Backtracking and Branch & Bound

Saturday, December 10, 2022 10:40 AM

Backtracking -

- It is a problem solving strategy
- It uses a recursive approach
- Better version of brute force approach
- When we have set of choices and don't know which choice leads to solution, we use backtracking
- Almost every CSP can be solved using backtracking
- If the choices satisfies given constrains then they are marked as Partial solutions
- These partial solutions then explored using DFS
- If complete soln found then we have found one of the possible solutions
- If we do not get complete soln then we backtrack from the partial soln and explore another
- We can visualize backtracking using State Space Tree
- A node in a tree is Promising if it represents partial soln otherwise non-promising
- Ex. N-Queens problem, Graph coloring problem, TSP, etc.
- A soln is represented as a tuple which is a finite set of choices

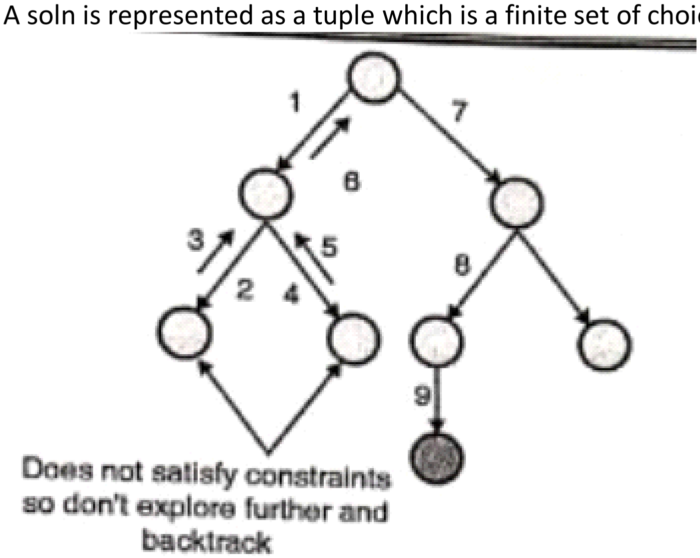


Fig. 5.1.1 : Process of backtracking

- For algorithm or control abstraction of backtracking consider refer example of N-queens

Constrains -

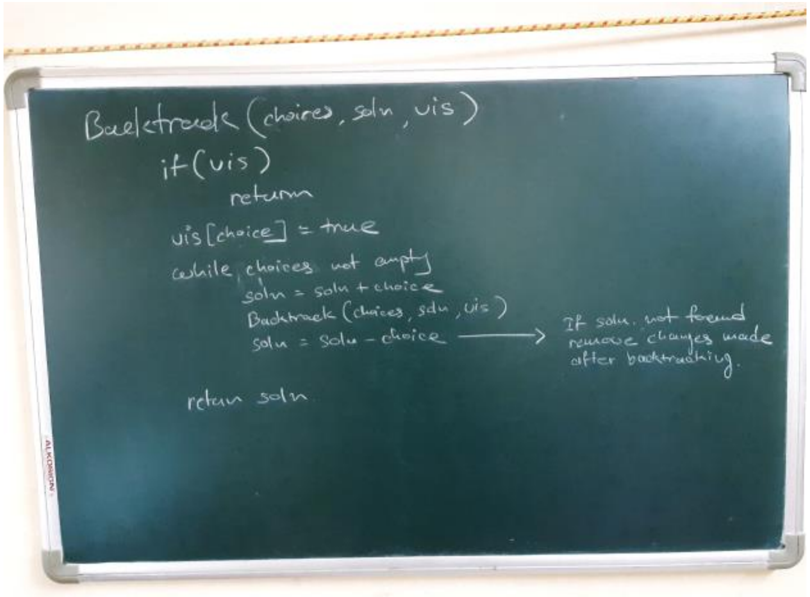
Implicit constraints -

These are the rules that states that how the selected elements in the tuple are related

Explicit constraints -

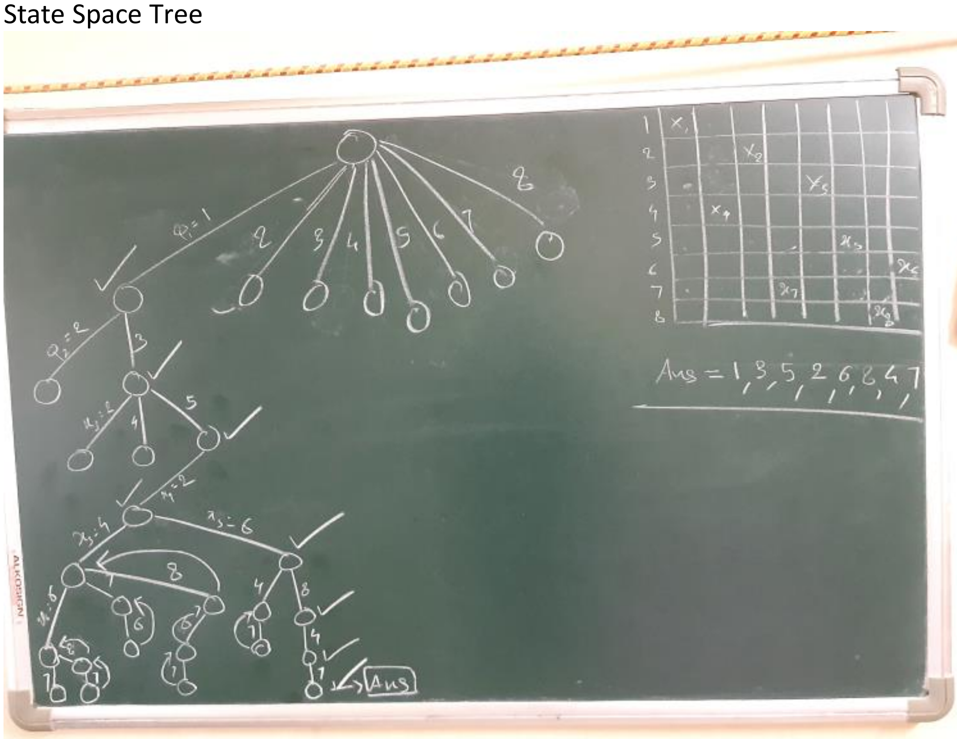
These are the rules that states that how to select an element from set of choices

Control abstraction for Backtracking -



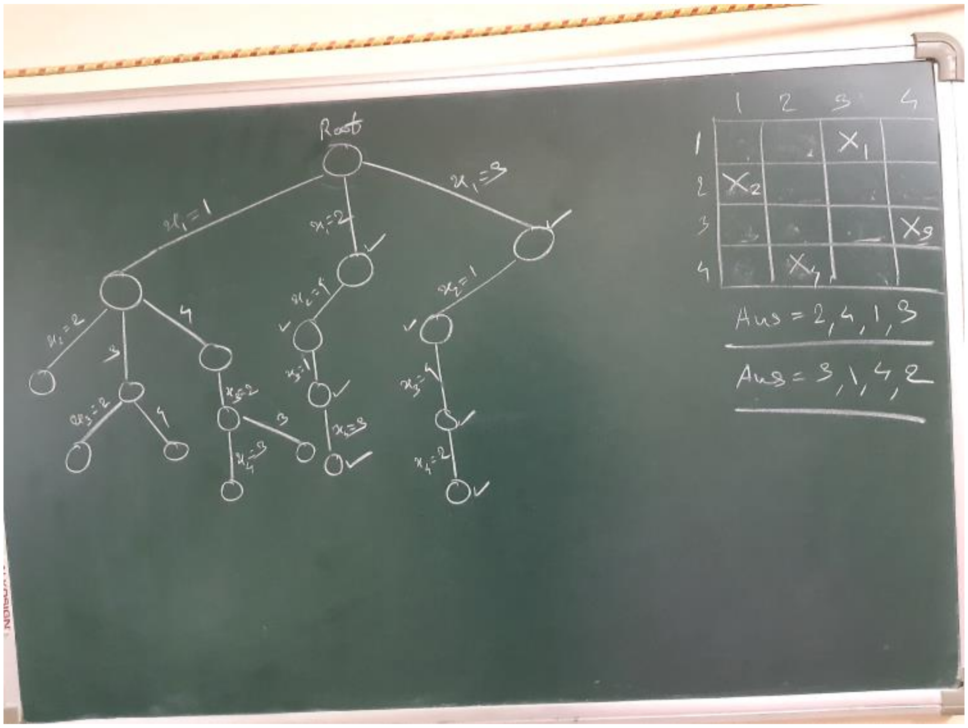
8 Queens -

State Space Tree



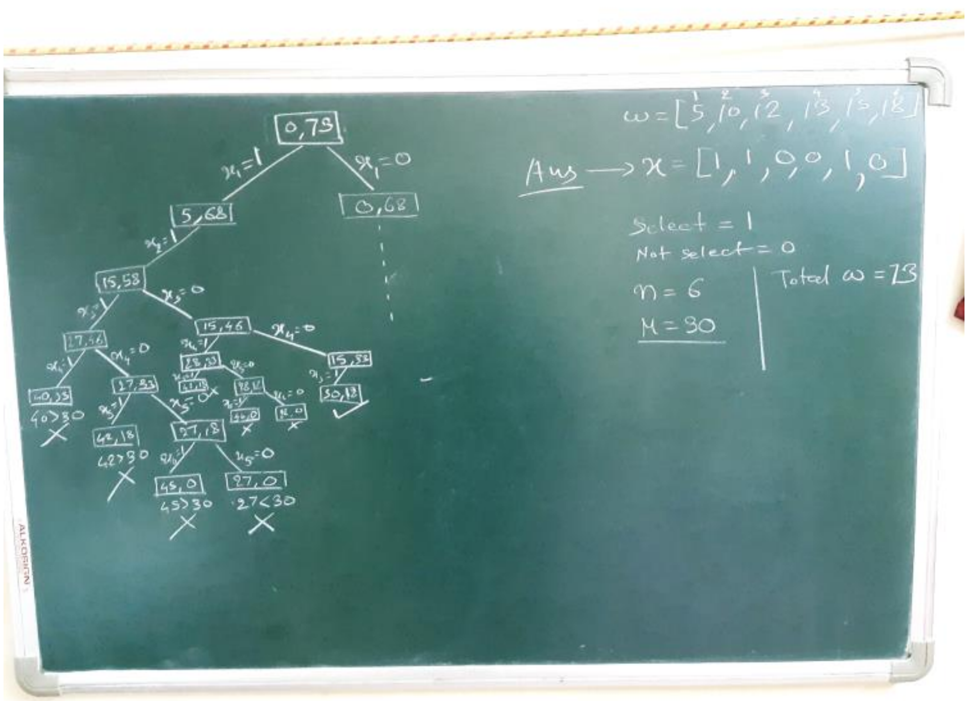
4 Queens -

State Space Tree

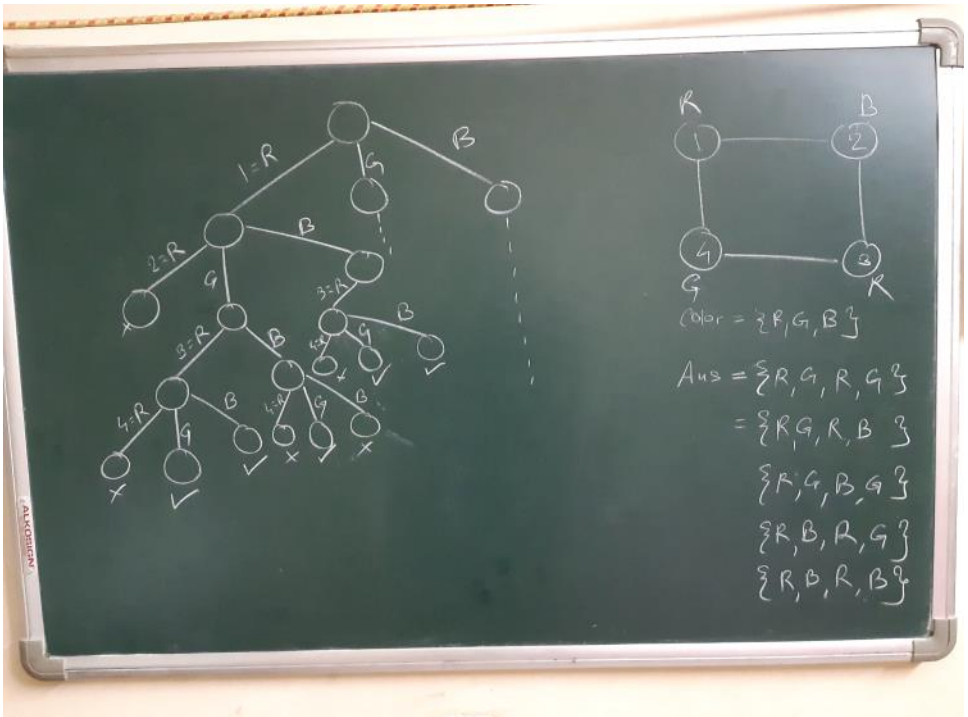


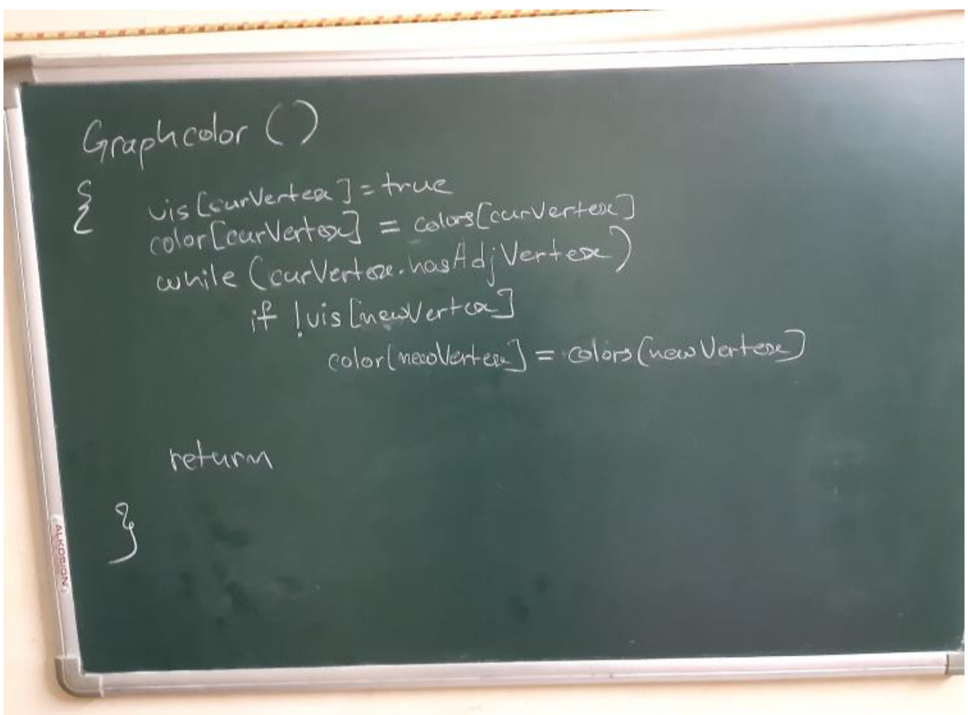
Sum of Subsets -

State Space Tree



Graph Coloring -

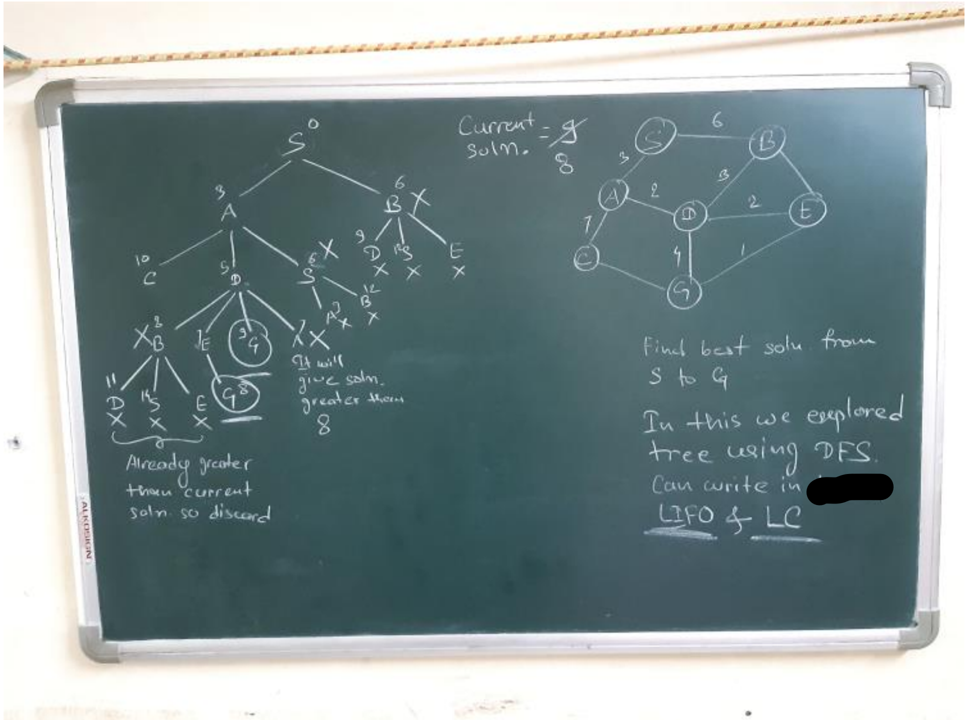


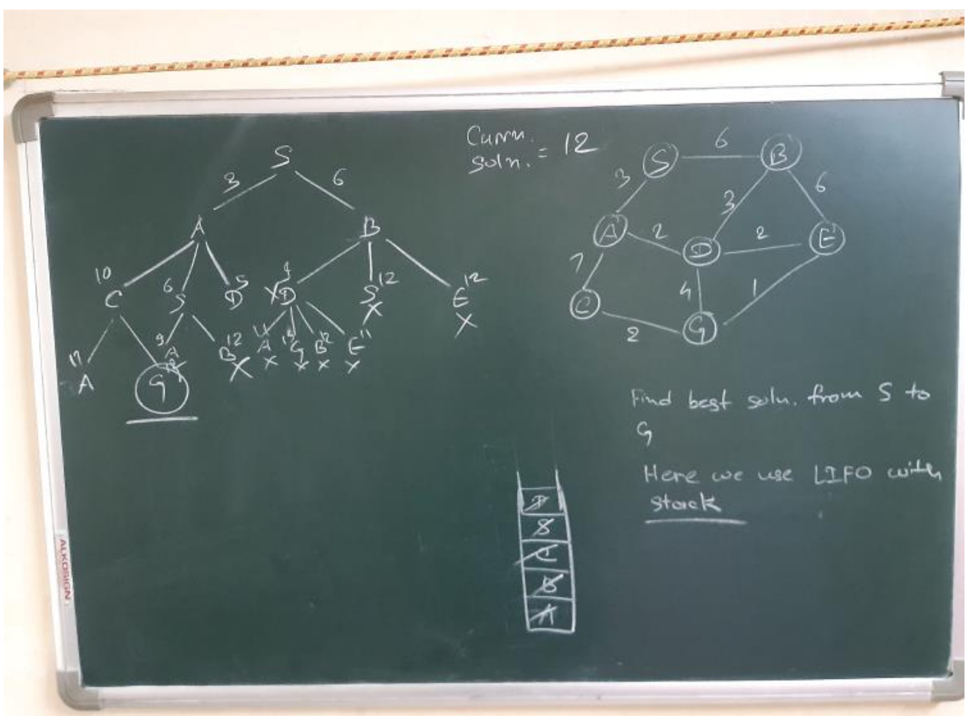


Branch and Bound -

- Very similar to backtracking
- Problem solving strategy
- Backtracking is used for decision problems while BB is used for optimization problems
- In this we limit our search to few branches of state space tree and apply some bounds or conditions to optimize the solution
- If a node having more cost than then we prune/discard that node or branch of the tree
- This is called as bounding
- Three strategies are used to solve BB problems FIFO, LIFO and LC

- FIFO uses queue to store the nodes of the state space tree
- LIFO uses stack to store the nodes
- In Branch and bound we only use BFS stategy
- Least Cost (LC) uses min cost of the node as the condition or bound to explore the node to find the solution
- Node having min cost is only explored to get an optimal solution.

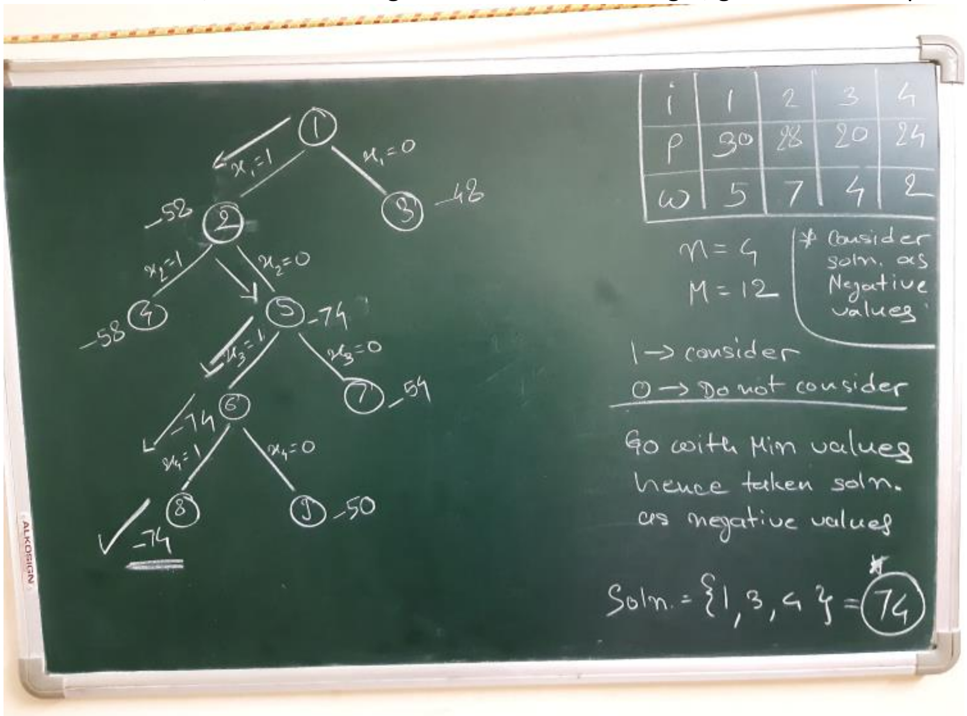




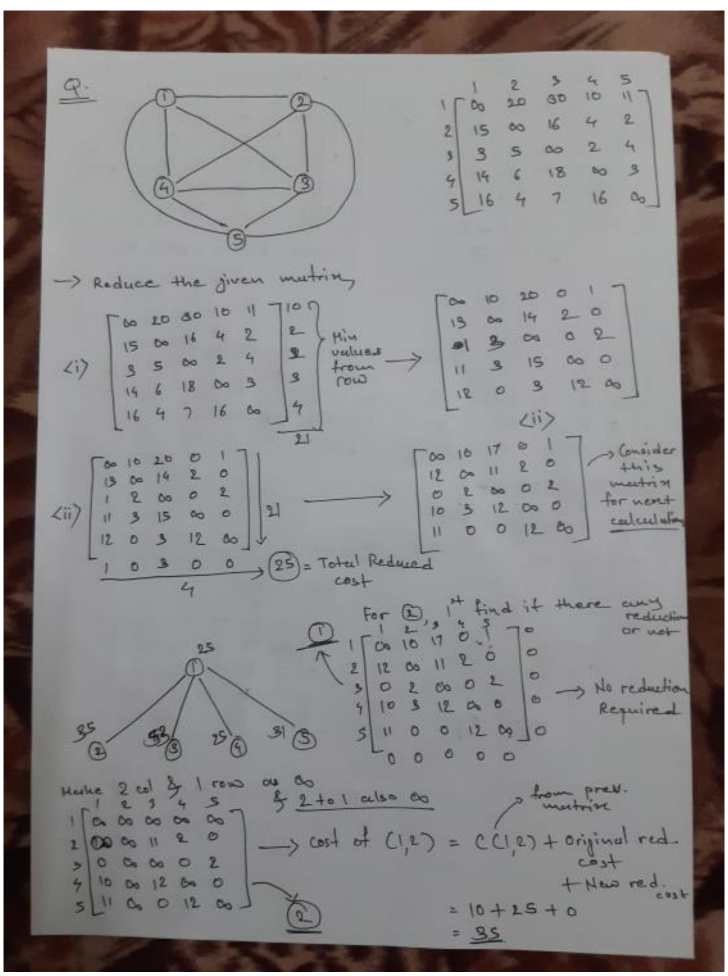
0/1 Knapsack using LCBB -

- We consider one node and move on with next consecutive node
- We will either select a node or not select it
- If selected then try to find soln by considering next nodes also until soln<M

- Give -ve value to soln because BB is used for minimization problems only
- Whichever value, either selecting a node or not selecting it, gives min value proceed with that path



Travelling Salesman Problem using BB -



For 3,

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 12 & \infty & \infty & 2 & 0 \\ 3 & \infty & 3 & \infty & 0 & 2 \\ 4 & 15 & 3 & \infty & \infty & 0 \\ 5 & 11 & 0 & \infty & 12 & \infty \\ & 11 & & & & \end{array} \longrightarrow \cancel{3}$$

$C(1,3) = 17 + 25 + 11$
$= 42 + 11 = 53$

For 4,

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 12 & \infty & 11 & \infty & 0 \\ 3 & 0 & 3 & \infty & \infty & 2 \\ 4 & \infty & 3 & 12 & \infty & 0 \\ 5 & 11 & 0 & 0 & \infty & \infty \end{array} \longrightarrow 4$$

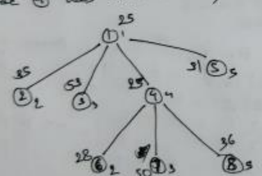
$cost\ C(1,4) = 0 + 25 + 0$
$= 25$

For 5,

$$\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & \infty & \infty & \infty & \infty & \\ 2 & 12 & \infty & 11 & 2 & \infty & 2 \\ 3 & 0 & 3 & \infty & 0 & \infty & \\ 4 & 15 & 3 & 12 & \infty & \infty & 3 \\ 5 & \infty & 0 & 0 & 12 & \infty & \end{array} \longrightarrow \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 10 & \infty & 9 & 0 & \infty \\ 3 & 0 & 3 & \infty & 0 & \infty \\ 4 & 12 & 0 & 9 & \infty & \infty \\ 5 & \infty & 0 & 0 & 12 & \infty \end{array} \longrightarrow 5$$

$C(1,5) = 1 + 25 + 5$
$= 26 + 5 = 31$

Node 4 has min value, Consider 4 Matrix.



for (4,2),

$$\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & 11 & \infty & 0 & 0 \\ 3 & 0 & \infty & \infty & \infty & 2 & 0 \\ 4 & \infty & \infty & \infty & \infty & \infty & 0 \\ 5 & 11 & \infty & 0 & \infty & \infty & 0 \\ \hline & 0 & 0 & 0 & 0 & 0 & \end{array} \longrightarrow 6$$

$c(4,2) = c(4,2) + c(4) + 0$
$= 3 + 25 + 0$
$= 28$

for (4, 3),

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 12 & \infty & \infty & \infty & 0 \\ 3 & \infty & 3 & \infty & \infty & 2 \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & 11 & 0 & \infty & \infty & \infty \end{array} \; 2 \quad \longrightarrow \quad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 1 & \infty & \infty & \infty & 0 \\ 3 & \infty & 1 & \infty & \infty & 0 \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & 0 & 0 & \infty & \infty & \infty \end{array} \; 7$$

11 → 13

$c(4,3) + c(4) + 13$
$= 12 + 25 + 13 = \underline{\underline{50}}$

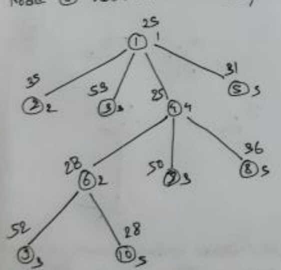
for (4, 5),

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 12 & \infty & 11 & \infty & \infty \\ 3 & 0 & 3 & \infty & \infty & \infty \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & \infty & 0 & 0 & \infty & \infty \end{array} \; 11 \quad \longrightarrow \quad \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & 1 & \infty & 0 & \infty & \infty \\ 3 & 0 & 3 & \infty & \infty & \infty \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & \infty & 0 & 0 & \infty & \infty \end{array} \; 8$$

11

$c(4,5) + c(4) + 11$
$= 0 + 25 + 11 = \underline{36}$

Node 6 has min value, consider 6 Matrix



for (2, 3),

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty & \infty & \text{[illegible]} \\ 4 & \infty & \infty & \infty & \infty & \text{[illegible]} \\ 5 & 11 & \infty & \infty & \infty & \infty \end{array} \; \begin{matrix} 2 \\ 11 \end{matrix} \; \downarrow 13$$

$$\downarrow$$

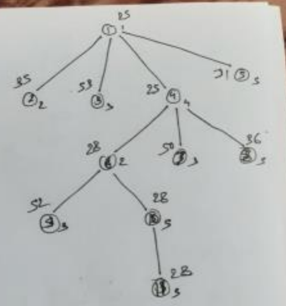
$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty & \infty & 0 \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & 0 & \infty & \infty & \infty & \infty \end{array} \; 9$$

$c(2,3) + c(2) + 13$
$= 11 + 28 + 13$
$= \underline{52}$

for (2,5),

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & \infty & \infty \\ 3 & 0 & \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & \infty & \infty & 0 & \infty & \infty \end{array} \rightarrow 10$$

$$c(2,5) + c(2) + 0$$
$$= 0 + 28 + 0$$
$$= \underline{28}$$

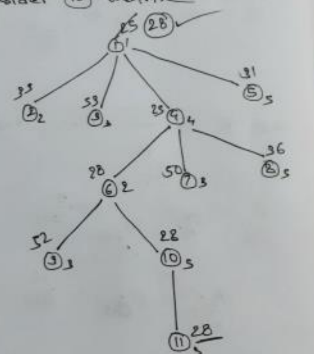


Node (10) has min value, consider (10) matrix

for (5,3)

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & \infty & \infty & \infty & \infty \\ 2 & \infty & \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & \infty & \infty & \infty \\ 5 & \infty & \infty & \infty & \infty & \infty \end{array}$$

$$c(5,3) + c(5) + 0$$
$$= 0 + 28 + 0$$
$$= \underline{28}$$



All other nodes have values greater than 28 hence the final ans is 28 & path is

[1 -> 4 -> 2 -> 5 -> 3 -> 1]