0/1 knapsack problem using FIFO Branch & Bound:-

→ 0/1 knapsack problem is a maximization problem but FIFO Branch and Bound is a minimization problem. To convert this maximization problem into minimization problem, change the +ve profits into -ve profits by putting the -ve sign infront of each profit.

→ In 0/1 knapsack FIFO BB problem, we have to calculate two bounds.

1. lower bound ($\hat{c}$)
   - lower bound allows fractions.
2. Upper bound ($\hat{u}$)
   - upper bound doesnot allow fractions.

→ while solving the 0/1 knapsack problem using FIFO BB, the following steps we have to follow.

1) Set the node 1 upperbound as global upper bound ($\hat{U}$)

2) If the lower bound of any node is greater than the global upperbound, kill that node, otherwise the node can be expanded.

3) If any node upperbound is less than the global upperbound, then update the global upper bound value.

4) If the objects are considered one by one, some of those objects greater than the capacity of the knapsack, then the node is killed. Hence, that node has infeasible solution.

1) Draw the state space tree for the following 0/1 knapsack instance using FIFO BB

$n = 4 \quad (P_1, P_2, P_3, P_4) = (10, 10, 12, 18)$

$(w_1, w_2, w_3, w_4) = (2, 4, 6, 9) \quad M = 15$

0/1 knapsack problem is a maximization problem but we require FIFO BB knapsack problem is a minimization problem.

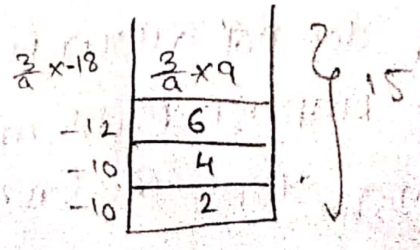
Hence, we convert the +ve profits into -ve profits. Hence, the problem is converted from maximization problem to minimization problem.

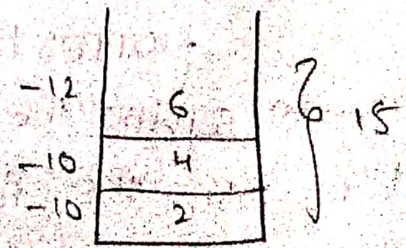
$(P_1, P_2, P_3, P_4) = (-10, -10, -12, -18)$

$(w_1, w_2, w_3, w_4) = (2, 4, 6, 9)$

$M = 15 \quad n = 4$

for node 1:

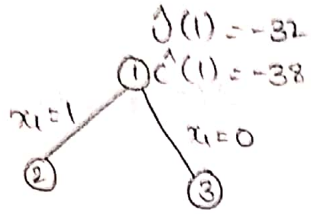




$\hat{c}(1) = -10-10-12-6$
$= -38$

$\hat{u}(1) = -10-10-12$
$= -32$

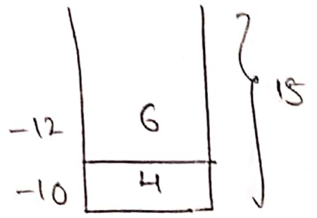
$\hat{u}$ = -32 as global upper bound. $-38 > -32$
False
so extend node(1)

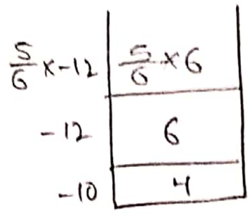


for node 2:-

$\hat{c}(2) = -38$
$\hat{u}(2) = -32$

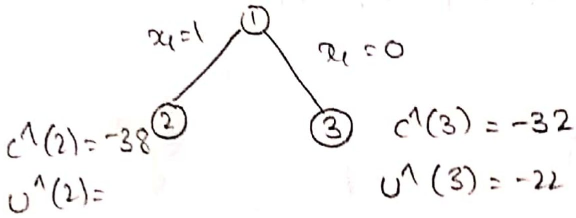
for node 3:- $x_1 = 0$



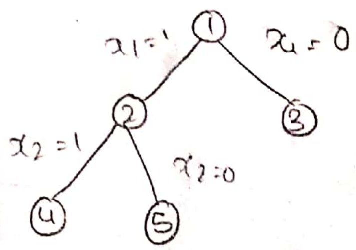


$\hat{u}(3) = -10-12 = -22$

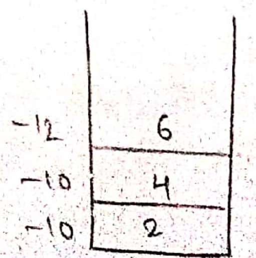
$\hat{c}(3) = -10-12-10 = -32$

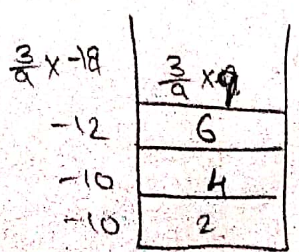


$-38 > -32$ False. Extend node 2



For node 4:- $x_1 = 1$, $x_2 = 1$

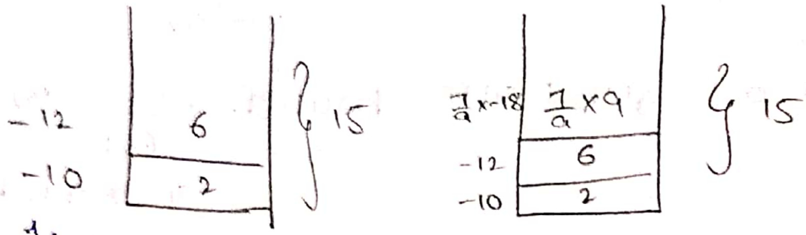




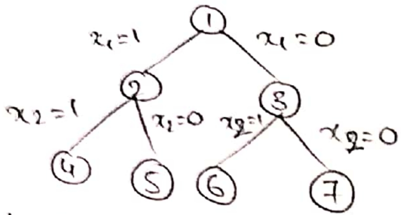
$\hat{u}(4) = -12-10-10$
$= -32$

$\hat{c}(4) = -10-10-12-6$
$= -38$

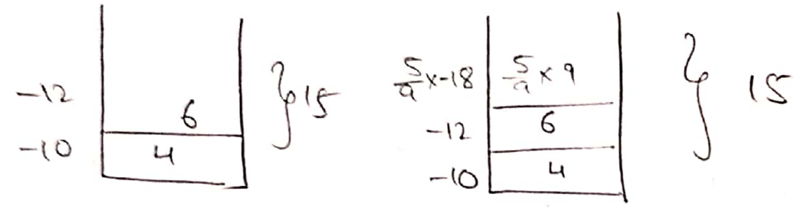
For node 5:- $x_1 = 1, \; x_2 = 0$



$\hat{c}(5) = -12-10 = -22$ $\qquad$ $c(5) = -12-10-14 = -36$

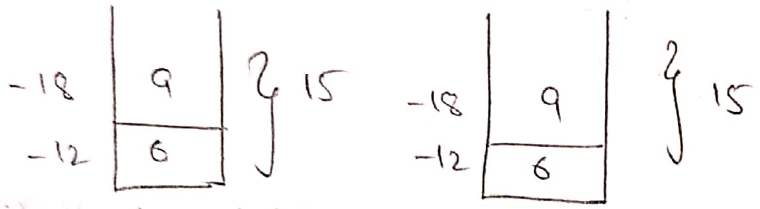


For node 6:- $x_1 = 0, \; x_2 = 1$ $\qquad$ $-32 > -32 \rightarrow$ False.



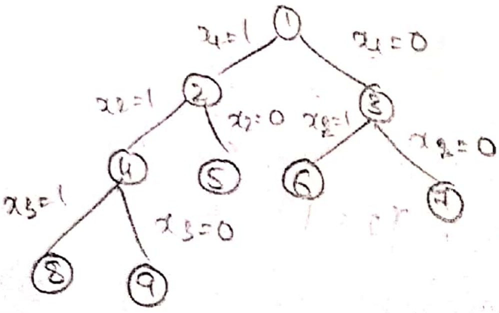
$\hat{c}(6) = -12-10 = -22$ $\qquad$ $c(6) = -10-12-10 = -32$

For node 7:- $x_1 = 0, \; x_2 = 0$

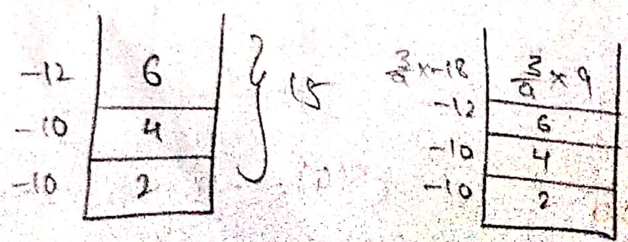


$\hat{c}(7) = -18-12$

$= -30$ $\qquad$ $c(7) = -12-18 = -30$



For node 8:-

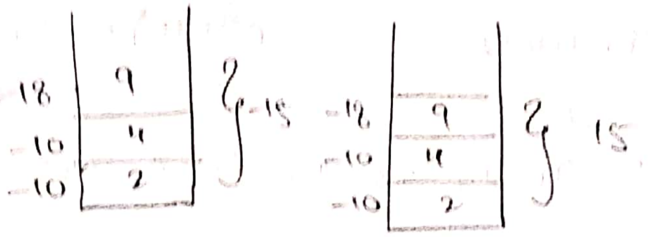


Scanned with CamScanner

$\hat{u}(8) = -10-10-12$ $\quad\quad$ $\hat{c}(8) = -10-10-12-6$

$= -32$ $\quad\quad$ $= -38$

for node 9: $x_1 = 1,\ x_2 = 1,\ x_3 = 0$

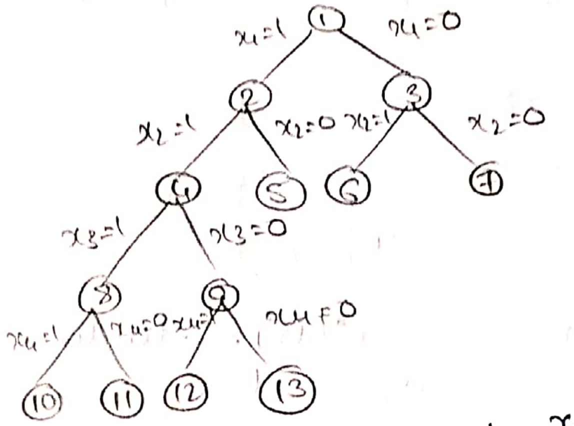


$\hat{u}(9) = -10-10-18$ $\quad\quad$ $\hat{c}(9) = -10-10-18$

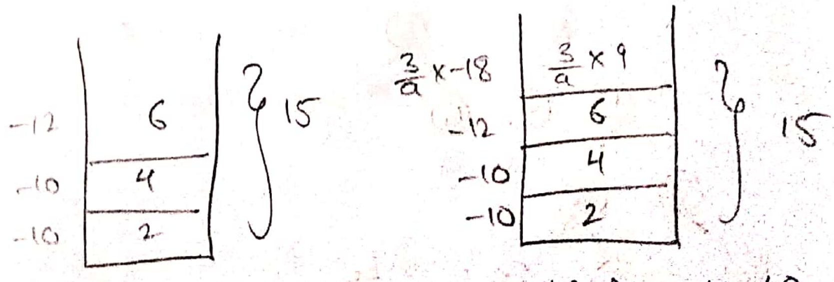
$= -38$ $\quad\quad$ $= -38$

Here the upper bound -38 is greater than global upperbound -32. So, we have to update the global upper bound.

$\hat{u} = -38$ as global upper bound



for node 10:- $x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 1$



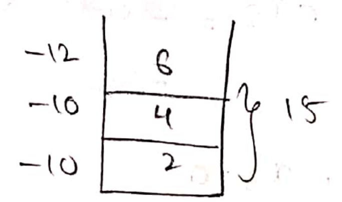
$\hat{u}(10) = -10-10-12$ $\quad\quad$ $\hat{c}(10) = -10-10-12-6$

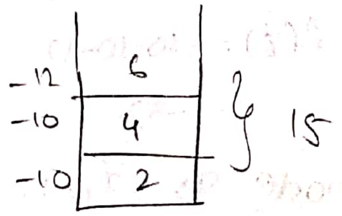
$= -32$ $\quad\quad$ $= -38$

Infeasible solution

for node 11:-

$x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0$

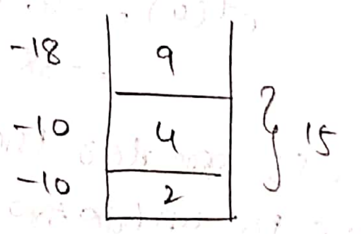


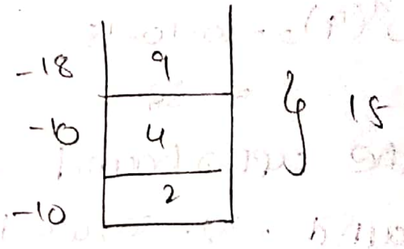


$u^\wedge(11) = -10-10-12$
$= -32$

$c^\wedge(11) = -10-10-12$
$= -32$

For node 12:- $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 1$

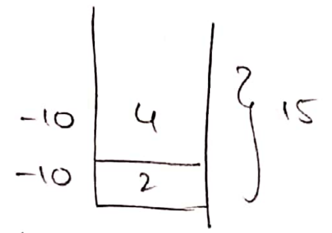


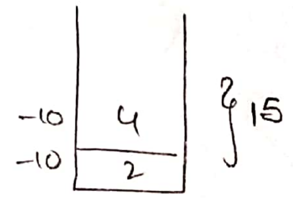


$u^\wedge(12) = -10-10-18$
$= -38$

$c^\wedge(12) = -10-10-18$
$= -38$

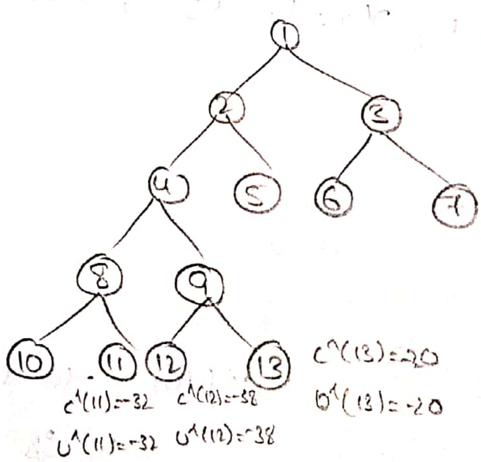
For node 13:- $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$





$u^\wedge(13) = -10-10$
$= -20$

$c^\wedge(13) = -10-10 = -20$



→ If the lower bound is greater than global upper bound kill that node.

→ If lower bound is less than global upper bound ~~the~~ extend that node.

Here the active node is 12

solution vector $[x_1, x_2, x_3, x_4] = [1, 1, 0, 1]$

max profit $= -10 - 10 - 18 = -38$

weight $= 2 + 4 + 0 + 9 = 15$

Now, convert the minimization problem into maximization problem by changing -ve profit sighs to +ve profit sighs

solution vector $[x_1, x_2, x_3, x_4] = [1, 1, 0, 1]$

max profit $= 10 + 10 + 18 = 38$

2) Draw the portion of the state space tree generated by the FIFO B B technique for the knapsack instance of $n = 5$, $m = 12$, $(P_1 - P_5) = (10, 15, 6, 8, 4)$ $(w_1 - w_5) = (4, 6, 3, 4, 2)$

3) Apply the least cost Branch and Bound technique for the following adjacency matrix using travelling sales person problem.

$$\begin{bmatrix} \infty & 7 & 3 & 12 & 8 \\ 3 & \infty & 6 & 14 & 9 \\ 5 & 8 & \infty & 6 & 18 \\ 9 & 3 & 5 & \infty & 11 \\ 18 & 14 & 9 & 8 & \infty \end{bmatrix}$$

Scanned with CamScanner

UNIT - III

Dynamic programming:

* Travelling sales person problem:-(TSP)

→ Let $G=(V,E)$ be a directed graph with edge cost c_{ij} where 'V' denotes the set of vertices and 'E' denotes the set of Edges. The edges are given along with their costs c_{ij} the cost $c_{ij} > 0$ for all i and j.

$$\text{cost}[i,j] = \begin{cases} 0 & \text{if } i=j \\ c_{ij} & (i,j) \in E \\ \infty & (i,j) \notin E \end{cases}$$

→ A tour for the graph G is a directed cycle that includes every vertex in 'V' exactly once. The cost of the tour is the sum of the cost of the edges on the tour and the cost must be minimum.

→ The travelling salesperson problem objective is to find out the tour of the minimum cost.

→ Let us assume that the tour starts from vertex '1' and ends at vertex 1. Every tour consists of an edge $(1,k)$ for some $k \in V-\{1\}$ and a path from vertex k to vertex 1.

→ The path from vertex k to vertex 1 goes through each vertex in $V-\{1,k\}$ exactly once.

→ Let $G(i,S)$ be the length of a shortest path starting at vertex i, going through all the vertices. S and terminating edge vertex 1.

→ $g(i,S) = \min\limits_{j \in S} \{c_{ij} + g(j, S-\{j\})\}$

where $S = V-\{i\}$

Example:-

→ suppose we have route a postal van to pickup mails from mail boxes located at 'n' different locations in a city.

→ n+1 vertex graph can be used to represent this situation. one vertex represents the post office from which the postal van starts and to which it must return.

→ Edge (i,j) is assigned a cost equal to the distance from location i to location j. The root taken by the postal van is a tour and we are interested in finding a tour of minimum cost.

→ let us consider a graph with 4 vertices, whose cost adjacency matrix of the graph given below

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

Find out the optimal tour cost.

Sol

$$g(i,S) = \min_{j \in S} \{ c_{ij} + g(j, S - \{j\}) \}$$

i) $|S| \cancel{\neq} = \phi$

$$g(1,\phi) = c_{11} = c(i,1) = c(1,1) = 0$$

$$g(2,\phi) = c_{21} = c(2,1) = 5$$

$$g(3,\phi) = c_{31} = c(3,1) = 6$$

$$g(4,\phi) = c_{41} = c(4,1) = 8$$

2) $|S| = 1$

$g(2, \{3\}) = \min_{j \in 3} \{C_{23} + g(3, \{3\} - \{3\})\}$

$= \min \{\{ 9 + g(3, \phi)\}$

$= \min \{9+6\}$

$= 15$

$g(2, \{4\}) = \min_{j \in 4} \{C_{24} + g(4, \{4\} - \{4\})\}$

$= \min \{10 + g(4, \phi)\}$

$= \min \{10+8\}$

$= 18$

$g(3, \{2\}) = \min_{j \in 2} \{C_{32} + g(2, \{2\} - \{2\})\}$

$= \min \{13 + g(2, \phi)\}$

$= \min \{13+5\}$

$= 18$

$g(3, \{4\}) = \min_{j \in 4} \{C_{34} + g(4, \{4\} - \{4\})\}$

$= \min \{12 + g(4, \phi)\}$

$= \min \{12+8\}$

$= 20$

$g(4, \{2\}) = \min_{j \in 2} \{C_{42} + g(2, \{2\} - \{2\})\}$

$= \min \{8 + g(2, \phi)\}$

$= \min \{8+5\}$

$= 13$

$g(4, \{3\}) = \min_{j \in 3} \{C_{43} + g(3, \{3\} - \{3\})\}$

$= \min \{9 + g(3, \phi)\}$

$= \min \{9+6\}$

$= 15$

Scanned with CamScanner

3) $|S| = 2$

$$g(2,\{3,4\}) = \min_{j\in 3,4} \begin{cases} c_{2,3} + g(3,\{3,4\} - \{3\}), \\ c_{2,4} + g(4,\{3,4\} - \{4\}) \end{cases}$$

$$= \min\{9 + g(3,\{4\}),\ 10 + g(4,\{3\})\}$$

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

$$= \min\{29,\ 15\}$$

$$= 25$$

$$g(2,\{4,3\}) = \min_{j\in 4,3} \{ c_{2,4} + g(4,\{3,4\} - \{4\}),\ c_{2,3} + g(3,\{3,4\} - \{3\}) \}$$

$$= \min\{10 + g(4,\{3\}),\ 9 + g(3,\{4\})\}$$

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

$$= \min\{25,\ 29\}$$

$$= 25$$

$$g(3,\{2,4\}) = \min_{j\in 2,4} \{ c_{32} + g(2,\{2,4\} - \{2\}),\ c_{34} + g(4,\{2,4\} - \{4\}) \}$$

$$= \min\{13 + g(2,\{4\}),\ 12 + g(4,\{2\})\}$$

$$= \min\{13+18,\ 12+13\}$$

$$= \min\{32,\ 25\}$$

$$= 25$$

$$g(3,\{4,2\}) = \min_{j\in 4,2} \{ c_{34} + g(4,\{2,4\} - \{4\}),\ c_{32} + g(2,\{2,4\} - \{2\}) \}$$

$$= \min\{12 + g(4,\{2\}),\ 13 + g(2,\{4\})\}$$

$$= \min\{12+13,\ 13+18\}$$

$$= \min\{25,\ 32\}$$

$$= 25$$

$$g(4,\{2,3\}) = \min_{j\in 2,3} \{ c_{42} + g(2,\{2,3\} - \{2\}),\ c_{43} + g(3,\{2,3\} - \{3\}) \}$$

$$= \min\{8 + g(2,\{3\}),\ 9 + g(3,\{2\})\}$$

$$= \min\{8+15,\ 9+18\} = \min\{23,\ 27\}$$

$$= 23$$

$$g(4,\{3,2\}) = \min_{j \in 3,2} \{ c_{43} + g(3, \{2,3\} - \{3\}), \; c_{42} + g(2, \{2,3\} - \{2\}) \}$$

$$= \min\{9 + g(3,\{2\}), 8 + g(2,\{3\})\}$$

$$= \min\{9+18, 8+15\}$$

$$= \min\{27, 23\}$$

$$= 23$$

4) $|S| = 3$

$$g(1,\{2,3,4\}) = \min_{j \in 2,3,4} \begin{cases} c_{12} + g(2, \{2,3,4\} - \{2\}) \\ c_{13} + g(3, \{2,3,4\} - \{3\}) \\ c_{14} + g(4, \{2,3,4\} - \{4\}) \end{cases}$$

$$= \min \begin{cases} 10 + g(2, \{3,4\}) \\ 15 + g(3, \{2,4\}) \\ 20 + g(4, \{2,3\}) \end{cases}$$

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

$$= \min\{35, 40, 43\}$$

$$= 35$$

Optimal tour = 35

Path = 1-2-4-3-1 with cost = 35

2)
$$\begin{array}{c|c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 20 & 30 & 10 & 11 \\ 15 & 0 & 16 & 4 & 2 \\ 3 & 5 & 0 & 2 & 4 \\ 19 & 6 & 18 & 0 & 3 \\ 16 & 4 & 7 & 16 & 0 \end{bmatrix} \end{array}$$

1. $|S| = \phi$

$g(1,\phi) = c_{11} = c(1,1) = c(1,1) = 0$

$g(2,\phi) = c_{21} = c(2,1) = 15$

$g(3,\phi) = c_{31} = c(3,1) = 3$

$g(4,\phi) = c_{41} = c(4,1) = 19$

$g(5,\phi) = c_{51} = c(5,1) = 16$

Scanned with CamScanner

2. $|S| = 1$

$g(2, \{3\}) = \min_{j \in 3} \{ c_{2,3} + g(3, \{3\} - \{3\}) \}$

$= \min \{ 16 + g(3, \phi) \}$

$= \min \{ 16 + 3 \}$

$= 19$

$g(2, \{4\}) = \min_{j \in 4} \{ c_{2,4} + g(4, \{4\} - \{4\}) \}$

$= \min \{ 4 + g(4, \phi) \}$

$= \min \{ 4 + 19 \}$

$= 23$

$g(2, \{5\}) = \min_{j \in 5} \{ c_{2,5} + g(5, \{5\} - \{5\}) \}$

$= \min \{ 2 + g(5, \phi) \}$

$= \min \{ 2 + 16 \}$

$= 18$

$g(3, \{2\}) = \min_{j \in 2} \{ c_{3,2} + g(2, \{2\} - \{2\}) \}$

$= \min \{ 5 + g(2, \phi) \}$

$= \min \{ 5 + 15 \}$

$= 20$

$g(3, \{4\}) = \min_{j \in 4} \{ c_{3,4} + g(4, \{4\} - \{4\}) \}$

$= \min \{ 2 + g(4, \phi) \}$

$= \min \{ 2 + 19 \}$

$= 21$

$g(3, \{5\}) = \min_{j \in 5} \{ c_{3,5} + g(5, \{5\} - \{5\}) \}$

$= \min \{ 4 + g(5, \phi) \}$

$= \min \{ 4 + 16 \}$

$= 20$

$g(4, \{2\}) = \min_{j \in 2} \{ c_{4,2} + g(2, \{2\} - \{2\}) \}$

$= \min \{ 6 + g(2, \phi) \}$

$= \min \{ 6 + 15 \}$

$= 21$

$$g(4,\{3\}) = \min_{j\in 3} \{ c_{4,3} + g(3, \{3\} - \{3\}) \}$$
$$= \min \{ 18 + g(3, \phi) \}$$
$$= \min \{ 18 + 3 \}$$
$$= 21$$

$$g(4,\{5\}) = \min_{j\in 5} \{ c_{4,5} + g(5, \{5\} - \{5\}) \}$$
$$= \min \{ 3 + g(5, \phi) \}$$
$$= \min \{ 3 + 16 \}$$
$$= 19$$

$$g(5,\{2\}) = \min_{j\in 2} \{ c_{5,2} + g(2, \{2\} - \{2\}) \}$$
$$= \min \{ 4 + g(2, \phi) \}$$
$$= \min \{ 4 + 15 \}$$
$$= 19$$

$$g(5,\{3\}) = \min_{j\in 3} \{ c_{5,3} + g(3, \{3\} - \{3\}) \}$$
$$= \min \{ 7 + g(3, \phi) \}$$
$$= \min \{ 7 + 3 \}$$
$$= 10$$

$$g(5,\{4\}) = \min_{j\in 4} \{ c_{5,4} + g(4, \{4\} - \{4\}) \}$$
$$= \min \{ 16 + g(4, \phi) \}$$
$$= \min \{ 16 + 19 \}$$
$$= 35$$

3. $|S| = 2$

$$g(2,\{3,4\}) = \min_{j\in 3,4} \{ c_{2,3} + g(3, \{3,4\} - \{3\}),\ c_{2,4} + g(4, \{3,4\} - \{4\}) \}$$
$$= \min \{ 16 + g(3, \{4\}),\ 4 + g(4, \{3\}) \}$$
$$= \min \{ 16 + 21,\ 4 + 21 \}$$
$$= \min \{ 37, 25 \}$$
$$= 25$$

$g(2,\{4,3\}) = \min\{25, 37\}$

$= 25$

$g(2,\{4,5\}) = \min_{j\in 4,5} \{c_{2,4} + g(4,\{4,5\}-\{4\}), c_{2,5} + g(5,\{4,5\}, -\{5\})\}$

$= \min\{4+g(4,\{5\}), 2+g(5,\{4\})\}$

$= \min\{4+19, 2+35\}$

$= \min\{23, 37\}$

$= 23$

$g(2,\{5,4\}) = \min\{37, 23\}$

$= 23$

$g(2,\{3,5\}) = \min_{j\in 3,5} \{c_{2,3} + g(3,\{3,5\}-\{3\}), c_{2,5} + g(5,\{3,5\}-\{5\})$

$= \min\{16+g(3,\{5\}, 2+g(5,\{3\})\}$

$= \min\{16+20, 2+10\}$

$= \min\{36, 12\}$

$= 12$

$g(2,\{5,3\}) = \min\{12, 36\}$

$= 12$

$g(3,\{2,4\}) = \min_{j\in 2,4} \{c_{3,2} + g(2,\{2,4\}-\{2\}), c_{3,4} + g(4,\{2,4\}-\{4\})\}$

$= \min\{5+g(2,\{4\}), 2+g(4,\{2\})\}$

$= \min\{5+23, 2+21\}$

$= \min\{28, 23\}$

$= 23$

$g(3,\{4,2\}) = \min(23, 28\}$

$= 23$

$g(3,\{2,5\}) = \min_{j\in 2,5} \{c_{3,2} + g(2,\{2,5\}-\{2\}), c_{3,5} + g(5, \{2,5\}-\{5\})\}$

$= \min\{5+g(2,\{5\}), 4+g(5,\{2\})\}$

$= \min\{5+18, 4+19\}$

$= \min\{23, 23\}$

$= 23$

$g(3, \{5,2\}) = \min\{23, 23\}$

$= 23$

$g(3, \{4,5\}) = \min_{j \in 4,5} \{ c_{3,4} + g(4, \{4,5\} - \{4\}), c_{3,5} + g(5, \{4,5\} - \{5\}) \}$

$= \min\{2 + g(4, \{5\}), 4 + g(5, \{4\})\}$

$= \min\{2+19, 4+35\}$

$= \min\{21, 39\}$

$= 21$

$g(3, \{5,4\}) = \min\{39, 21\}$

$= 21$

$g(4, \{2,3\}) = \min_{j \in 2,3} \{ c_{4,2} + g(2, \{2,3\} - \{2\}), c_{4,3} + g(3, \{2,3\} - \{3\}) \}$

$= \min\{6 + g(2, \{3\}), 18 + g(3, \{2\})\}$

$= \min\{6+19, 18+20\}$

$= \min\{25, 38\}$

$= 25$

$g(4, \{3,2\}) = \min\{38, 25\}$

$= 25$

$g(4, \{2,5\}) = \min_{j \in 2,5} \{ c_{4,2} + g(2, \{2,5\} - \{2\}), c_{4,5} + g(5, \{2,5\} - \{5\}) \}$

$= \min\{6 + g(2, \{5\}), 3 + g(5, \{2\})\}$

$= \min\{6+18, 3+19\}$

$= \min\{24, 22\}$

$= 22$

$g(4, \{5,2\}) = \min\{22, 24\}$

$= 22$

$g(4, \{3,5\}) = \min_{j \in 3,5} \{ c_{4,3} + g(3, \{3,5\} - \{3\}), c_{4,5} + g(5, \{3,5\} - \{5\}) \}$

$= \min\{18 + g(3, \{5\}), 3 + g(5, \{3\})\}$

$= \min\{18+20, 3+10\} = \min\{38, 13\}$

$= 13$

Scanned with CamScanner

$g(4,\{5,3\}) = \min\{13, 32\}$

$= 13$

$g(5,\{2,3\}) = \min_{j \in 2,3} \{c_{5,2} + g(2, \{2,3\} - \{2\}), c_{5,3} + g(3, \{2,3\} - \{3\})\}$

$= \min\{4 + g(2,\{3\}), 7 + g(3,\{2\})\}$

$= \min\{4+19, 7+20\}$

$= \min\{23, 27\}$

$= 23$

$g(5, \{3,2\}) = \min\{27, 23\}$

$= 23$

$g(5,\{2,4\}) = \min_{j \in 2,4} \{c_{5,2} + g(2, \{2,4\} - \{2\}), c_{5,2} + g(4, \{2,4\} - \{4\})\}$

$= \min\{4 + g(2,\{4\}), 16 + g(4,\{2\})\}$

$= \min\{4+23, 16+21\}$

$= \min\{27, 37\}$

$= 27$

$g(5, \{4,2\}) = 27$

$g(5,\{3,4\}) = \min_{j \in 3,4} \{c_{5,3} + g(3, \{3,4\} - \{3\}), c_{5,4} + g(4, \{3,4\} - \{4\})\}$

$= \min\{7 + g(3,\{4\}), 16 + g(4,\{3\})\}$

$= \min\{7+21, 16+21\}$

$= \min\{28, 37\}$

$= 28$

$g(5, \{4,3\}) = \min\{37, 28\}$

$= 28$

4. $|S| = 3$

$g(2,\{3,4,5\}) = \min_{j \in 3,4,5} \{c_{2,3} + g(3, \{3,4,5\} - \{3\}), c_{2,4} + g(4, \{3,4,5\} - \{4\}), c_{2,5} + g(5, \{3,4,5\} - \{5\})\}$

$= \min\{16 + g(3, \{4,5\}), 4 + g(4, \{3,5\}),$
$2 + g(5, \{3,4\})\}.$

$= \min\{16+21, 4+13, 2+28\}$

$= \min\{37, 17, 30\}$

$= 17.$

$g(2, \{4,3,5\}) = 17$

$g(2, \{5,3,4\}), g(2, \{4,5,3\}), g(2, \{5,4,3\}),$

$g(2, \{5,3,4\}) = 17$

$g(3, \{2,4,5\}) = \min_{j \in 2,4,5} \{C_{3,2} + g(2, \{2,4,5\} - \{2\}),$
$C_{3,4} + g(4, \{2,4,5\} - \{4\}),$
$C_{3,5} + g(5, \{2,4,5\} - \{5\})\}$

$= \min\{5 + g(2, \{4,5\}), 2 + g(4, \{2,5\}), 4 + g(5, \{2,4\})$

$= \min\{5+23, 2+22, 4+27\}$

$= \min\{28, 24, 31\}$

$= 24$

$g(3, \{2,5,4\}), g(3, \{4,2,5\}), g(3, \{4,5,2\}), g(3, \{5,2,4\})$

$g(3, \{5,4,2\}) = 24$

$g(4, \{2,3,5\}) = \min_{j \in 2,3,5} \{C_{4,2} + g(2, \{2,3,5\} - \{2\}),$
$C_{4,3} + g(3, \{2,3,5\} - \{3\}),$
$C_{4,5} + g(5, \{2,3,5\} - \{5\})\}$

$= \min\{6 + g(2, \{3,5\}), 18 + g(3, \{2,5\}),$
$3 + g\{5, \{2,3\})\}$

$= \min\{6+12, 18+23, 3+23\}$

$= \min\{18, 41, 26\}$

$= 18$

$g(4, \{2,5,3\}), g(4, \{3,2,5\}), g(4, \{3,5,2\}),$

$g(4, \{5,2,3\}), g(4, \{5,3,2\}) = 18$

$g(5, \{2,3,4\}) = \min_{j \in 2,3,4} \{ c_{5,2} + g(2, \{2,3,4\} - \{2\}),$

$c_{5,3} + g(3, \{2,3,4\} - \{3\}),$

$c_{5,4} + g(4, \{2,3,4\} - \{4\})\}$

$= \min\{4 + g(2, \{3,4\}),\ 7 + g(3, \{2,4\}),\ 16 + g(4, \{2,3\})\}$

$= \min\{4+25,\ 7+23,\ 16+25\}$

$= \min\{29, 30, 41\}$

$= 29$

$g(5, \{2,4,3\}),\ g(5, \{3,2,4\}),\ g(5, \{3,4,2\}),\ g(5, \{4,2,3\})$

$g(5, \{4,3,2\}) = 29$

5. $|S| = 4$

$g(1, \{2,3,4,5\}) = \min_{j \in 2,3,4,5} \{ c_{12} + g(2, \{2,3,4,5\} - \{2\}),$

$c_{13} + g(3, \{2,3,4,5\} - \{3\}),$

$c_{14} + g(4, \{2,3,4,5\} - \{4\}),$

$c_{15} + g(5, \{2,3,4,5\} - \{5\})\}$

$= \min\{20 + g(2, \{3,4,5\}),\ 30 + g(3, \{2,4,5\}),$

$10 + g(4, \{2,3,5\}),\ 11 + g(5, \{2,3,4\})\}$

$= \min\{20+17,\ 30+24,\ 10+18,\ 11+29\}$

$= \min\{37, 54, 28, 40\}$

$= 28$

1-4-2-5-3-1 with cost 28.

* All pairs shortest path problem (Floyd's /warshall's Algorithm) :-

Let $G = (V, E)$ be a directed graph with n vertices. Let cost be the cost of a adjacency matrix for graph G such that $c(i,i) = 0$, $i \le j \le n$.

cost(i,j) be the length (or) cost of the edge (i,j) if $(i,j) \in E(G)$ and cost$(i,j) = \alpha$, if $i \ne j$ and $(i,j) \notin E(G)$.

All pairs shortest path problem is to determine a matrix A such that A(i,j) is the length of a shortest path from i to j.

* The time complexity of all pairs shortest path problem is $O(n^3)$ using the principle of optimality.

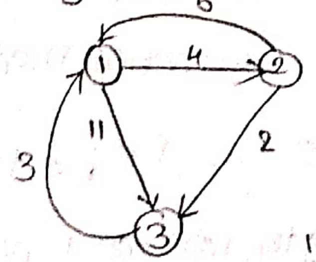
* let us Examine a shortest path from i to j in G (i ≠ j). This path originates at vertex i and goes through some intermediate vertices and terminates at vertex j. we can assume that this path contains no cycle. If 'k' is an intermediate vertex on this shortest path, then the subpath from i to k and k to j must be shortest paths respectively. Hence, the principle of optimality holds.

* If 'k' is an intermediate vertex with highest index, then i to k path is a shortest path in G going through no vertex with index > k-1.

* $A^k(i,j)$ is the length of a shortest path from i to j going through no vertex of index > k.

$$A^k(i,j) = \min\{A^{k-1}(i,j), A^{k-1}(i,k) + A^{k-1}(k,j)\}$$

consider the following graph and find out the shortest distance between every pair of vertices using dynamic programming.



Take k=0

$$A^0 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & \infty & 0 \end{array}$$

Take $k=1$

$A^1(i,j) = \min\{A^{1-1}(i,j),\ A^{1-1}(i,k) + A^{1-1}(k,j)\}$

$A^1(1,2) = \min\{A^{1-1}(1,2),\ A^{1-1}(1,1) + A^{1-1}(1,2)\}$

$= \min\{A^0(1,2),\ A^0(1,1) + A^0(1,2)\}$

$= \min\{4,\ 0+4\}$

$= \min\{4, 4\}$

$= 4$

$A^1(1,3) = \min\{A^0(1,3),\ A^0(1,1) + A^0(1,3)\}$

$= \min\{11,\ 0+11\}$

$= 11$

$A^1(2,1) = \min\{A^0(2,1),\ A^0(2,1) + A^0(1,1)\}$

$= \min\{6,\ 6+0\}$

$= 6$

$A^1(2,3) = \min\{A^0(2,3),\ A^0(2,1) + A^0(1,3)\}$

$= \min\{2,\ 6+11\}$

$= \min\{2, 17\}$

$= 2$

$A^1(3,1) = \min\{A^0(3,1),\ A^0(3,1) + A^0(1,2)\}$

$= \min\{3,\ 3+0\}$

$= 3$

$A^1(3,2) = \min\{A^0(3,2),\ A^0(3,1) + A^0(1,2)\}$

$= \min\{\alpha,\ 3+4\}$

$= \min\{\alpha, 7\}$

$= 7$

$$A^1 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

Scanned with CamScanner

Take $k=2$

$A^2(i,j) = \min\{A^{2-1}(i,j),\ A^{2-1}(i,k) + A^{2-1}(k,j)\}$

$A^2(1,2) = \min\{A^{2-1}(1,2),\ A^{2-1}(1,2) + A^{2-1}(2,2)\}$

$= \min\{A^1(1,2),\ A^1(1,2) + A^1(2,2)\}$

$= \min\{4,\ 4+0\}$

$= 4$

$A^2(1,3) = \min\{A^1(1,3),\ A^1(1,2) + A^1(2,3)\}$

$= \min\{11,\ 4+2\}$

$= 6$

$A^2(2,1) = \min\{A^1(2,1),\ A^1(2,2) + A^1(2,1)\}$

$= \min\{6,\ 0+6\}$

$= 6$

$A^2(2,3) = \min\{A^1(2,3),\ A^1(2,2) + A^1(2,3)\}$

$= \min\{2,\ 0+2\}$

$= 2$

$A^2(3,1) = \min\{A^1(3,1),\ A^1(3,2) + A^1(2,1)\}$

$= \min\{3,\ 7+6\}$

$= 3$

$A^2(3,2) = \min\{A^1(3,2),\ A^1(3,2) + A^1(2,2)\}$

$= \min\{7,\ 7+0\}$

$= 7$

$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

Take $k=3$

$A^3(i,j) = \min\{A^{3-1}(i,j),\ A^{3-1}(i,k) + A^{3-1}(k,j)\}$

$A^3(1,2) = \min\{A^2(1,2),\ A^2(1,3) + A^2(3,2)\}$

$= \min\{4,\ 6+7\}$

$= 4$

$A^3(1,3) = \min\{A^2(1,3),\ A^2(1,3) + A^2(3,3)\}$

$= \min\{6, 6+0\}$

$= 6$

$A^3(2,1) = \min\{A^2(2,1),\ A^2(2,3) + A^2(3,1)\}$

$= \min\{6, 2+3\}$

$= \min\{6, 5\}$

$= 5$

$A^3(2,3) = \min\{A^2(2,3),\ A^2(2,3) + A^2(3,3)\}$

$= \min\{2, 2+0\}$

$= 2$

$A^3(3,1) = \min(A^2(3,1),\ A^2(3,3) + A^2(3,1)\}$

$= \min\{3, 0+3\}$

$= 3$

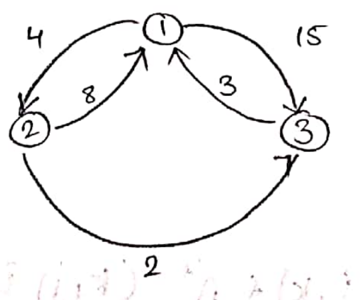
$A^3.(3,2) = \min\{A^2(3,2),\ A^2(3,3) + A^2(3,2)\}$

$= \min\{7, 0+7\}$

$= 7$

$$A^3 = \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

2)



Take $k = 0$

$$A^0 = \begin{bmatrix} 0 & 4 & 15 \\ 8 & 0 & 2 \\ 3 & \alpha & 0 \end{bmatrix}$$

Take $k = 1$

$A^1(i,j) = \min\{A^{1-1}(i,j),\ A^{1-1}(i,k) + A^{1-1}(k,j)\}$

$A^1(1,2) = \min\{A^0(1,2), A^0(1,1) + A^0(1,2)\}$

$= \min\{4, 0+4\}$

$= 4$

$A^1(1,3) = \min\{A^0(1,3), A^0(1,1) + A^0(1,3)\}$

$= \min\{15, 0+15\}$

$= 15$

$A^1(2,1) = \min\{A^0(2,1), A^0(2,1) + A^0(1,1)\}$

$= \min\{8, 8+0\}$

$= 8$

$A^1(2,3) = \min\{A^0(2,3), A^0(2,1) + A^0(1,3)\}$

$= \min\{2, 8+15\}$

$= \min\{2, 23\}$

$= 2$

$A^1(3,1) = \min\{A^0(3,1), A^0(3,1) + A^0(1,1)\}$

$= \min\{3, 3+0\}$

$= 3$

$A^1(3,2) = \min\{A^0(3,2), A^0(3,1) + A^0(1,2)\}$

$= \min\{\alpha, 3+4\}$

$= 7$

$$A^1 = \begin{bmatrix} 0 & 4 & 15 \\ 8 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

Take $k=2$

$A^2 = \min\{A^{2-1}(i,j), A^{2-1}(i,k) + A^{2-1}(k,j)\}$

$A^2(1,2) = \min\{A^1(1,2), A^1(1,2) + A^1(2,2)\}$

$= \min\{4, 4+0\}$

$= 4$

$A^2(1,3) = \min\{A^1(1,3), A^1(1,2) + A^1(2,3)\}$

$= \min\{15, 4+2\}$

$= 6$

$A^2(2,1) = \min\{A^1(2,1),\ A^1(2,2) + A^1(2,1)\}$

$= \min\{8, 0+8\}$

$= 8$

$A^2(2,3) = \min\{A^1(2,3),\ A^1(2,2) + A^1(2,3)\}$

$= \min\{2, 0+2\}$

$= 2$

$A^2(3,1) = \min\{A^1(3,1),\ A^1(3,2) + A^1(2,3)\}$

$= \min\{3, 7+8\}$

$= 3$

$A^2(3,2) = \min\{A^1(3,2),\ A^1(3,2) + A^1(2,2)\}$

$= \min\{7, 7+0\}$

$= 7$

$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ 8 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

Algorithm:-

```
Algorithm AllPairsshortest (cost, A, n)
// cost[1:n, 1:n] is the cost adjacency matrix of a
   graph with n vertices.
// A[i,j] is the cost of a shortest path from vertex i
   to vertex j.
// cost[i,i] = 0 for 1 ≤ i ≤ n
{
    for i:=1 to n do
        for j:=1 to n do                   // cost adjacency
            A[i,j] = cost[i,j];               matrix
    for k:=1 to n do
        for i:=1 to n do
            for j:=1 to n do
                A[i,j] := min {A(i,j), A[i,k] + A[k,j]};
}
```

Time complexity:-

→ In the above algorithm the first two for loops are used for construction of cost adjacency matrix is $O(n^2)$

→ The next 3 for loops are used for finding out the shortest path between every pair of vertices that take is $O(n^3)$

∴ The time complexity of all pairs shortest path is

$$= \{O(n^2) + O(n^3)$$

$$= O(n^3)$$

Multistage graph:-

A multistage graph $G = (V, E)$ is a directed weighted graph. In this graph, all the vertices are partitioned into k stages where $k \geq 2$

In multistage graph problem, we have to findout the shortest path from source to sink.

The cost of the path is calculated by using weights given along the edges. The cost of a path from source (S) to sink (T) is the sum of the cost of the edges on the path.

→ In multistage graph problem, we have to find out the shortest path from S to T. There is a set of vertices in each stage.

→ Edges are connecting vertices from one stage to next stage. starting stage contains only one vertex is called source vertex (S) and ending stage contains only one vertex is called sink (T) vertex).

→ Various paths from source to sink, now we have to select the optimal path.

The multistage graph problem can be solved by using two approaches.

1) Forward approach.
2) Backward approach.

forward approach:-

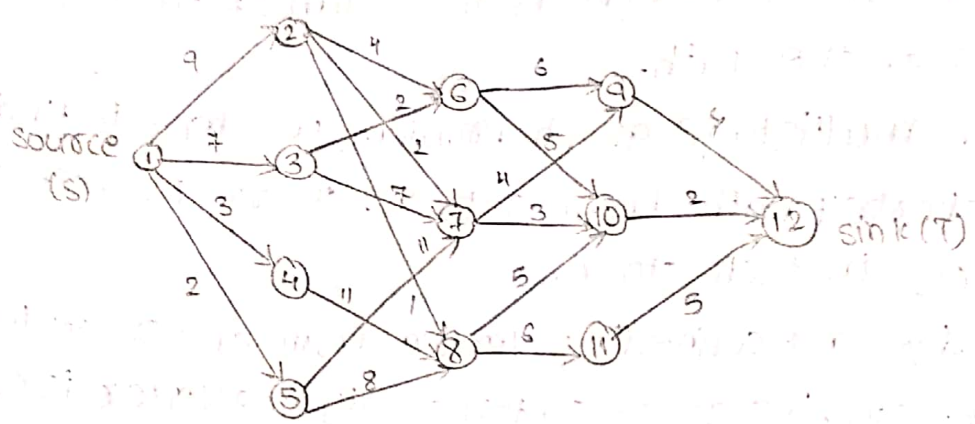
Let $c(i,j)$ be a minimum cost path from vertex j in stage i to vertex sink. Let $cost(i,j)$ be the cost of its path, then

$$cost(i,j) = \min_{\substack{(j,l) \in E \\ l \in V_{i+1}}} \{ c(j,l) + cost(i+1,l) \}$$

where $<j,l>$ is the cost to reach a node l in the level $i+1$ $D(i,j)$ be the value of l that minimises $c(j,l)$ and $cost(i+1,l)$

$$D(i,j) = \min \{ c(j,l) + cost(i+1,l) \}$$

1) Find out the minimum shortest path for the following graph using forward approach.



stages:- 1 2 3 4 5

5^{th} stage:-

cost(5, 12) = 0

4^{th} stage:-

cost (4, 9) = c(9, 12) = 9 4

cost(4, 10) = c(j, l) = c(10, 12) = 2

cost (4, 11) = c(j, l) = c(11, 12) = 5

3^{rd} stage:-

$$\text{cost}(3,6) = \min \begin{cases} c(6,9) + \text{cost}(4,9) \\ \cancel{\text{cost}} \\ c(6,10) + \text{cost}(4,10) \end{cases}$$

= min {6+4, 5+2}

= 7.

D (3,6) = 10

$$\text{cost}(3,7) = \min \begin{cases} c(7,9) + \text{cost}(4,9) \\ c(7,10) + \text{cost}(4,10) \end{cases}$$

= min { 4+4, 3+2 }

= 5

D (3,7) = 10

$$\text{cost}(3,8) = \min \begin{cases} c(8,10) + \text{cost}(4,10) \\ c(8,11) + \text{cost}(4,11) \end{cases}$$

= min { 5+2, 6+5 }

= 7

D(3,8) = 10

2^{nd} stage:-

$$cost(2,2) = \min \begin{Bmatrix} c(2,6) + cost(3,6) \\ c(2,7) + cost(3,7) \\ c(2,8) + cost(3,8) \end{Bmatrix}$$

$= \min\{4+7, 2+5, 1+7\}$

$= 7$

$D(2,2) = 7$

$$cost(2,3) = \min \begin{Bmatrix} c(3,6) + cost(3,6) \\ c(3,7) + cost(3,7) \end{Bmatrix}$$

$= \min\{2+7, 7+5\}$

$= 9$

$D(2,3) = 6$

$cost(2,4) = \min\{c(4,8) + cost(3,8)\}$

$= \min\{11+7\}$

$= 18$

$D(2,4) = 18$

$$cost(2,5) = \min \begin{Bmatrix} c(5,7) + cost(3,7) \\ c(5,8) + cost(3,8) \end{Bmatrix}$$

$= \min\{11+5, 8+7\}$

$= 15$

1^{st} stage:-

$$cost(1,1) = \min \begin{Bmatrix} c(1,2) + cost(2,2) \\ c(1,3) + cost(2,3) \\ c(1,4) + cost(2,4) \\ c(1,5) + cost(2,5) \end{Bmatrix}$$

$= \min\{9+7, 7+9, 3+18, 2+15\}$

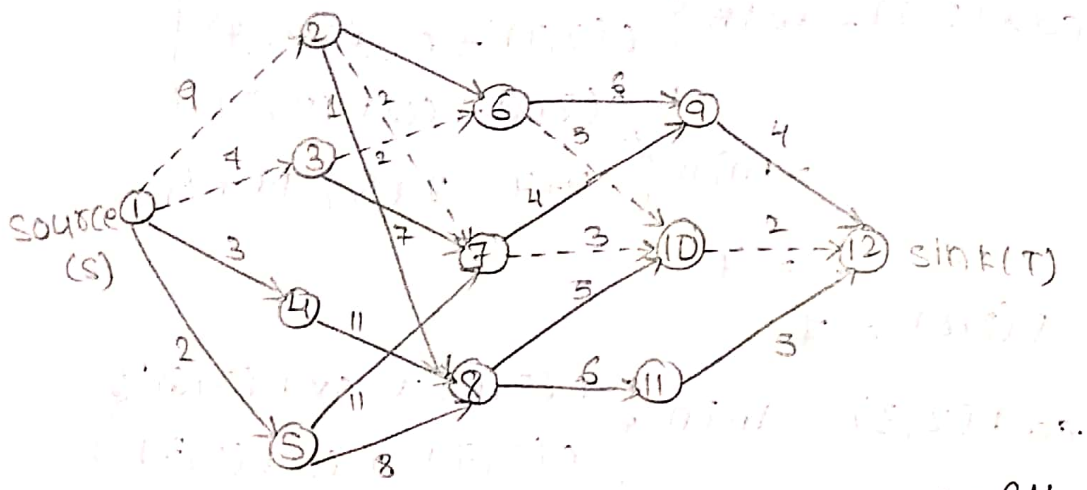
$= \min\{16, 16, 21, 17\}$

$= 16$

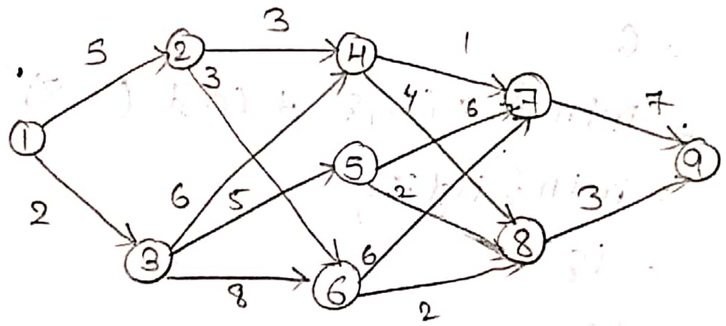
$D(1,1) = 2$, $D(1,1) = 3$

Shortest path

$D(1,1) = 2$	$D(1,1) = 3$
$D(2,2) = 7$	$D(2,3) = 6$
$D(3,7) = 10$	$D(3,6) = 10$
$D(4,10) = 12$	$D(4,10) = 12$
$1-2-7-10-12 = 16$	$1-3-6-10-12 = 16$



2) Find the minimum shortest paths for the following multistage graph using forward approach.



Backward Approach:-

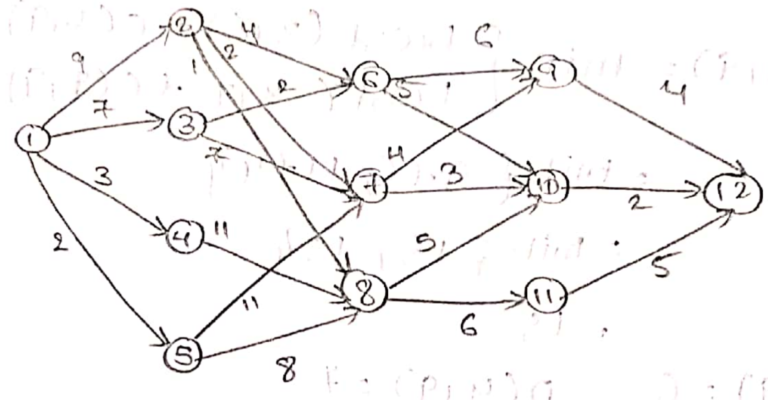
Let c(i,j) be a minimum cost path from vertex j in stage i to vertex source. Let cost(i,j) be the cost of this path then

$$bcost(i,j) = \min_{\substack{l \in V_{i-1} \\ (l,j) \in E}} \{bcost(i-1,l) + c(l,j)\}$$

where

(l,j) is an edge cost to reach a node j in the level i-l. D(i,j) = l.

1) Find out the minimum shortest paths for the following multistage graph using backward approach.



Stages:- 1 2 3 4 5

1^{st} stage:-

bcost(1,1) = 0

2^{nd} stage:-

bcost(2,2) = c(i,j) = c(1,2) = 9.

bcost(2,3) = c(1,3) = 7

bcost(2,4) = c(1,4) = 3

bcost(2,5) = c(1,5) = 2

3^{rd} stage:-

$$bcost(3,6) = \min \begin{cases} bcost(2,2) + c(2,6) \\ bcost(2,3) + c(3,6) \end{cases}$$

$= \min\{9+4, 7+2\}$

$= 9$

D(3,6) = 3

bcost(3,7) = min { bcost (2,2) + C(2,7)
bcost(2,3) + C(3,7)
bcost (2,5) + C (5,7 }

= min { 9+2, 7+7, 2+11 }

= 11

D(3,7) = 2

bcost (3,8) = min { bcost (2,2)+C(2,8)
bcost(2,4)+C(4,8)
bcost (2,5)+C(5,8) }

= min { 9+1, 3+11, 2+8 }

= 10

D(3,8) = 2, D(3,8) = 5

4th stage:-

bcost(4,9) = min { bcost (3,6) + C(6,9)
bcost(3,7 + C(7,9) }

= min { 9+6, 11+4 }

= min { 15, 15 }

= 15

D(4,9) = 6, D(4,9) = 7

bcost (4,10) = min { bcost (3,6) + C(6,10)
bcost(3,7) + C(7,10)
bcost(3,8) + C(8,10 }

= min { 9+5, 11+3, 10+5 }

= 14

D(4,10) = 6, D(4,10) = 7

bcost(4,11) = min { bcost (3,8) + C(8,11) }

= min { 10+6 }

= 16

D(4,11) = 8.

5th stage:-

bcost(5,12) = min { bcost(4,9) + C(9,12)
bcost (4,10) + C(10,12)
bcost(4,11) + C(11,12) }

$= \min\{15+4, 14+2, 16+5\}$

$= \min\{19, 16, 21\}$

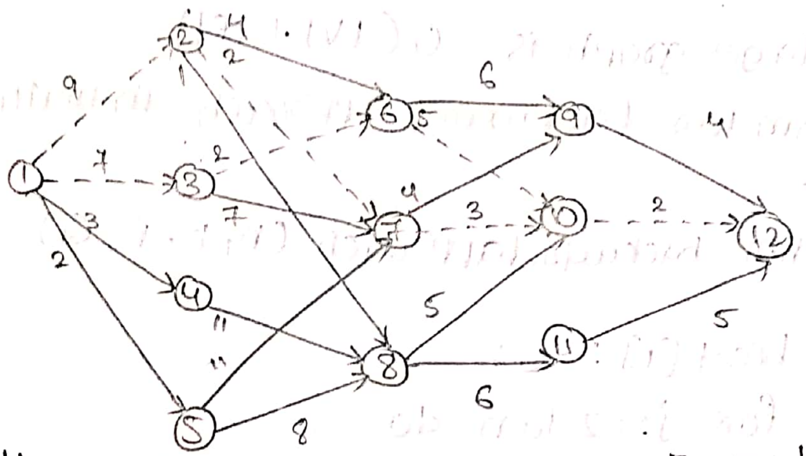
$= 16$

$D(5,12) = 10$

shortest Path:

$D(5,12) = 10$	$D(5,12) = 10$
$D(4,10) = 6$	$D(4,10) = 7$
$D(3,6) = 3$	$D(3,7) = 2$
$D(2,3) = 1$	$D(2,2) = 1$
1-3-6-10-12	1-2-7-10-12

∴ shortest paths are 1-3-6-10-12, 1-2-7-10-12



Algorithm for Forward approach in multistage graph:-

Algorithm Forwardapproach (n, k, P, G)

// The input for the multistage graph is a k-stage graph G(V,E) with n vertices indexed in order of stages E is a set of edges and c[i,j] is the cost(i,j). P[1:k] is a minimum cost path

```
{
    cost[n] := 0 ;
    for j:n-1 to 1 do
    {
        // compute cost[j]
        // let l be a vertex such that <j,l> is
           an edge of an G and c[j,l]+cost[l]
```

is minimum

cost [j] : = c[j, l] + cost (l) ;

D[j] : = l ;

}

// find a minimum cost path

P[1] : = 1 ;

P[k] : = n ;

for j : = 2 to k-1 do

P[j] : = d[P[j-1]] ;

}

Time complexity:

The Time complexity of forward approach in multistage graph is O(|V| + |E|)

* Algorithm for backward approach in multistage graph:

Algorithm backwardapproach (n, k, P, G)

{

bcost [1] : = 0 ;

for j : = 2 to n do

{

// compute bcost [j]

// let l be a vertex such that <l, j> is an edge of G and bcost(l) + c(l, j) is minimum.

b cost[j] = c(l, j) + bcost (l) ;

D[j] : = l ;

}

// find a minimum cost path

P[1] : = 1 ;

P[k] : = n ;

for j : = k-1 to 2 do

P[j] : = d[P[j+1]] ;

}

Time complexity :-

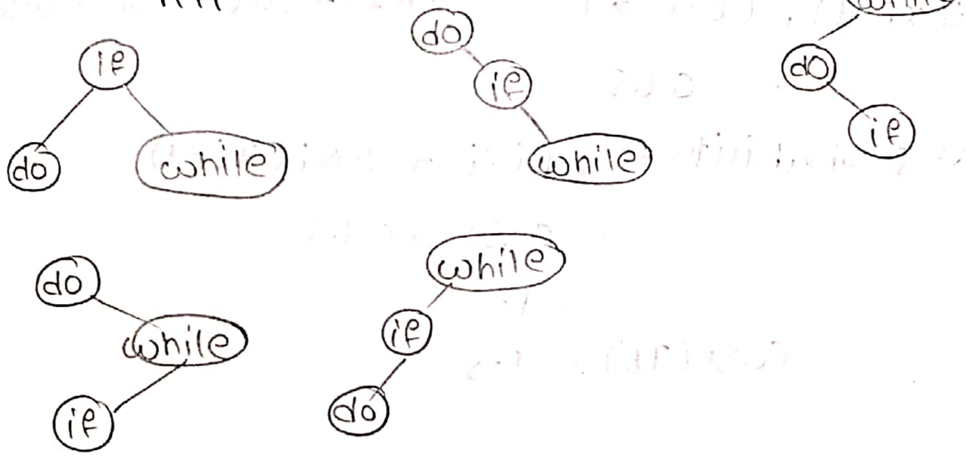
The time complexity of backward approach in multistage graph is $O(|V| + |E|)$

Optimal Binary search Tree (OBST) :-

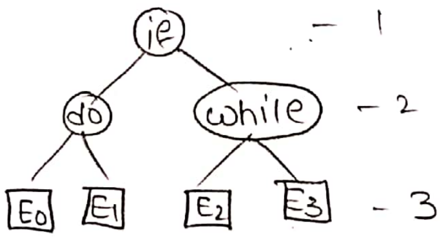
Find out optimal binary search tree for a set $(a_1, a_2, a_3) = (do, if, while)$ $(P_1, P_2, P_3) = (0.5, 0.1, 0.05)$ $(Q_0, Q_1, Q_2, Q_3) = (0.15, 0.1, 0.05, 0.05)$

The possible number of binary search tree with $n = 3$ is

$$\frac{2nC_n}{n+1} = \frac{2 * 3C_3}{3+1} = \frac{6C_3}{4} = \frac{20}{4} = 5$$



i)



$P(1) = 0.1 * 1 + 0.5 * 2 + 0.05 * 2$

$= 0.1 + 1 + 0.1$

$= 1.2$

$Q(1) = 0.15 * 2 + 0.1 * 2 + 0.05 * 2 + 0.05 * 2$

$= 0.7$

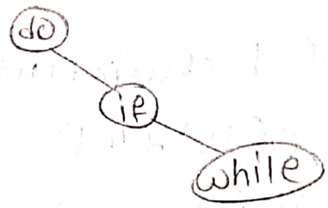
Total probability $= \Sigma P_i x_i + \Sigma Q_i (x_i - 1)$

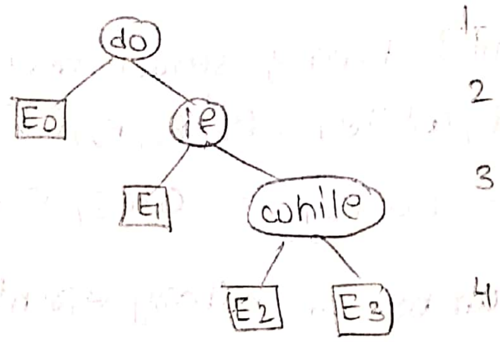
$= 1.2 + 0.7$

$= 1.9$

$Cost(T_a) = 1.9$

(ii)





$\Sigma P_i x_i = (0.5*1) + (0.1*2) + (0.05*3)$

$= 0.85$

$\Sigma Q_i(x_i - 1) = (0.15*1) + (0.1*2) + (0.05*3) + (0.05*$

$= 0.65$

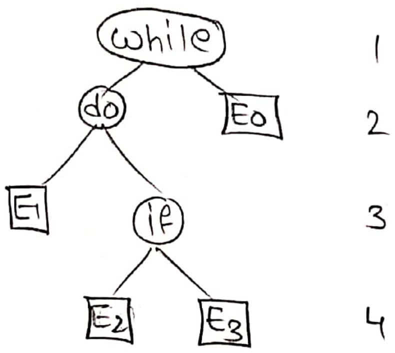
Total probability $= \Sigma P_i x_i + \Sigma Q_i(x_i - 1)$

$= 0.85 + 0.65$

$= 1.5$

$Cost(T_b) = 1.5$

(iii)



$\Sigma P_i x_i = (0.05*1) + (0.5*2) + (0.1*3)$

$= 1.35$

$\Sigma Q_i(x_i - 1) = (0.15*1) + (0.1*2) + (0.05*3) + (0.05*3)$

$= 0.65$

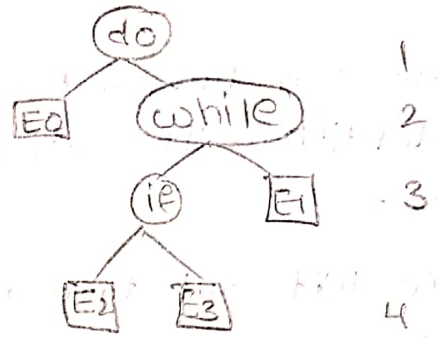
Total probability $= \Sigma P_i x_i + \Sigma Q_i(x_i - 1)$

$= 1.35 + 0.65$

$= 2$

$Cost(T_c) = 2$

(iv)



$\Sigma P_i x_i = (0.5*1) + (0.05*2) + (0.1*3)$

$= 0.9$

$\Sigma Q_i(x_i - 1) = (0.15*1) + (0.1*2) + (0.05*3) + (0.05*3)$

$= 0.65$

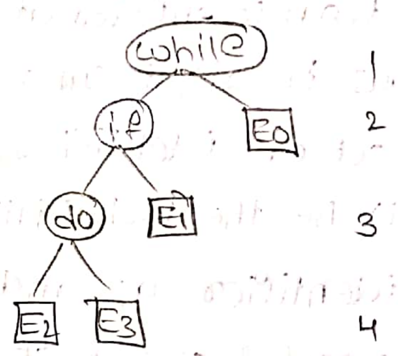
Total probability $= \Sigma P_i x_i + \Sigma Q_i(x_i - 1)$

$= 0.9 + 0.65$

$= 1.55$

$cost(T_4) = 1.55$

(v)



$\Sigma P_i x_i = (0.05*1) + (0.1*2) + (0.5*3)$

$= 1.75$

$\Sigma Q_i(x_i - 1) = (0.15*1) + (0.1*2) + (0.05*3) + (0.05*3)$

$= 0.65$

Total probability $= \Sigma P_i x_i + \Sigma Q_i(x_i - 1)$

$= 1.75 + 0.65$

$= 2.4$ $\quad cost(T_5) = 2.4$

$cost(T_1) = 1.9$

$cost(T_2) = 1.5$

$cost(T_3) = 2$

$cost(T_4) = 1.55$

$cost(T_5) = 2.4$

Among all the 5 binary search trees tree-2 is optimal tree with cost = 1.5

Theory:-

A binary search tree is a binary tree in which each node is an identifier. It satisfies the following properties:-

1) All identifiers in left sub tree are less than identifier at root node.

2) All identifiers in right sub tree are greater than the identifier at root node.

3) To find whether an identifier x is present or not using the following steps

(i) x is compared with the root.

(ii) If x is less than identifier at root node, then search continuously in left sub tree.

(iii) If x is greater than identifier at root node, then search continuous in right sub tree.

→ Let us assume given set of identifiers $\{a_1, a_2, a_3 \dots a_n\}$ and let P(i) be the probability of successful search of an identifier a_i and Q(i) be the probability of unsuccessful search of an identifier a_i.

$$\therefore \ \Sigma P(i) + \Sigma Q(i) = 1$$

→ To obtain a cost function for binary search tree, it is useful to add an imaginary node or empty node.

→ If n identifiers are there, there will be 'n' internal nodes and n+1 external nodes. Every external node represents an unsuccessful search.

→ If a successful search terminates at an internal node at level l. Hence, the expected cost contribution from the internal node for identifier a_i is

$P(i) * level(a_i)$.

→ unsuccessful search terminates at an external node at level l. Hence, the expected cost contribution for the external node is $Q(i) * level((E_i) - 1)$

→ The expected cost of a binary search tree is

$\Sigma(P(i) * level(a_i)) + \Sigma(Q(i) * level(E_i - 1))$

where $P(i)$ is the probability of successful search and $Q(i)$ is the probability of unsuccessful search

Definition of OBST:-

→ For a given set of identifiers, we have different binary search trees. Among all those which requires least number of comparisons for searching a particular identifier called OBST (or) optimal Binary search Tree.

→ Let $a_1, a_2, a_3, \ldots\ldots, a_n$ are different identifiers with $a_1 < a_2 < a_2 < \ldots\ldots < a_n$.

→ successful search probabilities $P(i)$, $1 \le i \le n$

→ unsuccessful search probabilities $Q(i)$, $0 \le i \le n$

→ For the given set of n identifiers we have to construct the optimal binary search tree T_{on}, where T_{on} is the optimal binary search tree

→ W_{ij} is the weight of each T_{ij}

Initial condition:-

$C(i,i) = 0$

$r(i,i) = 0$

$W(i,i) = q_i$ (or) $q(i)$

Add node by node to the BST by calculating

$$C(i,j) = \min_{i < k \le j} \{C(i, k-1) + C(k,j)\} + w(i,j)$$

$w(i,j) = w(i, j-1) + p(j) + q(j)$

$r(i,j) = k.$

1) consider $n = 4$ and $(a_1, a_2, a_3, a_4) = (do, if, int,$ while) $(P_1, P_2, P_3, P_4) = (3, 3, 1, 1)$, $q(0-4) = (2,3,1,1)$

construct the OBST.

we have to construct the OBST $T_{on} = T_{04}$

	0	1	2	3	4
$\lvert j-i\rvert=0$ 0	$w(0,0)=2$ $c(0,0)=0$ $r(0,0)=0$	$w(1,1)=3$ $c(1,1)=0$ $r(1,1)=0$	$w(2,2)=1$ $c(2,2)=0$ $r(2,2)=0$	$w(3,3)=1$ $c(3,3)=0$ $r(3,3)=0$	$w(4,4)=1$ $c(4,4)=0$ $r(4,4)=0$
$\lvert j-i\rvert=1$ 1	$w(0,1)=8$ $c(0,1)=8$ $r(0,1)=1$	$w(1,2)=7$ $c(1,2)=7$ $r(1,2)=2$	$w(2,3)=3$ $c(2,3)=3$ $r(2,3)=3$	$w(3,4)=3$ $c(3,4)=3$ $r(3,4)=4$	
$\lvert j-i\rvert=2$ 2	$w(0,2)=12$ $c(0,2)=19$ $r(0,2)=1$	$w(1,3)=9$ $c(1,3)=12$ $r(1,3)=2$	$w(2,4)=5$ $c(2,4)=8$ $r(2,4)=3,4$		
$\lvert j-i\rvert=3$ 3	$w(0,3)=14$ $c(0,3)=25$ $r(0,3)=2$	$w(1,4)=11$ $c(1,4)=19$ $r(1,4)=2$			
$\lvert j-i\rvert=4$ 4	$w(0,4)=2$ $c(0,4)=32$ $r(0,4)=16$				

Step-1:- $\lvert j-i\rvert = 0$

$c(i,i) = 0$

$r(i,i) = 0$

$w(i,i) = q(i)$

$c(0,0) = 0$
$r(0,0) = 0$
$w(0,0) = q(0) = 2$

$c(1,1) = 0$
$r(1,1) = 0$
$w(1,1) = q(1) = 3$

$c(2,2) = 0$
$r(2,2) = 0$
$w(2,2) = q(2) = 1$

$c(3,3) = 0$
$r(3,3) = 0$
$w(3,3) = q(3) = 1$

$c(4,4) = 0$
$r(4,4) = 0$
$w(4,4) = q(4) = 1$

Scanned with CamScanner

Step-2: $|j-i| = 1$

$\rightarrow w(0,1) = w(0,0) + P(1) + q(1)$

$= 2+3+3$

$= 8$

$c(0,1) = \min_{0<k\le 1} \{c(0,1-1) + c(1,1)\} + w(0,1)$

$= \min_{k=1} \{c(0,0) + c(1,1)\} + 8$

$= \min\{0+0\} + 8$

$= 8$

$r(0,1) = 1$

$\rightarrow w(1,2) = w(1,1) + P(2) + q(2) = 3+3+1 = 7$

$c(1,2) = \min_{1<k\le 2} \{c(1,2-1) + c(2,2)\} + w(1,2)$

$= \min_{k=2} \{c(1,1) + c(2,2)\} + 7$

$= \min_{k=2} \{0+0\} + 7$

$= 7.$

$r(1,2) = 2$

$\rightarrow w(\overset{i}{2},\overset{j}{3}) = w(2,2) + P(3) + Q(3) = 1+1+1 = 3$

$c(2,3) = \min_{2<k\le 3} \{c(2,2) + c(3,3)\} + w(2,3)$

$= \min_{k=3} \{0+0\} + 3$

$= 3$

$r(2,3) = 3$

$\rightarrow w(3,4) = w(3,3) + P(4) + Q(4) = 1+1+1 = 3$

$c(3,4) = \min_{3<k\le 4} \{c(3,3) + c(4,4)\} + w(3,4)$

$= \min_{k=4} \{0+0\} + 3 = 3$

$r(3,4) = 4$

Step-3:-

$|j-i| = 2$

$\Rightarrow w[0,2] = w(0,1) + p(2) + q(2) = 8+3+1 = 12$

$$c(0,2) = \min_{\substack{0<k\le 2 \\ k=1,2}} \begin{Bmatrix} c(0,0)+c(1,2) \\ c(0,1)+c(2,2) \end{Bmatrix} + w(0,2)$$

$$= \min\{0+7,\ 8+0\} + 12$$

$$= \min\{7, 8\} + 12$$

$$= \overset{k=1}{7} + 12$$

$$= 19$$

$r(0,2) = 1$

$\Rightarrow w(1,3) = w(1,2) + p(3) + q(3) = 7+1+1 = 9$

$$c(1,3) = \min_{\substack{1<k\le 3 \\ k=2 \\ k=3}} \begin{Bmatrix} c(1,1)+c(2,3) \\ c(1,2)+c(3,3) \end{Bmatrix} + w(1,3)$$

$$= \min\{\overset{k=2}{0+3},\ \overset{k=3}{7+0}\} + 9$$

$$= 3+9$$

$$= 12$$

$r(1,3) = 2$

$\Rightarrow w(2,4) = w(2,3) + p(4) + q(4) = 3+1+1 = 5$

$$c(2,4) = \min_{\substack{2<k\le 4 \\ k=3 \\ k=4}} \begin{Bmatrix} c(2,2)+c(3,4) \\ c(2,3)+c(4,4) \end{Bmatrix} + w(2,4)$$

$$= \min\{0+3,\ 3+0\} + 5$$

$$= \min\{3, 3\} + 5$$

$$= 3+5 = 8$$

$r(2,4) = 3, 4$

Scanned with CamScanner

Step-4:-

$\rightarrow \omega(0,3) = \omega(0,2) + P(3) + q(3) = 12+1+1 = 14$

$$c(\underset{i}{0},\underset{j}{3}) = \min_{\substack{0<k\le 3 \\ k=1 \\ k=2 \\ k=3}} \begin{Bmatrix} c(0,0) + c(1,3) \\ c(0,1) + c(2,3) \\ c(0,2) + c(3,3 \end{Bmatrix} + \omega(0,3)$$

$$= \min \begin{Bmatrix} 0+12 \\ 8+3 \\ 19+0 \end{Bmatrix} + 14$$

$$= \min\{12, 11, 19\} + 14$$

$$= 11 + 14$$

$$= 25$$

$r(0,3) = 2$

$\omega(1,4) = \omega(1,3) + P(4) + q(4) = 9+1+1 = 11$

$$\rightarrow c(\underset{i}{1},\underset{j}{4}) = \min_{\substack{1<k\le 4 \\ k=2 \\ k=3 \\ k=4}} \begin{Bmatrix} c(1,1) + c(2,4) \\ c(1,2) + c(3,4) \\ c(1,3) + c(4,4) \end{Bmatrix} + \omega(1,4)$$

$$= \min \begin{Bmatrix} 0+8 \\ 7+3 \\ 12+0 \end{Bmatrix} + 11$$

$$= \min\{8, 10, 12\} + 11$$

$$= 8 + 11 = 19$$

$r(1,4) = 2$

Step-5:-

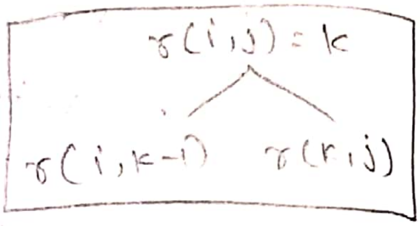
$\omega(\underset{i}{0},\underset{j}{4}) = \omega(0,3) + P(4) + q(4) = \cancel{12+1+1=14}$

$$= 14+1+1 = 16$$

$$c(\underset{i}{0},\underset{j}{4}) = \min_{\substack{0<k\le 4 \\ k=1 \\ k=2 \\ k=3 \\ k=4}} \begin{Bmatrix} c(0,0) + c(1,4) \\ c(0,1) + c(2,4) \\ c(0,2) + c(3,4) \\ c(0,3) + c(4,4) \end{Bmatrix} + \omega(0,4)$$

$$= \min \begin{Bmatrix} 0+19 \\ 8+8 \\ 19+3 \\ 25+0 \end{Bmatrix} + 16$$

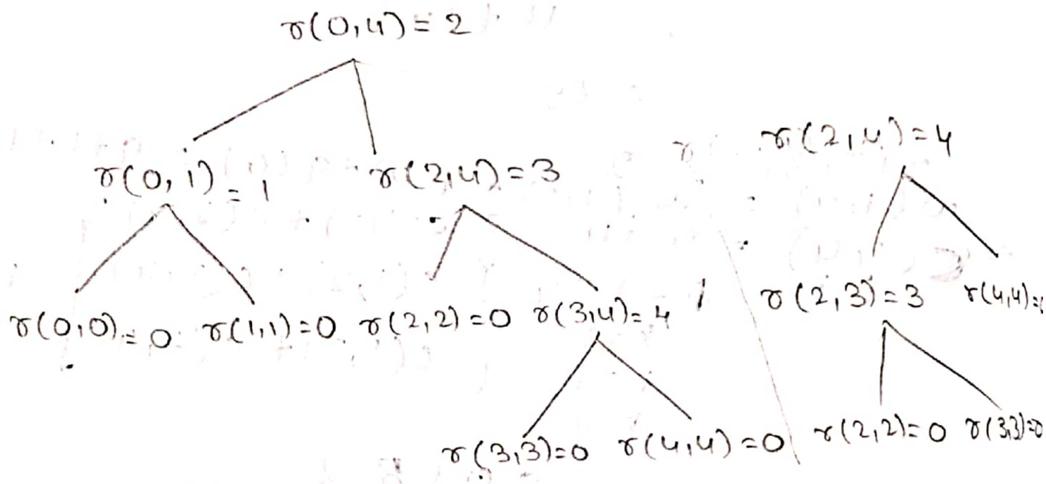
$$= \min \{19, 16, 22, 25\} + 16$$

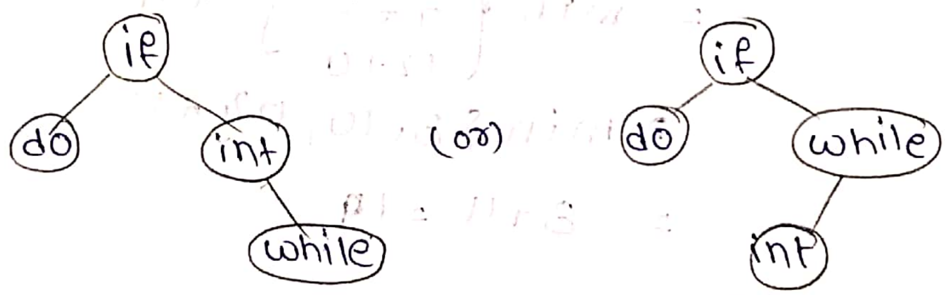


$$= 16 + 16$$

$$= 32$$

$r(0,4) = 2$





2) using OBST Algorithm findout the optimal Binary search tree for the identifier set (a_1, a_2, a_3, a_4) = (end, goto, print, stop) with $P(1) = \frac{1}{20}$, $P(2) = \frac{1}{5}$, $P(3) = \frac{1}{10}$, $P(4) = \frac{1}{20}$
$q(0) = \frac{1}{5}$, $q(1) = \frac{1}{10}$ $q(2) = \frac{1}{5}$ $q(3) = \frac{1}{20}$
$q(4) = \frac{1}{20}$

we will multiply the values of p's & q's by 20 (maximum) denominator.

find OBST for $n=4$, $(a_1, a_2, a_3, a_4) = $ (end, goto, print, stop) $p_1 = \frac{1}{4}$ $p_2 = \frac{1}{8}$ $p_3 = \frac{1}{16}$ $p_4 = \frac{1}{16}$

$q_0 = \frac{1}{4}$ $q_1 = \frac{3}{16}$ $q_2 = \frac{1}{16}$ $q_3 = \frac{1}{16}$ $q_4 = \frac{1}{16}$

16