S = {A, B, C, D, E}
R = {F, G}
A = {{A, B}, {B, E}, {B, C}, {B, D}}
Lightest edge = {{B, F}}

Step-6)



S = {A, B, C, D, E, F}
R = {G}
A = {{A, B}, {B, E}, {B, C}, {B, D}, {B, F}}
Lightest edge = {{F, G}}

Step-7)



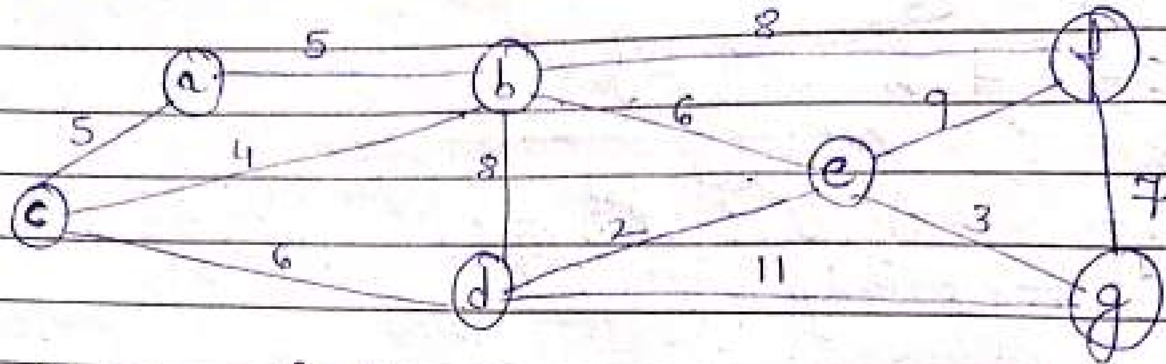
S = {A, B, E, C, D, F, G}
R = { }
A = {{A, B}, {B, E}, {B, C}, {B, D}, {B, F}, {F, G}}

Find Cost! w(A C) + w(BE —

Min Cost Spanning Tree = 27

Date :

Ques - 6) Obtain Minimum Cost Search Spanning with its given undirected graph using Prims algo Vertex 'a' as a root Vertex.

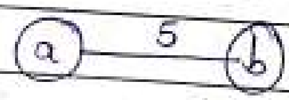


Step-1) $S = \{a\}$

$K = \{b, c, d, e, f, g\}$

$A = \{\ \}$

Lightest edge $= \{\{a, b\}\}$

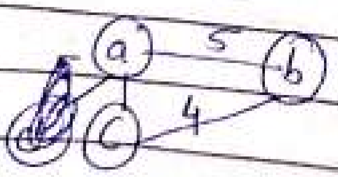


Step-2) $S = \{a, b\}$

$K = \{c, d, e, f, g\}$

$A = \{\{a, b\}\}$

Lightest edge $= \{\{b, c\}\}$

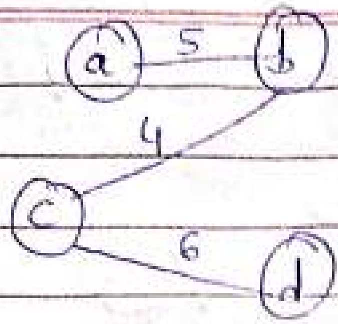


Step-3) $S = \{a, b, c\}$

$K = \{d, e, f, g\}$

$A = \{\{a, b\}, \{b, c\}\}$

Lightest Edge $= \{\{c, d\}$

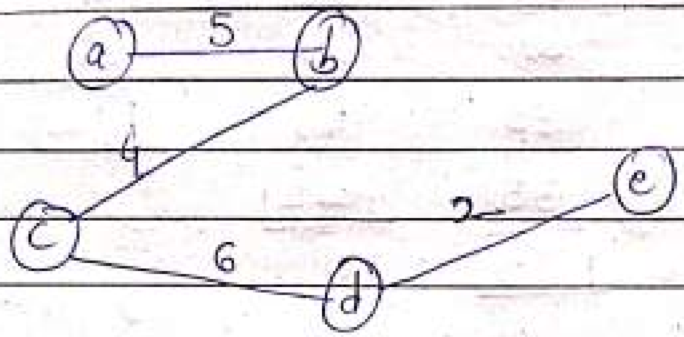


Step-4) $S = \{a, b, c, d\}$

$R = \{e, f, g\}$

$A = \{\{a,b\}, \{b,c\}, \{c,d\}\}$

Lightest edge $= \{\{d,e\}\}$

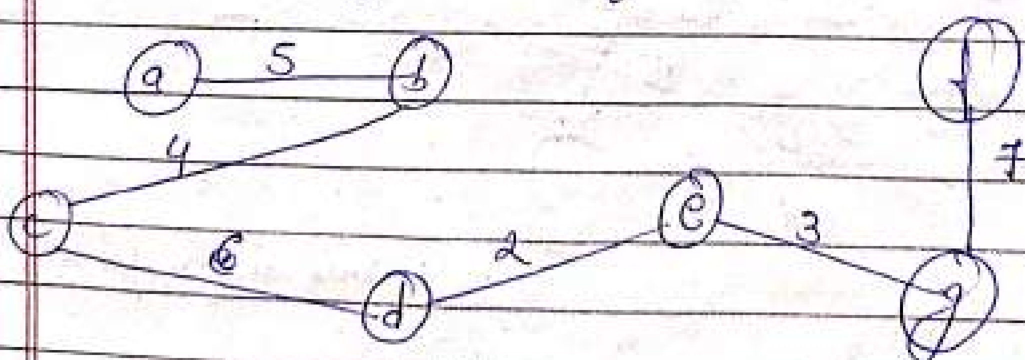


Step-5) $S = \{a, b, c, d, e\}$

$R = \{f, g\}$

$A = \{\{a,b\}, \{b,c\}, \{c,d\}, \{d,e\}\}$

Lightest edge $\{\{e, g\}\}$

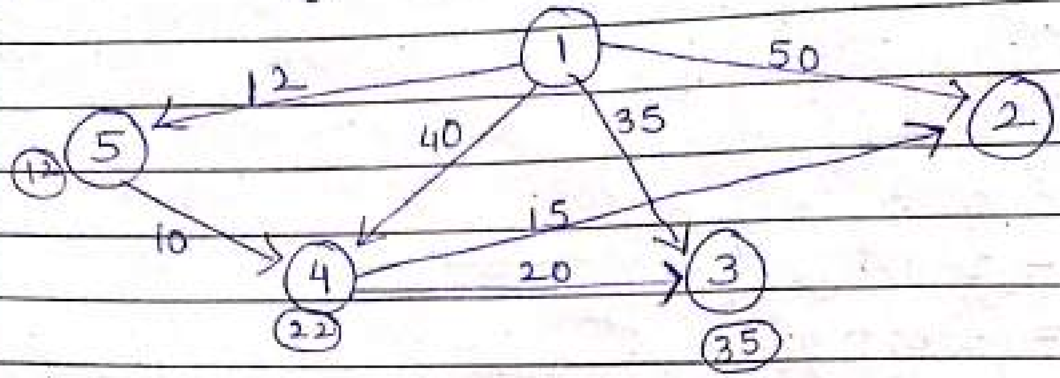


Min Cost of Spanning Tree = 27

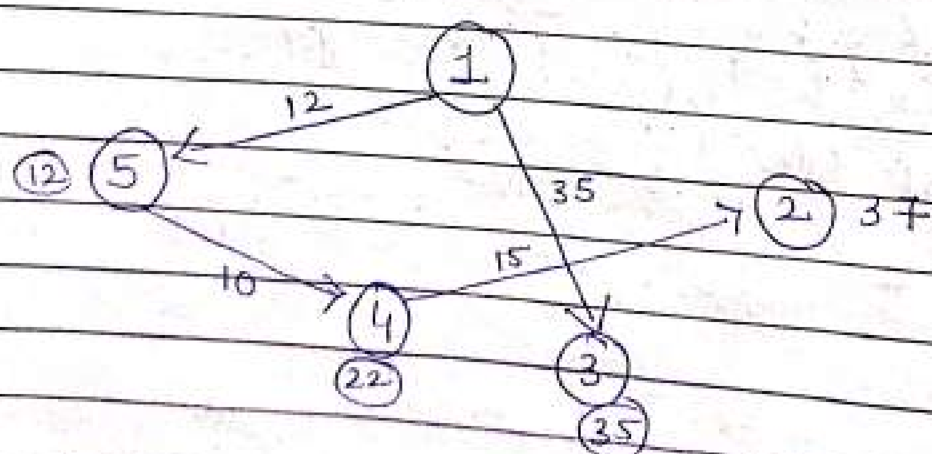
Page No. :
Date :

Dijikshtra Algo

Q-8) : Write Greedy based single source shortest path algorithm



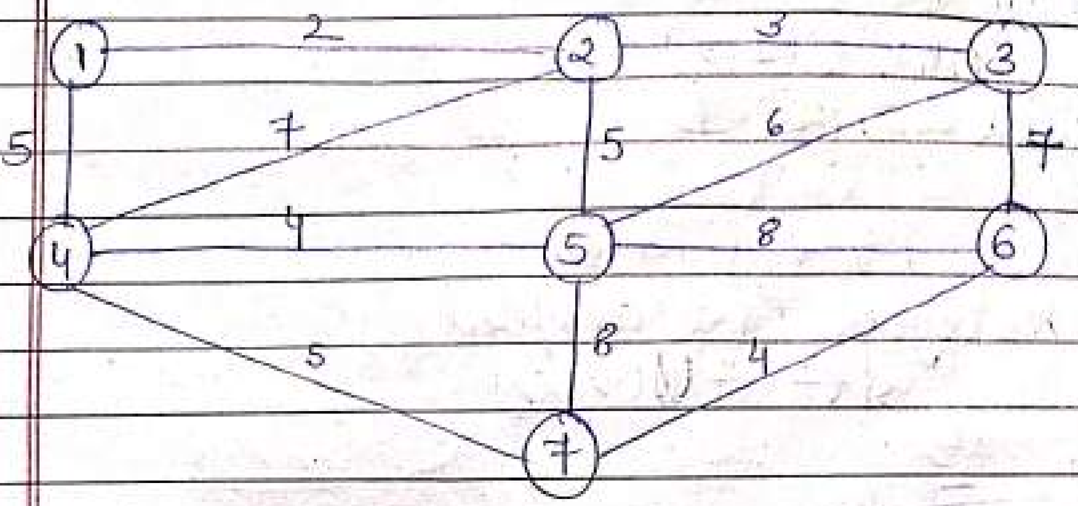
Source	Destination			
1	2	3	4	5
1	∞	∞	∞	∞
1, 5	50	35	40	$\boxed{12}$
1, 5, 4	50	35	$\boxed{22}$	
1, 5, 4, 3	37	$\boxed{35}$		
1, 5, 4, 3, 2	$\boxed{37}$			



* This is graph

Vertex	d(v)
1	0
2	37
3	35
4	22
5	12

Ques-9) Krushals Algorithm to find minimum Spanning Tree.



Solⁿ: Step-1) Make set of N vertices

① $A \leftarrow \phi$

① ② ③ ④ ⑤ ⑥ ⑦

1 2 3 4 5 6 7

Step-2) Sort edges in Increasing order of weight

edges	weight
1 2	2
2 3	3
4 5	4
6 7	4
2 5	5
4 7	5
1 4	5
3 5	6
3 6	7
4 2	7
5 7	8
5 6	8

Date :

step-3) Construction of Minimal Spanning Tree.

→ for edge 12

$(u,v) = (1,2)$

FIND-SET(1) = 1

" (2) = 2

$1 \neq 2$ (True)

No cycle (1 2) accepted

$A \leftarrow A \cup (V,V)$

$(1) \xrightarrow{2} (2)$

Union (u,v)

12	3	4	5	6	7
1	2	3	4	5	6

→ for edge 23

$(u,v) = (2,3)$

FD(2) = 1

FD(3) = 2

$1 \neq 2$ (True) no cycle (2,3) accepted

$A \leftarrow A \cup (V,v)$

$(1) \xrightarrow{2} (2) \xrightarrow{3} (3)$

123	4	5	6	7
1	2	3	4	5

→ for edge 45

$(u,v) = (4,5)$

FS(4) = 2

FS(5) = 3

$2 \neq 3$ (True)

No cycle (2 3) accepted

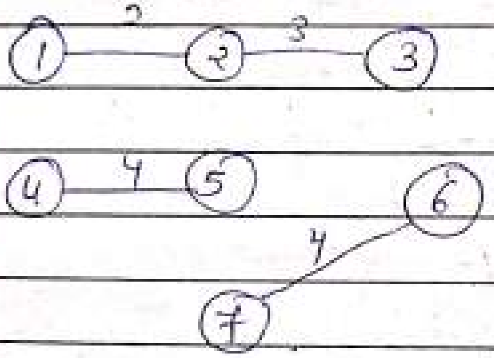
→ For edge 67

$(u, v) = (6, 7)$

$F(6) = 3$

$f(7) = 4$ $\quad 3 \neq 4$ No cycle

67 edge is accepted



123 45 67

1 2 3

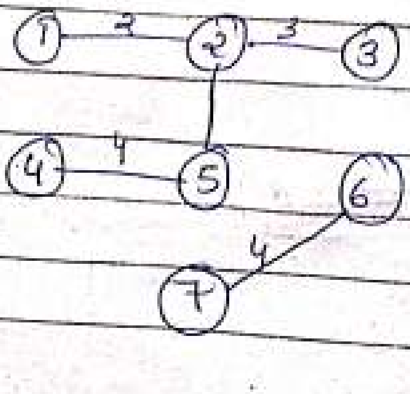
→ For edge 25

$(u, v) = (2, 5)$

$FS(2) = 1$, $FS(5) = 2$

$1 \neq 2$ (True)

25 is accepted



12345 67

1 2

→ for edge 47

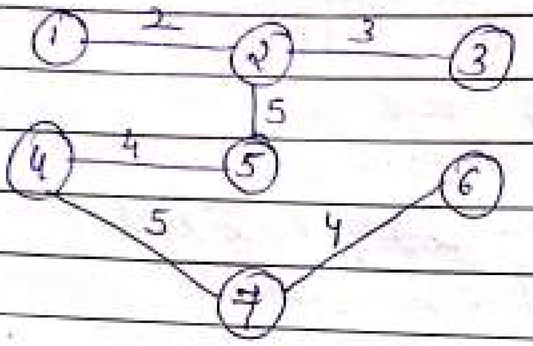
$(u,v) = (4,7)$

$fs(4) = 1$

$fs(7) = 2$

$1 \neq 2$ (True) No cycle

47 is accepted



union

12 34 5 6 7.

→ for edge 14

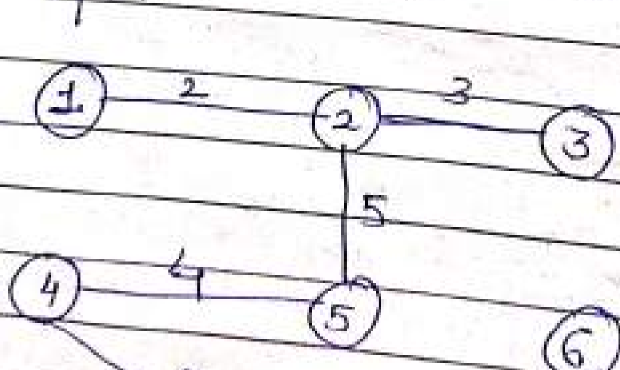
$fs(1) = 1$, $fs(4) = 1$

$1 \neq 1$ (false)

14 rejected

Similarly all the remaining edges are rejected

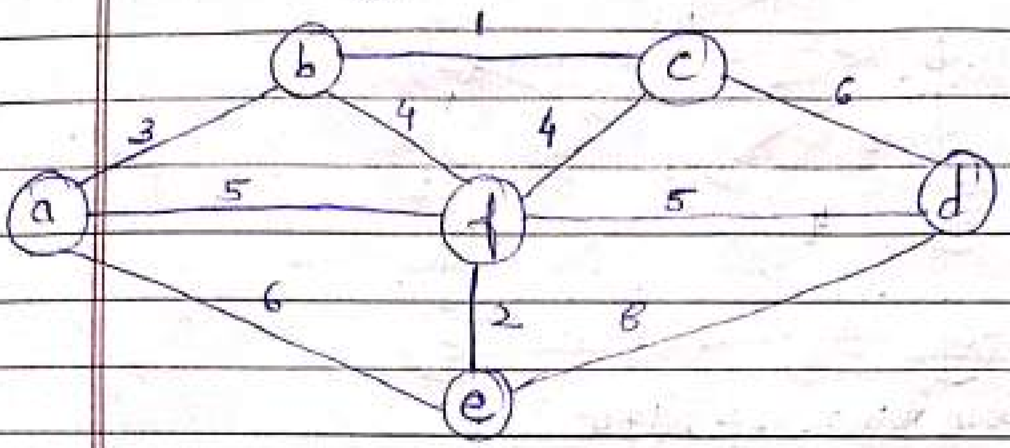
so final MST.



Find cost (Minimal Cost of Spanning Tree) = 23

Page No. :
Date :

Q-10) Prims algorithm determine Minimum Cost of Spanning Tree



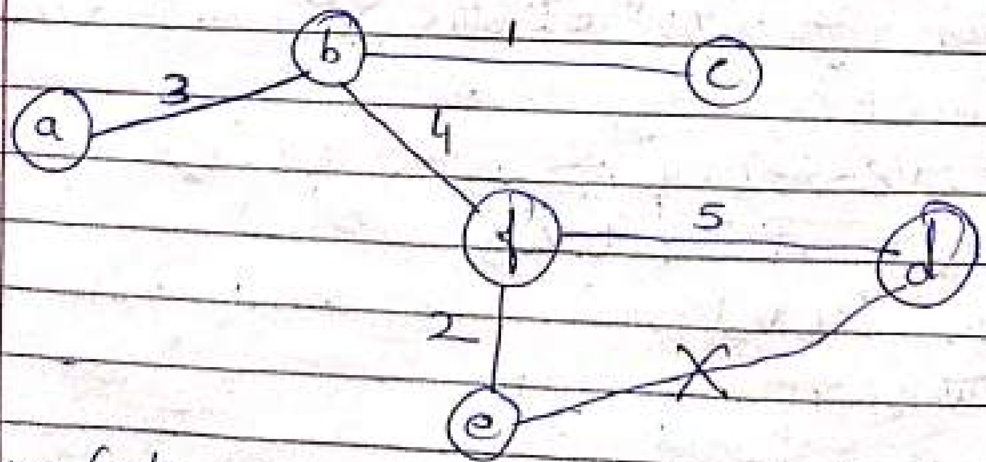
Soln:

(A)

$S = \{a\}$, $k = \{b, c, d, e, f\}$
$A = \{\}$ lightest edge $= \{\{a, b\}\}$

(a) —3— (b)

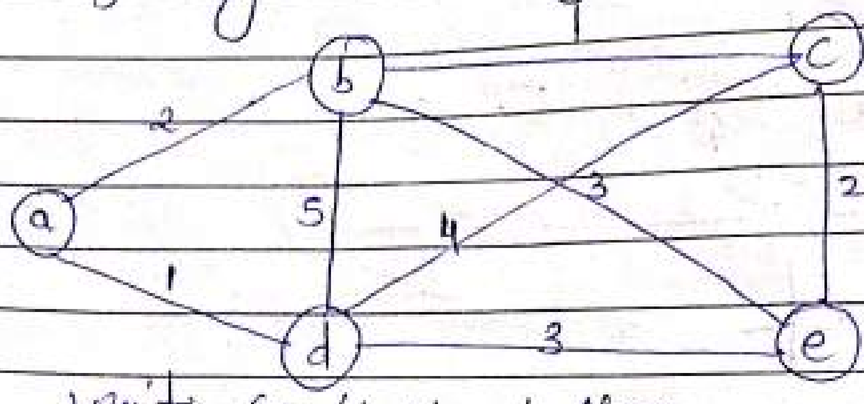
final step: $S = \{a, b, c, d, e, f\}$, $k = \{\}$
$A = \{\{a,b\}, \{b,c\}, \{b,f\}, \{f,d\}, \{f,e\}\}$



Minimum Cost of Spanning Tree $= 15$

Date :

Q-11) Write Krushkal's Algorithm to generate Spanning Tree. Apply algo on Graph



Write Complexity of Algo

Ans: Krushkal's Algorithm :

Step-1) 1) Make set of n vertices
$a \leftarrow \phi$
2) for each vertex $v \in V(G)$
3) Do make set (v)

Step-2) Sort the edges E into Increasing / non-decreasing order by weight ω

Step-3) Construct the MST by Merging of Trees
(1) For each edge $(u,v) \in E$, taken in non-decreasing order by weight.
(2) Do if FIND-SET $(u) \neq$ FINDSET(v)
(3) Then $A \leftarrow A \cup \{(u,v)\}$
(4) union (u,v)
(5) return A

Page No. :
Date :

Step-1) $a \leftarrow 0$,

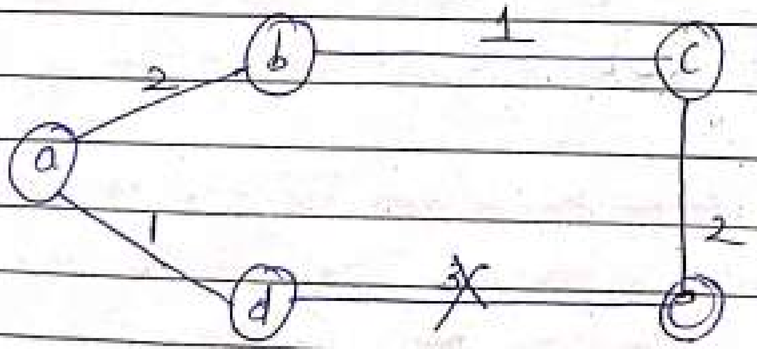
a b c d e

1 2 3 4 5

Step-2) sort edges

Edges	Weight
ad	1
bc	1
de	2
ec	2
	3
	3
	4
	5

Steps -> for lena!



ab cde

Minimal cost of Spanning Tree = 6

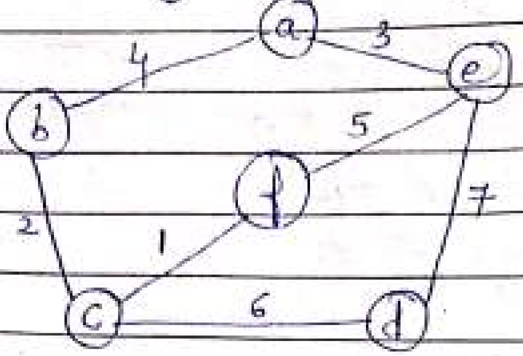
Time Complexity = $O(E \log V)$

edges

no. of Vertices

Time Complexity = ...

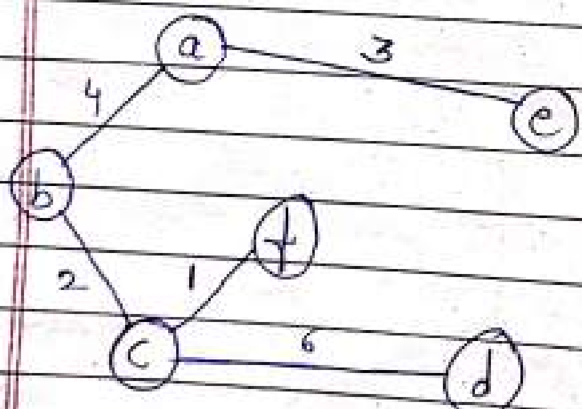
(12) Obtain Minimum Spanning Tree for given undirected graph using Prim's algorithm.



Soln: $S = \{a, b, c, d, e, f\}$

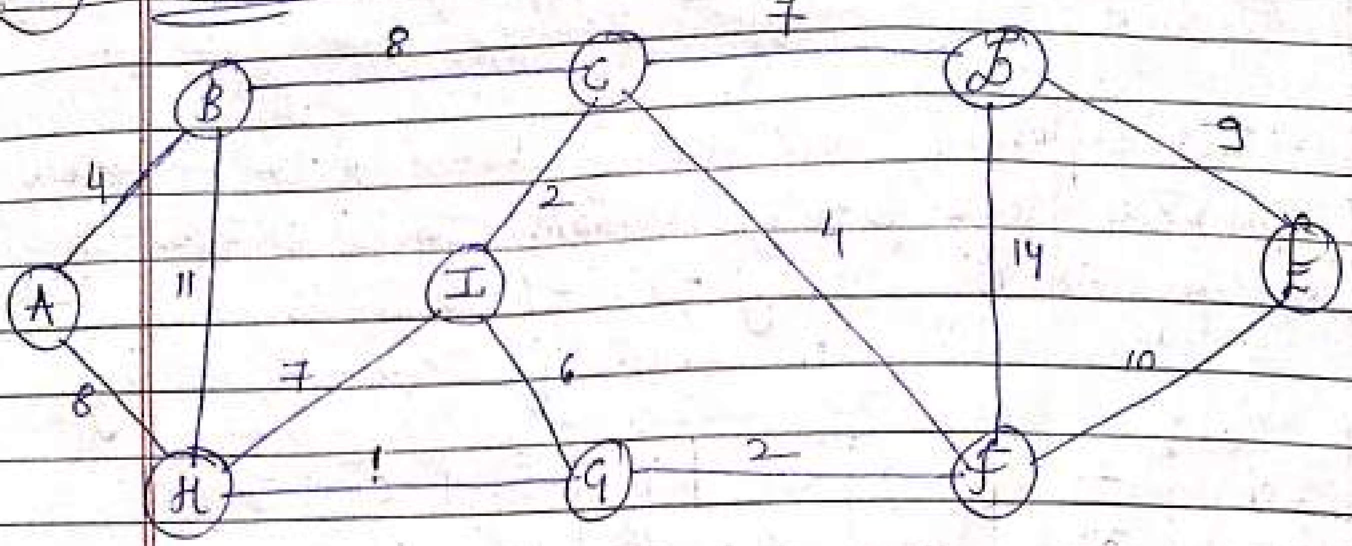
$R = \{ \}$

$A = \{\{a,b\}, \{a,e\}, \{b,c\}, \{c,f\}, \{c,d\}\}$



Cost of minimal Spanning Tree = 16

1.3 PRIMS

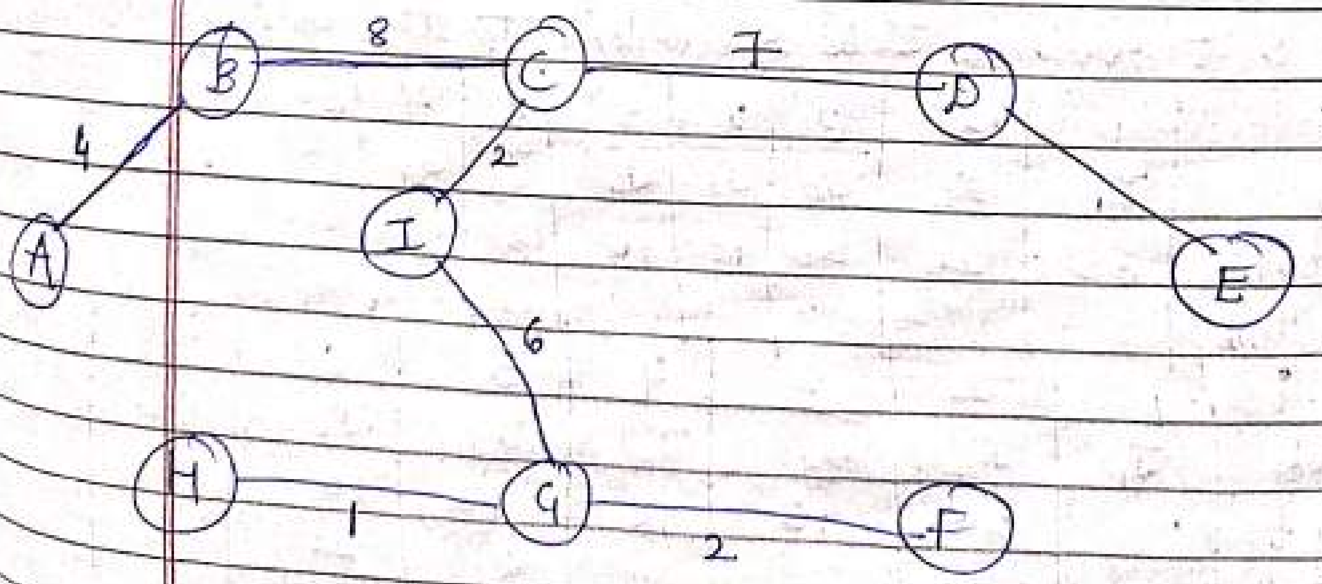


S = {A, B, C, D, E, F, G, H, I}

R = {φ

A = {

lightest edge = {{ φ}



Date :

Ques 4) : Explain Job Sequencing Approach. Find best Possible Sequence

Ans Job Sequencing is a scheduling technique used in various industries to prioritize & manage tasks or Jobs. It typically involves assigning priorities to Job based on certain criteria and then scheduling them accordingly

Job	1	2	3	4	5	6	7
Gain	35	20	18	16	12	10	8
Deadline	3	1	3	4	2	2	1

Step-1) Label the Jobs accordingly

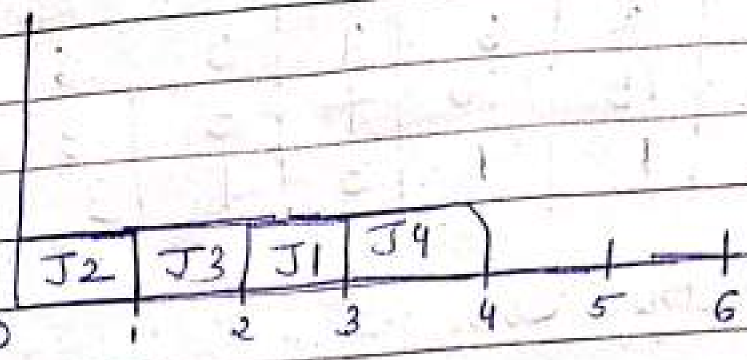
Job	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Gain	35	20	18	16	12	10	8
Deadline	3	1	3	4	2	2	1

Step-2) Sort / Arrange Jobs According to the Gain in descending order

i	1	2	3	4	5	6	7
Job	J_1	J_2	J_6	J_3	J_4	J_5	J_7
Gain	35	20	20	18	16	12	
Deadline							

i	1	2	3	4	5	6	7
Job	J_1	J_2	J_3	J_4	J_5	J_6	J_7
Gain Profit	35	20	18	16	12	10	8
Deadline	3	1	3	4	2	2	1

Step 1) Finding set of Job containing flexible Job to get a maximum profit



J	Assigned slots	Job considered	Action	Profit
ϕ	None	J1	Assign [2,3]	0
J1	[2, 3]	J2	Assign [0,1]	35
J1, J2	[2,3] [0,1]	J3	Assign [1, 2]	35+ 20=
J1, J2, J3	[2,3] [0,1] [1, 2]	J4	Accept [3,4]	55+
J1, J2, J3	[2,3] [0,1] [1, 2]	J5	~~Assign [3,4]~~ Reject	73 77

$J_1 + J_2 + J_3 + J_4$

89

Q-16) Explain Job scheduling approach find the optimal schedule for following Job with $n=6$.

Job(n)	1	2	3	4	5	6
Gain / Profit	20	15	10	7	5	3
Deadline	3	1	1	3	1	3

Sol^n **Step-1)** Label the Jobs

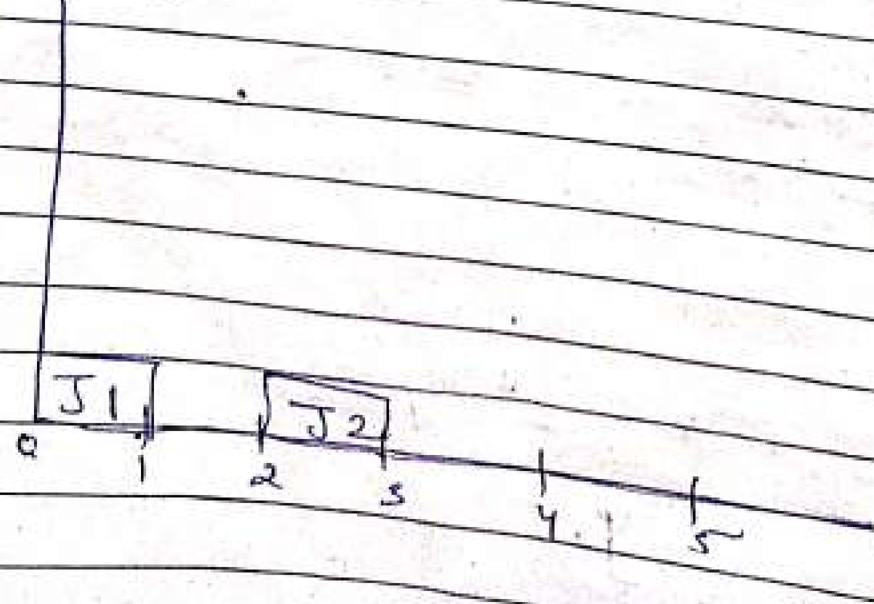
Job(n)	J1	J2	J3	J4	J5	J6
Profit	20	15	10	7	5	3
Deadline	3	1	1	3	1	3

Step-2) Sort in Decreasing order of Profit.

i	1	2	3	4	5	6
Job(n)	J1	J2	J3	J4	J5	J6
Profit	20	15	10	7	5	3
Deadline	3	1	1	3	1	3

Step-3) finding set

Q



J	Assign Slots	Job Considered	Action	Profit
φ	None	J1	Assign [2,3]	0
J1	[2,3]	J2	Assign [0,1]	0+20=20
J1,J2	[2,3] [0,1]	J3	reject	20+15=35
J1,J2	[2,3] [0,1]	J4	reject	35
		J5	reject	
		J6		

Total Profit = p(J1) + p(J2)

= 35.

Que-15) Explain Greedy Algorithm for scheduling problem with Deadline.

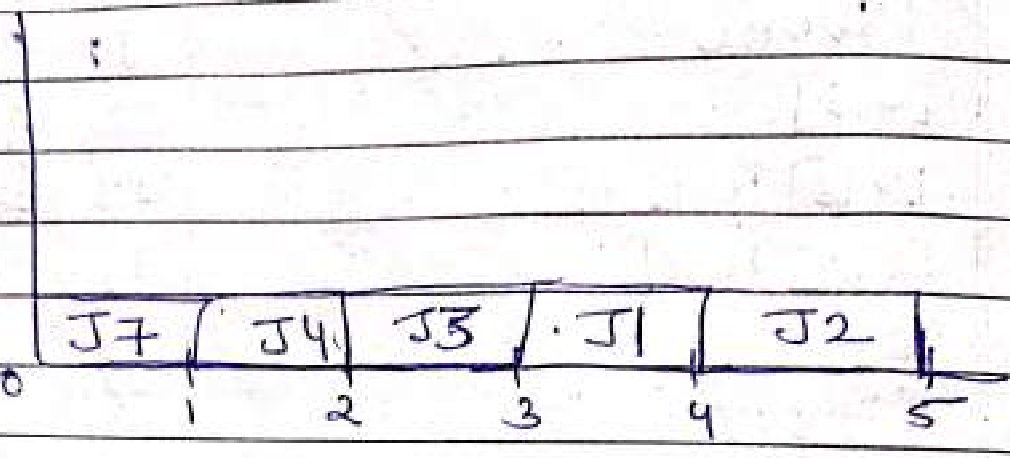
Job	J1	J2	J3	J4	J_5	J_6	J_7	J_8
Profit	102	91	39	55	25	18	40	05
Deadline	4	5	3	2	1	5	1	2

Step-1) Label the Jobs

Step-2) Arrange (sort the Jobs in the Decreasing order of Profit.

i	1	2	3	4	5	6	7	8
Job	J1	J2	J4	J7	J3	J5	J_6	J_8
Profit	102	91	55	40	39	25	18	05
Deadline	4	5	2	1	3	1	5	2

3) Finding set of Job Containing flexible Job to get of maximum profit



Assigned slots	Job Considered	Action	Profit
None	J1	Assign [3,4]	0
[3,4]	J2	Assign [4,5]	102
[3,4] [4,5]	J4	Assign [1,2]	102+91
[3,4] [4,5] [1,2]	J7	Assign [0,1]	102+91+
[3,4] [4,5] [1,2]	J3	Assign [2,3]	102+91+ +40
[3,4] [4,5] [1,2] [2,3]	J5	Reject	102+91+ 40+19
	Reject	hogi	

Total profit $= w(J_1) + w(J_2) + w(J_8) + w(J_7) + w(J_4)$

$= 327$

Page No. :
Date :

Step-3) Finding set of Job Containing flexible Job to get of maximum profit

J7	J4	J5	J1	J2

0 1 2 3 4 5

J	Assigned slots
ϕ	None
J1	[3,4]
J2	[3,4] [4,5]
J4	[3,4] [4,5] [1,2]
J7	[3,4] [4,5] [1,2]
J3	[3,4] [4,5] [1,2] [2,3]

Job Considered	Action	Profit
J1	Assign [3,4]	0
J2	Assign [4,5]	102
J4	Assign [1,2]	102+91
J7	Assign [0,1]	102+91+[illegible]
J3	Assign [2,3]	102+91+[illegible] +40
J5	Reject	102+91+[illegible] 40+19
Reject	hogi .	

Total Profit $= w(J_1) + w(J_2) + w(J_3) + w(J_7) + w(J_4)$

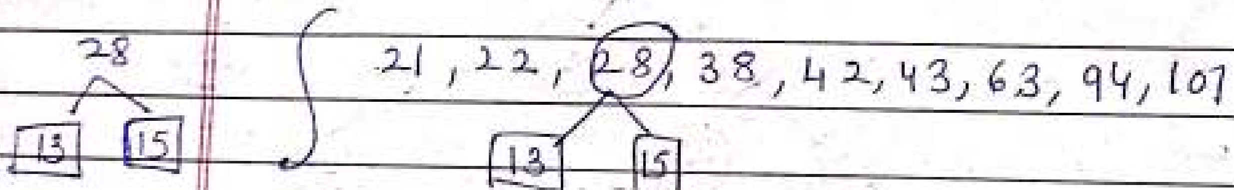
$= 327$

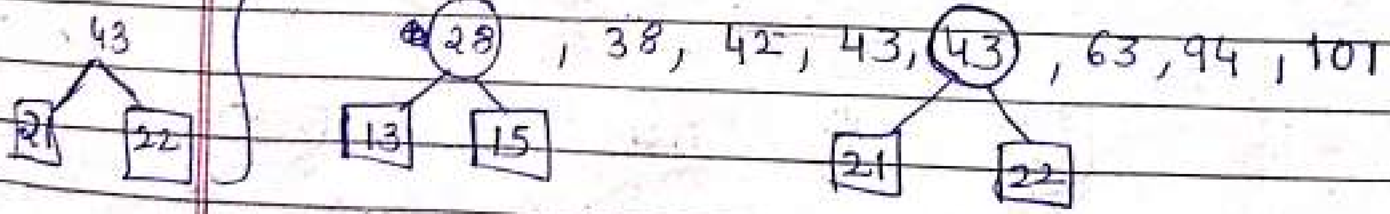
Ques-19) what is Optimal Merge Pattern? Implement optimal binary merge pattern for ten files whose Length are 38, 42, 22, 15, 94, 63, 101, 43, 13, 21

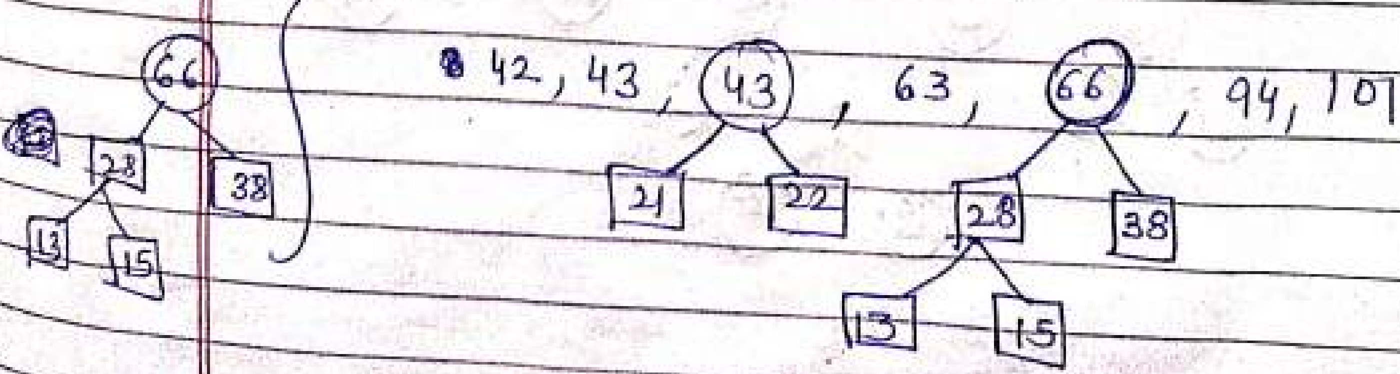
Ans The "optimal merge pattern" is a Concept in Computer science, specifically in Context of sorting & Merging data. It's a technique used for efficiently merging multiple sorted sequences into a single sorted sequence, often used in external sorting & Database Management systems

Step-1) Sort in Ascending order

13, 15, 21, 22, 38, 42, 43, 63, 94, 101

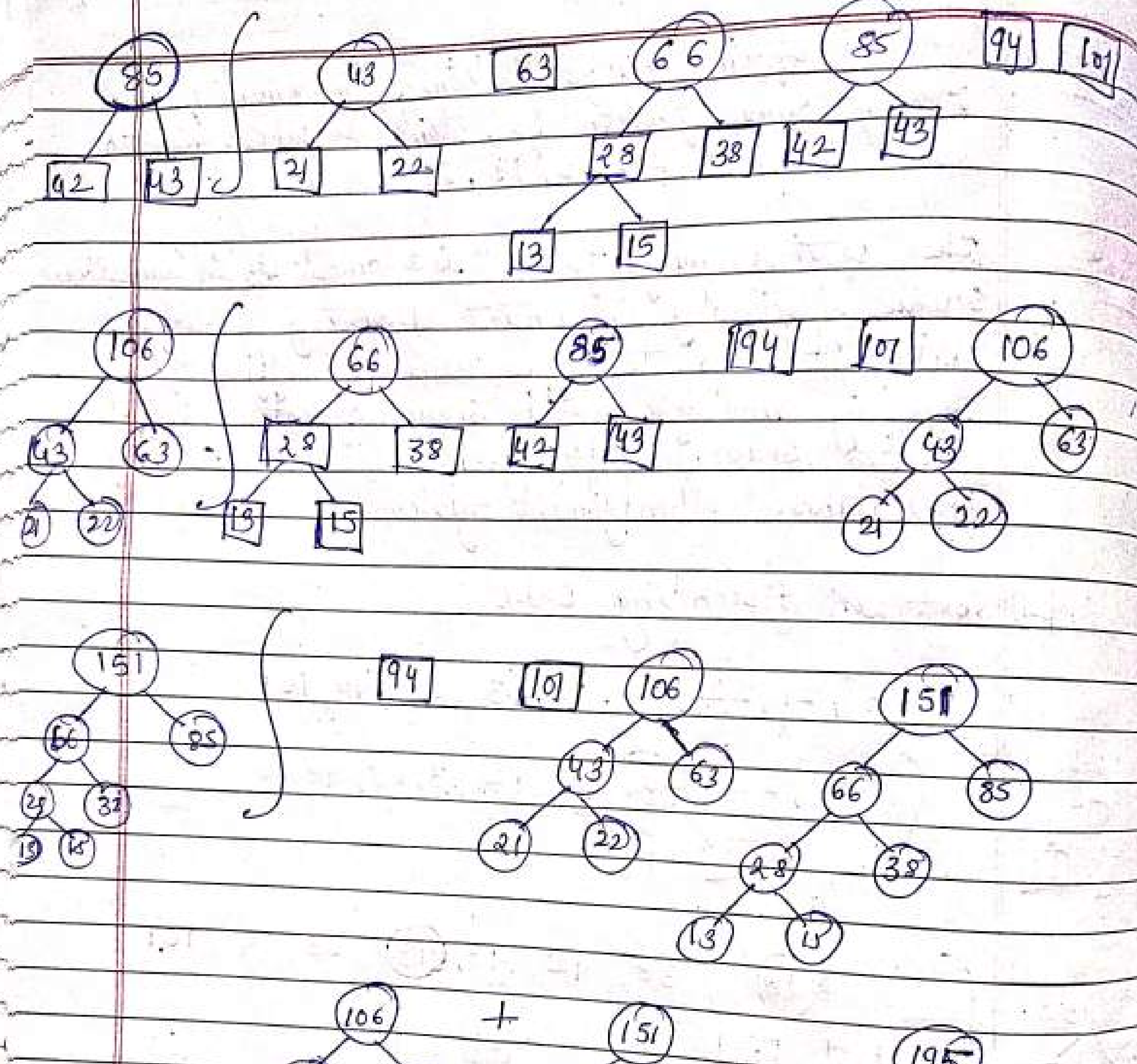




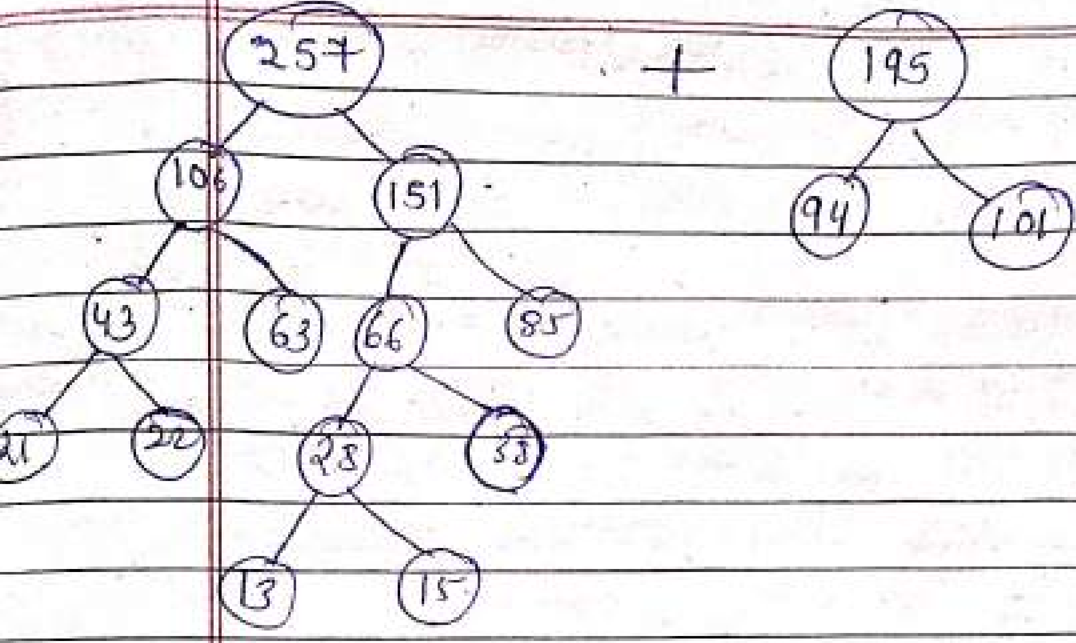


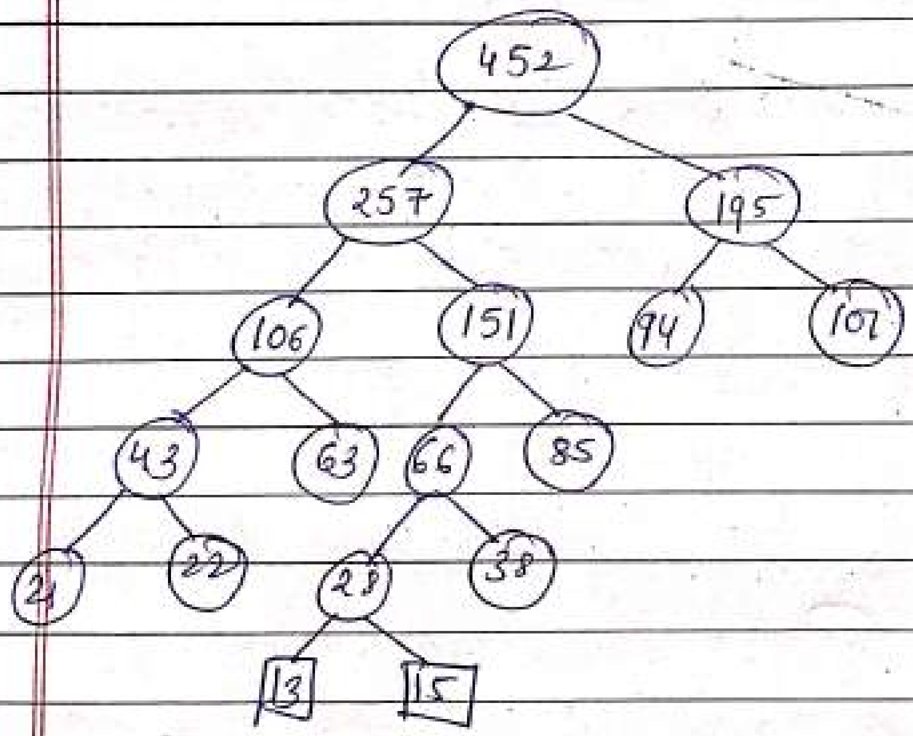
Page No. :

Date :



85 42 43 43 21 22 63 66 28 38 13 15 85 42 43 94 101
106 43 63 21 22 66 28 38 13 15 85 42 43 94 107 106 43 63 21 22
151 66 85 28 38 13 15 94 101 106 43 63 21 22 151 66 85 28 38 13 15
106 43 63 21 22 + 151 66 85 28 38 13 15 195 94 101





Total values = 452 + 257 + 195 + 106 + 151 + 43 + 66 + 85 + 28 = 1383

7. The