classmate

Date 10/4/2023

Page 107

* BRANCH & BOUND

Intro :-

It is algo design paradigm which is generally used for solving optimization problems. These problems are exponentials in terms of time complexity. B&B solves them in relatively less time.

Ex :- Job sequencing with deadline

$J = \{ J_1, J_2, J_3, J_4 \}$

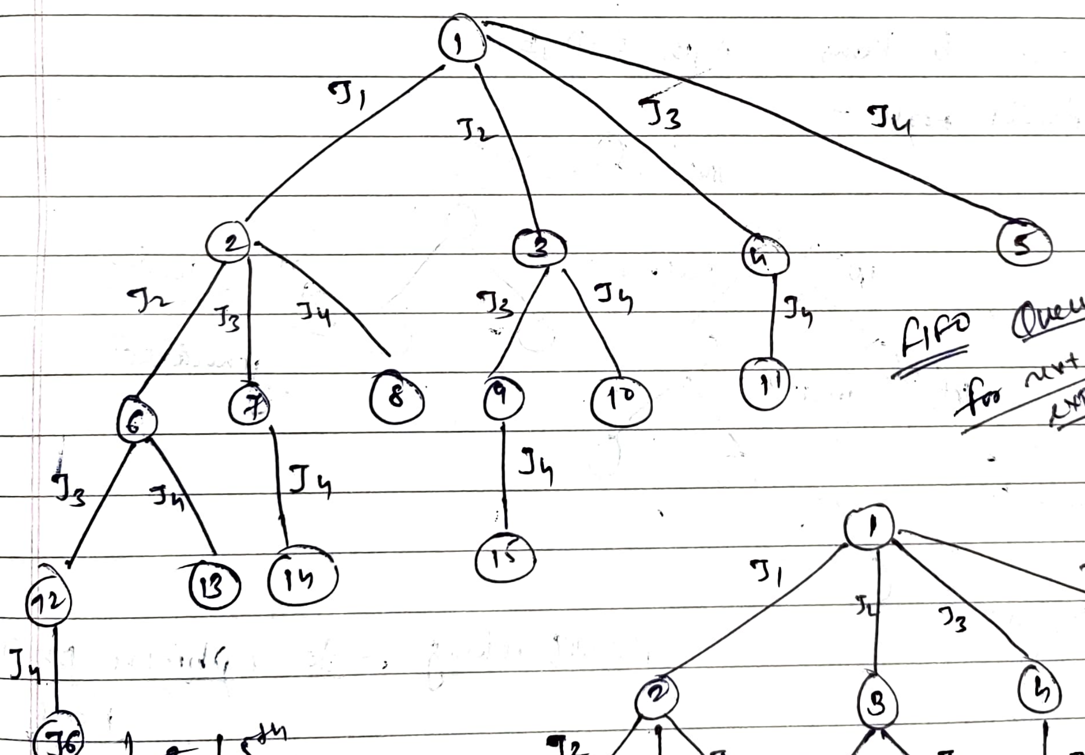
$P = \{ 10, 5, 8, 3 \}$

$D = \{ 1, 2, 1, 2 \}$

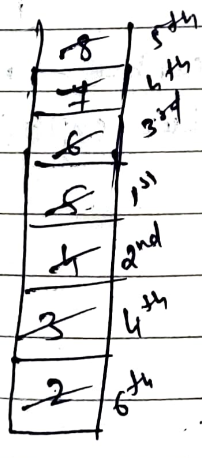
$\{ J_1, J_2 \} \rightarrow$ FIFO B&B

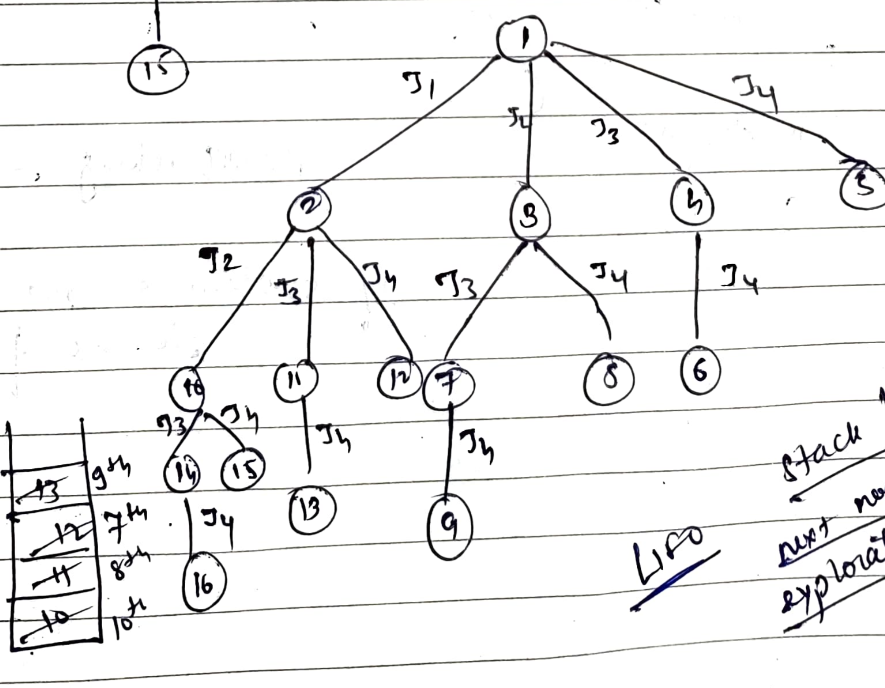
$\{ 1, 0, 0, 1 \} \rightarrow$ LIFO B&B

Least cost B&B (LC)



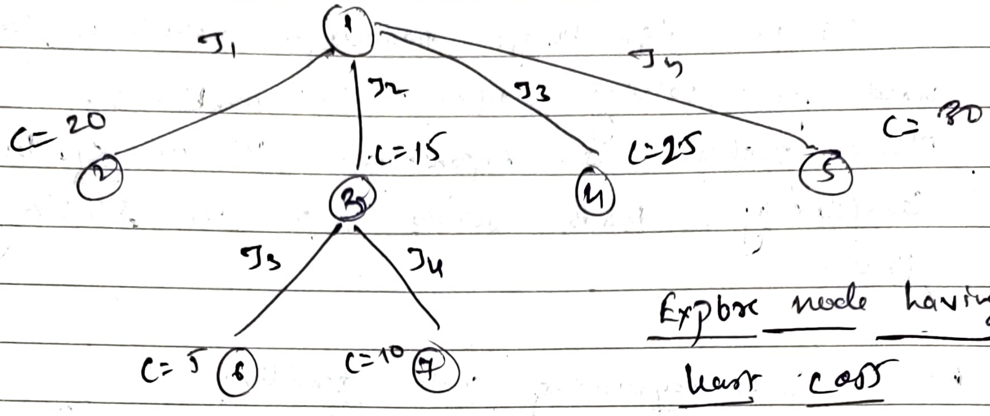
FIFO Queue for next node exploration





LIFO stack for next node exploration

Least cost B&B (LCB&B)



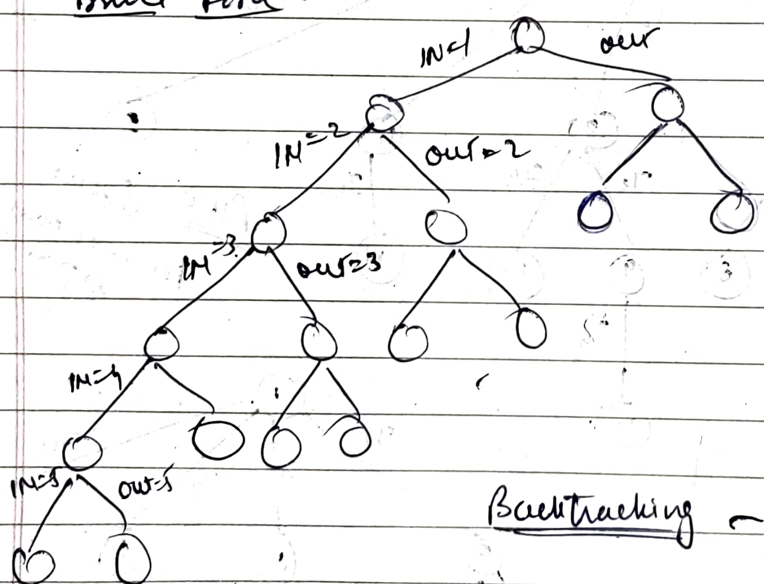
Explore node having least cost

---

* 0/1 knapsack using B&B

For 5 items A, B, C, D, E

Brute force -



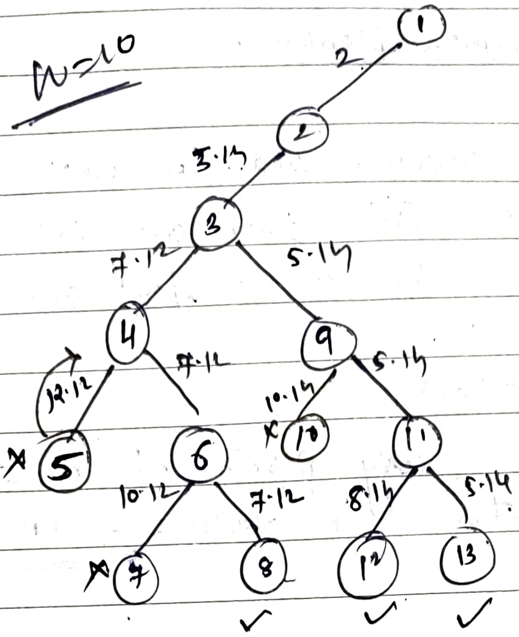
2^n solutions generated

Backtracking - to optimize Brute force

If solution is not feasible ∴ no need to do further exploring.

A = 2
B = 3.14
C = 1.98
D = 5
E = 3

W=10



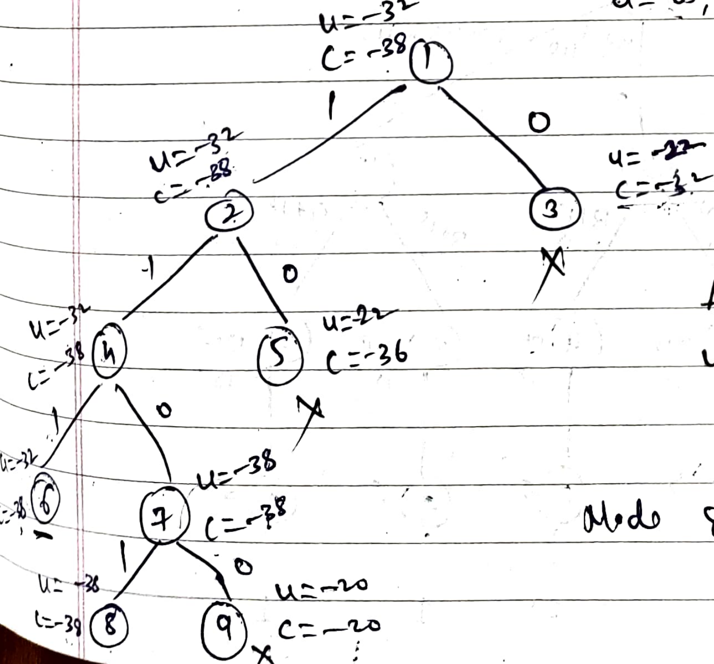
* **Branch & Bound** - Backtracking works better than Brute Force. We can do even better if we know bound on best possible solution subtree.

Logic : If the best in subtree is worse than current Best, we can simply ignore this node and its subtrees. So, we compute bound for every node and compare it with current best before exploring the node.

u = ∞, -32, -38

1	2	3	4
10	10	12	18
2	4	6	9

W=15



At node '7', we get smaller upper bound -38 so we update it

Node 8 = final answer
{1, 1, 0, 1}.

* Travelling Salesman Problem.

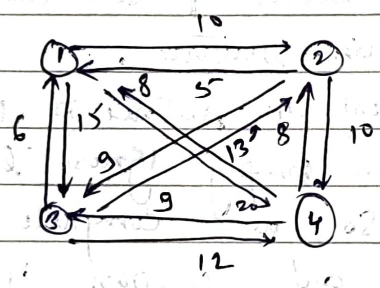
Given a salesman and 'n' cities, where distance between all cities are known and each city should be visited just once.

Brute force Method

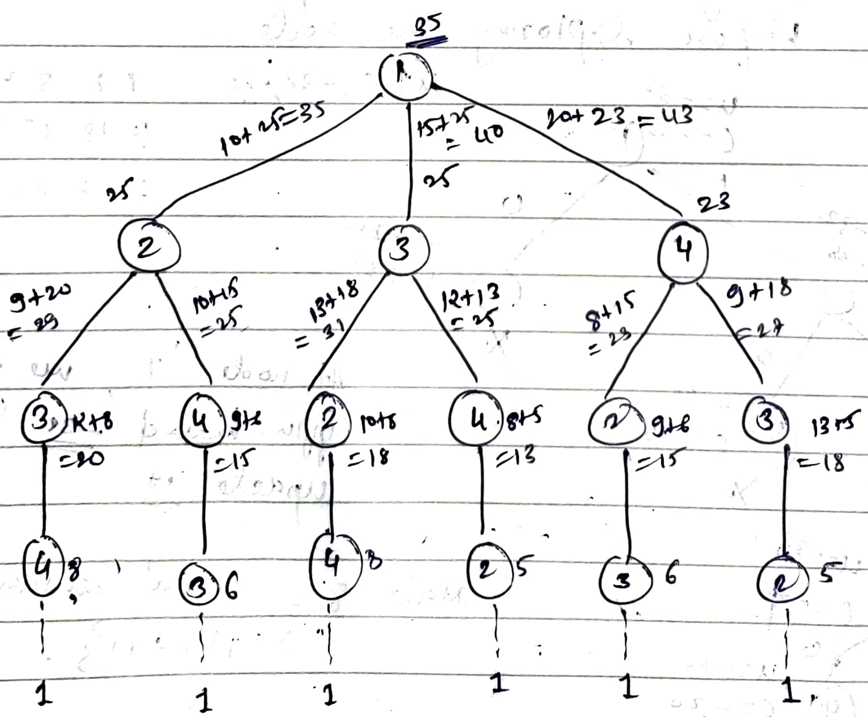
Evaluate every possible tour and select the best one. For 'n' vertices – (n-1)! possibilities

Dynamic =>

This can be implemented using less time as compared to brute force, though it is not polynomial



	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0



classmate

Date ___ Page 111

~~$C(1,$~~

$C(\{2,3,4\}, 1\} \Rightarrow$ length of shortest path visiting each node in $\{2,3,4\}$ from 1 upto 1.

$C(S, j) \Rightarrow$ length of shortest path visiting each node in S starting at 1 ending at j.

$$c(\{2,3,4\}, 1\} \Rightarrow \min_{k \in \{2,3,4\}} \{ d_{1k} + C(\{2,3,4\} - \{k\}, k) \}$$

<u>In General</u> $\Rightarrow$

$$C(S, j) = \min_{k \in S} \{ d_{jk} + C(S - \{k\}, k) \}.$$

<u>$S = \phi$</u>

$C(\phi, 2) = 0 + d(2,1) = 5$ $\quad C(2, \phi, 1)$

$C(\phi, 3) = 0 + d(3,1) = 6$ $\quad C(3, \phi, 1)$

$C(\phi, 4) = 0 + d(4,1) = 8$ $\quad C(4, \phi, 1)$.

<u>$S = 1$</u>

$C(3, \{4\}, 1) = d(3,4) + cost(4, \phi, 1) = 12 + 8 = 20$

$C(4, \{3\}, 1) = d(4,3) + cost(3, \phi, 1) = 9 + 6 = 15$

$C(2, \{4\}, 1) = d(2,4) + cost(4, \phi, 1) = 10 + 8 = 18$

$C(4, \{2\}, 1) = d(4,2) + C(2, \phi, 1) = 8 + 5 = 13$

$C(2, \{3\}, 1) = d(2,3) + C(3, \phi, 1) = 9 + 6 = 15$

$C(3, \{2\}, 1) = d(3,2) + C(2, \phi, 1) = 13 + 5 = 18$

<u>$S = 2$</u>

$C(2, \{3,4\}, 1) = \min \{ d(2,3) + C(3, \{4\}, 1)$
$\quad d(2,4) + C(4, \{3\}, 1) \}$

$= \min \{ 9 + 20, 10 + 15 \}$

$= \underline{25}$

$C(3, \{2,4\}, 1) = \min \{ d(3,2) + C(2, \{4\}, 1)$
$\quad d(3,4) + C(4, \{2\}, 1) \} = 25$

$C(4, \{2,3\}, 1) = \min \{ d(4,2) + C(2, \{3\}, 1)$
$\quad d(4,3) + C(3, \{2\}, 1 \} = 23$

S=3

$$c\{1, \{2,3,4\}, 1\} = \min \begin{cases} d(1,2) + c(2\{3,4\},1) \\ d(1,3) + c(3,\{2,4\},1) \\ d(1,4) + c(4,\{2,3\},1) \end{cases}$$

$$= \min\{10 + 25, 15+25, 20+23\}$$

$$= \underline{35}$$

To get Path :-

$c\{1, \{2,3,4\},1)$. min for $d(1,2) = \underline{10}$

$1 \to 2$

$c\{2, \{3,4\},1\}$ min for $d(2,4) = \underline{10}$

$1 \to 2 \to 4$

$c\{4, \{3\},1)$ min for $d(4,3) = \underline{9}$

$1 \to 2 \to 4 \to 3.$

$c(3, \phi, 1)$ min for $d(3,1) = \underline{6}$

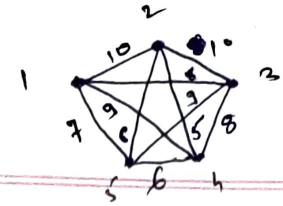
$\underline{1 \to 2 \to 4 \to 3 \to 1}$

* **Using Branch and Bound**

Example 1 **Method 1**

Given Matrix -

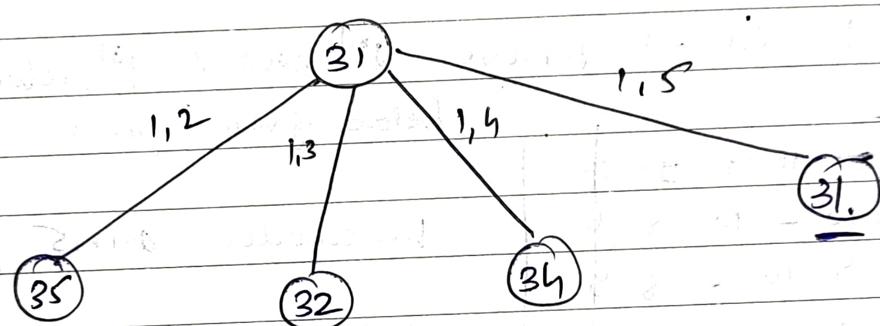
$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 10 & 15 & 20 \\ 2 & 5 & 0 & 9 & 10 \\ 3 & 6 & 13 & 0 & 12 \\ 4 & 8 & 8 & 9 & 0 \end{array}$$



Example 2 — Method 2

$$\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & - & 10 & 8 & 9 & 7 & 7 \\ 2 & 10 & - & 10 & 5 & 6 & 5 \\ 3 & 8 & 10 & - & 8 & 9 & 8 \\ 4 & 9 & 5 & 8 & - & 6 & 5 \\ 5 & 7 & 6 & 9 & 6 & - & 6 \\ \hline & & & & & & 31 \end{array}$$

~~Row Reduction~~
Sum of Row minimum

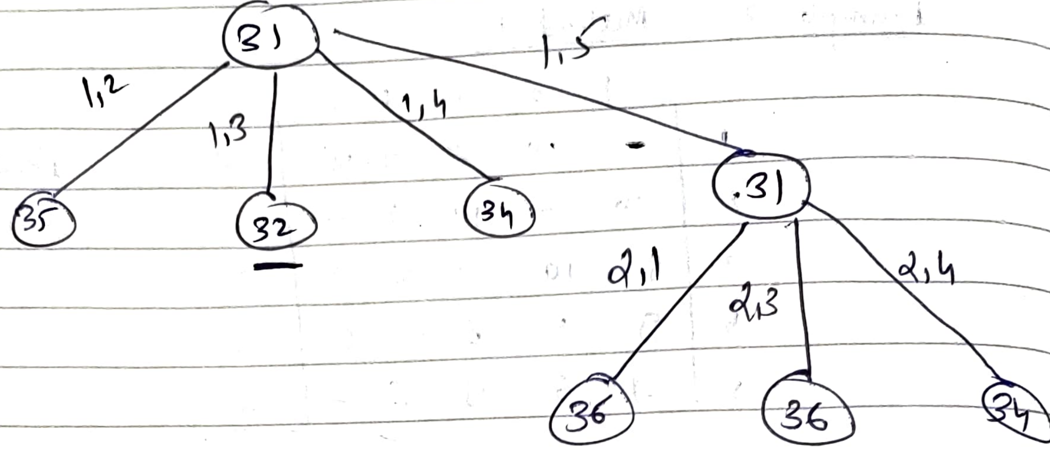


for 1,2 → Remove 1st row & 2nd column
Find sum of row min. = 5 + 8 + 6 + 6 = 25
Add d(1,2) = 10
35

for 1,3 → Remove 1st row & 3rd column
Sum of Rows min = 5 + 8 + 5 + 6 = 24
d(1,3) = 8
32

for 1,4 → Remove 1st row & 4th column
Sum of Row mins = 6 + 8 + 5 + 6 = 25
d(1,4) = 9
34

for 1,5 → Remove 1st row & 5th column
Sum of Row mins = 5 + 8 + 5 + 6 = 24
d(1,5) = 7
31



for 2,1 → Remove 2nd row & 1st column from below given matrix

$$\begin{bmatrix} \cancel{-} & \cancel{10} & \cancel{8} & \cancel{9} & \cancel{7} \\ \cancel{10} & - & 10 & 5 & \cancel{6} \\ 8 & 10 & - & 8 & \cancel{9} \\ 9 & 5 & 8 & - & \cancel{6} \\ 7 & 6 & 9 & 6 & \cancel{-} \end{bmatrix}$$

we consider 2-1-5, remove 5-2

Add Row min $= 8+5+6 = 19$

Add $d(1,5)$ & $d(2,1) = 7+10$

$= 36$

for 2,3 → Remove 2nd Row & 3rd column

Also remove 3,2

$$\begin{array}{c|ccc} & 1 & 2 & 4 \\ 3 & 8 & - & 8 \\ 4 & 9 & 5 & - \\ 5 & \cancel{7} & 6 & 6 \end{array}$$

Add Row min $= 8+\cancel{8}5+6 = \cancel{22}\ 19$

$d(1,5)+d(2,3) = 7+10$

$\cancel{40}\ 36$

for 2,4 → Remove 2nd Row & 4th column

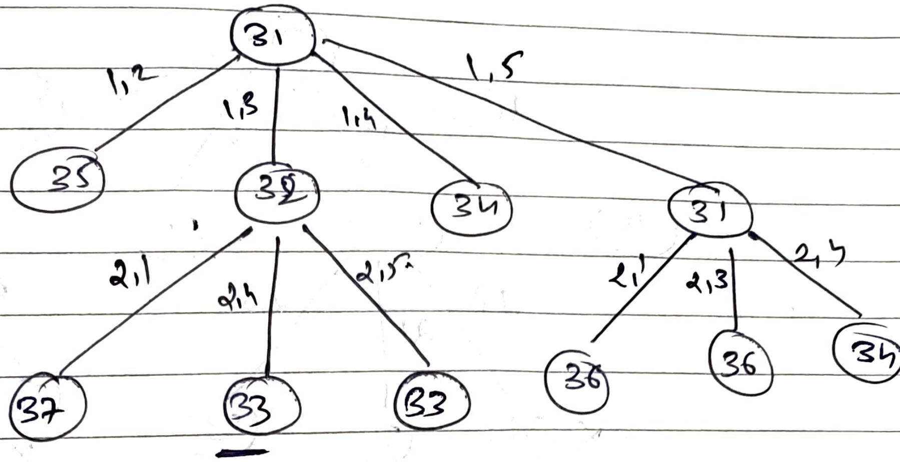
Also remove 4,2

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ 3 & 8 & 10 & - \\ 4 & 9 & - & 8 \\ 5 & 7 & 6 & 9 \end{array}$$

Add Row min $= 8+8+6 = 22$

Add $d(1,5)+d(2,4) = 7+5$

$= 34$



for 2-1 — Remove 2nd row 1st column
d-1-3 → Remove 3,2

$$\begin{bmatrix} \cancel{-} & \cancel{10} & \cancel{8} & \cancel{9} & \cancel{7} \\ 10 & - & 10 & 5 & 6 \\ 8 & 10 & - & 8 & 9 \\ 9 & 5 & 8 & - & 6 \\ 7 & 6 & 9 & 6 & - \end{bmatrix} \rightarrow \begin{array}{c|ccc} & 2 & 4 & 5 \\ \hline 3 & \cancel{10} & 8 & 9 \\ 4 & 5 & - & 6 \\ 5 & 6 & 6 & - \end{array}$$

Add Row min = 8 + 5 + 6 = 19
d (1,3) + d (2,1) = 8 + 10 = 18
37

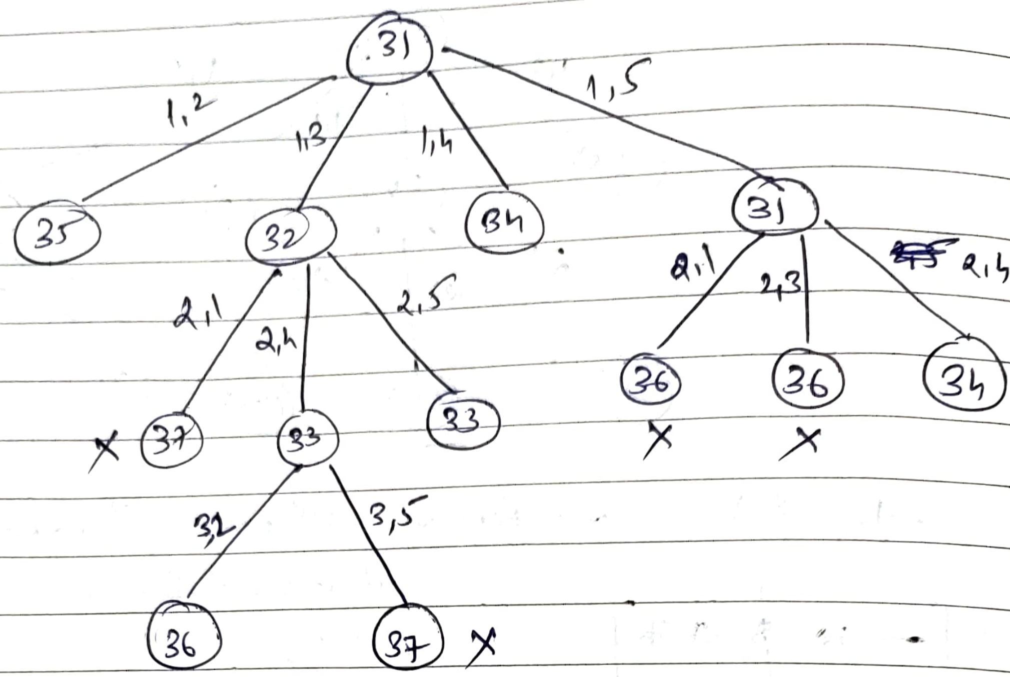
for 2-4 → Remove 2nd row 4th column
Remove 4,2

$$\begin{array}{c|ccc} & 1 & 2 & 5 \\ \hline 3 & 8 & 10 & 9 \\ 4 & 9 & \cancel{5} & 6 \\ 5 & 7 & 6 & - \end{array}$$

Add Row min = 8 + 6 + 6 = 20
d(1,3) + d(2,4) = 8 + 5 = 13
33

for 2-5 — Remove 2nd row 5th column
Remove 5,2
Add Row min = 8 + 5 + 6 = 19
d(1,3) + d(2,5) = 8 + 6 = 14
33

$$\begin{array}{c|ccc} & 1 & 2 & 4 \\ \hline 3 & 8 & 10 & 8 \\ 4 & 9 & 5 & - \\ 5 & 7 & - & 6 \end{array}$$



=(Now we have fixed (3) nodes and only (2) nodes to go.

so rather than finding Lower Bound we find upper Bound

(3,2) fixed - 1,3 , 2-4 3,2

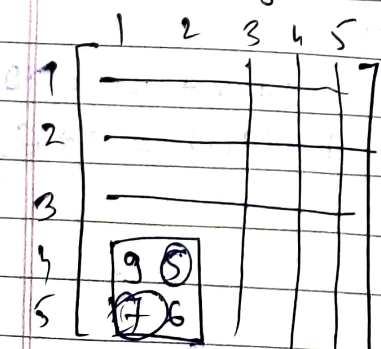
merge ⇒ 1-3-2-4-5-1

8 + 10 + 5 + 6 + 7 = 36

kill nodes ≥ 36 value.

(3,5) fixed 1,3 2-4 3-5

merge 1-3-5 2-4.



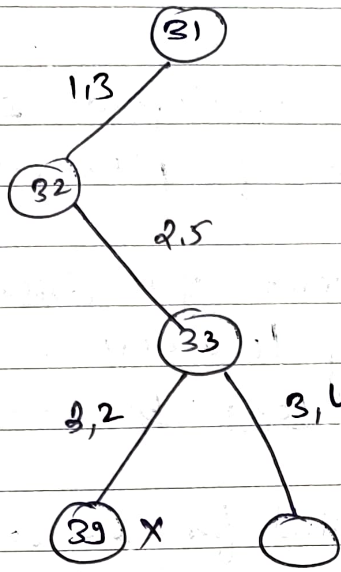
we don't want - 5,1 & 4,2

so we can only take 4-1 & 5-2

final path = 1-3-5-2-4-1

8 + 9 + 6 + 5 + 9 = 37

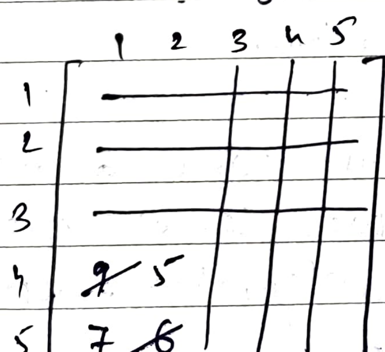
kill this node



Again we find upper Bound rather than lower Bound

(3,2) fixed ⇒ 1,3 2,5 3,2
merge ⇒ 1-3-2-5-4-1
8+10+6+6+9 = 39. ⟶ kill this

(3,4) fixed ⇒ 1,3 2,5 3,4
merge ⇒ 1-3-4 2-5
Remove 4,1 5,2
we can take 4,2 & 5,1
path 1-3-4-2-5-1
8+ 8+5+6+7 = 34



This kills all remaining nodes.

Final Ans → 34
Final Path → 1-3-4-2-5-1.