UNIT-IV

Dynamic Programming : the General Method, All pairs shortest path problem, optimal binary search tree, 0/1 knapsack problem, Reliability design, the traveling salesperson problem, Matrix chain multiplication.

General Method :-

Dynamic programming is an algorithm design method that solves a given complex problem by breaking it into subproblems, solving each of those sub problems just once and storing their result in an array.

Dynamic programming follows a principle called "principle of optimality".

principle of optimality states that an optimal sequence of decisions has the property that what ever the initial state and decision are, but the remaining decisions must constitute an optimal decision sequence with regard to the state resulting from the first decision.

The two properties of a problem that suggest that the given problem can be solved using dynamic programming

1. optimal substructure
2. overlapping sub problems.

optimal substructure :-

A given problem has optimal substructure property, when optimal solution of the given problem can be obtained by using optimal solution of its sub problems.

Scanned with OKEN Scanner

Ex:- The shortest path problem has the optimal substructure property.

overlapping subproblems :-

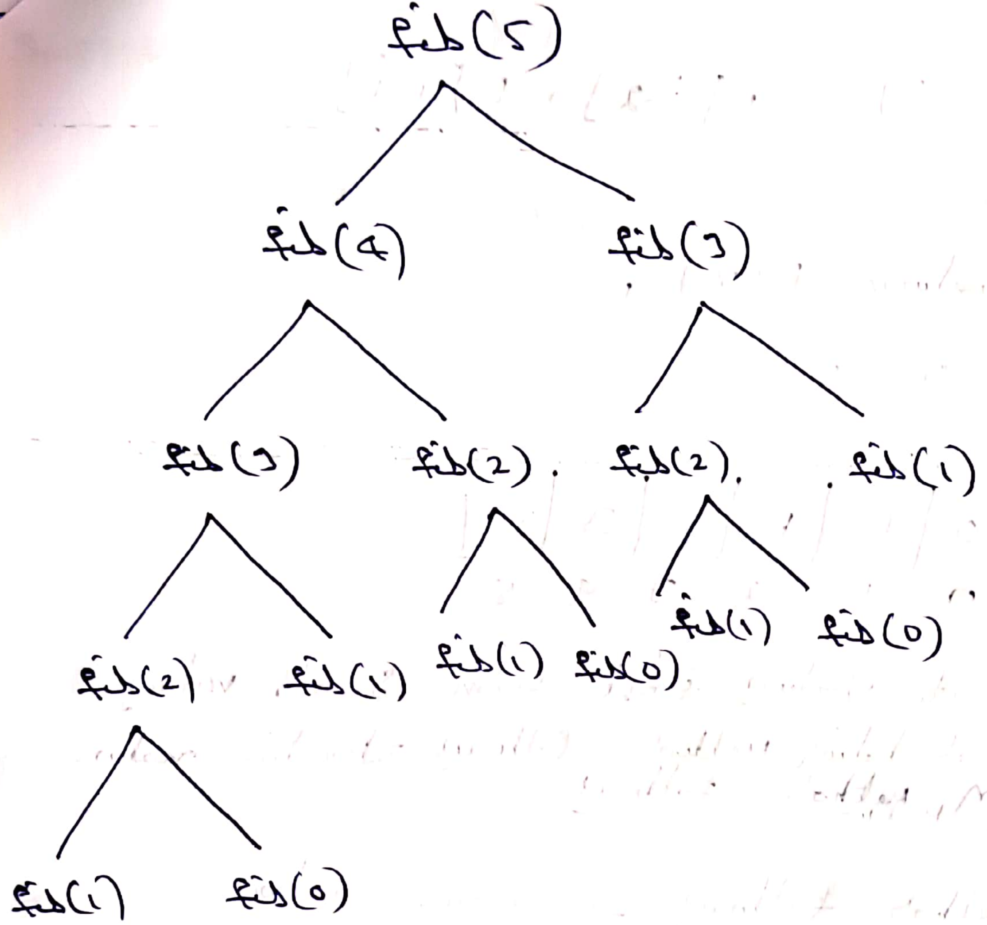
→ Like divide and Conquer, dynamic programming combines the solutions of subproblems.

→ Dynamic programming is mainly used when the solutions of same (overlapping) subproblems are needed again and again. The computed solutions of subproblems are stored in a table so that these subproblems donot have to be recomputed.

Ex:- Consider recursive program for fibonacci numbers, there are many subproblems which are solved again and again.

we can see that the function fib(3) is being called 2 times. If we would have stored the value of fib(3), then instead of computing it again, we could have reused the old stored value

```
Int fib (int n)
{
    If (n <= 1) then
        return n;
    else
        return fib(n-1) + fib(n-2);
}
```



There are two different ways to store the values, so that the values can be reused in the problem.

(1) Tabulation, (2) Memorization.

Tabulation Method :-

For a given problem, Tabulation method builds a table in bottom-up approach and returns the last entry from the table.

Ex :-

```
int fib (int n)
{
   if (n <= 1)
     return n;
   F(0) = 0;
   F(1) = 1;
for (i=2; i<=n; i++)
```

{

$F(i) = F[i-2] + F[i-1]$

}

return $F(n)$;

}

Suppose $n = 5$

F	0	1	1	2	3	5
	0	1	2	3	4	5

filling of values first from small values.

Tabulation method follows iterative nature.

<u>Memorization Method</u> :-

→ This method follows recursive nature

→ This method follows Top-down approach

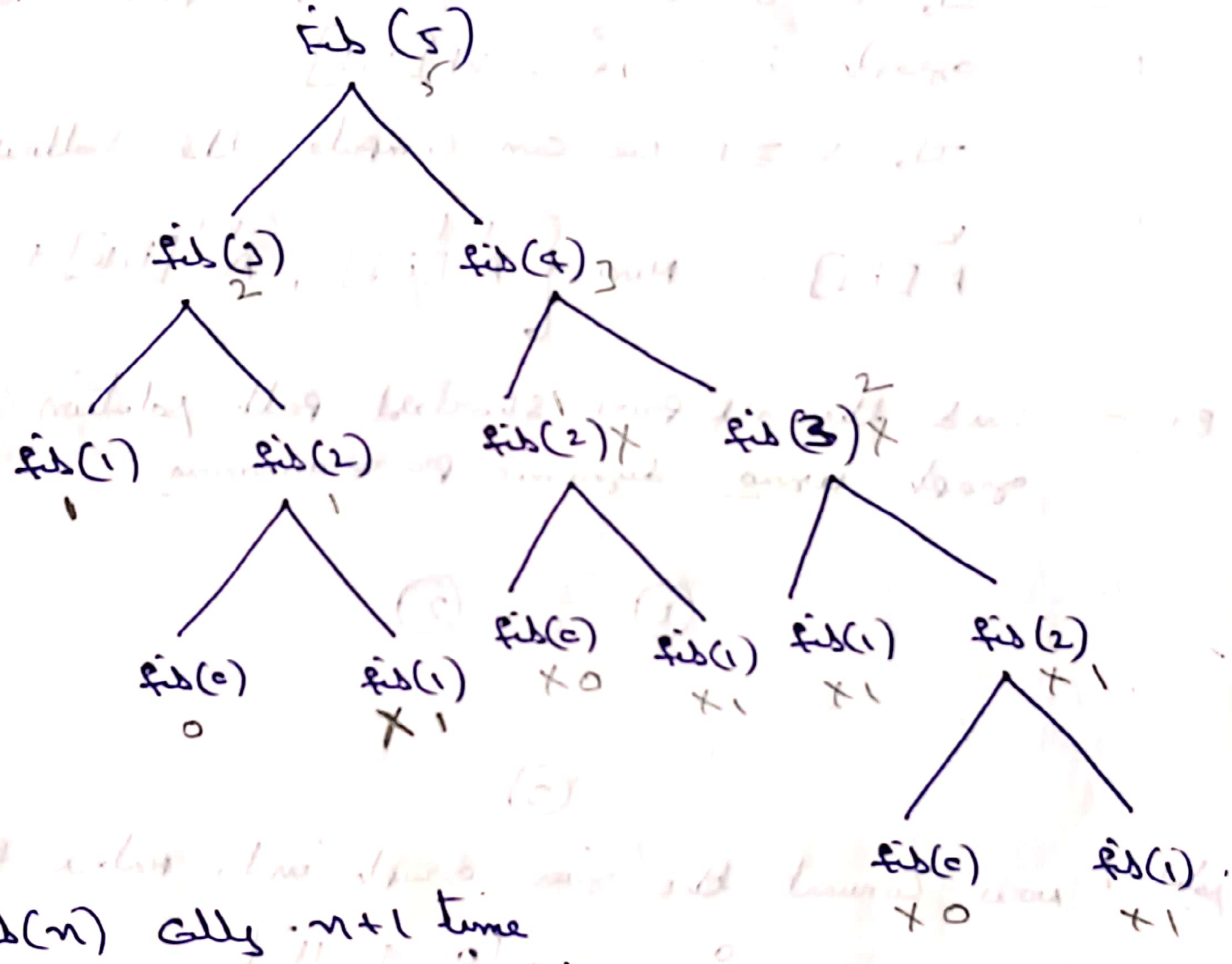
→ In this method, we initialize a look up table with initial values as -1. whenever we need the solution to a subproblem, we first look into the look-up table; If the precomputed value is there then we return that value, otherwise we calculate the value and put the result in the look-up table, so that it can be reused later.

<u>Ex</u>:-

```
int fib(int n)
{
    if (n <= 1) then
        return n;
    else
    return fib(n-2) + fib(n-1);
}
```

Suppose $n = 5$

F	0	1	2	3	4	5
	~~-1~~ 0	~~-1~~ 1	~~-1~~ 1	~~-1~~ 2	~~-1~~ 3	~~-1~~ 5



fib(n) calls n+1 time

fib(5) calls 5+1 = 6 times.

All pairs shortest path problem (FLOYd - Warshall)

→ Suppose the given graph is weighted and connected graph.

→ the objective of all pairs shortest path problem is to find shortest path between each and every pair of nodes present in the given graph.

→ To find shortest path between every pair nodes using dynamic programming we need to find $A^k[i,j]$ where $k = 0$ to n.

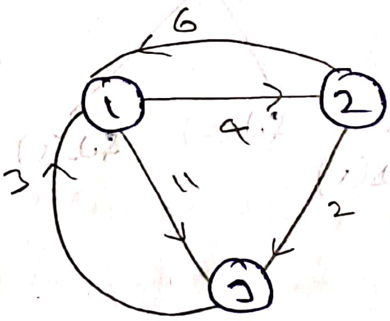
$A^k[i,j]$ represents path between i to j via k.

→ If $k = 0$, we can write A^0 from the given graph i.e $A^0 = W[i,j]$

If $k \geq 1$ we can compute the following

$$\boxed{A^k[i,j] = \text{Mini}\left\{A^{k-1}[i,j],\ A^{k-1}[i,k] + A^{k-1}[k,j]\right\}}$$

Ex:- Find the all pairs shortest path solution for the graph using dynamic programming.



Soln:- Now convert the given graph into matrix form

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

Now we have to find A^1 matrix

i.e $A^1 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & - \\ 3 & - & 0 \end{bmatrix}$

Now $A^1[2,3] = \text{Mini}\left\{A^{1-1}[2,3],\ A^{1-1}[2,1] + A^{1-1}[1,3]\right\}$

$= \text{Mini}\left\{A^0[2,3],\ A^0[2,1] + A^0[1,3]\right\}$

$= \text{Mini}\left\{2,\ 6+11\right\}$

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

$= 2$

$A^1[3,2] = \min\{A^{1-1}[3,2], A^{1-1}[3,1] + A^{1-1}[1,2]\}$

$= \min\{A^0[3,2], A^0[3,1] + A^0[1,2]\}$

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

$= 7$

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

Now we have to find A^2 matrix

$$\text{i.e } A^2 = \begin{bmatrix} 0 & 4 & - \\ 6 & 0 & 2 \\ - & 7 & 0 \end{bmatrix}$$

Now $A^2[1,3] = \min\{A^{2-1}[1,3], A^{2-1}[1,2] + A^{2-1}[2,3]\}$

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

$= 6$

$A^2[3,1] = \min\{A^{2-1}[3,1], A^{2-1}[3,2] + A^{2-1}[2,1]\}$

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

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

$= 3$

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

Now we have to find A^3 Matrix

$$\text{i.e. } A^3 = \begin{bmatrix} 0 & - & 6 \\ - & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

Now $A^3[1,2] = \text{Mini}\{A^{3-1}[1,2],\ A^{3-1}[1,3] + A^{3-1}[3,2]\}$

$= \text{Mini}\{4,\ 6+7\}$

$= 4$

$A^3[2,1] = \text{Mini}\{A^{3-1}[2,1],\ A^{3-1}[2,3] + A^{3-1}[3,1]\}$

$= \text{Mini}\{6,\ 2+3\}$

$= 5$

$$\therefore A^3 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 6 \\ 2 & 5 & 0 & 2 \\ 3 & 3 & 7 & 0 \end{array}$$

Hence the distances between All pairs obtained.

Ex Find the all pairs shortest path solution for the graph represented by below adjacency matrix

$$\begin{bmatrix} 0 & 6 & 5 & 4 \\ 3 & 0 & 2 & 6 \\ 18 & 6 & 0 & 7 \\ 8 & 12 & 10 & 0 \end{bmatrix}$$

Solu: Let

$$A^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 6 & 5 & 4 \\ 2 & 3 & 0 & 2 & 6 \\ 3 & 18 & 6 & 0 & 7 \\ 4 & 8 & 12 & 10 & 0 \end{array}$$

we have to find A^1 Matrix

$$A^1 = \begin{bmatrix} 0 & 6 & 5 & 4 \\ 3 & 0 & - & - \\ 18 & - & 0 & - \\ 8 & - & - & 0 \end{bmatrix}$$

$$A^1[2,3] = \text{mini}\{A^{1-1}[2,3],\ A^{1-1}[2,1] + A^{1-1}[1,3]\}$$
$$= \text{mini}\{2,\ 3+5\}$$
$$= 2$$

$$A^1[2,4] = \text{mini}\{A^{1-1}[2,4],\ A^{1-1}[2,1] + A^{1-1}[1,4]\}$$
$$= \text{mini}\{6,\ 3+4\}$$
$$= 6$$

$$A^1[3,2] = \text{mini}\{A^{1-1}[3,2],\ A^{1-1}[3,1] + A^{1-1}[1,2]\}$$
$$= \text{mini}\{6,\ 18+6\}$$
$$= 6$$

$$A^1[3,4] = \text{mini}\{A^{1-1}[3,4],\ A^{1-1}[3,1] + A^{1-1}[1,4]\}$$
$$= \text{mini}\{7,\ 18+4\}$$
$$= 7$$

$$A^1[4,2] = \text{mini}\{A^{1-1}[4,2],\ A^{1-1}[4,1] + A^{1-1}[1,2]\}$$
$$= \text{mini}\{12,\ 8+6\}$$
$$= 12$$

$$A^1[4,3] = \text{mini}\{A^{1-1}[4,3],\ A^{1-1}[4,1] + A^{1-1}[1,3]\}$$
$$= \text{mini}\{10,\ 8+5\}$$
$$= 10$$

$$\therefore \quad A^1 = \begin{bmatrix} 0 & 6 & 5 & 4 \\ 3 & 0 & 2 & 6 \\ 18 & 6 & 0 & 7 \\ 8 & 12 & 10 & 0 \end{bmatrix}$$

now we have to find A^2 Matrix

$$\text{i.e } A^2 = \begin{bmatrix} 0 & 6 & - & - \\ 3 & 0 & 2 & 6 \\ - & 6 & 0 & - \\ - & 12 & - & 0 \end{bmatrix}$$

$$A^2[1,3] = \text{Mini}\{A^{2-1}[1,3],\ A^{2-1}[1,2] + A^{2-1}[2,3]\}$$

$$= \text{Mini}\{5,\ 6+2\}$$

$$= 5$$

$$A^2[1,4] = \text{Mini}\{A^{2-1}[1,4],\ A^{2-1}[1,2] + A^{2-1}[2,4]\}$$

$$= \text{Mini}\{4,\ 6+6\}$$

$$= 4$$

$$A^2[3,1] = \text{Mini}\{A^{2-1}[3,1],\ A^{2-1}[3,2] + A^{2-1}[2,1]\}$$

$$= \text{Mini}\{18,\ 6+3\}$$

$$= 9$$

$$A^2[3,4] = \text{Mini}\{A^{2-1}[3,4],\ A^{2-1}[3,2] + A^{2-1}[2,4]\}$$

$$= \text{Mini}\{7,\ 6+6\}$$

$$= 7$$

$$A^2[4,1] = \text{Mini}\{A^{2-1}[4,1],\ A^{2-1}[4,2] + A^{2-1}[2,1]\}$$

$$= \text{Mini}\{8,\ 12+3\}$$

$$= 8$$

$$A^{2}[4,3] = \text{Mini}\{A^{2-1}[4,3],\ A^{2-1}[4,2] + A^{2-1}[2,3]\}$$
$$= \text{Mini}\{10,\ 12+2\}$$
$$= 10$$

$$\therefore\ A^{2} = \begin{bmatrix} 0 & 6 & 5 & 4 \\ 3 & 0 & 2 & 6 \\ 9 & 6 & 0 & 7 \\ 8 & 12 & 10 & 0 \end{bmatrix}$$

Now we have to find A^3 matrix

$$\text{i.e } A^{3} = \begin{bmatrix} 0 & - & 5 & - \\ - & 0 & 2 & - \\ 9 & 6 & 0 & 7 \\ - & - & 10 & 0 \end{bmatrix}$$

$$A^{3}[1,2] = \text{Mini}\{A^{3-1}[1,2],\ A^{3-1}[1,3] + A^{3-1}[3,2]\}$$
$$= \text{Mini}\{6,\ 5+6\}$$
$$= 6$$

$$A^{3}[1,4] = \text{Mini}\{A^{3-1}[1,4],\ A^{3-1}[1,3] + A^{3-1}[3,4]\}$$
$$= \text{Mini}\{4,\ 5+7\}$$
$$= 4$$

$$A^{3}[2,1] = \text{Mini}\{A^{3-1}[2,1],\ A^{3-1}[2,3] + A^{3-1}[3,1]\}$$
$$= \text{Mini}\{3,\ 2+9\}$$
$$= 3$$

$$A^{3}[2,4] = \text{Mini}\{A^{3-1}[2,4],\ A^{3-1}[2,3] + A^{3-1}[3,4]\}$$
$$= \text{Mini}\{6,\ 2+7\}$$
$$= 6.$$

$$A^{3}[4,1] = \text{Mini}\{A^{3-1}[4,1],\ A^{3-1}[4,3] + A^{3-1}[3,1]\}$$
$$= \text{Mini}\{8,\ 10+9\}$$
$$= 8$$

$A^3[4,2] = \text{Mini}\{A^{3-1}[4,2], A^{3-1}[4,3] + A^{3-1}[3,$

$= \text{Mini}\{12, 10+6\}$

$= 12$

$$\therefore A^3 = \begin{bmatrix} 0 & 6 & 5 & 4 \\ 3 & 0 & 2 & 6 \\ 9 & 6 & 0 & 7 \\ 8 & 12 & 10 & 0 \end{bmatrix}$$

now we have to find A^4 matrix

$$\text{i.e} \quad A^4 = \begin{bmatrix} 0 & - & - & 4 \\ - & 0 & - & 6 \\ - & - & 0 & 7 \\ 8 & 12 & 10 & 0 \end{bmatrix}$$

$A^4[1,2] = \text{Mini}\{A^{4-1}[1,2], A^{4-1}[1,4] + A^{4-1}[4,2]\}$

$= \text{Mini}\{6, 4+12\}$

$= 6$

$A^4[1,3] = \text{Mini}\{A^{4-1}[1,3], A^{4-1}[1,4] + A^{4-1}[4,3]\}$

$= \text{Mini}\{5, 4+10\}$

$= 5$

$A^4[2,1] = \text{Mini}\{A^{4-1}[2,1], A^{4-1}[2,4] + A^{4-1}[4,1]\}$

$= \text{Mini}\{3, 6+8\}$

$= 3$

$A^4[2,3] = \text{Mini}\{A^{4-1}[2,3], A^{4-1}[2,4] + A^{4-1}[4,3]\}$

$= \text{Mini}\{2, 6+10\}$

$= 2$

$A^4[3,1] = \text{Mini}\{A^{4-1}[3,1], A^{4-1}[3,4] + A^{4-1}[4,1]\}$

$= \text{Mini}\{9, 7+8\}$

$= 9$

Scanned with OKEN Scanner

$A^4[3,2] = \min\{A^{4-1}[3,2],\ A^{4-1}[3,4] + A^{4-1}[4,2]\}$ 7

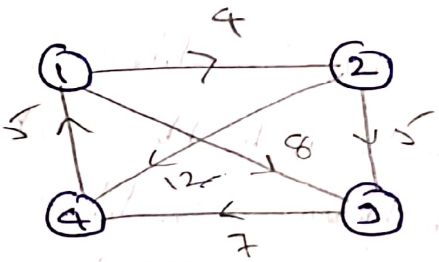
$= \min\{6,\ 7+12\}$

$= 6$

$$\therefore A^4 = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 6 & 5 & 4 \\ 3 & 0 & 2 & 6 \\ 9 & 6 & 0 & 7 \\ 8 & 12 & 10 & 0 \end{bmatrix} \end{array}$$

Hence the distances between All pairs obtained.

Ex:- Find the all pairs shortest path solution for the graph using dynamic programming



Ans
$$A^0 = \begin{bmatrix} 0 & 4 & 8 & \infty \\ \infty & 0 & 5 & 12 \\ \infty & \infty & 0 & 7 \\ 5 & \infty & \infty & 0 \end{bmatrix} \Rightarrow A^4 = \begin{bmatrix} 0 & 4 & 8 & 15 \\ 17 & 0 & 5 & 12 \\ 12 & 16 & 0 & 7 \\ 5 & 9 & 13 & 0 \end{bmatrix}$$

All pairs shortest paths Algorithm

```
Algorithm All paths (cost, A, n)
{
    for i = 1 to n do
     for j = 1 to n do
        A[i,j] = cost[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]}
}
```

Note:- Time Complexity of all pairs shortest path Algorithm is $O(n^3)$.

Optimal Binary Search Tree (OBST)

→ A Binary Search Tree is a binary tree, in which each node is an identifier

(i) All the identifiers in left sub tree are less than identifier at root node.

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

→ To find whether an identifier "x" is present or not

(i) "x" is compared with the root node

(ii) if "x" is less than identifier at root node, then search continues in left sub tree

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

→ Let us assume that given set of identifiers $\{a_1, a_2, \ldots, a_n\}$ where $a_1 < a_2 < a_3 < \ldots < a_n$ and let $p[i]$ be the probability of successful search of an identifier a_i, $1 \leq i \leq n$ and $q[i]$ be the probability of unsuccessful search of an identifier a_i, $0 \leq i \leq n$

→ To obtain a cost function for binary search tree, it is useful to add an external node in

→ If n identifiers are there, there will be n internal nodes and $n+1$ external nodes.

→ successful search terminates at an internal node.

→ unsuccessful search terminates at an external node

→ If a successful search terminates at an internal node at level "l". Hence, the expected cost of a internal node "a_i" is $\boxed{P(i) * level(a_i)}$

→ unsuccessful search terminates at an external node. i.e if the failure of search a node for E_i at level "l". the expected cost of a external node E_i is $\boxed{q(i) * (level(E_i) - 1)}$

→ the Expected cost of a binary search tree is

$$cost = \sum P(i) * level(a_i) + \sum q(i) * (level(E_i) - 1)$$

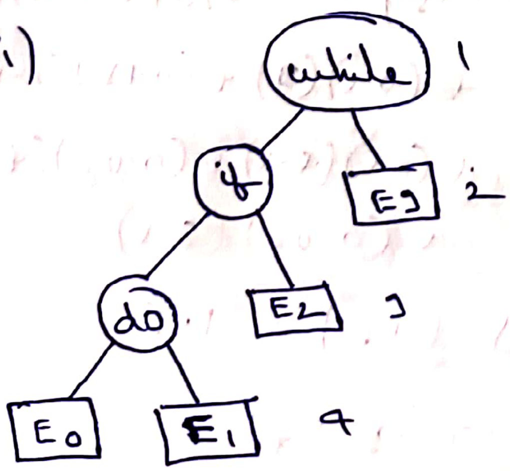
Ex:- Find OBST for a set of identifiers (a_1, a_2, a_3) = (do, if, while); $(P_1, P_2, P_3) = (0.5, 0.1, 0.05)$ and $(q_0, q_1, q_2, q_3) = (0.15, 0.1, 0.05, 0.05)$

solu) Given Identifiers = $n = 3$

⇒ Number of possible binary search trees are $\frac{{}^{2n}C_n}{n+1}$

$$= \frac{{}^6C_3}{3+1} = \frac{20}{4} = 5$$

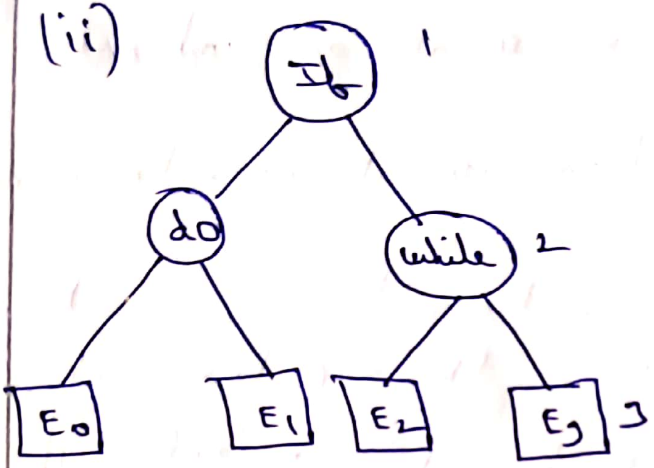
(i)



$$cost(T_1) = (0.05)(1) + (0.1)(2) + (0.5)(3) + (0.15)(4-1) + (0.1)(4-1) + (0.05)(3-1) + (0.05)(2-1)$$

$$cost(T_1) = 2.65$$

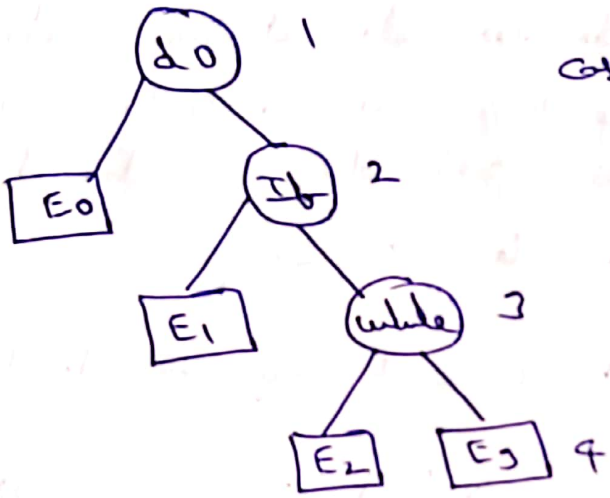
(ii)



$$Cost(T_2) = (0.1)(1) + (0.5)(2) + (0$$
$$(2) + (0.15)(3-1)$$
$$+ (0.1)(3-1) + (0.05)(3-1)$$
$$+ (0.05)(3-1)$$

$$Cost(T_2) = 1.9$$

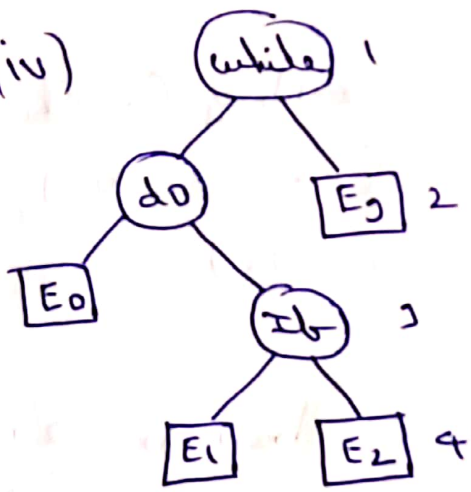
(iii)



$$Cost(T_3) : (0.5)(1) + (0.1)(2)$$
$$+ (0.05)(3) + (0.15)(2-1)$$
$$+ (0.1)(3-1) + (0.05)(4-1)$$
$$+ (0.05)(4-1)$$

$$Cost(T_3) : 1.5$$

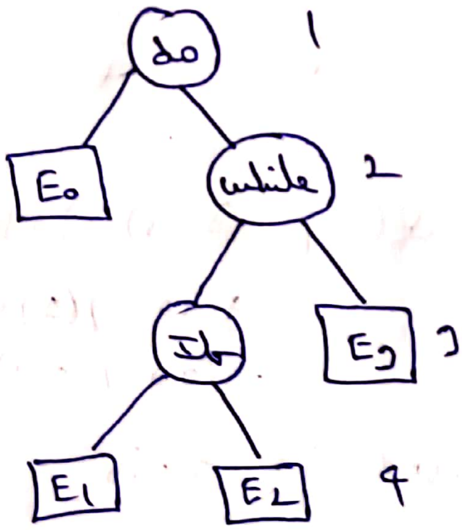
(iv)



$$Cost(T_4) = (0.05)(1) + (0.5)(2)$$
$$+ (0.1)(3) + (0.15)(3-1)$$
$$+ (0.1)(4-1) + (0.05)(4-1) +$$
$$(0.05)(2-1)$$

$$Cost(T_4) : 2.15$$

(v)



$$Cost(T_5) = (0.5)(1) + (0.05)(2)$$
$$+ (0.1)(3) + (0.15)(2-1)$$
$$+ (0.1)(4-1) + (0.05)(4-1)$$
$$+ (0.05)(3-1)$$

$$Cost(T_5) = 1.6$$

$\boxed{\text{OBST is } T_3}$

(2) Q:- Consider $n = 4$ and $(a_1, a_2, a_3, a_4) = (\text{do}, \text{if}, \text{int}, \text{while})$

$P(1:4) = (3,3,1,1)$ and $q(0:4) = (2,3,1,1,1)$

Construct optimal Binary search tree.

Solun) we have to construct $T_{0n} = T_{04}$

	0	1	2	3	4
0	$w(0,0) = 2$ $c(0,0) = 0$ $r(0,0) = 0$	$w(1,1) = 3$ $c(1,1) = 0$ $r(1,1) = 0$	$w(2,2) = 1$ $c(2,2) = 0$ $r(2,2) = 0$	$w(3,3) = 1$ $c(3,3) = 0$ $r(3,3) = 0$	$w(4,4) = 1$ $c(4,4) = 0$ $r(4,4) = 0$
1	$w_{01} = 8$ $c_{01} = 8$ $r_{01} = 1$	$w_{12} = 7$ $c_{12} = 7$ $r_{12} = 2$	$w_{23} = 3$ $c_{23} = 3$ $r_{23} = 3$	$w_{34} = 3$ $c_{34} = 3$ $r_{34} = 4$	
2	$w_{02} = 12$ $c_{02} = 19$ $r_{02} = 1$	$w_{13} = 9$ $c_{13} = 12$ $r_{13} = 2$	$w_{24} = 5$ $c_{24} = 8$ $r_{24} = 3 \text{ or } 4$		
3	$w_{03} = 14$ $c_{03} = 25$ $r_{03} = 2$	$w_{14} = 11$ $c_{14} = 19$ $r_{14} = 2$			
4	$w_{04} = 16$ $c_{04} = 32$ $r_{04} = 2$				

$|j-i|$

(i) $|j-i| = 0$

initially $w[i,i] = q[i]$

$c(i,i) = 0$

$r(i,i) = 0$

$w(0,0) = q(0) = 2$ | $w(1,1) = q(1) = 3$ | $w(2,2) = q($
$c(0,0) = 0$ | $c(1,1) = 0$ | $c(2,2) = 0$
$r(0,0) = 0$ | $r(1,1) = 0$ | $r(2,2) = 0$

$w(3,3) = q(3) = 1$ | $w(4,4) = q(4) = 1$
$c(3,3) = 0$ | $c(4,4) = 0$
$r(3,3) = 0$ | $r(4,4) = 0.$

(ii) $|j-i| = 1$

$$w[i,j] = w[i,j-1] + p[j] + q[j]$$

$$c[i,j] = \min_{i<k\leq j} \{c[i,k-1] + c[k,j]\} + w[i,j]$$

$$r[i,j] = k$$

Now $w[0,1] = w[0,1-1] + p[1] + q[1]$
$= w(0,0) + p(1) + q[1]$
$= 2 + 3 + 3$
$= 8$

$$c[0,1] = \min_{\substack{0<k\leq 1 \\ k=1}} \{c[0,1-1] + c[1,1]\} + w[0,1]$$
$$= \min_{k=1} \{c[0,0] + c[1,1]\} + w[0,1]$$
$$= \min_{k=1} \{0 + 0\} + 8$$
$$= 0 + 8$$
$$= 8$$

$r[0,1] = 1$

$$w[1,2] = w[1,1] + p[2] + q[2]$$
$$= 3 + 3 + 1$$
$$= 7$$

$$c[1,2] = \min_{1<k\le 2} \{ c[1,2-1] + c[2,2] \} + w[1,2]$$
$$k=2$$
$$= \min \{ c[1,1] + c[2,2] \} + w[1,2]$$
$$= \min \{ 0 + 0 \} + 7$$
$$= 0 + 7$$
$$= 7$$

$$r[1,2] = 2$$

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

$$c[2,3] = \min_{2<k\le 3} \{ c[2,3-1] + c[3,3] \} + w[2,3]$$
$$k=3$$
$$= \min \{ 0 + 0 \} + 3$$
$$= 3$$

$$r[2,3] = 3$$

$$w[3,4] = w[3,4-1] + p[4] + q[4]$$
$$= 1 + 1 + 1$$
$$= 3$$

$$c[3,4] = \min_{3<k\le 4} \{ c[3,4-1] + c[4,4] \} + w[3,4]$$
$$k=4$$
$$= \min \{ 0 + 0 \} + 3 = 0 + 3 = 3$$

$$r[3,4] = 4$$

(iii) $|j-i| = 2$

$$w[0,2] = w[0,2-1] + p[2] + q[2]$$
$$= w[0,1] + 3 + 1$$
$$= 8 + 3 + 1$$
$$= 12$$

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

$$= \min_{k=1,2} \{0+7,\ 8+0\} + 12$$

$$= \min_{k=1,2} \{7, 8\} + 12$$

$$= 7 + 12 = 19$$

$$r[0,2] = 1$$

$$w[1,3] = w[1,3-1] + p[3] + q[3]$$
$$= 7 + 1 + 1 = 9$$

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

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

$$= 3 + 9 = 12$$

$$r[1,3] = 2$$

$$w[2,4] = w[2,4-1] + p[4] + q[4] = 3 + 1 + 1 = 5$$

$$c[2,4] = \min_{\substack{2<k\le 4 \\ k=3,4}} \begin{Bmatrix} c[2,3-1] + c[3,4] \\ c[2,4-1] + c[4,4] \end{Bmatrix} + w[2,4]$$

$$= \min_{k=3,4} \{0+3,\ 3+0\} + 5$$

$$= 3+5 = 8$$

$r[2,4] = 3$ (or) 4.

(iv) $|j-i| = 3$

$$w[0,3] = w[0,3-1] + p[3] + q[3] = 12+1+1 = 14$$

$$c[0,3] = \min_{\substack{0<k\le 3\\ k=1,2,3}} \left\{ \begin{array}{l} c[0,1-1]+c[1,3] \\ c[0,2-1]+c[2,3] \\ c[0,3-1]+c[3,3] \end{array} \right\} + w[0,3]$$

$$= \min_{k=1,2,3} \{0+12,\ 8+3,\ 19+0\} + 14$$

$$= 11+14 = 25$$

$r[0,3] = 2$

$$w[1,4] = w[1,4-1] + p[4] + q[4] = 9+1+1 = 11$$

$$c[1,4] = \min_{\substack{1<k\le 4\\ k=2,3,4}} \left\{ \begin{array}{l} c[1,2-1]+c[2,4] \\ c[1,3-1]+c[3,4] \\ c[1,4-1]+c[4,4] \end{array} \right\} + w[1,4]$$

$$= \min_{k=2,3,4} \{0+8,\ 7+3,\ 12+0\} + 11$$

$$= \min_{k=2,3,4} \{8,\ 10,\ 12\} + 11$$

$$= 08 + 11$$

$$= 19$$

$r[1,4] = 2$

(V) $|j-i| = 4$

$$w[0,4] = w[0,4-1] + p[4] + q[4]$$

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

$$c[0,4] = \min_{\substack{0<k\leq 4 \\ k=1,2,3,4}} \begin{Bmatrix} c[0,1-1] + c[1,4] \\ c[0,2-1] + c[2,4] \\ c[0,3-1] + c[3,4] \\ c[0,4-1] + c[4,4] \end{Bmatrix} + w[0,4]$$

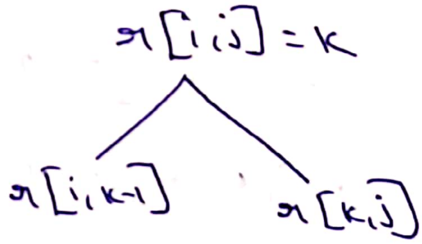
$$= \min_{k=1,2,3,4} \{0+19,\ 8+8,\ 19+3,\ 25+0\} + 16$$

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

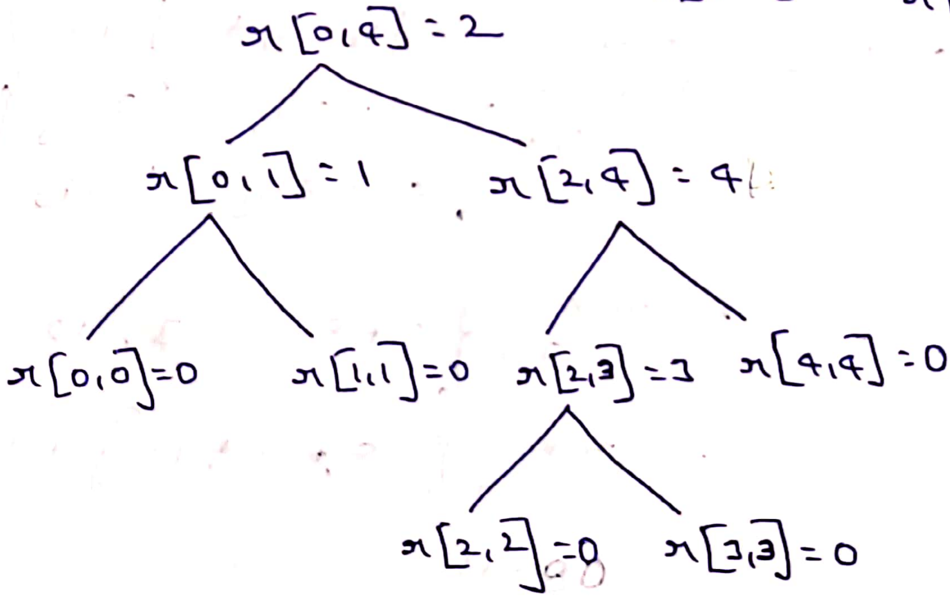
$$= 16 + 16 = 32$$

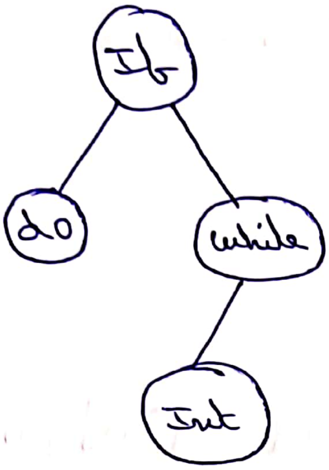
$r[0,4] = 2$

For construction of OBST we have

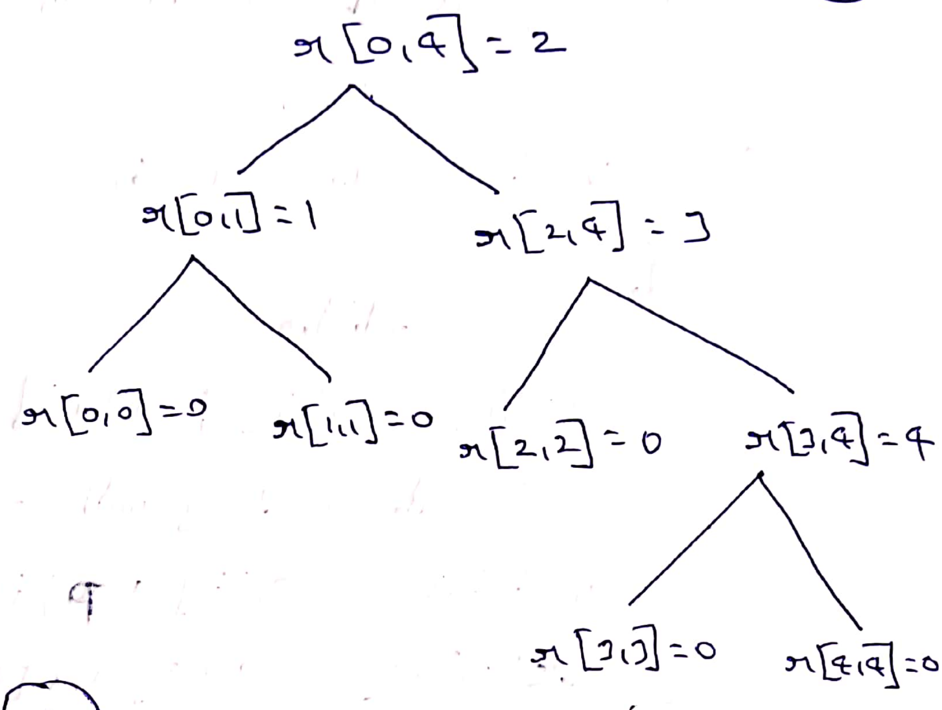


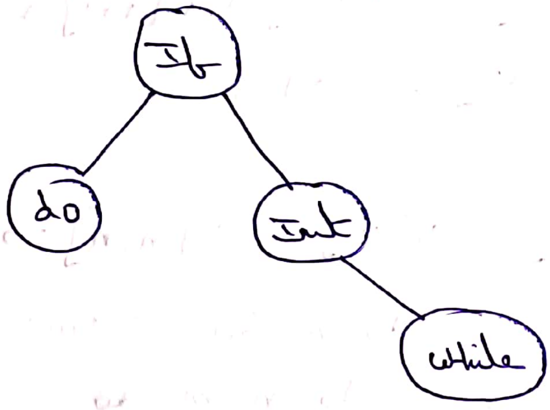
Now





OBST-1





OBST-2

Ex: Apply dynamic programming to obtain OBST for the identifier set $(a_1, a_2, a_3, a_4) = (\text{cin}, \text{for}, \text{int}, \text{while})$ with $(p_1, p_2, p_3, p_4) = (1, 4, 2, 1)$, $(q_0, q_1, q_2, q_3, q_4) = (4, 2, 4, 1, 1)$

Algorithm of OBST

Algorithm OBST (p, q, n)

// Given n distinct identifiers $a_1, a_2, \ldots, a_n$ where $a_1 < a_2 < \ldots < a_n$

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

// $c[i,j]$ is the cost of optimal binary search tree.

// $r[i,j]$ is the root of t_{ij}

//

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

{

for i = 0 to n-1 do

{

// Initialize

w[i,i] = q[i]; r[i,i] = 0; c[i,i] = 0;

// optimal trees with one node

w[i,i+1] = q[i] + p[i+1] + q[i+1];

c[i,i+1] = q[i] + p[i+1] + q[i+1];

r[i,i+1] = i+1;

}

w[n,n] = q[n]; c[n,n] = 0; r[n,n] = 0;

for m = 2 to n do // Find optimal tree 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]);

}

n = 4

i = 0 to 3

m = 2 to 4

i = 0 to 4-m

m = 2	m = 3	m = 4
i = 0 to 2	i = 0 to 1	i = 0
j = 2,3,4	j = 3,4	j = 4

Algorithm Find (c, r, i, j)

{

min = ∞

for m = r[i, j-1] to r[i+1, j] do

if (c[i, m-1] + c[m, j]) < min then

{

min = c[i, m-1] + c[m, j];

l = m;

}

return l;

}

0/1 Knapsack problem :-

→ 0/1 Knapsack problem represents "0" means not consider particular object total weight, "1" means consider total weight.

→ Maximum profit is $\sum_{i=1}^{n} p_i x_i$

where x_i is exact 0 (or) 1

→ Total weight is $\sum_{i=1}^{n} w_i x_i$

then maximized $\sum_{i=1}^{n} p_i x_i$ subject to $\sum_{i=1}^{n} w_i x_i \leq m$

Purging rule (or) Dominance rule :-

→ If S^{i+1} contains (p_j, w_j) and (p_k, w_k) is two pairs such that $p_j \leq p_k$ and $w_j \geq w_k$ then (p_j, w_j) can be eliminated.

→ In dominance rule remove the pair with less profit and more weight.

Ex:- Solve knapsack instance $m = 6$ and $n = 3$, $(P_1, P_2, P_3) = (1, 2, 5)$, $(W_1, W_2, W_3) = (2, 3, 4)$.

Solu) we have to build the sequence of decisions. S^0, S^1, S^2, S^3

$$S^{i+1} = S^i \cup S^i_1$$

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

$S^0_1 = \{$ select next (P_1, W_1) pair $\}$

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

$\therefore S^1 = S^0 \cup S^0_1 = \{(0,0)\} \cup \{(1,2)\}$

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

$S^1_1 = \{$ next select the (P_2, W_2) pair and add it with $S^1\}$

$S^1_1 = \{(0+2, 0+3), (1+2, 2+3)\}$

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

$\therefore S^2 = S^1_1 \cup S^1$

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

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

$S^2_1 = \{$ next select the (P_3, W_3) pair and add it with $S^2\}$

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

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

$$S^3 = \tilde{S} \cup \tilde{S}_1$$

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

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

Consider the pairs $(7,7)$ and $(8,9)$, the weights are 7 & 9 are greater than the capacity of the Knapsack $m=6$ so we can remove these two pairs from S^3

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

By applying the purging rule, the pair $(3,5)$ can be removed.

Purging rule: (P_j, W_j) and (P_K, W_K) then $P_j \le P_K$ and $W_j \ge W_K$ then eliminate (P_j, W_j) pair

Consider $(3,5)$ and $(5,4)$ pairs, we have $3 \le 5$ and $5 \ge 4$ are true

so we have to eliminate $(3,5)$ pair from S^3

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

As $m=6$, we will find the pair, that contains weight 6.

$$\Rightarrow (6,6) \in S^3$$

now $(6,6) \in S^{i-1} \Rightarrow (6,6) \in S^{3-1} \Rightarrow (6,6) \notin S^2$

$$\therefore \boxed{x_3 = 1}$$

Now $(6-p_3, 6-w_3) = (6-5, 6-4)$

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

Also $(1,2) \in S^{i-1} \Rightarrow (1,2) \in S^{2-1} \Rightarrow (1,2) \in S^1$

$\therefore \boxed{x_2 = 0}$

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

$\Rightarrow (1,2) \in S^{i-1} \Rightarrow (1,2) \in S^{1-1} \Rightarrow (1,2) \notin S^0$

$\therefore \boxed{x_1 = 1}$

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

Maximum profit $= \sum_{i=1}^{3} p_i x_i$

$= (1)(1) + (2)(0) + (5)(1)$

$= 1 + 0 + 5$

$= 6.$

Maximum weight $= \sum_{i=1}^{3} w_i x_i$

$= (2)(1) + (3)(0) + 4(1)$

$= 6.$

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

Ex:- Solve knapsack instance $m = 8$ and $n = 4$,

$(p_1, p_2, p_3, p_4) = (1, 2, 5, 6)$; $(w_1, w_2, w_3, w_4) = (2, 3, 4, 5)$

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

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

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

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

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

Reliability Design :-

→ The problem is to design a system which is composed of several devices connected in series

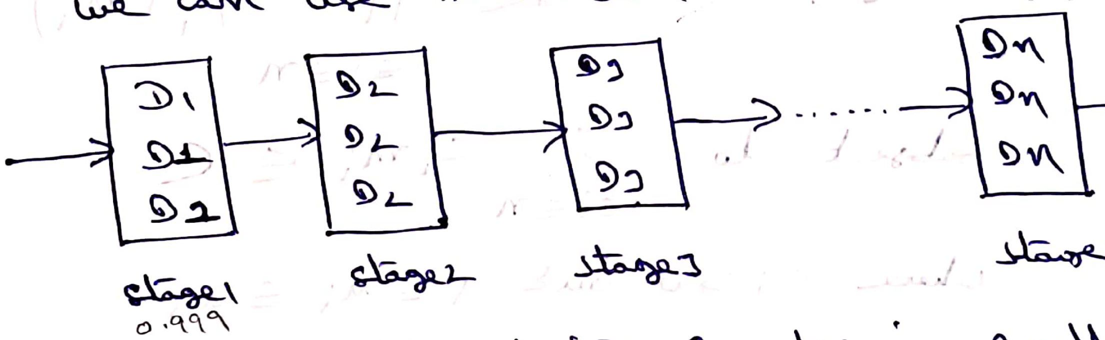
4 of n=4

$(0.9)^4 = 0.6561$

fig: n devices connected in series

from the above fig, the cost and reliability is less because one of the device among n devices failed, the total work is stopped.

→ To increase the reliability of the entire system, we can use more than one device of $D_1, D_2, \ldots, D_n$.



$r_1 = 0.9$

$1-r_1 = 0.1$

$(1-r_1)^3 = 0.001$

$1-(1-r_i)^3 = 1-0.001$
$= 0.999$

fig: Multiple devices connected in parallel in each stage

→ Let r_i be the reliability of the device D_i, then the reliability of the entire system is $\prod r_i$

→ Let m_i be the number of devices of D_i are connected in parallel with reliability r_i

→ $(1-r_i)$ is the probability that one copy of the device will malfunction

→ $(1-r_i)^{m_i}$ is the probability of all the devices malfunction at the same time

→ Hence the reliability of device D_i can be

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

→ The main objective of reliability design is to maximize the reliability using device duplication. The maximization is to be carried out under a cost constraint.

Let C be the cost of the entire system

c_i be the cost of the i^{th} device

m_i be the number of copies of device D_i

objective is : $\text{Maximize} \prod_{1\le i\le n} \phi_i(m_i)$

$$\text{subject to} \sum_{1\le i\le n} c_i m_i \le C$$

where $c_i > 0$ and $1 \le m_i \le u_i$.

where u_i – How many number of copies of each device can be consider.

$$\text{and} \quad u_i = \left\lfloor \frac{C + c_i - \sum_{j=1}^{n} c_j}{c_i} \right\rfloor$$

Ex:- We are to design a three stage system with device types D1, D2, D3. The costs are 30, 15 and 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.

Soln Given data- $C = 105$

$c_1 = 30$; $c_2 = 15$; $c_3 = 20$

$r_1 = 0.9$; $r_2 = 0.8$; $r_3 = 0.5$

Now $u_i = \left\lfloor \dfrac{c + c_i - \sum_{j=1}^{n} c_j}{c_i} \right\rfloor$

where $\sum_{j=1}^{n} c_j = \sum_{j=1}^{3} c_j = c_1 + c_2 + c_3 = 65$

then $u_1 = \left\lfloor \dfrac{105 + 30 - 65}{30} \right\rfloor = \left\lfloor \dfrac{70}{30} \right\rfloor = 2$

$u_2 = \left\lfloor \dfrac{105 + 15 - 65}{15} \right\rfloor = \left\lfloor \dfrac{55}{15} \right\rfloor = 3$

$u_3 = \left\lfloor \dfrac{105 + 20 - 65}{20} \right\rfloor = \left\lfloor \dfrac{60}{20} \right\rfloor = 3$

Hence D_1 —— 2 Copies

D_2 —— 3 Copies

D_3 —— 3 Copies

$i \rightarrow$ device no.

$S^i_j \rightarrow$ no. of duplicates

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

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

Now S^1 can be obtained from S^1_1 and S^1_2

Here $1 \leq m_i \leq u_i \Rightarrow 1 \leq m_1 \leq u_1$

$\Rightarrow 1 \leq m_1 \leq 2$

$\therefore m_1 = 1$ and 2

S^1_1 calculated as $i = 1, j = 1, m_1 = 1$

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

$= 1 - (0.1)^1$

$= 1 - 0.1$

$= 0.9$

$s_1^1 = \{(1*0.9, 0+30)\}$

$\therefore s_1^1 = \{(0.9, 30)\}$

s_2^1 calculated as $i=1; j=2, m_1=2$

$\phi_1(m_1) = 1-(1-r_1)^{m_1} = 1-(1-0.9)^2$

$= 1-(0.1)^2$

$= 1-0.01$

$= 0.99$

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

$\therefore s_2^1 = \{(0.99, 60)\}$

$\Rightarrow s^1 = s_1^1 \cup s_2^1$

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

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

Now s^2 can be obtained from s_1^2, s_2^2, s_3^2.

$1 \le m_i \le u_i \Rightarrow 1 \le m_2 \le u_2$

$\Rightarrow 1 \le m_2 \le 3$

$\Rightarrow m_2 = 1, 2, 3.$

s_1^2 calculated as $i=2, j=1, m_2=1$

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

$= 1-0.2$

$= 0.8$

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

$s_1^2 = \{(0.72, 45), (0.792, 75)\}$

S_2^2 calculated as $i=2;\ j=2;\ m_2=2$

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

$$\therefore S_2^2 = \{(0.9 * 0.96,\ 30+30),\ (0.99 * 0.96,\ 60+30)\}$$
$$= \{(0.864,\ 60),\ \underbrace{(0.9504,\ 90)}_{\times}\}$$ we can't purchase D3

S_3^2 calculated as $i=2;\ j=3,\ m_2=3$

$$\phi_2(m_2) = 1-(1-r_2)^{m_2}$$
$$= 1-(1-0.8)^3$$
$$= 1-(0.2)^3$$
$$= 1-0.08$$
$$= 0.992$$

$$\therefore S_3^2 = \{(0.9 * 0.992,\ 30+45),\ (0.99 * 0.992,\ 60+45)\}$$
$$= \{(0.8928,\ 75),\ \underbrace{(0.98208,\ 105)}_{\times}\}$$ we can't purchase D3

Now
$$S^2 = S_1^2 \cup S_2^2 \cup S_3^2$$
$$= \{(0.72,\ 45),\ (0.792,\ 75),\ (0.864,\ 60),\ \underbrace{(0.9504,\ 90)}_{\times},\ (0.8928,\ 75),\ \underbrace{(0.98208,\ 105)}_{\times}\}$$

Applying the dominance rule for the pair

$$(0.792,\ 75),\ (0.864,\ 60).$$

Here $0.792 \le 0.864$ and $75 \ge 60$ are true so we have to eliminate $(0.792,\ 75)$ from S^2

$$\therefore S^2 = \{(0.72,45),\ (0.864,60),\ \overset{\times}{\overline{(0.9504,90)}},\ (0.8928,75),\ \overset{\times}{\overline{(0.98208,105)}}\}$$

remove (0.9504, 90) because costs are always in increasing order

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

Now S^3 can be obtained from S_1^3, S_2^3 and S_3^3.

$1 \le m_i \le u_i \Rightarrow 1 \le m_3 \le u_3$

$\Rightarrow 1 \le m_3 \le 3$

$\Rightarrow m_3 = 1, 2, 3.$

S_1^3 calculated as $i = 3, j = 1, m_3 = 1$

$\phi_3(m_3) = 1-(1-x_3)^{m_3} = 1-(1-0.5)^1 = 1-(0.5)^1$

$= 1-0.5$

$= 0.5$

$S_1^3 = \{(0.72 * 0.5, 45+20), (0.864 * 0.5, 60+20),$

$(0.8928 * 0.5, 75+20), \underset{X}{\underline{(0.98208 * 0.5, 105+20)}}\}$

remove $(0.98208 * 0.5, 125)$ pair, it exceeds $C = 105$)

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

S_2^3 calculated as $i = 3, j = 2, m_3 = 2$

$\phi_3(m_3) = 1-(1-x_3)^{m_3} = 1-(1-0.5)^2 = 1-(0.5)^2$

$= 1-0.25$

$= 0.75$

$S_2^3 = \{(0.72 * 0.75, 45+40), (0.864 * 0.75, 60+40),$

$(0.8928 * 0.75, 75+40), \underset{X}{\underline{(0.9820 * 0.75, 105+40)}}\}$

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

we removed the pair $(0.8928 * 0.75, 115), (0.982 * 0.75, 145)$

because $C = 105.$

S_3^3 calculated as $i = 3, j = 3, m_3 = 3$

18

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

$= 1-(0.5)^3 = 1-0.125$

$= 0.875$

$S_3^3 = \{(0.875 * 0.72, 45+60), (0.864 * 0.875, 60+60),$
$(0.8928 * 0.875, 75+60), \underset{\times}{\underline{(0.98208 * 0.875, 105+60)}}\}$

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

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

$S^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95),$
$(0.54, 85), (0.648, 100), (0.63, 105)\}$

Apply dominance for the pairs (0.4464, 95), (0.54, 85).

Here $0.4464 \le 0.54$ and $95 \ge 85$ are true

so we have to eliminate (0.4464, 95) from S^3

$\therefore S^3 = \{(0.36, 65), (0.432, 80), (0.54, 85),$
$\underline{(0.648, 100), (0.63, 105)}$

now remove the pair (0.63, 105).

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

$\Rightarrow$ The Best design is 0.648 is the reliability with cost 100.

$(0.648, 100) \in S_2^3$ Here $i=3$; $j=2$; $m_3=2$

(0.648, 100) can be obtained from (0.864, 60) which is present in S_2^2 Here $i=2$; $j=2$; $m_2=2$

$(0.864, 60)$ can be obtained from $(0.9, 30) \in S_1^1$

Here $i=1$; $j=1$; $m_1=1$

Hence $m_1 = 1$; $m_2 = 2$; $m_3 = 2$

we require 1 copy of device D_1

2 copies of device D_2

2 copies of device D_3

Ex:- 2 Given D_1, D_2, D_3, devices

and $C = 110$; $C_1 = 40$; $C_2 = 30$; $C_3 = 20$

$r_1 = 0.9$; $r_2 = 0.7$; $r_3 = 0.5$

Soln $(r, x) = (0.4725, 110)$. and $D_1 = 1$; $D_2 = 1$; $D_3 = 2$

Matrix chain multiplication

→ Matrix chain multiplication problem is stated as follows.

Given a chain $\{A_1, A_2, \dots, A_n\}$ of n matrices where $i = 1, 2, \dots, n$. Matrix A_i has dimension $P_{i-1} * P_i$. To compute the matrix product $A_1 \cdot A_2 \cdots A_n$ in such a way that it minimizes the number of scalar multiplications. i.e our goal is only to determine an order of multiplying matrices that has the lowest cost.

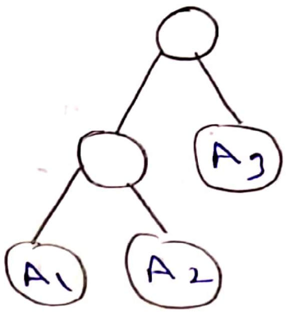
→ Let $A = [a_{ij}]_{p \times q}$; $B = [b_{ij}]_{q \times r}$ then

$$AB = [C]_{p \times r}$$

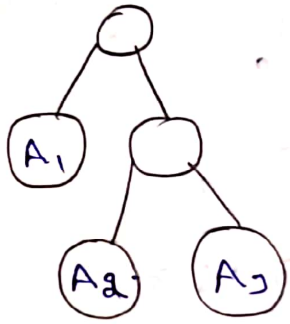
Ex:- Consider $A_1 = [a_{ij}]_{5 \times 4}$; $A_2 = [a_{ij}]_{4 \times 6}$

$$A_3 = [a_{ij}]_{6 \times 2}$$

1. $(A_1 \cdot A_2) \cdot A_3 = (5 \times 4 \times 6) + (5 \times 6 \times 2) = 120 + 60 = 180$
2. $A_1 \cdot (A_2 \cdot A_3) = (5 \times 4 \times 2) + (4 \times 6 \times 2) = 40 + 48 = 88$



Tree-1



Tree-2

$\therefore$ the optimal cost $= 88$ and
optimal order sequence $= A_1 \cdot (A_2 \cdot A_3)$

Ex:- Consider the following matrices $A_1 = 5 \times 4$; $A_2 = 4 \times 6$; $A_3 = 6 \times 2$, $A_4 = 2 \times 7$. Find out the optimal order sequence and optimal cost for multiplying these matrices using dynamic programming?

Soln) Given matrices dimensions are
$A_1 = 5 \times 4$; $A_2 = 4 \times 6$; $A_3 = 6 \times 2$; $A_4 = 2 \times 7$
Matrix A_i has dimension $P_{i-1} * P_i$

$A_1 = 5 \times 4 = P_{1-1} * P_1 = P_0 * P_1$
$A_2 = 4 \times 6 = P_{2-1} * P_2 = P_1 * P_2$
$A_3 = 6 \times 2 = P_{3-1} * P_3 = P_2 * P_3$
$A_4 = 2 \times 7 = P_{4-1} * P_4 = P_3 * P_4$
$\therefore P_0 = 5$; $P_1 = 4$; $P_2 = 6$; $P_3 = 2$; $P_4 = 7$

The recursive definition for the minimum cost of parenthesizing the product is

$$m[i,j] = \begin{cases} 0 & \text{if } i=j \\ \min\limits_{i \le k < j} \{ m[i,k] + m[k+1,j] + p_{i-1} p_k p_j \} & \text{if } i<j \end{cases}$$

$\longrightarrow j$, $i \downarrow$

$i \backslash j$	1	2	3	4
1	0	120	88	158
2		0	48	104
3			0	84
4				0

$m[i,j]$

$\longrightarrow j$, $i \downarrow$

$i \backslash j$	1	2	3	4
1		1	1	3
2			2	3
3				3
4				

$s[i,j]$

Now $m[1,1] = m[2,2] = m[3,3] = m[4,4] = 0$

$$m[1,2] = \min_{\substack{1 \le k < 2 \\ k=1}} \{ m[1,1] + m[2,2] + p_{1-1} p_1 p_2 \}$$

$$= \min_{k=1} \{ m[1,1] + m[2,2] + p_0 p_1 p_2 \}$$

$$= \min_{k=1} \{ 0 + 0 + 5 \cdot 4 \cdot 6 \}$$

$$= 120$$

$s[1,2] = 1$

$$m[2,3] = \min_{\substack{2 \le k < 3 \\ k=2}} \{ m[2,2] + m[3,3] + p_{2-1} p_2 p_3 \}$$

$$= \min_{k=2} \{ m[2,2] + m[3,3] + p_1 p_2 p_3 \}$$

$$= \min_{k=2} \{ 0 + 0 + 4 \cdot 6 \cdot 2 \}$$

$$= 48$$

$s[2,3] = 2$

(20)

$$m[3,4] = \min_{\substack{3 \le k < 4 \\ k=3}} \{ m[3,3] + m[4,4] + P_{3-1} P_3 P_4 \}$$

$$= \min_{k=3} \{ m[3,3] + m[4,4] + P_2 P_3 P_4 \}$$

$$= \min_{k=3} \{ 0 + 0 + 6 \cdot 2 \cdot 7 \}$$

$$= 84$$

$$s[3,4] = 3$$

$$m[1,3] = \min_{\substack{1 \le k < 3 \\ k=1,2}} \begin{cases} m[1,1] + m[2,3] + P_{1-1} P_1 P_3 \\ m[1,2] + m[3,3] + P_{1-1} P_2 P_3 \end{cases}$$

$$= \min_{k=1,2} \{ 0 + 48 + 5 \cdot 4 \cdot 2 ,\ 120 + 0 + 5 \cdot 6 \cdot 2 \}$$

$$= \min_{k=1,2} \{ 88 ,\ 180 \}$$

$$= 88$$

$$s[1,3] = 1$$

$$m[2,4] = \min_{\substack{2 \le k < 4 \\ k=2,3}} \begin{cases} m[2,2] + m[3,4] + P_{2-1} P_2 P_4 \\ m[2,3] + m[4,4] + P_{2-1} P_3 P_4 \end{cases}$$

$$= \min_{k=2,3} \begin{cases} 0 + 84 + 4 \cdot 6 \cdot 7 \\ 48 + 0 + 4 \cdot 2 \cdot 7 \end{cases}$$

$$= \min_{k=2,3} \{ 252 ,\ 104 \}$$

$$= 104$$

$$s[2,4] = 3$$

$$m[1,4] = \min_{\substack{1 \le k < 4 \\ k=1,2,3}} \begin{cases} m[1,1] + m[2,4] + P_{1-1} P_1 P_4 \\ m[1,2] + m[3,4] + P_{1-1} P_2 P_4 \\ m[1,3] + m[4,4] + P_{1-1} P_3 P_4 \end{cases}$$

$$= \min_{k=1,2,3} \begin{cases} 0 + 104 + 5 \cdot 4 \cdot 7 \\ 120 + 84 + 5 \cdot 6 \cdot 7 \\ 88 + 0 + 5 \cdot 2 \cdot 7 \end{cases}$$

$$= \min_{k=1,2,3} \{ 244, 414, 158 \}$$

$$= 158$$

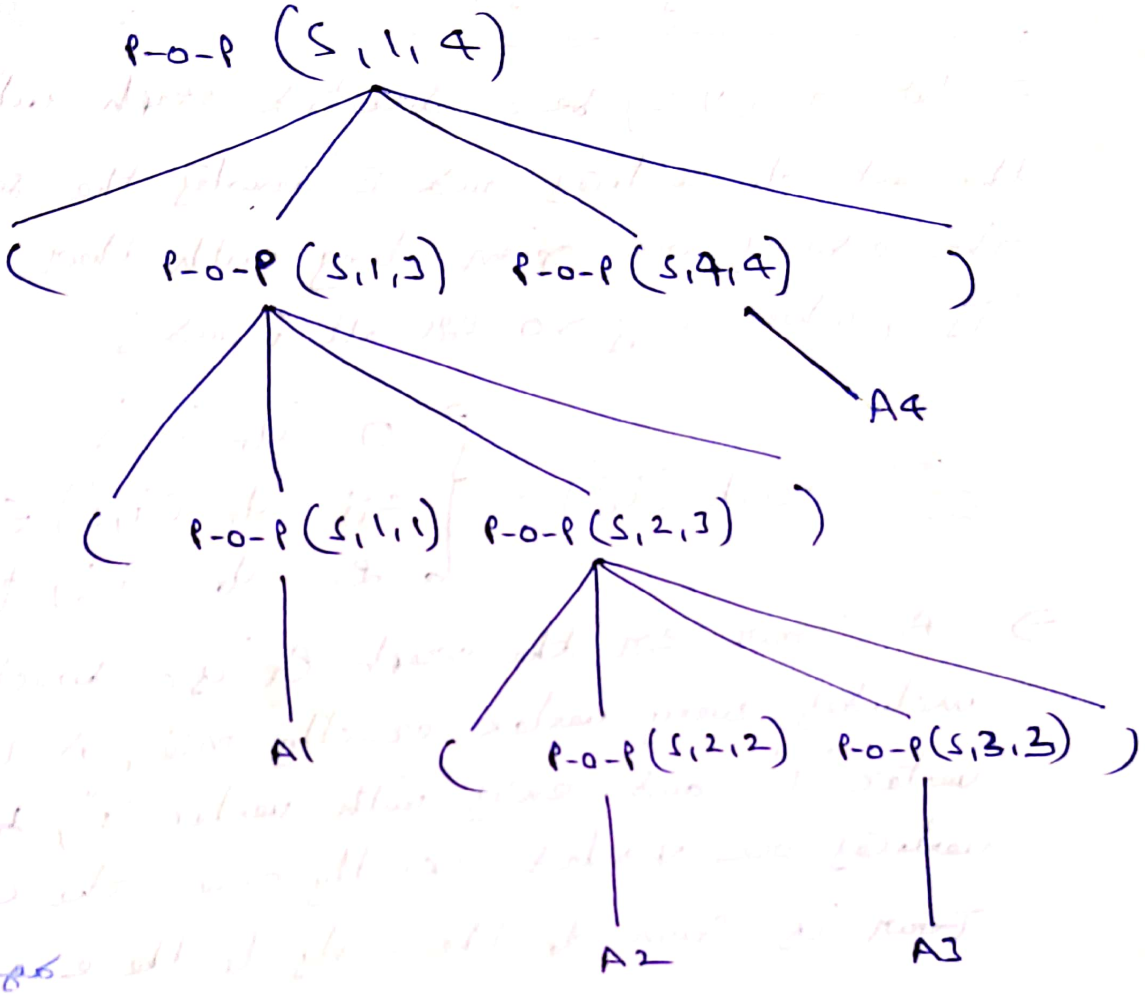
$s[1,4] = 3$

$\therefore$ the optimal cost $= m[1,4] = 158$.

print - optimal - parenthesis (s, i, j)

```
{
    if i = j then print " A_i "
    else
    print " ( "
    P-O-P (s, i, s[i,j]);
    P-O-P (s, s[i,j]+1, j);
    print " ) "
}
```

for the given problem, the initial call to print - optimal - parenthesis algorithm is $(s, 1, n)$ i.e $(s, 1, 4)$



∴ optimal order sequence = $((A_1(A_2 \cdot A_3))A_4)$

Ex:- Consider the following matrices $A_1 = 2 \times 1$; $A_2 = 1 \times 3$; $A_3 = 3 \times 4$; $A_4 = 4 \times 5$. find out the optimal order sequence and optimal cost for multiplying these matrices using dynamic programming?

Sol: the optimal cost = $m[1,4] = 42$

optimal order sequence = $A(B(CD))$ $(A_1((A_2 \cdot A_3)A_4))$

The Traveling sales person problem

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

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

→ A Tour for the graph G is a directed cycle that includes every vertex exactly once, A Tour starts at vertex "i" and ends with vertex "i", but remaining vertices are visited exactly once. The cost of the Tour is sum of the costs of the edges on the tour.

→ Let $g(i, S)$ be the length of the minimum shortest path, starting at vertex "i", going through all the vertices in "S" exactly once, and Terminating at vertex "i".

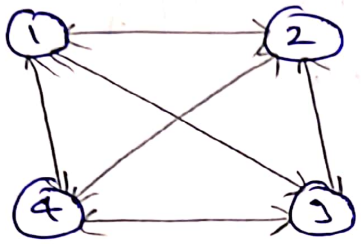
$$g(i, S) = \min_{j \in S} \left\{ c_{ij} + g\left(j, S - \{j\}\right) \right\}$$

where $S = V - \{1\}$

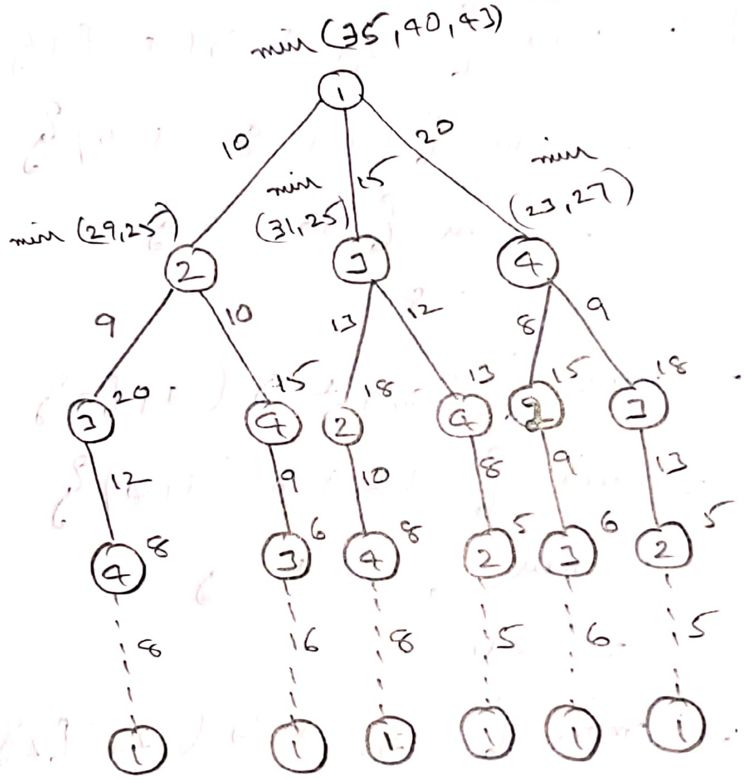
→ our main objective is to find the tour of optimal or Minimum cost.

$\boxed{\text{Ex:- postal van}}$

Consider the directed graph do the following. The edge lengths are given by matrix



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



shortest path = 1 — 2 — 4 — 3 — 1 = 10 + 10 + 9 + 6 = 35

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

now $|S| = \phi$

$g(i, \phi) = c_{i1}$

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

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

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

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

$|S| = 1$

$$g(2, \{3\}) = \min\{c_{2,3} + g(3, \{3\} - 3)\}$$
$$= \min\{9 + g(3, \phi)\}$$
$$= \min\{9 + 6\}$$
$$= \min\{15\}$$
$$= 15$$

$$g(2, \{4\}) = \min\{c_{2,4} + g(4, \{4\} - 4)\}$$
$$= \min\{10 + g(4, \phi)\}$$
$$= \min\{10 + 8\}$$
$$= 18$$

$$g(3, \{4\}) = \min\{c_{3,4} + g(4, \{4\} - 4)\}$$
$$= \min\{12 + g(4, \phi)\}$$
$$= \min\{12 + 8\}$$
$$= 20$$

$$g(3, \{2\}) = \min\{c_{3,2} + g(2, \{2\} - 2)\}$$
$$= \min\{13 + g(2, \phi)\}$$
$$= \min\{13 + 5\}$$
$$= 18$$

$$g(4, \{2\}) = \min\{c_{4,2} + g(2, \{2\} - 2)\}$$
$$= \min\{8 + g(2, \phi)\}$$
$$= \min\{8 + 5\}$$
$$= 13$$

$$g(4, \{3\}) = \min\{c_{4,3} + g(3, \{3\} - 3)\}$$
$$= \min\{9 + \underline{g(3, \phi)}\}$$

Scanned with OKEN Scanner

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

$= 15$

$|S| = 2$

$$g(2, \{3,4\}) = \min \begin{Bmatrix} c_{23} + g(3, \{3,4\} - 3) \\ c_{24} + g(4, \{3,4\} - 4) \end{Bmatrix}$$

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

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

$$= 25$$

$$g(3, \{2,4\}) = \min \begin{Bmatrix} c_{32} + g(2, \{2,4\} - 2) \\ c_{34} + g(4, \{2,4\} - 4) \end{Bmatrix}$$

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

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

$$= 25$$

$$g(4, \{2,3\}) = \min \begin{Bmatrix} c_{42} + g(2, \{2,3\} - 2) \\ c_{43} + g(3, \{2,3\} - 3) \end{Bmatrix}$$

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

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

$$= 23$$

$|S| = 3$

$$g(1, \{2,3,4\}) = \min \begin{Bmatrix} c_{12} + g(2, \{2,3,4\} - 2) \\ c_{13} + g(3, \{2,3,4\} - 3) \\ c_{14} + g(4, \{2,3,4\} - 4) \end{Bmatrix}$$

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

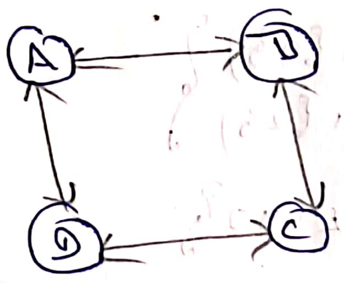
$$= \min \{10+25,\ 15+25,\ 20+23\}$$

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

$$= 35$$

Hence $1-2-4-3-1 = 10+10+9+6 = 35$

Ex:-2 Consider the directed graph & the following. the edge lengths are given by the matrix.



$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 16 & 11 & 6 \\ B & 8 & 0 & 13 & 16 \\ C & 4 & 2 & 0 & 9 \\ D & 5 & 12 & 2 & 0 \end{array}$$

Ans:- 23