UNIT-4

Dynamic programming

* The Drawback of Greedy method is, we will make one decision at a time, this can be overcome in dynamic programming. In this, we will make more than one decision at a time

* Dynamic programming is an algorithm design Technique that can be used when the solution to a problem may be viewed as the result of a sequence of decisions.

* Dynamic programming is Applicable when the subproblems are not independent, that is when subproblems share subproblems.

* In this every problem solved by using Bellman's principle of optimality. that is the output of stage 1 will be given as input to stage 2 the output of stage 2 will be given as input to stage 3. And so on.

* In this first we will Apply initial conditions as input to stage 1, then Apply principle of optimality.

All pair shortest path problem

* Let $G = (V, E)$ be a directed graph consisting of n vertices And each edge is Associated with a weight. To finding the shortest path b/w all pairs of vertices in a graph is called All pair shortest path problem.

* This problem can be solved by using dynamic programming Technique. The All pair shortest path problem is to determine a matrix A such that $A(i,j)$ is the length of a shortest path from vertex i to vertex j. Assume that this path contains no cycles.

* If K is an intermediate vertex on this path, then the subpaths from i to K and from K to j ~~are~~ ~~the~~ respectively

* Dijkstra Algorithm requires $O(n^2)$ time, All pair shortest path problem require $O(n^3)$.

* The shortest path can be computed using following recursive method

$$A^{K}(i,j) = w(i,j) \text{ ; if } K=0$$

$$= \min\{A^{K-1}(i,j),\ A^{K-1}(i,K) + A^{K-1}(K,j)\} \text{ if } K \geq 1$$

②

* The Floyd's Algorithm for all pair shortest path problem

Algorithm Allpair (A, n, ~~cost~~ w)

{

// w is weight of the Array, n is the no. of
// vertices, A is the shortest path from vertex i to j

for i := 1 to n do

for j := 1 to n do

A[i,j] := ~~cost~~ w(i,j)

for k := 1 to n do

for i := 1 to n do

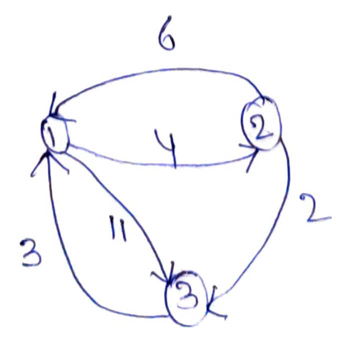
for j := 1 to n do

A[i,j] := min{ A(i,j), A(i,k) + A(k,j) } ;

}

∴ Time complexity is $O(n^3)$

Q :- Find the All pair shortest path for the graph



Algorithm :(w, A, n)

{

for i = 1 to 3 do

{ for j = 1 to 3 do

{ A[i, j] = w[i, j]

}

}

for k = 1 to 3 do

{ for i = 1 to 3 do

{ for j = 1 to 3 do

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

} } } }

∴ Time complexity is $O(n^3)$

3

cost Adjacency matrix, $A^0(i,j) = w(i,j) = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & \infty & 0 \end{array}$

step 1:- for $k=1$, i.e going from i to j through the intermediate vertex '1'.

$i=1, j=1$

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

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

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

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

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

$A^1(1,1) = 0$

$i=1, j=2$

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

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

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

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

$A^1(1,2) = 4$

$i=1, j=3$

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

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

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

$A^1(1,3) = 11$

$i=2, j=1$

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

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

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

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

$A^1(2,1) = 6$

$i=2, j=2$

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

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

$= \min\{0, 6+4\}$

$= \min\{0, 10\}$

$A^1(2,2) = 0$

$i=2, j=3$

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

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

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

$A^1(2,3) = 2$

$i=3, j=1$

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

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

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

$A^1(3,1) = 3$

$i=3, j=2$

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

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

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

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

$A^1(3,2) = 7$

$i=3, j=3$

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

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

$= \min\{0,\ 3+11\}$

$= \min\{0, 14\}$

$A^1(3,3) = 0$

$$\therefore A^1(i,j) = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

step 2:- for $k=2$, i.e going from i to j through vertex 2

$i=1, j=1, k=2$

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

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

$= \min\{0,\ 4+6\}$

$= \min\{0, 10\}$

$A^2(1,1) = 0$

$i=1, j=2$

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

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

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

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

$A^2(1,2) = 4$

$i=1, j=3$

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

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

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

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

$A^2(1,3) = 6$

$i=2, j=1$

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

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

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

$A^2(2,1) = 6$

$i=2, j=2$

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

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

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

$A^2(2,2) = 0$

$i=2, j=3$

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

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

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

$A^2(2,3) = 2$

$i=3, j=1$

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

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

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

$= \min\{3, 13\}$

$A^2(3,1) = 3$

$i=3, j=2$

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

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

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

$A^2(3,2) = 7$

$i=3, j=3$

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

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

$= \min\{0, 7+2\}$

$= \min\{0, 9\}$

$A^2(3,3) = 0$

$$\therefore A^2(i,j) = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

Step 3:- for $k=3$, i.e, going from i to j through vertex 3

$i=1, j=1$

$$A^3(1,1) = \min\{A^{3-1}(1,1), A^{3-1}(1,3) + A^{3-1}(3,1)\}$$
$$= \min\{A^2(1,1), A^2(1,3) + A^2(3,1)\}$$
$$= \min\{0, 6+3\}$$
$$= \min\{0, 9\}$$
$$A^3(1,1) = 0$$

$i=1, j=2$

$$A^k(1,2) = \min\{A^{3-1}(1,2), A^{3-1}(1,3) + A^{3-1}(3,2)\}$$
$$= \min\{A^2(1,2), A^2(1,3) + A^2(3,2)\}$$
$$= \min\{4, 6+7\}$$
$$= \min\{4, 13\}$$
$$A^3(1,2) = 4$$

$i=1, j=3$

$$A^3(1,3) = \min\{A^{3-1}(1,3), A^{3-1}(1,3) + A^{3-1}(3,3)\}$$
$$= \min\{A^2(1,3), A^2(1,3) + A^2(3,3)\}$$
$$= \min\{6, 6+0\}$$
$$A^3(1,3) = 6$$

$i=2, j=1$

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

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

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

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

$A^3(2,1) = 5$

$i=2, j=2$

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

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

$= \min\{0, 2+7\}$

$A^3(2,2) = 0$

$i=2, j=3$

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

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

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

$A^3(2,3) = 2$

$i=3, j=1$

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

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

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

$A^3(3,1) = 3$

$i=3,\ j=2$

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

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

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

$A^3(3,2) = 7$

$i=3,\ j=3$

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

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

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

$A^3(3,3) = 0$

$$\therefore A^3(i,j) = \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

Travelling sales person problem

* Here the sales man should start at a point and travels All the places and come back to starting point. The problem is to minimize the travelling cost. The main Requirement is there should be communication b/w nodes

* suppose we have to Route a postal van to pickup mails from mail boxes located at n different places.

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

* the Route taken by the postal van is a tour And it should have minimum length (minimum cost)

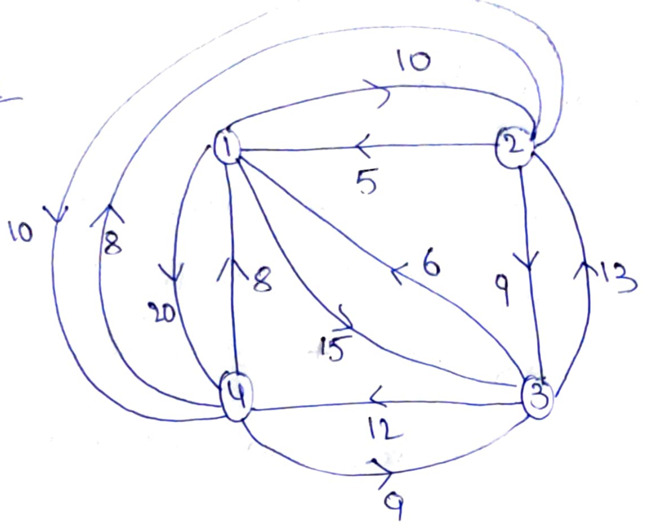
1) $g(i, \phi) = c_{i,1}, \quad 1 \leq i \leq n$

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

* Here $g(i, S)$ means i is starting node And the nodes in S are to be traversed. $\min_{j \in S}$ is considered as the intermediate node

* $g(j, S-\{j\})$ means j is Already traversed. so next we have to traversed $S-\{j\}$ with j is starting point.

Ex



sol:-

cost Adjacent matrix is

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

The formula for solving this problem is

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

$$g(i,\phi) = c_{i,1}, \quad 1 \le i \le n$$

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

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

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

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

* $|S| = 1$, it means set contains only one element,

Before solving this problem, we make an Assumption that the salesperson starts at vertex 1, from that he can move to vertex 2.

* If he visits vertex 2, he can next visit either vertex 3 or vertex 4.

$|S| = 1$

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

$$= \min_{j \in S}\{9+6\}$$

$$= 15$$

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

$$= \min_{j \in S}\{10+8\}$$

$$= \min_{j \in S}\{18\}$$

$$= 18$$

* From vertex 1, next he can visit vertex 3 instead of vertex 2, in this case from vertex 3, next he can visit either vertex 2 or vertex 4.

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

$$= \min_{j \in S}\{13+5\}$$

$$= \min_{j \in S}\{18\}$$

$$= 18$$

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

$$= \min_{j \in S}\{12+8\}$$

$$= 20$$

* from vertex 1, he can visit vertex 4 instead of vertex 3, In this case from vertex 4, next he can visit either vertex 2 or 3.

$$g(4,2) = \min_{j \in S}\{C_{42} + g(2,\phi)\}$$
$$= \min_{j \in S}\{8+5\}$$
$$= \min_{j \in S}\{13\}$$
$$= 13$$

$$g(4,3) = \min_{j \in S}\{C_{43} + g(3,\phi)\}$$
$$= \min_{j \in S}\{9+6\}$$
$$= \min_{j \in S}\{15\}$$
$$= 15$$

* $|S| = 2$, here set containg two values, so we can place two vertices in S. from starting vertex 1, next he can visit either vertex 2 or 3 or 4.

* If he visits vertex 2, then from that vertex he can visit either 3 or 4, or vertex 4 or 3

$$g(2,\{3,4\}) = \min_{j \in S}\{C_{23} + g(3,4),\ C_{24} + g(4,3)\}$$
$$= \min_{j \in S}\{9+20,\ 10+15\}$$
$$= \min_{j \in S}\{29,\ 25\}$$
$$= 25$$

* If he visit vertex 3 instead of vertex 2. next he vist either vertex 2 or 4

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

$$= \min_{j \in S} \{ 13 + 18, 12 + 13 \}$$

$$= \min_{j \in S} \{ 31, 25 \}$$

$$= 25$$

* If he visit vertex 4 instead of vertex 3, next he visit either vertex 2 or 3

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

$$= \min_{j \in S} \{ 8 + 15, 9 + 18 \}$$

$$= \min_{j \in S} \{ 23, 27 \}$$

$$= 23$$

* $|S| = 3$, here set contains 3 elements, After starting from vertex 1, next he can visit 2 or 3 or 4. So determine $g(1, \{2,3,4\})$.

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

$$= \min_{j \in S} \{ 10 + 25, 15 + 25, 20 + 23 \}$$

$$= \min_{j \in S} \{ 35, 40, 43 \} = 35$$

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

$10 + 10 + 9 + 6 = \underline{35}$

<u>Time Complexity</u>

$N = (n-1) \sum_{k=0}^{n-2} \binom{n-2}{k} \quad \left(\binom{n}{k} \text{ is same as } {}^{n}C_{k} \right)$

$= (n-1) \left({}^{n-2}C_0 + {}^{n-2}C_1 + {}^{n-2}C_2 + \cdots + {}^{n-2}C_{n-2} \right)$

$= (n-1) \left(2^{n-2} \right) \quad \left[\because {}^{n}C_0 + {}^{n}C_1 + {}^{n}C_2 + \cdots + {}^{n}C_n = 2^n \right)$

$n \cdot N = n(n-1) \left(\frac{2^n}{4} \right)$

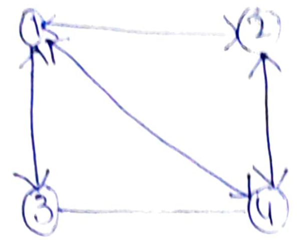
$= n^2 \cdot \frac{2^n}{4} - n \frac{2^n}{4}$

$= O(n^2 \cdot 2^n)$

$\therefore$ Time complexity is $O(n^2 \cdot 2^n)$

Gr

G:-



Sol

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

Step 1:- For $|S| = \phi$, $g(i,\phi) = c(i,1)$

$g(1,\phi) = c(1,1) = 0$

$g(2,\phi) = c(2,1) = 5$

$g(3,\phi) = c(3,1) = 6$

$g(4,\phi) = c(4,1) = 8$

Step 2:- $|S| = 1$, the no. of elements in S is one.

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

* from vertex 1 he can move from vertex 1 to vertex 2, from vertex 2 either he can move 3 or 4.

$g(2,\{3\}) = \{ c(2,3) + g(3,\phi) \} = \min\limits_{j \in S} \{9+6\} = 15$

$g(2,\{4\}) = \min\limits_{j \in S} \{ c(2,4) + g(4,\phi) \} = \min\limits_{j \in S} \{10+8\} = 18$

* ~~g~~

*

0/1 Knapsack problem

* Now we will compare knapsack problem with general problem. In olden days there was a store, Assume that which contains different types of ornaments, which are made up of gold. let n_1, n_2, n_3 be ornaments
* cost and weight of these ornaments are c_1, c_2, c_3 and w_1, w_2, w_3 pounds respectively.
* Now A thief wants to rob the ornaments so that he will brought the Empty bag (knapsack) with size m.
* In what way he has to place the items in the bag such that he should get the maximum profit. In this the thief can't place fraction of ornaments in the bag.
* Either he can place complete ornament in the bag, or he can't place ornament in the bag. so, ($x_i = 0$ or 1)

1) If $x_i = 0$ means we can't place ornament in the bag
2) If $x_i = 1$ means we can place ornament in the bag.

* This problem contains 0 or 1, that is why this problem is called as 0/1 knapsack problem.

* In this we can't place fractions of weights in the bag. Consider the weights $w_1, w_2, \ldots w_n$ And fraction of weights to be put in the bag should be $x_1, x_2 \ldots x_n = 0$ or 1, The dynamic programming solution for 0/1 Knapsack is

$$F_n(m) = \max[\underbrace{F_{n-1}(m)}_{\text{when } x_n=1}, \underbrace{F_{n-1}(m-w_n)+P_n}_{\text{when } x_n=0}]$$

where m = size of Knapsack

* when $x_n=1$ then the size of the bag is reduced by w_n which is the weight of the n^{th} item. As we are placing the n^{th} item we should add the profit for the last item.

* In Dynamic programming 0/1 Knapsack can be solved by using merging And purging Rule.

Initially we can take $S^0 = \{0,0\}$

$$S_1^i = S^{i-1} + (P_i\, w_i)$$

$$S^i = S^{i-1} + S_1^i$$

* purging Rule (Dominance Rule): If one S^{i-1} and S_1^i has a pair (P_j, w_j) and other has a pair (P_k, w_k) and $P_j \leq P_k$ while $w_j \geq w_k$ then the pair (P_j, w_j) is discarded.

* After Applying Purging Rule, we will check the following condition inorder to find solution.

If $(p_i, w_i) \in S^n$ and $(p_i, w_i) \notin S^{n-1}$ then

$x_n = 1$

otherwise $x_n = 0$

Q:- $n=3$, $m=6$, $(p_1\, p_2\, p_3) = (1, 2, 5)$

$(w_1\, w_2\, w_3) = (2, 3, 4)$

Sol:- $S^0 = \{0, 0\}$

$S_1^i = \underbrace{S^{i-1} + (p_i\, w_i)}$ → Addition

$S_1^1 = \underbrace{S^{1-1} + (p_1\, w_1)}$ → Addition

$= S^0 + (1, 2)$

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

$S_1^1 = \{(1, 2)\}$

$S^i = \underbrace{S^{i-1} + S_1^i}$ → merging

$S^1 = S^{1-1} + S_1^1$

$= S^0 + S_1^1$

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

$S^1 = \{(0, 0)\, (1, 2)\}$

$S_1^2 = S^{i-1} + (p_2, w_2)$

$= S^{2-1} + (2, 3)$

$= S^1 + (2, 3)$

$= S^1 + (2,3)$

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

$S_1^2 = \{(2,3)\ \ (3,5)\}$

$S^2 = S^{i-1} + S_1^i$

$= S^{2-1} + S_1^2$

$= S^1 + S_1^2$

$= \{(0,0)\ (1,2)\} + \{(2,3)\ (3,5)\}$

$= \{(0,0)\ (1,2)\ (2,3)\ (3,5)\}$

$S_1^3 = S^{3-1} + (P_3\ W_3)$

$= S^2 + (P_3\ W_3)$

$= \{(0,0)\ (1,2)\ (2,3)\ (3,5) + (5,4)\}$

$S_1^3 = \{(5,4)\ (6,6)\ (7,7)\ (8,9)\}$

$S^3 = S^{3-1} + S_1^3$

$= S^2 + S_1^3$

$= \{(0,0)\ (1,2)\ (2,3)\ (3,5)\} + \{(5,4)\ (6,6)\ (7,7)\ (8,9)\}$

$= \{(0,0)\ (1,2)\ (2,3)\ \underline{(3,5)\ (5,4)}\ (6,6)\ (7,7)\ (8,9)\}$

<u>purging Rule (Dominance Rule)</u> :- If one of S^{i-1} and S_1^i has a pair (P_j, W_j) And other has a pair (P_K, W_K) And $P_j \le P_K$ and $W_j \ge W_K$ then the pair (P_j, W_j) is discarded.

$\therefore$ (3,5) (5,4) are two pair (p_j, w_j) and (p_k, w_k)

$p_j \le p_k$, and $w_j \ge w_k$

$3 \le 5$, $5 \ge 4$ (False), then the pair

(3,5) has discarded.

then

$S^3 = \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)\ (7,7)\ (8,9)\}$

After Applying purge Rule, we will check the following condition inorder to find solution

* $(p_i, w_i) \in S^n$ and $(p_i, w_i) \notin S^{n-1}$ then.

$x_n = 1$ otherwise $x_n = 0$

* $(6,6) \in S^3$ then

$(6,6) \notin S^2$ — True

then $x_3 = 1$

$(6\,6) - (5,4) = (1,2) \in S^2$

$= (1,2) \in S^{2-1}$

$= (1,2) \in S^1$ — false

$x_2 = 0$

$(1,2) \in S^1$.

$(1,2) \notin S^0$ — ~~false~~ True

$x_1 = 1$

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

$$\sum p_i x_i = p_1 x_1 + p_2 x_2 + p_3 x_3$$
$$= 1 * 1 + 2 * 0 + 5 * 1$$
$$= 1 + 0 + 5$$
$$= \underline{\underline{6}}$$

* Here $(p_i, w_i) = (6, 6)$ in the above problem, so $(p_i, w_i) \notin S^2$. condition True so $x_3 = 1$, third item entirely placed in the bag.

* To get the next item to be placed in the bag subtract (p_3, w_3) from $(6, 6)$ because w_3 is placed in the bag.

* If $(1, 2)$ is in S^1, condition becomes false, so $x_2 = 0$, inorder to find get next item to be placed in bag. $(p_1, w_1) \in (1, 2)$, $(1, 2) \notin S^0$ here condition becomes true so $\underline{x_1 = 1}$

optimal solution $(x_1, x_2, x_3) = (1, 0, 1)$

$\therefore$ max profit $= \sum p_i x_i = 6$

14

Q:- $n=4,\ m=8,\ (P_1\ P_2\ P_3\ P_4) = (1,2,5,6)$

$(w_1\ w_2\ w_3\ w_4) = (2,3,4,5)$

Sol

$S^0 = \{(0,0)\}$

i=1 $\quad S_1^1 = S^{1-1} + (P_1, w_1)$

$= S^0 + (1,2)$

$= (0,0) + (1,2)$

$S_1^1 = \{(1,2)\}$

$S^1 = S^{1-1} + S_1^1$

$= S^0 + S_1^1$

$= (0,0) + (1,2)$

$S^1 = \{(0,0)\ (1,2)\}$

i=2 $\quad S_1^2 = S^{2-1} + (P_2\ w_2)$

$= S^1 + (2,3)$

$= (0,0)\ (1,2) + (2,3)$

$= (2,3)\ (3,5)$

$S^2 = S^{2-1} + S_1^2$

$= S^1 + (2,3)\ (3,5)$

$= (0,0)\ (1,2) + (2,3)\ (3,5)$

$S^2 = \{(0,0)\ (1,2)\ (2,3)\ (3,5)\}$

i=3 $\quad S_1^3 = S^{3-1} + (P_3\ w_3)$

$= S^2 + (P_3\ w_3)$

$= (0,0)\ (1,2)\ (2,3)\ (3,5) + (5,4)$

$S_1^3 = (5,4)\ (6,6)\ (7,7)\ (8,9)$

$S^3 = S^{3-1} + S_1^3$

$= S^2 + S_1^3$

$= \{(0,0)\ (1,2)\ (2,3)\ (3,5) + (5,4)\ (6,6)\ (7,7)\ (8,9)\}$

$S^3 = \{(0,0)\ (1,2)\ (2,3)\ (3,5)\ (5,4)\ (6,6)\ (7,7)\ (8,9)\}$

<u>purging Rule</u> :- $(P_j\ w_j)\ (P_k\ w_k)$ are the two pairs then discard $(P_j\ w_j)$

$P_j \leq P_k$ while $w_j \geq w_k$

* $(3,5)\ (5,4) \Rightarrow 3 \leq 5,\ 5 \geq 4$

$S^3 = \{(0,0)\ (1,2)\ (2,3)\ \cancel{(3,5)}\ (5,4)\ (6,6)\ (7,7)\ (8,9)\}$

<u>i=4</u> $S_1^4 = S^{4-1} + (P_4\ w_4)$

$= S^3 + (P_4\ w_4)$

$= \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)\ (7,7)\ (8,9) + (6,5)\}$

$S_1^4 = \{(6,5)\ (7,7)\ (8,8)\ (11,9)\ (12,11)\ (13,12)\ (14,14)\}$

$S^4 = S^{4-1} + S_1^4$

$= S^3 + S_1^4$

$= \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)\ (7,7)\ (8,9) + (6,5)\ (7,7)$
$(8,8)\ (11,9)\ (12,11)\ (13,12)\ (14,14)\}$

bag size is $m=8$, so we are eliminating Remaining weights from S^4.

15

$S^4 = (0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)(7,7)\ (6,5)\ (7,7)\ (8,8)$

$S^4 = \{(0,0)\ (1,2)\ (2,3)\ (5,4)\ (6,6)\ (7,7)\ (6,5)\ (7,7)\ (8,8)\}$

$\therefore$ bag size $m=8$ the pair is $(8,8)$, we will search for the tuple in which value of w is 8.
we obtain such tuple $(8,8)$ in S^4.

$(8,8) \in S^4$

$(8,8) \notin S^3$ True

then $x_4 = 1$

* so we are eliminating $(8,8) - (6,5)$. because of x_4 is placed in the bag.

$(8,8) - (6,5) \Rightarrow (2,3) \in S^3$

$\notin S^2$ false

so $x_3 = 0$

$\Rightarrow (2,3) \in S^2$

$\notin S^1$ True

then $x_2 = 1$

* so we do $(2,3) - (2,3)$, because of x_2 is placed in the bag

$(2,3) - (2,3) \Rightarrow 0$

$\therefore$ there is no space in the bag then

$x_1 = 0$

optimal solution is $(x_1 x_2 x_3 x_4) = (0 1 0 1)$

max profit $\sum p_i x_i = p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4$

$= 1*0 + 2*1 + 5*0 + 6*1$

$= 0 + 2 + 0 + 6$

$\sum p_i x_i = 8$

Ex:- $n = 4$, $m = 21$, $(p_1 p_2 p_3 p_4) = (2, 15, 8, 1)$

$(w_1 w_2 w_3 w_4) = (10, 15, 6, 9)$

optimal binary search trees (OBST)

* A binary search tree T is a binary tree, either it is empty or each node in the tree contains an identifier and.

1. All identifiers in the left subtree are less than the identifier in the root node
2. All identifiers in the Right subtree are greater than the identifier in the root node
3. The left and Right subtree are also binary search trees.

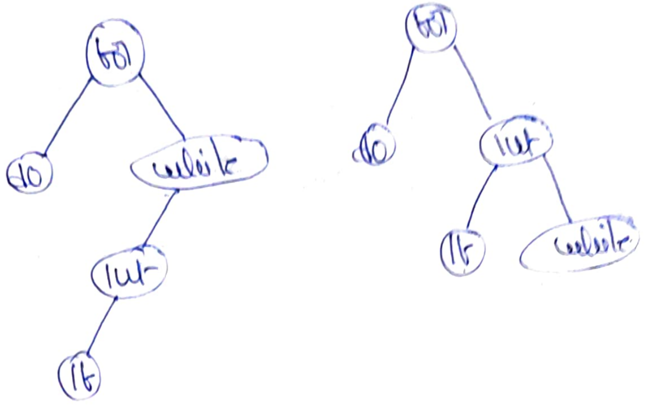
* If we want to search an element in binary search tree, first that element is compared with root node. If element is less than the root node then search continue in left subtree.

* If element is greater than the root node then search continue in right subtree. If element is equal to the root node then print the successful search and terminate search procedure

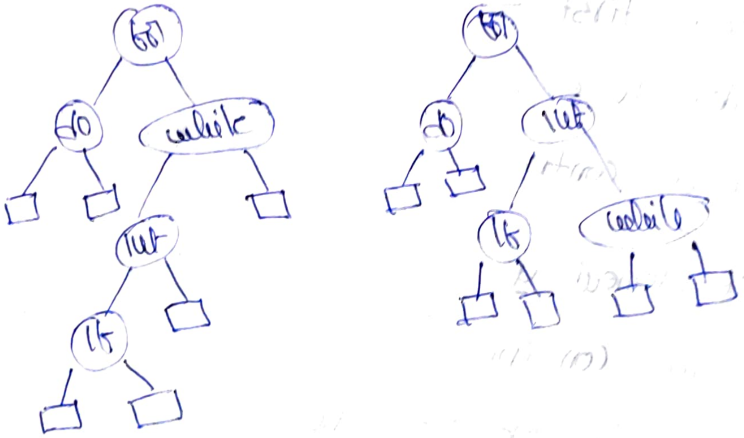
* To give set of identifiers, different binary search trees can be created with different performance characteristics. An optimal binary search tree offers the most economical binary search tree with the lowest average cost of

searching. The problem of obtaining optimal binary search tree as the result of sequence of decisions

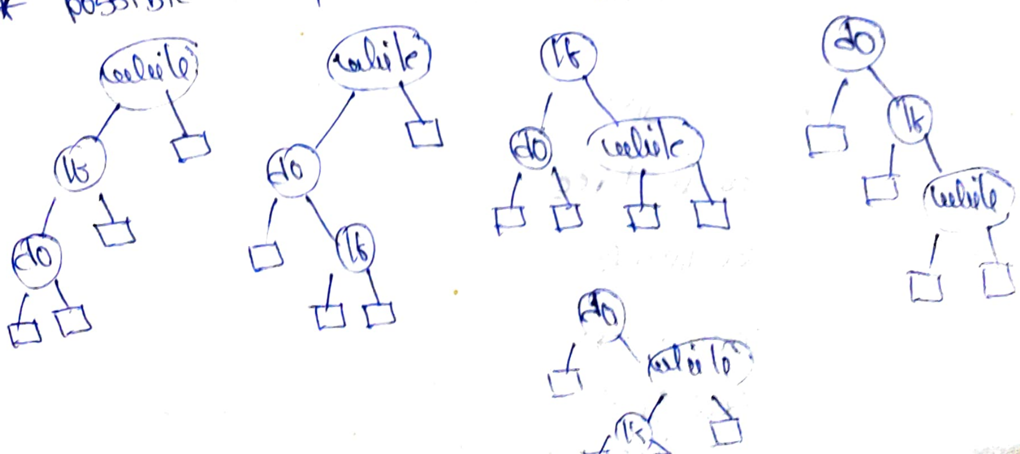
* possible binary search trees {for, do, while, int, if}



* If a binary search tree represents n identifiers then there will be exactly n internal nodes and n+1 external nodes.



* possible binary search trees {while, if, do}



(17)

Q:- using algorithm OBST compute $w(i,j)$, $r(i,j)$ and $c(i,j)$ $0 \le i \le j \le 4$ for the Identifier set $(a_1, a_2, a_3, a_4) =$ (end, goto, print, stop) with $p(1) = 3$, $p(2) = 3$, $p(3) = 1$ $p(4) = 1$, $q(0) = 2$, $q(1) = 3$, $q(2) = 1$, $q(3) = 1$, $q(4) = 1$ using $r(i,j)$ construct the OBST

Sol:- Initially $c(i,i) = 0$, $r(i,i) = 0$, $0 \le i \le 4$

$w(i,i) = \cancel{q(i)}$

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

$w(1,1) = q(1) = 3$

$w(2,2) = q(2) = 1$

$w(3,3) = q(3) = 1$

$w(4,4) = q(4) = 1$

Now optimal trees with one node

$\cancel{c(i,j)} =$

$w(i,i+1) = q[i] + q[i+1] + p(i+1)$

$c(i,i+1) = q[i] + q[i+1] + p(i+1)$

$r(i,i+1) = i+1$

i=0 $w(0,0+1) = q(0) + q(0+1) + p(0+1)$

$w(0,1) = q(0) + q(1) + p(1)$

$= 2 + 3 + 3$

$w(0,1) = 8$

$c(0,0+1) = q(0) + q(0+1) + p(0+1)$

$= q(0) + q(1) + p(1)$

$c(0,1) = 2 + 3 + 3$

$c(0,1) = 8$

Algorithm OBST (p, q, n)

{ // Give n distinct identifiers $a_1 < a_2 < \ldots < a_n$ and

// probabilities p[i], $1 \le i \le n$, and q[i], $0 \le i \le n$, this

// Algorithm computes the cost c[i, j] of optimal binary

// search trees t_{ij} for identifiers $a_{i+1} \ldots a_j$, it also

// computes r[i, j] the root of t_{ij}, w[i, j] is the weight

// of t_{ij}.

```
for i := 1 to n do
{ // initialize
  w[i, i] := q[i] ; r[i, i] = 0 ; c[i, i] = 0.0 ;
  // optimal trees with one node
  w[i, i+1] = q[i] + q[i+1] + p[i+1] ;
  r[i, i+1] = i+1 ;
  c[i, i+1] = q[i] + q[i+1] + p[i+1] ;
}
w[n, n] = q[n] ; r[n, n] = 0 , c[n, n] = 0.0 ;
for m := 2 to n do // find optimal trees with m nodes
  for i = 0 to n-m do
  {
     j = i+m ;
     w[i, j] = w[i, j-1] + p[j] + q[j] ;
     k = find(c, r, i, j) ;
     c[i, j] = w[i, j] + c[i, k-1] + c[k, j] ;
  } r[i, j] = k ;
write (c[0, n], w[0, n], r[0, n]) ;
}
```

$r(0, 0+1) = 0+1$

$r(0, 1) = 1$

$\underline{i=1}$ $w(1, 1+1) = v(1) + v(1+1) + p(1+1)$

$w(1,2) = v(1) + v(2) + p(2)$

$= 3+1+3$

$w(1,2) = 7$

$c(1, 1+1) = v(1) + v(1+1) + p(1+1)$

$c(1,2) = v(1) + v(2) + p(2)$

$= 3+1+3$

$c(1,2) = 7$

$r(1, 1+1) = 1+1$

$r(1,2) = 2$

$\underline{i=2}$ $w(2, 2+1) = v(2) + v(2+1) + p(2+1)$

$w(2,3) = v(2) + v(3) + p(3)$

$= 1+1+1$

$w(2,3) = 3$

$c(2, 2+1) = v(2) + v(2+1) + p(2+1)$

$c(2,3) = v(2) + v(3) + p(3)$

$= 1+1+1$

$c(2,3) = 3$

$r(2, 2+1) = 2+1$

$r(2,3) = 3$

$\underline{i=3}$ $w(3, 3+1) = v(3) + v(3+1) + p(3+1)$

$w(3,4) = v(3) + v(4) + p(4)$

$= 1+1+1$

$w(3,4) = 3$

$c(3,3+1) = v(3) + v(3+1) + p(4)$

$c(3,4) = v(3) + v(4) + p(4)$

$= 1 + 1 + 1$

$c(3,4) = 3$

$r(3,3+1) = 3+1$

$r(3,4) = 4$

// find optimal trees with n-nodes

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

$c(i,j) = \min\limits_{i < k \leq j} \{ c(i,k-1) + c(k,j) + w(i,j) \}$

$r(i,j) = k \quad i < k \leq j$ (k is choosen such that where cost is minimum)

for $j-i = 1$

$w(0,1) = p(1) + v(1) + w(0,1-1)$

$= p(1) + v(1) + w(0,0)$

$= 3 + 3 + 2$

$w(0,1) = 8$

$c(0,1) = \min\limits_{0 < k \leq 1} \{ c(0,1-1) + c(1,1) + w(0,1) \}$

$= \min\limits_{0 < k \leq 1} \{ c(0,0) + c(1,1) + w(0,1) \}$

$= \min\limits_{0 < k \leq 1} \{ 0 + 0 + 8 \}$

$c(0,1) = 8$

$r(0,1) = 1$

$w(1,2) = p(2) + v(2) + w(1,2-1)$

$= p(2) + v(2) + w(1,1)$

$= 3 + 1 + 3$

$w(1,2) = 7$

(19)

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

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

$$= \min_{1<K\le 2} \{ 0 + 0 + 7 \}$$

$$c(1,2) = 7$$

$$\gamma(1,2) = 2$$

$$w(2,3) = q(3) + p(3) + w(2,3-1)$$

$$= q(3) + p(3) + w(2,2)$$

$$= 1 + 1 + 1$$

$$w(2,3) = 3$$

$$c(2,3) = \min_{2<K\le 3} \{ c(2,3-1) + c(3,3) + w(2,3) \}$$

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

$$= \min_{2<K\le 3} \{ 0 + 0 + 3 \}$$

$$c(2,3) = 3$$

$$\gamma(2,3) = 3$$

$$w(3,4) = p(4) + q(4) + w(3,4-1)$$

$$= p(4) + q(4) + w(3,3)$$

$$= 1 + 1 + 1$$

$$= 3$$

$$c(3,4) = \min_{3<K\le 4} \{ c(3,4-1) + c(4,4) + w(3,4) \}$$

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

$$= \min_{3<K\le 4} \{ 0 + 0 + 3 \}$$

$$c(3,4) = 3$$

$r(3,4) = 4$

for $j-i = 2$

$w(0,2) = p[2] + q[2] + w(0,2-1)$

$= p(2) + q(2) + w(0,1)$

$= 3 + 1 + 8$

$w(0,2) = 12$

$c(0,2) = \min_{0<k\leq 2} \{ c[0,1-1] + c[1,2],\ c(0,2-1) + c(2,2) \} + w(0,2) \}$

$= \min_{0<k\leq 2} \{ \{ c(0,0) + c(1,2),\ c(0,1) + c(2,2) \} + w(0,2) \}$

$= \min_{0<k\leq 2} \{ \{ 0 + 7,\ 8 + 0 \} + 12 \}$

$= \min_{0<k\leq 2} \{ \{ 7, 8 \} + 12 \}$

$= \min_{0<k\leq 2} \{ 7 + 12 \}$

$c(0,2) = 19$

$r(0,2) = 1$

$w(1,3) = p(3) + q(3) + w(1,3-1)$

$= p(3) + q(3) + w(1,2)$

$= 1 + 1 + 7$

$w(1,3) = 9$

$c(1,3) = \min_{1<k\leq 3} \{ c(1,2-1) + c(2,3),\ c(1,3-1) + c(3,3) \} + w(1,3) \}$

$= \min_{1<k\leq 3} \{ \{ c(1,1) + c(2,3),\ c(1,2) + c(3,3) \} + w(1,3) \}$

$= \min_{1<k\leq 3} \{ 0 + 3,\ 7 + 0 \} + 9 \}$

$= \min_{1<k\leq 3} \{ \{ 3, 7 \} + 9 \}$

$= \min_{1<k\leq 3} \{ 3 + 9 \}$

$c(1,3) = 12$

$r(1,3) = 2$

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

$$c(2,4) = \min_{2<k\le 4} \{ \{ \underbrace{c(2,3-1) + c(3,4)}_{k=3}, \underbrace{c(2,4-1) + c(4,4)}_{k=4} \} + w(2,4) \}$$
$$= \min_{2<k\le 4} \{ c(2,2) + c(3,4), c(2,3) + c(4,4) \} + w(2,4) \}$$
$$= \min_{2<k\le 4} \{ 0+3, 3+0 \} + 5 \}$$
$$= \min_{2<k\le 4} \{ 3, 3 \} + 5 \}$$
$$= \min_{2<k\le 4} \{ 3+5 \}$$
$$c(2,4) = 8$$

$$r(2,4) = \underline{\underline{3}}$$

For $j-i=3$

$$w(0,3) = p(3) + q(3) + w(0,3-1)$$
$$= p(3) + q(3) + w(0,2)$$
$$= 1+1+12$$
$$w(0,3) = 14$$

$$c(0,3) = \min_{0<k\le 3} \{ \{ \underbrace{c(0,1-1) + c(1,3)}_{k=1}, \underbrace{c(0,2-1) + c(2,3)}_{k=2}, \underbrace{c(0,3-1) + c(3,3)}_{k=3} + w(0,3) \}$$
$$= \min_{0<k\le 3} \{ c(0,0) + c(1,3), c(0,1) + c(2,3), c(0,2) + c(3,3) \} + w(0,3)$$
$$= \min_{0<k\le 3} \{ 0+12, 8+3, 19+0 \} + 14 \}$$
$$= \min_{0<k\le 3} \{ 12, 11, 19 \} + 14 \}$$
$$= \min_{0<k\le 3} \{ 11+14 \} \Rightarrow 25$$

$r(0,3) = 2$

$w(1,4) = p(4) + q(4) + w(1,4-1)$

$= p(4) + q(4) + w(1,3)$

$= 1 + 1 + 9$

$w(1,4) = 11$

$c(1,4) = \min_{1<k\le 4} \{ \underbrace{c(1,2-1) + c(2,4)}_{k=2}, \underbrace{c(1,3-1) + c(3,4)}_{k=3}, \underbrace{c(1,4-1) + c(4,4)}_{k=4} + w(1,4) \}$

$= \min_{1<k\le 4} \{ c(1,1) + c(2,4), c(1,2) + c(3,4), c(1,3) + c(4,4) \} + 11 \}$

$= \min_{1<k\le 4} \{ 0 + 8, 7 + 3, 12 + 0 \} + 11 \}$

$= \min_{1<k\le 4} \{ 8, 10, 12 \} + 11 \}$

$= \min_{1<k\le 4} \{ 8 + 11 \}$

$= \underline{19}$

$r(1,4) = \underline{2}$

$\underline{j-i=4}$

$w(0,4) = p(4) + q(4) + w(0,4-1)$

$= 1 + 1 + w(0,3)$

$= 1 + 1 + 14$

$w(0,4) = 16$

$c(0,4) = \min_{0<k\le 4} \{ \underbrace{c(0,1-1) + c(1,4)}_{k=1}, \underbrace{c(0,2-1) + c(2,4)}_{k=2}, \underbrace{c(0,3-1) + c(3,4)}_{k=3}, \underbrace{c(0,4-1) + c(4,4)}_{k=4} \} + w(0,4) \}$

$= \min_{0<k\le 4} \{ c(0,0) + c(1,4), c(0,1) + c(2,4), c(0,2) + c(3,4), c(0,3) + c(4,4) + w(0,4) \}$

$= \min_{0<k\le 4} \{ 0 + 19, 8 + 8, 19 + 3, 25 + 0 \} + 16 \}$

$= \min_{0<k\le 4} \{ 19, \underline{16}, 22, 25 \} + 16 \}$

$= \min_{0<k\le 4} \{ 16 + 16 \} = \underline{32}$

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

$r(0,4) = 2$

To build OBST, $r(0,4) = 2$

$k = 2$

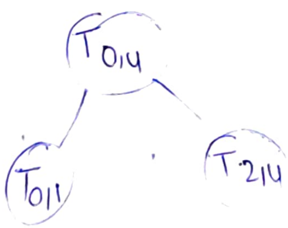
Hence a_2 becomes root node

Let T be OBST, $\boxed{T_{i,j} = T_{i,k-1},\ T_{k,j}}$

So $T_{0,4}$ is divided into two parts

$T_{0,4} = T_{0,2-1},\ T_{2,4}$

$= T_{0,1},\ T_{2,4}$



$T_{0,1} = r(0,1) = 1 \Rightarrow k = 1$

$T_{2,4} = r(2,4) = 3 \Rightarrow k = 3$

$T_{0,1}$ is divided into two parts

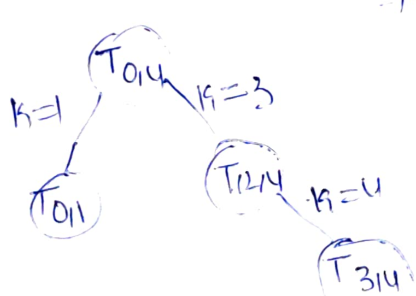
$T_{0,1} = T_{0,1-1},\ T_{1,1}$

$= T_{0,0},\ T_{1,1}$ [$\because k = 1$]

$T_{2,4}$ is divided into two parts

$T_{2,4} = T_{2,3-1},\ T_{3,4}$

$= T_{2,2},\ T_{3,4}$



$\therefore$ Since $r_{00}, r_{11}, r_{22}, r_{33}, r_{44}$ is 0 they are external nodes. And can be neglected.

Q:- OBST compute $w(i,j), r(i,j), c(i,j)$ $0 \le i \le j \le 4$ for the identifier set (a_1, a_2, a_3, a_4) = (cout, float, if, while) with $p_1 = \frac{1}{20}$, $p_2 = \frac{1}{5}$, $p_3 = \frac{1}{10}$, $p_4 = \frac{1}{20}$, $q_0 = \frac{1}{5}$, $q_1 = \frac{1}{10}$, $q_2 = \frac{1}{5}$, $q_3 = \frac{1}{20}$, $q_4 = \frac{1}{20}$ using $r(i,j)$ to construct OBST

Sol) ~~$P_1 = \frac{1}{20} \times$~~ every value is multiply with 20 (P & Q values are multiply with 20) because we will get only numerical values only.

$$P_1 = \frac{1}{\cancel{20}} \times \cancel{20} = 1 \qquad Q_0 = \frac{1}{\cancel{5}} \times \cancel{20}^{4} = 4$$

$$P_2 = \frac{1}{\cancel{5}} \times \cancel{20}^{4} = 4 \qquad Q_1 = \frac{1}{\cancel{10}} \times \cancel{20}^{2} = 2$$

$$P_3 = \frac{1}{\cancel{10}} \times \cancel{20} = 2 \qquad Q_2 = \frac{1}{\cancel{5}} \times \cancel{20}^{4} = 4$$

$$P_4 = \frac{1}{\cancel{20}} \times \cancel{20} = 1 \qquad Q_3 = \frac{1}{\cancel{20}} \times \cancel{20} = 1$$

$$Q_4 = \frac{1}{\cancel{20}} \times \cancel{20} = 1$$

$\therefore$ $P_1 = 1$, $P_2 = 4$, $P_3 = 2$, $P_4 = 1$

$Q_0 = 4$, $Q_1 = 2$, $Q_2 = 4$, $Q_3 = 1$, $Q_4 = 1$

Reliability design

* we will discuss the problem based on multiplicative optimization function. we will consider such a system in which many devices are connected in series

I/P → D_1 → D_2 → ...... → D_n → O/P

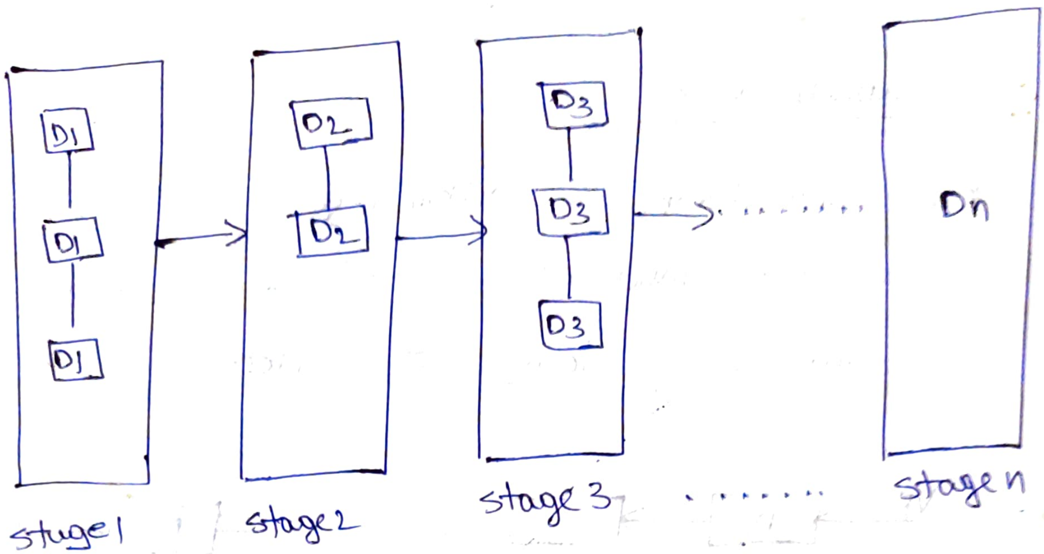
* when devices are connected together then it is a necessity that each device should work properly. the probability that device 'i' will work properly is called reliability of that device.

* let r_i be the reliability of device D_i then reliability of entire system is $\prod r_i$.

* it may happen that even if reliability of individual device is very good but reliability of entire system may not be good.

* Hence we can obtain the good performance from entire system we can duplicate individual devices and can connect them in a series.

* To do so, we will Attach switching circuits. The job of switching circuit is to determine which device in a group is working properly.



multiple devices connected together

* Let m_i be the copies of device D_i then $(1-r_i)^{m_i}$ be the probability that all m_i have a malfunction.

* Hence the stage reliability is $1-(1-r_i)^{m_i}$.

$$\therefore \phi_i(m_i) = 1-(1-r_i)^{m_i}$$

* In reliability design problem we except to get maximum reliability using device duplication.

* the upperbound u_i on the cost can be determined as.

$$u_i = \left[\left[C + C_i - \sum_{1}^{n} C_j \right] / C_i \right]$$

Q1:- Design a three stage system with device types D_1, D_2, D_3. The costs are Rs. 30, 15, 20 Respectively. The cost of the system is to be no more than 105. The Reliability of each device type is 0.9, 0.8 and 0.5 Respectively.

Sol:- let us compute u_1, u_2 and u_3 first.

$$u_i = \left[\left[c + c_i - \sum_{1}^{n} c_j \right] / c_i \right]$$

$$u_1 = (105 + 30 - (30+15+20))/30$$

$$= ((135 - 65)/30)$$

$$= 2.333$$

$$u_1 = 2$$

$$u_2 \left[(105 + 15 - (30+15+20)) / 15 \right]$$

$$= ((120 - 65)/15)$$

$$= 3.666$$

$$u_2 = 3$$

$$u_3 = \left[(105 + 20 - (30+15+20)) / 20 \right]$$

$$= ((125 - 65)/20)$$

$$u_3 = 3$$

$\therefore u_1 = 2, u_2 = 3, u_3 = 3$

* $\phi_i(m_i) = 1 - (1 - r_i)^{m_i}$, $\phi_i(m_i)$ is a Reliability function using S_j^i to represent All tuples of obtainable from S^{i-1} by choosing $m_i = j$

Beginning with $S^0 = \{(1,0)\}$

* So by using S_j^i calculate S_1^1 here $i=1, j=1, m_1=1$

$$\phi_1(m_1) = 1-(1-r_1)^1 = 1-(1-0.9)^1 = 1-0.1 = 0.9$$

$$S^0 = \{(1,0)\}$$

* In each tuple (r, x) reliability is multiplied with previous reliability and cost is added to previous cost

$$S_1^1 = \{(1 * 0.9, 0+30)\}$$

$$S_1^1 = \{(0.9, 30)\}$$

* Here reliability (0.9) is multiplied with 1 and cost $c_1 = 30$ is added to 0. it means the operation is performed for previous pair $(1,0)$.

* Now S_2^1 is calculated as follows, $i=1, j=2, m_1=2$

$$\phi_1(m_1) = 1-(1-r_1)^2 = 1-(1-0.9)^2$$
$$= 1-(0.01)$$
$$= 0.99$$

* Now 0.99 is multiplied with 1 and cost $2c_1 = 2 \times 30 = 60$ is added to 0.

$$S_2^1 = \{(1 * 0.99, 0+60)\}$$

$$S_2^1 = \{(0.99, 60)\}$$

* We can't calculate S_1^3 since upper bound is 2 i.e $u_1 = 2$

* S^1 can be obtained by merging the sets S_1^1, S_2^1

$$S^1 = \{(0.9, 30) (0.99, 60)\}$$

25

$s^1_2 = \{(0.9, 30)\ (0.99, 60)\}$

* similarly s_1^2, s_2^2, s_3^2 can be calculated from s^1

* s_1^2 here $i=2, j=1$, i.e $m_2=1$

$$\phi_2(m_2) = 1-(1-r_2)^{m_2}$$
$$= 1-(1-0.8)^1$$
$$= 1-(0.2)$$
$$\phi_2(m_2) = 0.8$$

→ here the reliability (0.8) is multiplied with previous reliability in s^1, And cost (15) is added to previous costs in s^1

$$s_1^2 = \{(0.9 * 0.8,\ 30+15),\ (0.99 * 0.8,\ 60+15)\}$$
$$s_1^2 = \{(0.72, 45)\ (0.792, 75)\}$$

* s_2^2 here $i=2, j=2$ i.e $m_2 = 2$

$$\phi_2(m_2) = 1-(1-r_2)^{m_2}$$
$$= 1-(1-0.8)^2$$
$$= 1-(0.2)^2$$
$$= 1-0.04$$
$$\phi_2(m_2) = 0.96$$

* here the reliability (0.96) is multiplied with reliability in s^1 pair, And cost $2c_2 = 2*15 = 30$ is added to s^1 pair.

$S_2^2 = \{(0.9 * 0.96, 30+30), (0.99 * 0.96, 60+30)\}.$

$= \{(0.864, 60)(0.9504, 90)\}$

$S_2^2 = \{(0.864, 60)(0.9504, 90)\}$

* now S_3^2 can be calculated, here $i=2, j=3$, i.e $m_2 = 3$

$$q_2(m_2) = 1-(1-r_2)^{m_2}$$
$$= 1-(1-0.8)^3$$
$$= 1-(0.2)^3$$
$$= 0.992$$

* now reliability 0.992 is multiplied with the S^1 pair,

And cost $3C_2 = 3*15 = 45$

$S_3^2 = \{(0.9 * 0.992, 30+45)(0.99 * 0.992, 60+45)\}$

$S_3^2 = \{(0.8928, 75)(0.98208, 105)\}$

* $S^2 = S_1^2 \cup S_2^2 \cup S_3^2$

$= \{(0.72, 45)(0.792, 75), (0.864, 60), (0.9504, 90),$
$(0.8928, 75)(0.98208, 105)\}$

* observe the cost is in increasing order

$45 < 75 < 60 < 90 < 75 < 105$ here $75 < 60$ fail

and $90 < 75$ condition fail so remove from S^2

$(0.792, 75)(0.9504, 90)$ remove from S^2

$\therefore S^2 = \{(0.72, 45)(0.864, 60), (0.8928, 75), (0.98208, 105)\}$

* Similarly S_1^3, S_2^3, S_3^3 can be calculated from S^2 as follows

* S_1^3 here $i=3, j=1, m_3=1$

$\phi_3(m_3) = 1-(1-r_3)^{m_3}$
$= 1-(1-0.5)^1$
$= 1-0.5$
$= 0.5$

* $C_3 = 20$ is added to S^2 cost, 0.5 is multiplied with S^2 reliability

* $S_1^3 = \{(0.72*0.5, 45+20)(0.864*0.5, 60+20)$
$(0.8928*0.5, 75+20)(0.98208*0.5, 105+20)\}$

$S_1^3 = \{(0.36, 65)(0.432, 80)(0.4464, 95)(\quad, 125)\}$

* Last tuple is removed from S_1^3, since cost 125 exceeding the given cost 105.

$S_1^3 = \{(0.36, 65)(0.432, 80)(0.4464, 95)\}$

* S_2^3 here $i=3, j=2, m_3=2$

$\phi_3(m_3) = 1-(1-r_3)^{m_3}$
$= 1-(1-0.5)^2$
$= 1-(1-0.5)^2$
$= 1-(0.5)^2$
$= 1-0.25$
$= 0.75$

∴ The reliability 0.75 is multiplied with previous s^2 reliability, And the cost $2C_3 = 2 * 20 = 40$ is added to previous s^2 cost

$$S_2^3 = \{(0.72 * 0.75,\ 45+40)\ (0.864 * 0.75,\ 60+40)$$
$$(0.9504 * 0.75,\ 90+40) \cdot (0.8928 * 0.75,\ 75+40)$$
$$(0.98208 * 0.75,\ 105+40)\}$$

$$S_2^3 = \{(0.54,\ 85)\ (0.648,\ 100)\}\ (\cdot\ ,\ 130)\ (\cdot\ ,\ 115)$$
$$(\cdot\ ,\ 145)\}$$

* Last 3 tuples are removed from S_2^3, since cost 130, 115, 145 exceeding the given cost 105

$$\therefore S_2^3 = \{(0.54, 85)\ (0.648, 100)\}$$

* S_3^3, here $i=3,\ j=3,\ m_3=3$

$$\phi_3(m_3) = 1-(1-r_3)^{m_3}$$
$$= 1-(1-0.5)^3$$
$$= 1-(0.5)^3$$
$$= 1-0.125$$
$$= 0.875$$

* The reliability 0.875 is multiplied with previous s^2 Reliability, And the cost $3C_3 = 3 * 20 = 60$ is added to previous s^2 cost

$$\therefore S_3^3 = \{(0.72 * 0.875,\ 45+60)\}$$

$$S_3^3 = \{(0.63,\ 105)\}$$

* Remaining All costs are exceeding the given cost 105

$S^3 = S_1^3 \cup S_2^3 \cup S_3^3$

$= \{(0.36, 65), (0.432, 80) \cdot (0.4464, 95) (0.54, 85) (0.648, 100) (0.63, 105)\}$

Apply the purging and Domiance Rules

$S^3 = (0.36, 65) (0.432, 80) (0.54, 85) (0.648, 100)$

* the best design has reliablialitey of 0.648, and cost of 100 (0.648, 100) pair present in S_2^3, so i = 3, j = 2 and $m_3 = 2$.

* the (0.648, 100) obtained from (0.864, 60) which is present in S_2^2; so i = 2, j = 2 and $m_2 = 2$

* (0.864, 60) pair obtain from (0.9, 30) which is in S_1^1, i = 1, j = 1, $m_1 = 1$

$\therefore m_1 = 1, m_2 = 2, m_3 = 2$