Chapter-1 :- Greedy Method.

1) General Method.

2) Applications

(i) Job sequence with deadlines

(ii) 0/1 Knapsack problem.

(iii) Minimum cost spanning trees
[krushkal's, prims]

(iv) Single source shortest path problem.

chapter-2 :- Dynamic programming

1) General Method.

2) Applications

(i) Multistage graphs.

(ii) optimal binary search trees (OBST)

(iii) 0/1 knapsack problem.

(iv) All pairs shortest problem.

(v) Travelling sales person problem (TSP)

(vi) Reliability design.

Greedy Method :-

1) General Method :-

~~The~~ divide and conquer technique is applicable for only problems which can be divided. There exists some problems, which cannot be divisible, that type of problems we can solve by using greedy method.

Ex :- knapsack problem.

This is the draw back of divide & conquer technique which is rectified in greedy method.

2) Divide and conquer verses greedy method :-

⇒ In divide and conquer approach, a problem is divided recursively into sub problems until they are small enough to be solved and finally solutions of sub problems due combined to the original problem.

→ In greedy approach a problem is solved by determining a subset to satisfy the some constraints. Here, constraints means rules and regulations. If subset satisfies the given constraints is called as "feesable solution."

⇒ A feesable solution either maximized / minimized of an objective function is called optimal solution, or Best solution.

→ Difference between divide & conquer and Greedy method :-

Divide and conquer	Greedy Method
1) This approach is result oreinted approach.	1) In greedy method, there are some chances of getting an optimal solution to a specific problem.
2) This approach does not depend on constraints to solve a specific problem.	2) This approach depends on contraints to solve specific problem.
3) Time taken by this algorithm is efficient when compared to greedy method.	3) Time taken by this algorithm is not efficient when compared to divide and conquer method.
4) This approach is not efficient for large problems.	4) This approach is applicable and efficient for variety of problems as well.
5) Space requirement is very much high	5) Space requirement is less when compared to D &C method
6) This method is not applicable to problems which are not divisible.	6) This problem is rectified in greedy method.
Eg:- Knapsack problem.	Eg:- Knapsack problem.

→ Control Abstraction for greedy method :-

Algorithm Greedy (A,n)
{
Solution : =c
for i to n do
{

```
    r = select (A)
    if feasible (solution x) then
    {
      solution = union (solution, x);
    }
    else
      reject ( )
  }
  return solution;
}
```

2m

(i) Job Sequence with dead lines :-

1) Solve the job sequencing problem, given $N=5$, profits (1, 5, 20, 15, 10) on deadlines (1, 2, 3, 4, 1) by using greedy strategy.

Sol :- Given $N = 5$

$$\text{profits } (P_1, P_2, P_3, P_4, P_5) = (1, 5, 20, 15, 10)$$

$$\text{Dead Lines } (D_1, D_2, D_3, D_4, D_5) = (1, 2, 4, 1, 3)$$

In the above problem maximum dead line is 4 units.

∴ The feasible solution set must have less than or equal to 4 elements.
(Jobs)

Now, arrange the job | profits in decreasing order, we get

$$(P_3, P_4, P_5, P_2, P_1) = (20, 15, 10, 5, 1)$$

$$(D_3, D_4, D_5, D_2, D_1) = (4, 1, 3, 2, 1)$$

SNO	Feasible solution	processing sequence	profits.
1.	{3}	3	20
2.	{3,4}	3,4 / 4,3	15+20 = 35
3.	{3,4,5}	3,4,5 / 4,3,5.	15+20+10 = 45
4.	{3,4,5,2}	3,4,5,2 / 4,3,5,2	15+20+10+5 = 50
5.	{3,5,2,1}.	3,5,2,1 / 3,5,1,2	20+10+5+1 = 36

⇒ Solution 4 is optimal solution, the job must be processed in a the order 4,2,3,5 (or) 4,2,5,3.

∴ The value of optimal solution is 50.

Here we can take only 4 elements because the max deadline is 4 units.

2) Solve the job sequency deadline problem N = 4 and

$(P_1, P_2, P_3, P_4) = (100, 27, 15, 10)$

$(D_1, D_2, D_3, D_4) = (2, 1, 2, 1)$ using greedy strategy.

Sol:- Given N = 4

$(P_1, P_2, P_3, P_4) = (100, 27, 15, 10)$

$(D_1, D_2, D_3, D_4) = (2, 1, 2, 1)$

Now, arrange the jobs or profits in decreasing order, we get

$(P_1, P_2, P_3, P_4) = (100, 27, 15, 10)$

$(D_1, D_2, D_3, D_4) = (2, 1, 2, 1)$

*(Now, arrange the jobs or profits in decreasing order, we get

$(P_1, P_2, P_3, P_4) = (100, 27, 15, 10)$

$(D_1, D_2, D_3, D_4) = (2, 1, 2, 1)$ *

S.No	feasible solution	processing sequence	profits
1.	{1}	1	100
2.	{1,2}	1,2 / 2,1	100+27=127
3.	{1,3}	1,3 / 3,1	100+15=115
4.	{1,4}	1,4 / 4,1	100+10=110
5.	{2,3}	2,3 / 3,2	27+15=42

6.	{2,4}	2,4/4,2	27+10=37
7.	{3,4}	3,4/4,2	15+10=25

∴ Solution 2 is optimal solution, the job must be processed in the order 1,2 (or) 2,1.

∴ The value of optimal solution is 127

Here we can take only 2 elements because the max deadline is 2 units.

18/01/17.

5m
⇒ 0/1 Knapsack (→ empty bag) problem:-
Imp.

In olden days, their was a store which contains different types of gold coins, cost, profits & weights. Let the gold coins represents $n_1, n_2, n_3 ---- n_n$ and cost represent $C_1, C_2, C_3 ----, C_n$ and weight represent $w_1, w_2, w_3 --- w_n$ respectively. Now a thief wants to rob the store for that he brought a empty bag of size 'M'. Now his problem was, in what way he has to place gold coins in bag such that he should get maximum profit? This problem is called as knapsack problem. Here knapsack means empty bag. Knapsack is an empty bag. consider whose size is 'M', it contains 'n' items with weights $w_1, w_2, ---, w_n$ respectively and the profits of each weight is $P_1, P_2, P_3, ---, P_n$ and $x_1, x_2, x_3, ----, x_n$ be a fraction of items therefore the problem is to maximize $\boxed{\sum_{1 \le i \le n} P_i x_i}$

Scanned by CamScanner

wimp

5m

⇒ For given instance of knapsack problem $N=7$, $M=15$ (bag size),

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (10, 5, 15, 7, 6, 18, 3)$.

$(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (2, 3, 5, 7, 1, 4, 1)$ Find the optimal solution.

(i) Max. profit (iii) max. profit per unique weight.

(ii) min. weight.

Sol:- Given that $N=7$, $M=15$

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (10, 5, 15, 7, 6, 18, 3)$

$(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (2, 3, 5, 7, 1, 4, 1)$

(i) Maximum profit :-

Now bag size is 15. $x_6 = 1$, complete item weight '4' (w_4) kept in bag and 11 unit place is left i.e $15-4=11$. $x_3=1$, the complete item weight '5' (w_5) kept in bag and 6 units place is left i.e $11-5=6$. $x_1=1$, complete item w_2 (weight '2') kept in bag and 4 units place is left i.e $6-2=4$. $x_4 = 4/7$ means that the w_7 (weight '7') is not inserted into the bag of size '4'. $x_2=0$, $x_5=0$, $x_7=0$ because their is no bag size so, we can't place the items.

∴ The maximum profit $= \sum_{1 \le i \le n} p_i x_i$

$(N=7) \quad = P_1 x_1 + P_2 x_2 + P_3 x_3 + P_4 x_4 + P_5 x_5 + P_6 x_6 + P_7 x_7$

$= (10 \times 1) + (5 \times 0) + (15 \times 1) + (7 \times 4/7) + (6 \times 0) + (18 \times 1) + (3 \times 0)$

$= 10 + 15 + 4 + 18$

$= \underline{47}$ (OR)

$15-4=11$, $x_6=1$
$11-5=6$, $x_3=1$
$6-2=4$, $x_1=1$.

$\therefore$ Not possible , $x_4=4/7$

$x_2=0$

$x_5=0$

$x_7=0$.

(ii) minimum weight :-

$N=7$

$M=15$

$(P_1,P_2,P_3,P_4,P_5,P_6,P_7) = (10, 5, 15, 7, 6, 18, 3)$

$(w_1,w_2,w_3,w_4,w_5,w_6,w_7) = (2,3,5,7,1,4,1)$

$x_7=1$ i.e $15-1=14$

$x_5=1$ i.e $14-1=13$

$x_1=1$ i.e $13-2=11$

$x_2=1$ i.e $11-3=8$

$x_6=1$ i.e $8-4=4$

$x_3=4/5$.

$x_4=0$.

$\therefore$ The minimum profit $= \sum_{1\le i\le n} P_i x_i$

$= P_1x_1+P_2x_2+P_3x_3+P_4x_4+P_5x_5+P_6x_6+P_7x_7$

$= (10\times1)+(5\times1)+(\overset{3}{\cancel{15}}\times\frac{4}{\cancel{5}})+(7\times0)+(6\times1)+(18\times1)+(3\times1)$

$= 10+5+12+6+18+3$

$= 64$

(iii) Maximum profit per unique weight :

$N=7$

$M=15$

$(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (10, 5, 15, 7, 6, 18, 3)$

$(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (2, 3, 5, 7, 1, 4, 1)$

Now

$\frac{P_1}{w_1} = \frac{10}{2} = 5$ $\quad\quad$ $\frac{P_5}{w_5} = \frac{6}{1} = 6$ ✓

$\frac{P_2}{w_2} = \frac{5}{3} = 1.67$ $\quad\quad$ $\frac{P_6}{w_6} = \frac{18}{4} = 4.5$

$\frac{P_3}{w_3} = \frac{15}{5} = 3$ $\quad\quad$ $\frac{P_7}{w_7} = \frac{3}{1} = 3$

$\frac{P_4}{w_4} = \frac{7}{7} = 1.$

$x_5 = 1$ i.e $15 - 1 = 14$

$x_1 = 1$ i.e $14 - 2 = 12$

$x_6 = 1$ i.e $12 - 4 = 8$

$x_3 = 1$ i.e $8 - 5 = 3$

$x_7 = 1$ i.e $3 - 1 = 2$

$x_2 = 2/3$ i.e 2

$x_4 = 0$

10/01/17

⟹ Find instance of knapsack problem $N=3$, $M=20$

$(P_1, P_2, P_3) = (25, 24, 15)$, $(w_1, w_2, w_3) = (18, 15, 10)$ find

1m in mids (i) Maximum profit (ii) Minimum weight (iii) maximum profit per unit weight. 10

Sol :- Given that $N=3$, $M=20$

$(P_1, P_2, P_3) = (25, 24, 15)$

$(w_1, w_2, w_3) = (18, 15, 10)$

$\boxed{20}$ $-18 = 2$

$x_1 = 1$

$x_2 = \frac{2}{15}$

$18 \times 1 + 15 \times \frac{2}{15} + 0$

20

1m

(i) maximum profit :-

M (bag size)

$\boxed{20}$

$X_1 = 1$ i.e $20 - 18 = 2$

$X_2 = 2/15$

$X_3 = 0$.

$\therefore$ The maximum profit $\sum P_i x_i = P_1 x_1 + P_2 x_2 + P_3 x_3$

$= 25 \times 1 + 24 \times \frac{2}{15} + 15 \times 0$

$= 25 + \frac{48}{15}$

$= 28.2$

(ii) Minimum weight :-

$N=3$, $M=20$

$(P_1, P_2, P_3) = (25, 24, 15)$; $(w_1, w_2, w_3) = (18, 15, 10)$

$X_3 = 1$ i.e $20 - 10 = 10$.

$X_2 = 10/15 = 2/3$

$X_1 = 0$

$\therefore$ The minimum weight $= \Sigma p_i x_i = p_1 x_1 + p_2 x_2 + p_3 x_3$

$= 25 \times 0 + \overset{8}{\cancel{24}} \times \frac{2}{\cancel{3}} + 15 \times 1$

$= 0 + 16 + 15$

$= \underline{31}$.

(iii) Maximum profit per unit weight :-

$N = 3,\ M = 20$

$(P_1, P_2, P_3) = (25, 24, 15)$; $(w_1, w_2, w_3) = (18, 15, 10)$.

$\frac{P_1}{w_1} = \frac{25}{18} = 1.3$ $\qquad x_2 = 1$ i.e $20 - 15 = 5$

$\frac{P_2}{w_2} = \frac{24}{15} = 1.6$ $\qquad x_3 = 5/10 = 1/2$.

$\frac{P_3}{w_3} = \frac{15}{10} = 1.5$ $\qquad x_1 = 0$.

$\Sigma P_i x_i = P_1 x_1 + P_2 x_2 + P_3 x_3$

$X = \overset{12.5}{\cancel{25}} \times \frac{1}{\cancel{2}} + 24 \times 1 + 15 \times 0) = 25 \times 0 + 24 \times 1 + 15 \times \frac{1}{2}$

$= 12.5 + 24)$ $\qquad = 0 + 24 + 7.5$

$= 36.5) \times$ $\qquad = \underline{31.5}$

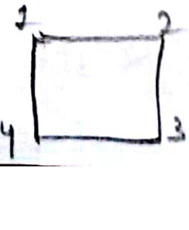
12.5
24.0
36.5

24
7.5

$\Rightarrow$ Minimal cost spanning tree :-

$\Rightarrow$ Spanning tree :- A graph 'G' is said to be a spanning tree. It contains 'n' vertices and $(n-1)$ edges.

Ex :-

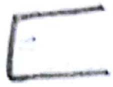


remove $(\overset{edges}{m} - \underset{vertices}{n} + 1)$ edges.

$m = 4,\ n = 4 \Rightarrow (4 - 4 + 1) = 1$ edge will be deleted from graph

(i) Delete (1,2) edge :-

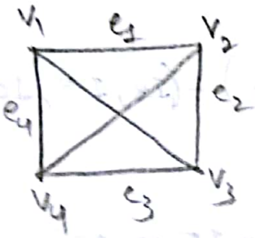
(ii) Delete (2,3) edge :-



(iii) Delete (3,4) edge :-

(iv) Delete (4,1) edge :-

$\Rightarrow$ Find spanning tree of graph 'G'.



The above graph is complete graph

$\therefore$ The no. of possible spanning trees are n^{n-2}

$= 4^{4-2}$

$= 4^2$

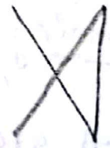
$= 16$ possible spanning trees.

Here we have to delete $(m-n+1)$ edges $= (6-4+1)$

$= 3$ edges will be deleted from graph G

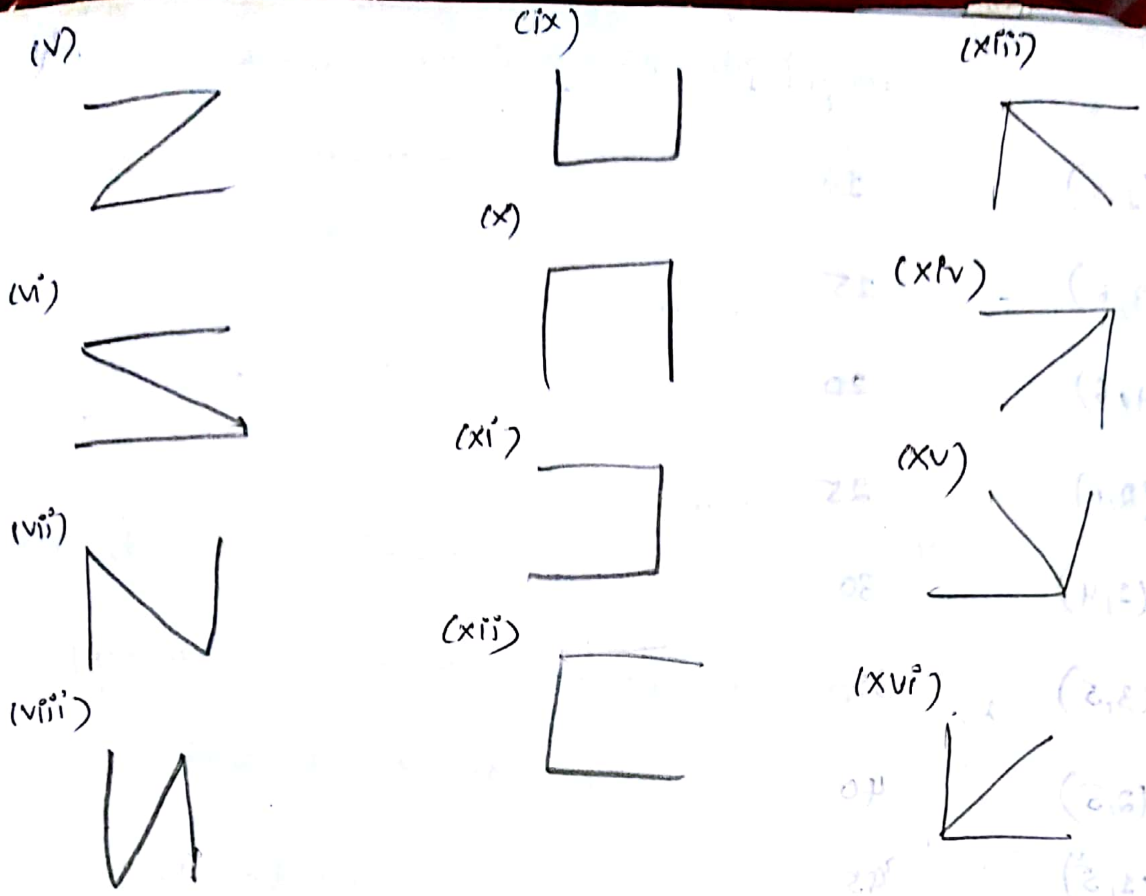
(i)

(ii)



(iii)

(iv)



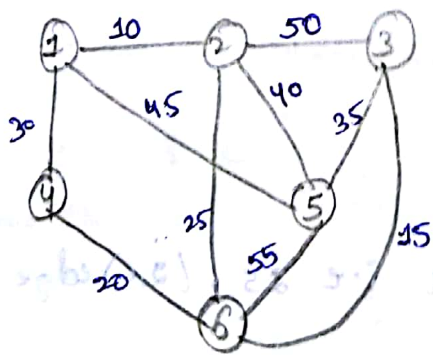
⇒ Here we mainly concentrate on 2 minimal cost spanning trees i.e

(i) krushkal's algorithm

(ii) prims algorithm.

(i) krushkal's algorithm :-

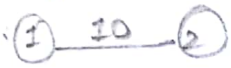
⇒ Find the minimal cost spanning tree of graph 'G' by using Krushkal's algorithm.



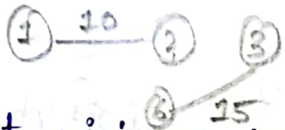
Sol :- orbitarly choose 1st vertex i.e root vertex and list the all weights in non decreasing order.

edge	weight.
(1,2)	10
(3,6)	15
(4,6)	20
(2,6)	25
(1,4)	30
(3,5)	35
(2,5)	40
(1,5)	45
(2,3)	50
(5,6)	55

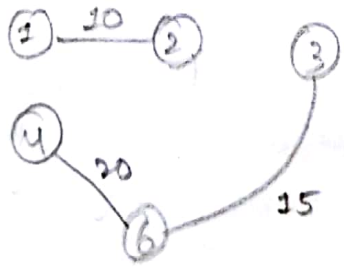
STEP 1 :- Select an edge with minimum weight i.e 10, (1,2)



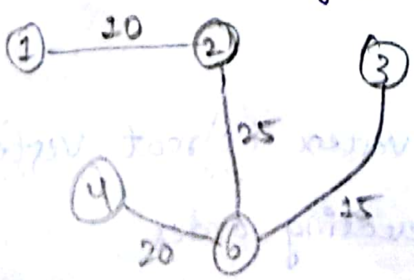
STEP 2 :- Select next minimum weight i.e 15, (3,6)



STEP 3 :- Select next minimum weight i-e 20, (4,6)

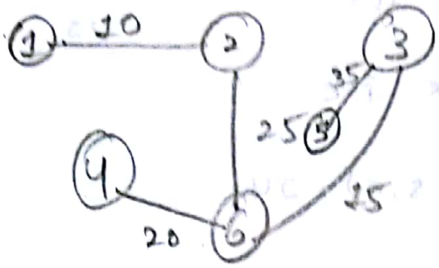


STEP 4 :- Select next minimum weight i.e 25, (2,6) edge.



(1,4) forms closed loop eliminate it.

Select next minimum weight i.e 35, (3,5)



~~(Select next minimum weight i.e 40 (2,5)~~

Here we got n vertices, (n-1) edges no loop.

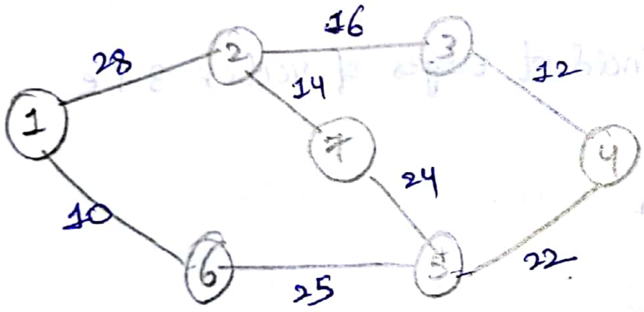
Hence it is a spanning tree.

∴ minimal cost = 10+20+15+25+35 = 105

21/01/17

⟹ prims algorithm :- F1

⟹ Find the minimal cost spanning tree of graph 'G' by using prims algorithm.

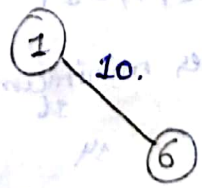


STEP 1 :- orbitarily choose 1st a vertex i.e root vertex.

$w(1,2) = 28$

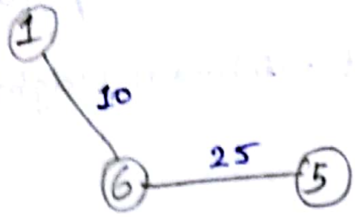
$w(1,6) = 10$

Here minimum weight is 10 then



STEP 2 :- Now list all the edges incident to vertex 6.

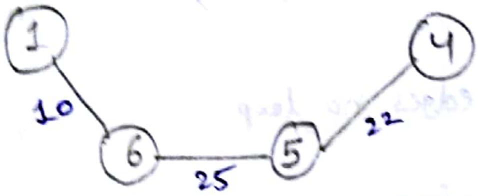
i.e $w(6,5) = 25$ then.



<u>STEP 3</u> :- Now consider all the incident edges of vertex 5 i.e

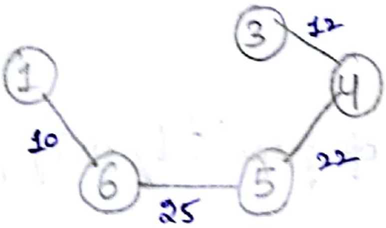
$w(5,4) = 22$, $w(5,7) = 24$.

Select least weight $w(5,4) = 22$ i.e



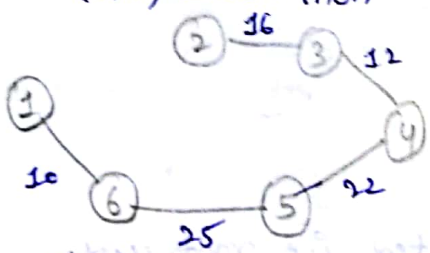
<u>STEP 4</u> :- consider all the incident edges of vertex 4 i.e

$w(4,3) = 12$. then



<u>STEP 5</u> :- consider all the incident edges of vertex 3 i.e

$w(3,2) = 16$. then

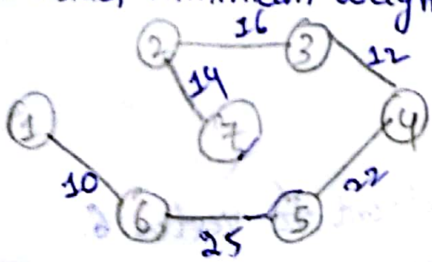


<u>STEP 6</u> :- Consider all the incident edges of vertex 2 i.e

$w(2,1) = 28$

$w(2,7) = 14$

considers minimum weight i.e $w(2,7) = 14$ then



Scanned by CamScanner

Here we got 7 vertices and 6 edges . n vertices , (n-1) edges and no closed loop so it is spanning tree.

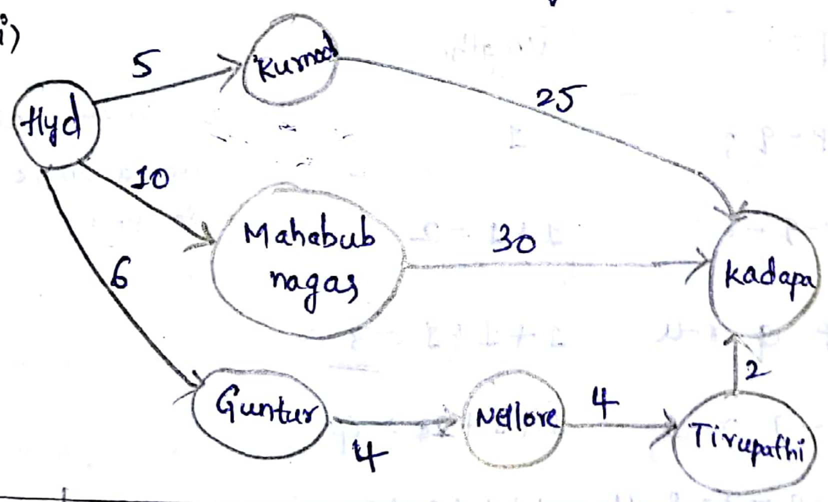
The minimal cost = 16+14+12+10+25+22

= 99

→ Single Source shortest path problem :-

Let G = (V, E) be a directed graph with weight function 'w'. Here the starting vertex of path is called as source and ending vertex of path is called as destination. The problem is to determine a shortest path to a given destination vertex 'v'. from source vertex 'v' is called single source shortest path problem

Ex :- (i)



S. NO	path	length.
1.	Hyd – ku	5
2.	H – ku – ka	5+25 = 30 .
3.	H – M	10
4.	H – M – ka	10+30 = 40
5.	H – G	6
6.	H – G – N	6+4 = 10
7.	H – G – N – T	6+4+4 = 14

Scanned by CamScanner

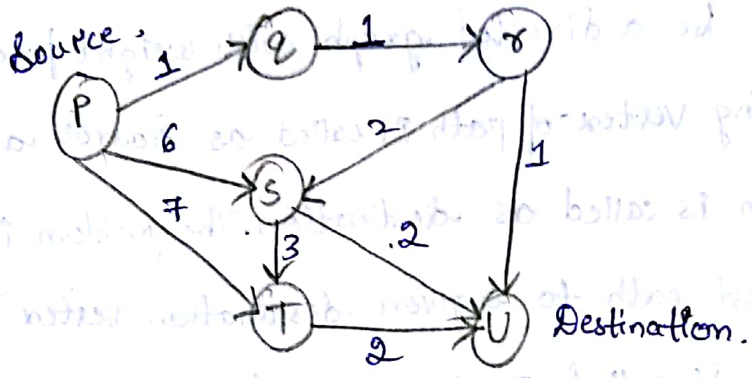
8.	H - G - N - T - Ka ✓	6+4+4+2 = 16.

(~~2~~, 4, 8) ⟶ feasible solutions becz reached last destination.

out of ~~2~~, 4, 8 select shortest path.

∴ Shortest path is 16 (H-G-N-T-Ka)

(ii) consider the directed graph of graph G'.



SNO	path	length.
1.	p - q	1
2.	p - q - r	1+1 = 2
3.	p - q - r - u	1+1+1 = 3 ✓
4.	p - q - r - s	1+1+2 = 4
5.	p - q - r - s - u	1+1+2+2 = 6 ✓
6.	p - q - r - s - T	1+1+2+3 = 7
7.	p - q - r - s - T - u	1+1+2+3+2 = 9. ✓
8.	p - s	6
9.	p - s - u	6+2 = 8 ✓
10.	p - s - T	6+3 = 9
11.	p - s - T - u	6+3+2 = 11 ✓
12.	p - T	7
13.	p - T - u	7+2 = 9 ✓

✓ → means reached destination i.e feasible solutions

Scanned by CamScanner

3, 6, 9, 8, 11, 9 lengths are feasible solutions.

∴ The shortest path is 3 (p-q-r-u).

06/02/16

chapter - 2

⇒ Dynamic programming:- (General Method):-

⇒ Dynamic programming, like divide & conquer method, solves problems by combining the solutions to sub problems. Dynamic programming is typically applied to optimization problem. This technique is invented by "Richard Bellman" in 1950's.

⇒ It is a technique for solving problems with overlapping sub problems. In this method each sub problem is solved and the result of each sub problem is recorded in a table from which we can obtain a solution to the original problem.

⇒ principle of optimality :- The dynamic programming algorithm obtains the solution using principle of optimality. The principle of optimality states that "In an optimal sequence of decisions, each subsequence must also be optimal." When it is not possible to apply the principle of optimality it is almost impossible to obtain the solution using dynamic programming approach.

Ex:- Finding of shortest path in a given graph uses the principle of optimality.

imp

⇒ * Multistage graph :- A graph is a collection of vertices and edges and there are two types of graphs i.e

(i) Directed graphs

(ii) Undirected graphs.

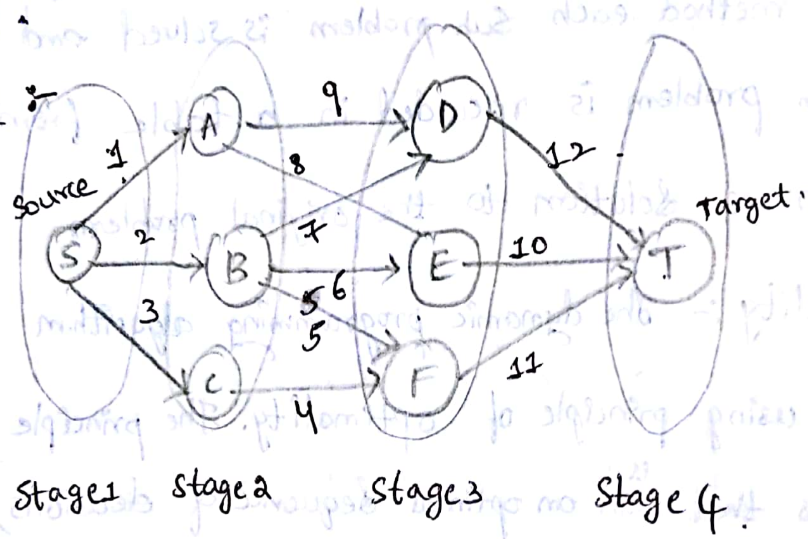
2pt

Binary tree :- A binary tree is a data struct

⇒ A multistage graph is a directed graph ~~the~~ in which the vertices are divided into two or more stages. A stage is a disjoint set of vertices.

⇒ Each stage consists of some vertices such that there is no edge between them. In multistage graph an edge is present between two vertices which are present in adjacent stages only.

Ex :-



⇒ In the above graph S & T are Source & target (destination) and 4 stages i.e stage 1, stage 2, stage 3 & stage 4

⇒ Stage 1 contains only 1 vertices i.e S

⇒ Stage 2 contains only 3 vertices i.e A, B and C

⇒ Stage 3 contains 3 vertices i.e D, E and F

⇒ Stage 4 contains 1 vertex i.e T

⇒ Here the multistage graph problem is to find the ~~minimal~~ minimum cost.

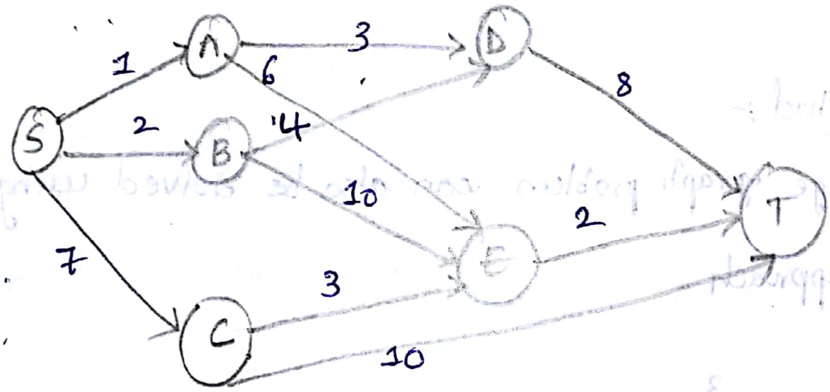
The multistage graph can be solved using 2 ways they are:

(i) Forward method

(ii) Backward method.

→ (i) Forward method :- Assume that there are 'k' stages in a graph. In this forward approach, we will find out the cost of each and every node starting from the 'k'th' stage to 1st stage. We will find out the minimal cost path from source to destination i.e stage 1, stage 2 ---- stage k. If a vertex having more than one path then we have to choose the minimum cost path.

Ex:-



Sol :- From the graph distance of $D(S,A) = 1$.

$D(S,B) = 2$.

$D(S,C) = 7$

$$D(S,D) = \min\{1 + D(A,D),\ 2 + D(B,D)\}$$

$$= \min\{1+3,\ 2+4\} = \min\{4,6\}$$

$$\boxed{D(S,D) = 4}$$

$$D(S,E) = \min\{1 + D(A,E),\ 2 + D(B,E),\ 7 + D(C,E)\}$$

$= \min\{1+6, 2+10, 7+3\}$

$= \min\{7, 12, 10\}$

$\boxed{D(S,E) = 7}$

$D(S,T) = \min\{D(S,D) + D(D,T), D(S,E) + D(E,T), D(S,C) + D(C,T)\}$

$= \min\{4+8, 7+2, 7+10\}$

$= \min\{12, 9, 17\}$

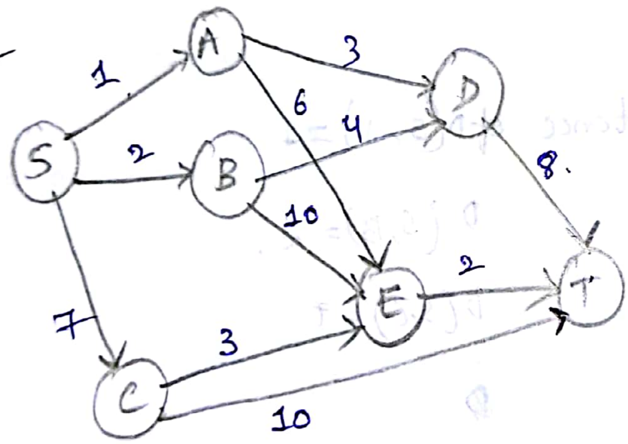
$\boxed{D(S,T) = 9}$

08/02/17

⇒ Backward method :-

The multistage graph problem can also be solved using the backward approach.

Ex:-



$D(S,T) = \min\{1 + D(A,T), 2 + D(B,T), 7 + D(C,T)\}$ — (1)

now compute $D(A,T)$.

$D(A,T) = \min\{3 + D(D,T), 6 + D(E,T)\}$ — (2)

Now compute

$$D(B,T) = \min\{4 + D(D,T),\ 10 + D(E,T)\} \longrightarrow \text{③}$$

Now compute

$$D(C,T) = \min\{3 + D(E,T),\ D(C,T)\} \longrightarrow \text{④}$$

$D(D,T) = 8$; $D(E,T) = 2$; $D(C,T) = 10$.

Substitute these values in eq's 2, 3, 4 then we get.

$$D(A,T) = \min\{3+8,\ 6+2\}$$
$$= \min\{11,\ 8\}$$

$$\boxed{D(A,T) = 8}$$

$$D(B,T) = \min\{4+8,\ 10+2\}$$
$$= \min\{12, 12\}$$

$$\boxed{D(B,T) = 12}$$

$$D(C,T) = \min\{3+2,\ 10\}$$
$$= \min\{5, 10\}$$

$$\boxed{D(C,T) = 5}$$

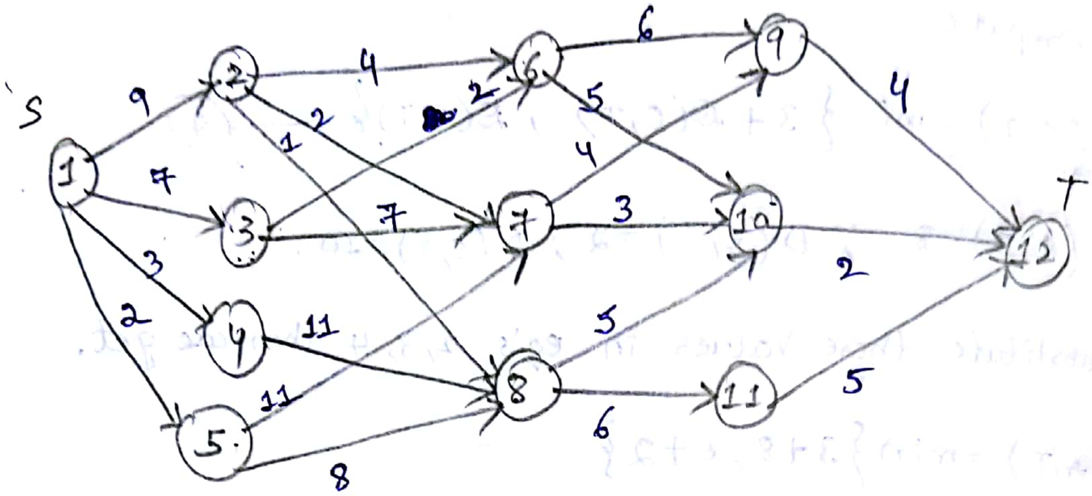
Again sub. $D(A,T)$, $D(B,T)$, $D(C,T)$ in eq ①

$$D(S,T) = \min\{1+8,\ 2+12,\ 7+5\}$$
$$= \min\{9,\ 14,\ 12\}$$

$$\boxed{D(S,T) = 9}$$

∴ ~~The~~ minimum cost for backward approach is **9**.

⇒ Find minimum cost of multistage graph as follows using dynamic programming.



Sol :- From the given graph

$D(1,2) = 9$

$D(1,3) = 7$

$D(1,4) = 3$

$D(1,5) = 2$

$D(1,6) = \min\{9 + D(2,6), 7 + D(3,6)\}$

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

$= \min\{13, 9\}$

$\boxed{D(1,6) = 9}$

$D(1,7) = \min\{9 + D(2,7), 7 + D(3,7), 2 + D(5,7)\}$

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

$= \min\{11, 14, 13\}$

$\boxed{D(1,7) = 11}$

$D(1,8) = \min\{9 + D(2,8), 3 + D(4,8), 2 + D(5,8)\}$

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

$= \min\{10, 14, 10\}$

$\boxed{D(1,8) = 10}$

$D(1,9) = \min\{D(1,6) + D(6,9), D(1,7) + D(7,9)\}$

$= \min\{9+6, 11+4\}$

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

$\boxed{D(1,9) = 15}$

$D(1,10) = \min\{D(1,6) + D(6,10), D(1,7) + D(7,10), D(1,8) + D(8,10)\}$

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

$= \min\{14, 14, 15\}$

$\boxed{D(1,10) = 14}$

$D(1,11) = \min\{D(1,8) + D(8,11)\}$

$= \min\{10+6\}$

$= \min\{16\}$

$\boxed{D(1,11) = 16}$

$D(1,12)$ (or) $D(S,T) = \min\{D(1,9) + D(9,12), D(1,10) + D(10,12), D(1,11) + D(11,12)\}$

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

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

$\boxed{D(1,12) = 16}$

∴ The minimum or shortest path is

$1 \rightarrow 2 \rightarrow 7 \rightarrow 10 \rightarrow 12$

(or)

$1 \rightarrow 3 \rightarrow 6 \rightarrow 10 \rightarrow 12$.

∴ minimum cost is 16 through forward approach.

⇒ **Backward method :**

$$D(S,T) = \min\{9 + D(2,12),\ 7 + D(3,12),\ 3 + D(4,12)\} \quad \text{— (1)}$$

Now compute

$$D(2,12) = \min\{4 + D(6,12),\ 2 + D(7,12),\ 1 + D(8,12)\} \quad \text{— (2)}$$

Now compute.

$$D(3,12) = \min\{7 + D(7,12),\ 2 + D(6,12)\} \quad \text{— (3)}$$

Now compute.

$$D(4,12) = \min\{11 + D(8,12)\} \quad \text{— (4)}$$

Now compute

$$D(6,12) = \min\{5 + D(10,12),\ 6 + D(9,12)\} \quad \text{— (5)}$$

Now compute.

$$D(7,12) = \min\{3 + D(10,12),\ 4 + D(9,12)\} \quad \text{— (6)}$$

Now compute

$$D(8,12) = \min\{5 + D(10,12),\ 6 + D(11,12)\} \quad \text{— (7)}$$

$D(10,12) = 2$; $D(9,12) = 4$; $D(11,12) = 5$.

Substitute all these values in (2), (3), (4), (5), (6), (7)

$$D(6,12) = \min\{5 + 2,\ 6 + 4\}$$

$$= \min\{7, 10\}$$

$$\boxed{D(6,12) = 7}$$

$D(7,12) = \min\{3+2, 4+4\} = \min\{5,8\}$

$\boxed{D(7,12) = 5}$

$D(8,12) = \min\{5+2, 6+5\} = \min\{7, 11\}$

$\boxed{D(8,12) = 7}$

$D(4,12) = \min\{11+7\} = \min\{18\}$

$\boxed{D(4,12) = 18}$

$D(3,12) = \min\{7+5, 2+7\} = \min\{12, 9\}$

$\boxed{D(3,12) = 9}$

$D(2,12) = \min\{4+7, 2+5, 1+7\}$
$= \min\{11, 7, 8\}$

$\boxed{D(2,12) = 7}$

Substitute all these values in eq (1).

$D(S,T) = \min\{9+7, 7+9, 3+18\} = \min\{16, 16, 21\}$

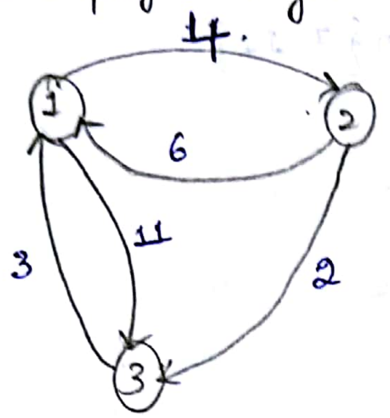
$\boxed{D(S,T) = 16}$

⇒ **All pairs shortest path problem :-** All pairs shortest path problem is to find the shortest distance between every pair of vertices of the given graph. Let $G=(V,E)$ be a directed graph where 'V' is a collection of vertices and 'E' is a collection of edges. And each edge has assigned a non negative weight. The problem is to calculate the length of shortest path between each pair of vertices. The shortest path can be calculated using the following recurrence method i.e

$$\boxed{\begin{array}{l} A^k(i,j) = W[i,j] \quad \text{if } k=0 \\ \qquad \min\{A^{k-1}(i,j),\ A^{k-1}(i,k) + A^{k-1}(k,j)\} \quad \text{if } k \geq 1 \end{array}}$$

Imp

Imp → Find all pairs shortest path problem of graph 'G' by using dynamic programming.



(1, 2, 3) vertices.

3 vertices so do till $k = 3$.

if their are 5 vertices do till $k = 5$.

Sol :- From the graph the cost adjacency matrix $A^0(i,j) = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 4 & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & \infty & 0 \end{array}$

$w[i,j] = k$

∞ infinity bcz 3, 2 no edge but adjacent.

our general formula is

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

STEP 1 :- $k = 1$ i.e going from 'i' to 'j' through the intermediate Vertex '1'. i.e $i = 1$.

$$A^1(\underset{i}{1},\underset{j}{1}) = \min\{A^0(1,1);\ A^0(1,1) + A^0(1,1)\}$$
$$= \min\{0, 0+0\}$$

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

$$A^1(\underset{i}{1},\underset{j}{2}) = \min\{A^0(1,2),\ A^0(1,1) + A^0(1,2)\}$$
$$= \min\{4, 0+4\}$$
$$= \min\{4,4\}$$

$$\boxed{A^1(1,2) = 4}$$

$$A^1(\underset{i}{1},\underset{j}{3}) = \min\{A^0(1,3),\ A^0(1,1) + A^0(1,3)\}$$
$$= \min\{11, 0+11\}$$

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

$\boxed{A^1(1,3) = 11}$

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

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

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

$\boxed{A^1(2,1) = 6}$

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

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

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

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

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

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

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

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

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

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

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

$\boxed{A^1(3,1) = 3}$

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

$= \min\{\infty, 3+4\} = \min\{\infty, 7\}$

$\boxed{A^1(3,2) = 7}$

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

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

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

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

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

STEP 2 :- Here $k=2$.

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

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

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

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

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

$\boxed{A^2(1,2) = 4}$

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

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

$\boxed{A^2(1,3) = 6}$

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

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

$\boxed{A^2(2,1) = 6}$

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

$$\boxed{A^2(2,2) = 0}$$

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

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

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

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

$$A^2(3,2) = \min\{A^1(3,2),\ A^1(3,2) + A^1(2,2)\}$$
$$= \min\{7,\ 7+0\}$$

$$\boxed{A^2(3,2) = 7}$$

$$A^2(3,3) = \min\{A^1(3,3),\ A^1(3,2) + A^1(2,3)\}$$
$$= \min\{0,\ 7+2\}$$

$$\boxed{A^2(3,3) = 0}$$

STEP3 :- Here $k=3$

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

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

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

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

$$\boxed{A^3(1,2) = 4}$$

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

$$\boxed{A^3(1,3) = 6}$$

$$A^3(2,1) = \min\{A^2(2,1), A^2(2,3)+A^2(3,1)\}$$
$$= \min\{6, 2+3\}$$
$$= \min\{6,5\}$$

$$\boxed{A^3(2,1) = 5}$$

$$A^3(2,2) = \min\{A^2(2,2), A^2(2,3)+A^2(3,2)\}$$
$$= \min\{0, 2+7\}$$
$$= \min\{0,9\}$$

$$\boxed{A^3(2,2) = 0}$$

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

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

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

$$\boxed{A^3(3,1) = 3}$$

$A^3(3,2) = \min\{\overset{i,j}{A^2(3,2)}, \overset{i,k}{A^2(3,3)} + \overset{k,j}{A^2(3,2)}\}$

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

$\boxed{A^3(3,2) = 7}$

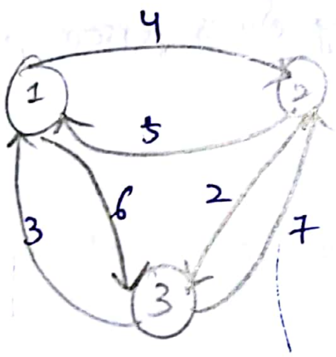
$A^3(3,3) = \min\{\underset{i,j}{A^2(3,3)}, \underset{i,k}{A^2(3,3)} + \underset{k,j}{A^2(3,3)}\}$

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

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

$\boxed{A^3(3,3) = 0}$

$$A^3(i,j) = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$



10/02/17

⇒ Travelling sales person problem ~~by~~ using dynamic programming:

Let $G = (V, E)$ be a directed graph with 'n' vertices and let $C_{i,j}$ is the cost of the edge between 'i' & 'j' and $c_{i,j} > 0$, $c_{ij} = \infty$ (their is no edge b/w $(i,j) \notin E$).

Let $V = n$ and assume that $n > 1$. A tower of 'G' is a directed simple cycle that includes every vertex in 'V'. The problem is to find tower of minimum cost. The cost of the tower is the sum of cost of the edges on the tours. In this shortest path starts and ends are the same vertex.

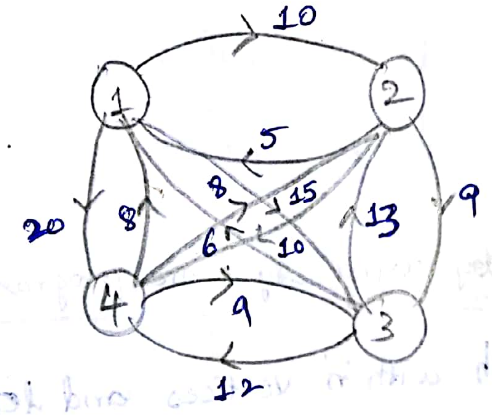
Travelling sales person problem is a permutation problem, these problems are harder to solve & we have to use dynamic programming approach, US mathematician "Richard Bellman" introduced

$$\boxed{\begin{array}{l} g(i, \phi) = c_{i1} \quad 1 \le i \le n \\ g(i, s) = \min \left\{ c_{ij} + g(j, s - \{j\}) \right\} \end{array}}$$

Here $g(i, s)$ means that 'i' is starting node and the nodes in 's' are to be traversed and 'j' means already traversed.

10m, mid comp, 4 times

Find the minimum cost of a travelling sales person problem of graph 'G' by using dynamic programming.



Sol:- From the graph the cost adjacency matrix is

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

Let us start the tour from vertex '1'.

The general formula is

$$\boxed{g(1,S) = \min\{c_{1j} + g(j, S-\{j\})\}} \quad \text{—} ①$$

And calculate $g(i,\phi) = c_{i1} \quad 1 \le i \le n$

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

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

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

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

Using eq ① we obtain

$$g(1,\{2,3,4\}) = \min\{c_{12} + g(2,(3,4)), c_{13} + g(3,(2,4)), c_{14} + g(4,(2,3))\} \quad \text{—} ②$$

Now compute $g(2,(3,4)) = \min\{c_{23} + g(3,4)\}, c_{24} + g(4,3)$ — ③

$g(3,4) = g(3,\{4\}) = \min\{c_{34} + g(4,\phi)\}$

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

$= \min\{20\}$

$$\boxed{g(3,\{4\}) = 20}$$

$g(4,3) = g(4,\{3\}) = \min\{c_{43} + g(3,\phi)\}$

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

$$\boxed{g(4,\{3\}) = 15}$$

Now substitute $g(3,\{4\})$ & $g(4,\{3\})$ in eq (3) we get.

$$g(2,(3,4)) = \min\{(9+20), 10+15\}$$

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

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

Now compute

$$g(3,\{2,4\}) = \min\{C_{32} + g(2,4), C_{34} + g(4,2)\} \quad (4)$$

Now compute $g(2,\{4\}) = \min\{C_{24} + g(4,\phi)\}$

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

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

$$g(4,\{2\}) = \min\{C_{42} + g(2,\phi)\}$$

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

$$\boxed{g(4,\{2\}) = 13}$$

Sub in (4)

$$g(3,\{2,4\}) = \min\{13+18, 12+13\}$$

$$= \min\{31, 25\}$$

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

Now compute $g(4,\{2,3\}) = \min\{C_{42} + g(2,3), C_{43} + g(3,2)\}$ (5)

Again compute $g(2,\{3\}) = \min\{C_{23} + g(3,\phi)\}$

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

$$\boxed{g(2,\{3\}) = 15}$$

Now compute $g(3,\{2\}) = \min\{c_{32} + g(2,\phi)\}$

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

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

Sub in 5

$g(4,\{2,3\}) = \min\{(8+15),(9+18)\}$

$= \min\{23, 26\}$

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

Sub $g(2,(3,4))$, $g(3,(2,4))$ and $g(4,(2,3))$ in eq 2.

$g(1,\{2,3,4\}) = \min\{(10+25),(15+25),(20+23)\}$

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

$\boxed{g(1,\{2,3,4\}) = 35}$

17/02/17

⇒ 0/1 knapsack problem by using dynamic programming :-

In olden days their was a store which contains different types of profits i.e $P_1, P_2, P_3 ---- P_n$ and cost $C_1, C_2, C_3 ----- C_n$ and weights $w_1, w_2, w_3 ---- w_n$ respectively. Now a thief wants to rob a store for that he brought a empty bag of size 'M'. Now his problem was in what way he can place the empty bag with maximum profit. This problem is called as knapsack problem. Here we have to use fractional values, 0's & 1's (binary no.'s)

Here knapsack means empty bag.

$\boxed{\text{Maximized } \Sigma P_i x_i \text{ subject to constraint } \Sigma w_i x_i \le w}$

18/02/17

STEP 1 :- Initially compute $S^0 = \{(0,0)\}$ and ~~$S_1^i = \{(P,w)/[P$~~

$$S_1^i = \{(P,w) / [(P-P_i), (w-w_i)] \in S^i\}$$

and S^{i+1} can be computed by merging S^i and S_1^i i.e

$$\boxed{S^{i+1} = S^i + S_1^i}$$

om vimp

STEP 2 :- Perging rule (Dominance rule) :-

If S^{i+1} contains P_j, w_j and (P_k, w_k) these two pairs satisfies the $P_j \le P_k$ and $w_j \ge w_k$ then we eliminate (P_j, w_j)

In perging rule basically the dominated tuples gets perged. In other words remove the pair with less profit and more weight

i.e $x_i = 1$ when $(P,w) \in S^i$ and $(P,w) \notin S^{i-1}$

$x_i = 0$ otherwise.

10m

vimp 7 times asked long

⟹ Find the knapsack problem by using dynamic programming $N=4$

$(P_1, P_2, P_3, P_4) = (1,2,5,6)$, $(w_1, w_2, w_3, w_4) = (2,3,4,5)$ size $M=8$

Sol :- Given that $N=4$

$$(P_1, P_2, P_3, P_4) = (1,2,5,6)$$

$$(w_1, w_2, w_3, w_4) = (2,3,4,5) \quad \& \text{ size } M=8$$

STEP 1 : Initially $S^0 = \{(0,0)\}$ i.e $i=0$

$$S_1^i = \{(P,W) \mid [(P-P_i),(W-W_i)] \in S^i\}$$

$\boxed{S_1^i = S_1^0 = \{(1,2)\}}$ means that we have selected first

(P,W) pair from given input i.e $(P_1, W_1) = (1,2)$

$\boxed{S^{i+1} = S^i + S_1^i}$

S^{i+1} = merging of S^0 and S_1^0

$S^{0+1} = S^1 = S^0 + S_1^0$

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

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

S_1^1 = {Select next pair and add it with S^1}

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

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

Since repetations are not allowed $\boxed{S_1^1 = \{(2,3)(3,5)\}}$

S^2 = {merging of S^1 and S_1^1}

$\boxed{S^2 = \{S^1 + S_1^1\} = \{(0,0)(1,2), (2,3), (3,5)\}}$ $(i=2)$

S_1^2 = {Select next pair and add it with S^2}

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

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

Eliminate repeated terms

$$\boxed{S_1^2 = \{(5,4)(6,6)(7,7)(8,9)\}}$$

$S^3 = \{\text{merging of } S^2 \text{ and } S_1^2\}$

$S^3 = S^2 + S_1^2$

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

By using dominance rule or perging rule. In this (3,5) pair is dominated by (5,4) pair because profit of (3,5) is less than (5,4) and weight is more so that (3,5) pair has been eliminated from S^3

i.e $(P_j, w_j), (P_k, w_k)$

$(3,5) \quad (5,4)$

$P_j \le P_k \quad w_j \ge w_k$

$3 \le 5$ (T) $\quad 5 \ge 4$ (T)

Eliminate (P_j, w_j) i.e (3,5)

∴ condition is satisfied then we can eliminate (3,5)

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

$S_1^3 = \{\text{select next pair and add it with } S^3\}$

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

$$\boxed{S_1^3 = \{(6,5), (6,5), (7,7), (8,8), (11,9), (12,11)(13,12)(14,14)\}}$$

$S^4 = \{\text{merging } S^3 + S_1^3\}$

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

our bag size is. $m=8$ so discard remaining.

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

Given size $M=8$ $\therefore (8,8) \in S^4$ and $(8,8) \notin S^3$, then

$X_i = 1$

$$\boxed{X_4 = 1}$$

($\because (P,w) \in S^i$ and $(P,w) \notin S^{i-1}$ then $X_i = 1$).

To select next pair i.e $(6,5) \Rightarrow (P,w) = (6,5)$

Now we evaluate $[P-P_i], [w-w_i]$

$[P-P_4], [w-w_4]$

$[8-6], [8-5]$

$[2,3]$

$\therefore (2,3) \in S^3$ and $(2,3) \in S^2$ then $X_3 = 0$

$(2,3) \in S^2$ and $(2,3) \notin S^1$ then $X_2 = 1$.

$X_1 = 0$

$\therefore (X_1, X_2, X_3, X_4) = (0,1,0,1)$

$P_i x_i = P_1x_1 + P_2x_2 + P_3x_3 + P_4x_4$

$= (1\times0) + (2\times1) + (5\times0) + (6\times1)$

$= 0+2+0+6 = 8$

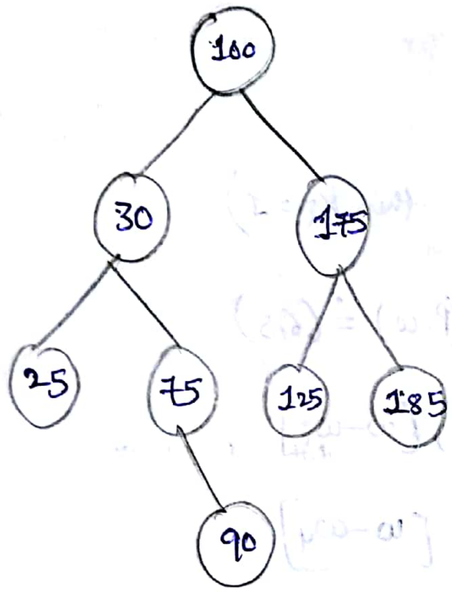
⇒ Optimal binary search tree :- (OBST)

⇒ Binary search tree :- A binary search tree is a binary tree which satisfies the following conditions.

(i) At every node its left child is smaller and right child is greater.

(ii) All elements in left subtree of root must be smaller than root and all elements in right subtree must be greater than root.

Ex :-



⇒ Optimal binary search tree are classified into 2 types i.e static and dynamic.

⇒ In static optimality problem, the tree cannot be modified after it has been constructed.

⇒ In dynamic optimality problem the tree can be modified at any time.

⇒ Here we have to calculate cost of the binary search tree is sum of successful node cost contribution and unsuccessful node cost contribution i.e

(i) cost contribution for successful node (internal node) =

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

(ii) cost contribution for unsuccessful node (external node) =

$$Q(i) * level(E_i - 1)$$

∴ cost contribution of binary search is :=

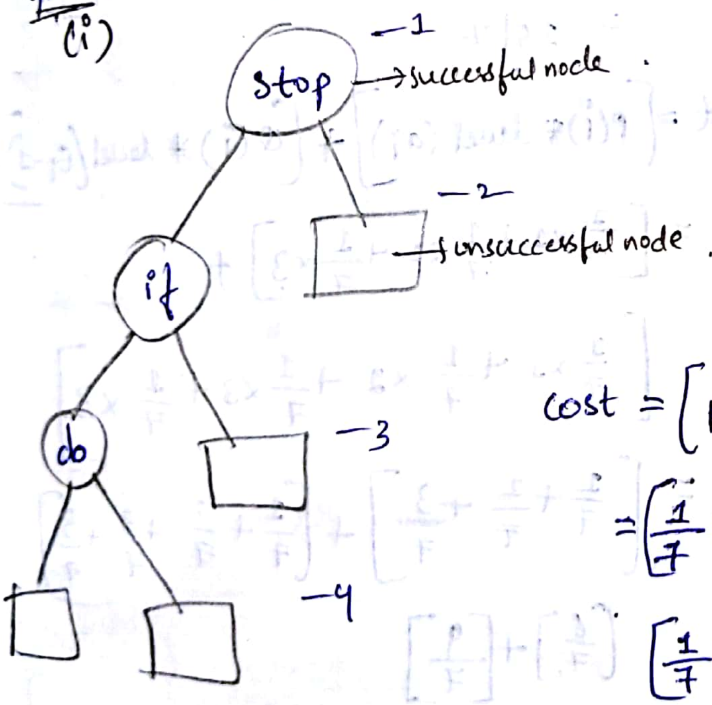
successful node + unsuccessful node cost contribution.

imp $\boxed{cost = [P(i) * level(a_i)] + [Q(i) * level[E_i - 1]].}$

Ex :- The possible binary search tree for identifying sets $(a_1, a_2, a_3) = (do, if, stop)$ are as follows the given equal probabilities $P(i) = 1/7$ and $Q(i) = 1/7 \ \forall i$ then

gate imp

(i)



$$cost = [P(i) * level(a_i)] + [Q(i) * level(E_i - 1)]$$

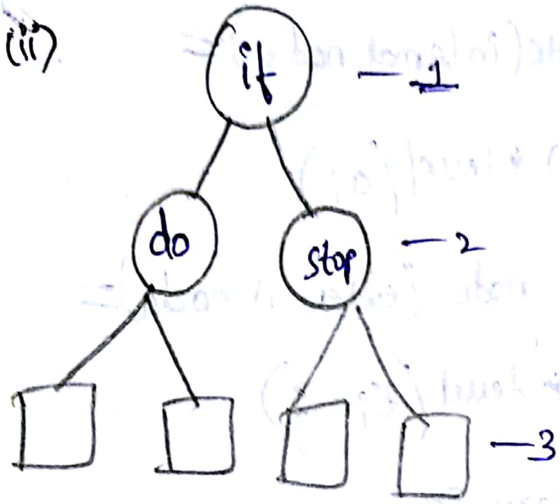
$$= \left[\frac{1}{7} \times 1 + \frac{1}{7} \times 2 + \frac{1}{7} \times 3\right] +$$

$$\left[\frac{1}{7} \times 1 + \frac{1}{7} \times 2 + \underset{\text{left}}{\frac{1}{7} \times 3} + \underset{\text{right}}{\frac{1}{7} \times 3}\right]$$

$$= \frac{15}{7}$$

Scanned by CamScanner

(ii)

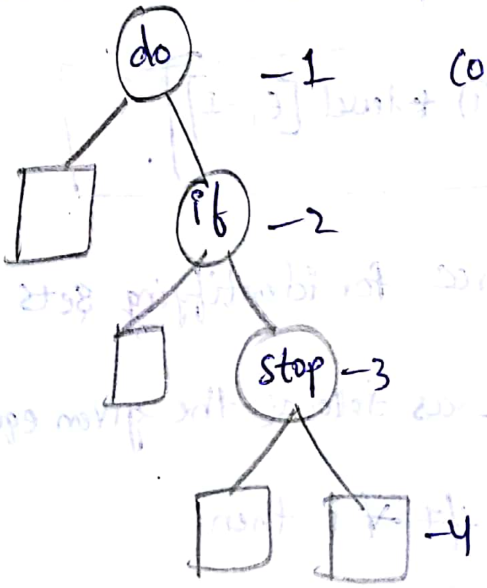


$$cost = [p(i) * level(a_i)] + [Q(i) * level(E_i - 1)]$$

$$= \left(\frac{1}{7} * 1 + \frac{1}{7} \times 2 + \frac{1}{7} \times 2\right) + \left(\frac{1}{7} \times 2 + \frac{1}{7} \times 2 + \frac{1}{7} \times 2 + \frac{1}{7} \times 2\right)$$

$$= \frac{13}{7}$$

(iii)



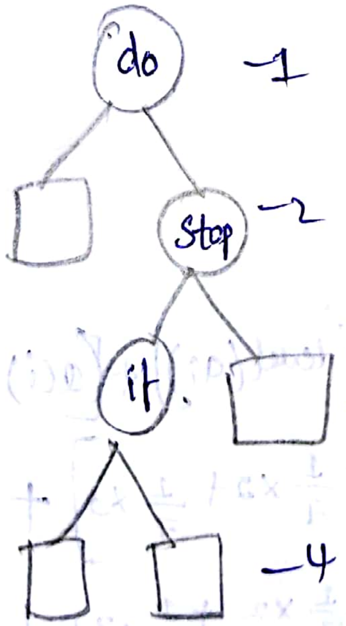
$$Cost = [p(i) * level(a_i)] + [Q(i) * level(E_i - 1)]$$

$$= \left[\frac{1}{7} \times 1 + \frac{1}{7} \times 2 + \frac{1}{7} \times 3\right] + \left[\frac{1}{7} \times 1 + \frac{1}{7} \times 2 + \frac{1}{7} \times 3 + \frac{1}{7} \times 3\right]$$

$$= \left[\frac{1}{7} + \frac{2}{7} + \frac{3}{7}\right] + \left[\frac{1}{7} + \frac{2}{7} + \frac{6}{7}\right]$$

$$= 15/7$$

(iv)



$$Cost = [p(i) * level(a_i)] + [Q(i) * level(E_i - 1)]$$

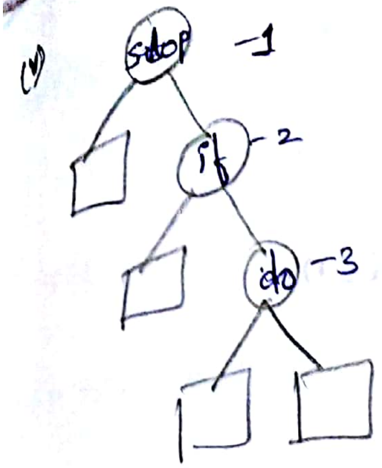
$$= \left[\frac{1}{7} \times 1 + \frac{1}{7} \times 2 + \frac{1}{7} \times 3\right] + \left[\frac{1}{7} \times 1 + \frac{1}{7} \times 2 + \frac{1}{7} \times 3 + \frac{1}{7} \times 3\right]$$

$$= \left[\frac{1}{7} + \frac{2}{7} + \frac{3}{7}\right] + \left[\frac{1}{7} + \frac{2}{7} + \frac{3}{7} + \frac{3}{7}\right]$$

$$= \left[\frac{6}{7}\right] + \left[\frac{9}{7}\right]$$

$$= 15/7.$$

(v)

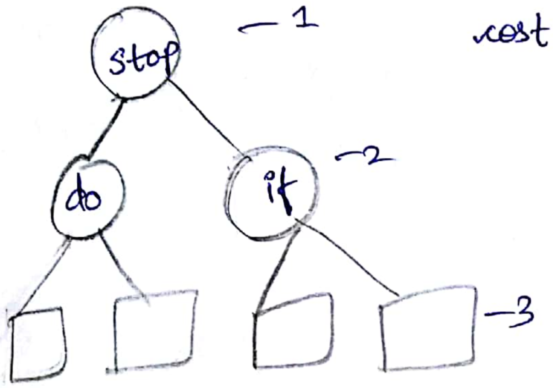


$$cost = [p(i) * level(a_i)] + [Q(i) * level(E_i - 1)]$$

$$= \left[\frac{1}{7}\times 1 + \frac{1}{7}\times 2 + \frac{1}{7}\times 3\right] + \left[\frac{1}{7}\times 1 + \frac{1}{7}\times 2 + \frac{1}{7}\times 3 + \frac{1}{7}\times 3\right]$$

$$= \left[\frac{16}{7}\right]$$

(vi)



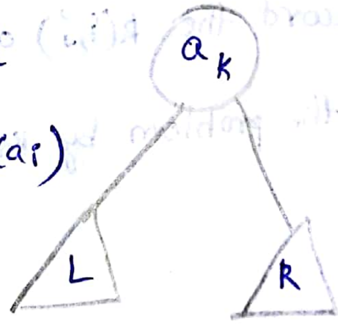
$$cost = [p(i) * level(a_i)] + [Q(i) * level[E_i - 1]]$$

$$= \left[1\times\frac{1}{7} + \frac{1}{7}\times 2 + \frac{1}{7}\times 2\right] + \left[\frac{1}{7}\times 2 + \frac{1}{7}\times 2 + \frac{1}{7}\times 2 + \frac{1}{7}\times 2\right]$$

$$= 13/7.$$

20/02/17

⇒ The structure of OBST is



$$cost(L) = \sum_{i=1}^{k} p(i) * level(a_i) + \sum_{i=0}^{k} Q(i) * Level(E_i - 1)$$

$$cost(R) = \sum_{i=1}^{k} P(i) * level(a_i) + \sum_{i=0}^{k} Q(i) * level(E_i - 1)$$

$$\boxed{\begin{array}{l} cost(i,j) = \omega(i,j) + \min\limits_{i \le k \le j} \{ c(i, k-1) + c(k,j) \} \\ \text{where } \omega(i,j) = P(j) + Q(j) + \omega(i, j-1) \end{array}}$$

⇒ procedure :-

STEP 1 :- Initially $C(i,i) = 0$

$w(i,i) = Q(i)$ for $0 \le i \le n$.

$R(i,i) = 0$

STEP 2 :- Solve $C(0,n)$ by first computing all $C(i,j)$ such that

(i) $j - i = 1$

(ii) $j - i = 2$

(iii) $j - i = 3$

⋮

STEP 3 :- $C(i,j)$ is the cost of the OBST "T_{ij}" and after computation we record the $R(i,j)$ of each tree T_{ij}.

STEP 4 :- We solve the problem by knowing
$$\left.\begin{array}{l} w(i, i+1) \\ C(i, i+1) \\ R(i, i+1) \end{array}\right\} 0 \le i < 4$$

$$\left.\begin{array}{l} w(i, i+2) \\ C(i, i+2) \\ R(i, i+2) \end{array}\right\} 0 \le i \le 3$$

⋮

till 1

and repeat until $w(0,n)$, $C(0,n)$, $R(0,n)$ are obtained.

⇒ Find the OBST by using dynamic programming

$n=4$, $(a_1, a_2, a_3, a_4) = (\text{do if read while})$

$p(1:4) = (3,3,1,1)$, $Q(0:4) = (2,3,1,1,1)$.

Sol: Given that $n=4$

$a_1 = \text{do}$	$P_1 = 3$	$Q_0 = 2$
$a_2 = \text{if}$	$P_2 = 3$	$Q_1 = 3$
$a_3 = \text{read}$	$P_3 = 1$	$Q_2 = 1$
$a_4 = \text{while}$	$P_4 = 1$	$Q_3 = 1$
		$Q_4 = 1$.

STEP 1 :- Initially $C(i,i) = 0$ i.e

$C(1,1) = 0$, $C(2,2) = 0$, $C(3,3) = 0$, $C(4,4) = 0$

$w(i,i) = Q(i)$ i.e

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

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

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

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

$w(4,4) = Q(4) = 1$.

$R(i,i) = 0$ i.e

$R(1,1) = 0$

$R(2,2) = 0$, $R(3,3) = 0$, $R(4,4) = 0$.

	0	1	2	3	4
	w c R	w c R	w c R	w c R	w c R
0	2, 0, 0	8, 8, 1	12, 19, 1	14, 25, 2	16, 32, 2
1		3, 0, 0	7, 7, 2	9, 12, 2	11, 19, 2
2			1, 0, 0	3, 3, 3	5, 8, 3
3				1, 0, 0	3, 3, 4
4					1, 0, 0

Compute all $c(i,j)$ such that

1) $j - i = 1$ i.e $j = 1 + i$ Now $0 \le i < 4$ i.e $i = 0,1,2,3$ and $i < k \le j$

for calculating 'k' value.

(i) $i = 0$, $j = i + 1$, $i < k \le j$

$j = 0 + 1$ $\quad$ $0 < k \le 1$

$\boxed{j = 1}$ $\quad$ $\boxed{k = 1}$

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

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

$= 3 + 3 + 2$

$\boxed{w(0,1) = 8}$

$$cost(i,j) = \omega(i,j) + \min\{c(i,k-1)+c(k,j)\}$$

$$cost(0,1) = \omega(0,1) + \min\{c(0,0)+c(1,1)\}$$

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

$$\boxed{cost(0,1) = 8}$$

$\boxed{R(0,1) = 1}$ means that value of 'k' is minimum in the above equation.

(ii) $i=1$, $j=i+1$, $i < k \le j$

$j = 1+1$ $\quad 1 < k \le 2$

$\boxed{j=2}$ $\quad \boxed{k=2}$

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

$$= p(2) + q(2) + \omega(1,1)$$

$$= 3+1+3$$

$$\boxed{\omega(1,2) = 7}$$

$$cost(i,j) = \omega(i,j) + \min\{c(i,k-1)+c(k,j)\}$$

$$cost(1,2) = \omega(1,2) + \min\{c(1,1)+c(2,2)\}$$

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

$$\boxed{cost(1,2) = 7}$$

$\boxed{R(1,2) = 2}$ ← k value

(iii) $i = 2, \quad j = i+1 \quad , \quad i < k \le j$

$j = 2+1 \qquad 2 < k \le 3$

$\boxed{j = 3} \qquad \boxed{k = 3}$

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

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

$= 1 + 1 + 1$

$\boxed{w(2,3) = 3}$

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

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

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

$\boxed{cost(2,3) = 3}$

$\boxed{R(2,3) = 3}$

(iv) $i = 3, \quad j = i+1 \quad , \quad i < k \le j$

$j = 3+1 \qquad 3 < k \le 4$

$\boxed{j = 4} \qquad \boxed{k = 4}$

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

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

$= 1 + 1 + 1$

$\boxed{w(3,4) = 3}$

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

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

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

$\boxed{cost(3,4) = 3}$

$\boxed{R(3,4) = 4}$

Step 2:- $j - i = 2$, $j = i+2$

$0 \le i < 3$

$i = 0,1,2$

(i) $i=0$, $j = i+2$, $i < k \le j$

$= 0+2$ $\quad 0 < k \le 2$

$j = 2$ $\quad \boxed{k = 1,2}$

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

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

$= 3 + 1 + 8$

$\boxed{w(0,2) = 12}$

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

$cost(0,2) = w(0,2) + \min\{\underbrace{c(0,0) + c(1,2)}_{k=1}, \underbrace{c(0,1) + c(2,2)}_{k=2}\}$

$cost(0,2) = 12 + \min\{0+7, 8+0\}$

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

$= 12 + 7$

$\boxed{c(0,2) = 19}$

$\boxed{R(0,2) = 1}$ → in 'k' take min

(ii) $\boxed{i=1}$, $j = i+2$ $\quad i < k \le j$

$j = 1+2$ $\quad 1 < k \le 3$

$\boxed{j=3}$ $\quad \boxed{k = 2,3}$

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

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

$$= 1+1+7$$

$$\boxed{w(1,3) = 9}$$

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

$$cost(1,3) = w(1,3) + \min\{c(1,1) + c(2,3),\ c(1,2) + c(3,3)\}$$

$$= 9 + \min\{(0+3), (7+0)\}$$

$$= 9 + \min\{3,7\}$$

$$= 9+3$$

$$\boxed{cost(1,3) = 12}$$

$$\boxed{R(1,3) = 2}$$

(iii) $\boxed{i=2}$, $j = i+2$ $\quad i < k \le j$

$j = 2+2$ $\quad 2 < k \le 4$

$\boxed{j=4}$ $\quad \boxed{k = 3,4}$

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

$$w(2,4) = p(4) + Q(4) + w(2,3)$$

$$= 1+1+3$$

$$\boxed{w(2,4) = 5}$$

$cost(i,j) = \omega(i,j) + \min\{c(i,k-1) + c(k,j)\}$

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

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

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

$= 5+3$

$\boxed{c(2,4) = 8}$

$\boxed{R(2,4) = 3}$

STEP 3:- $j-i = 3$, $j = 3+i$, $0 \le i < 2$

$i = 0, 1$.

(i) $\boxed{i=0}$, $j = i+3$, $i < k \le j$

$j = 0+3$ $\quad 0 < k \le 3$

$\boxed{j = 3}$ $\quad \boxed{k = 1,2,3}$

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

$\omega(0,3) = p(3) + q(3) + \omega(0,2)$

$= 1+1+12$

$\boxed{\omega(0,3) = 14}$

$cost(i,j) = \omega(i,j) + \min\{c(i,k-1) + c(k,j)\}$

$cost(0,3) = \omega(0,3) + \min\{c(0,0) + c(1,3),\ c(0,1) + c(2,3),$

$c(0,2) + c(3,3)\}$

$= 14 + \min\{(0+12),\ (8+3),\ c(19+0)\}$

$= 14 + \min\{12, \underline{11}, 19\}$ check it is at 2nd place so 2

$cost(0,3) = 14+11$

$\boxed{cost(0,3) = 25}$

$\boxed{R(0,3) = 2}$

(ii) $\boxed{i=1}$, $j = i+3$, $i < k \leq j$

$j = 1+3$ $\quad$ $1 < k \leq 4$

$\boxed{j=4}$ $\quad$ $\boxed{k = 2,3,4}$

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

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

$= 1 + 1 + 9$

$\boxed{w(1,4) = 11}$

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

$cost(1,4) = w(1,4) + \min\{c(1,1) + c(2,4),\ c(1,2) + c(3,4),$
$c(1,3) + c(4,4)\}$

$cost(1,4) = 11 + \min\{(0+8), (7+3), (12+0)\}$

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

↳ 8 in 1st position so in R write k value 1st position value.

$= 11 + 8$

$\boxed{cost(1,4) = 19}$

$\boxed{R(1,4) = 2}$

STEP 4: $j-i=4$, $j=4+i$, $0 \le i < 1$

$i=0$

(i) $\boxed{i=0}$, $j=i+4$, $i<k\le j$

$j=0+4$ $\quad 0<k\le 4$

$\boxed{j=4}$ $\quad \boxed{k=1,2,3,4}$

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

$w(0,4) = p(4)+q(4)+w(0,3)$

$w(0,4)= 1+1+14$

$\boxed{w(0,4)=16}$

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

$cost(0,4) = w(0,4)+\min\{c(0,0)+c(1,4),\ c(0,1)+c(2,4),$
$c(0,2)+c(3,4),\ c(0,3)+c(4,4)\}$

$cost(0,4) = 16+\min\{(0+19),\ (8+8),\ (19+3),\ (25+0)\}$

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

$=16+16$

$\boxed{cost(0,4)=32}$

$\boxed{R(0,4)=2}$

	0	1	2	3	4
0	2,0,0	8,8,1	12,19,1	14,25,2	16,32,2
1		3,0,0	7,7,2	9,12,2	11,19,2
			4,0,0	3,3,3	5,8,3.
				1,0,0	3,3,4
					1,0,0.

From the table we see that cost$(0,4)=32$ of OBST for (a_1, a_2, a_3, a_4). i.e the root tree.

T_{04} i.e $R_{04} = 2$ i.e a_2

$\therefore R_{\underset{i\,j}{04}} = \underset{k}{2}$

$i=0, j=4, k=2$.

∴ The tree is.

Now evaluate left node by using $R_{i\,k-1}$ and evaluate right node by using R_{kj}

⇒ Left node :-

R_{ik-1}

From root node $i=0, j=4, k=2$

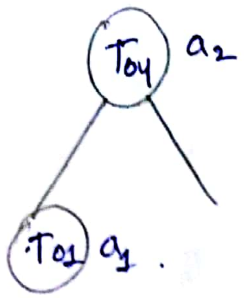
Sub. in above eq^n R_{ik-1}

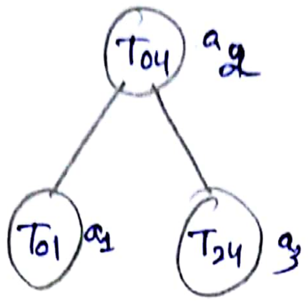
Right node

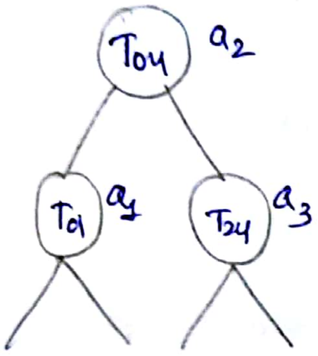
R_{kj}

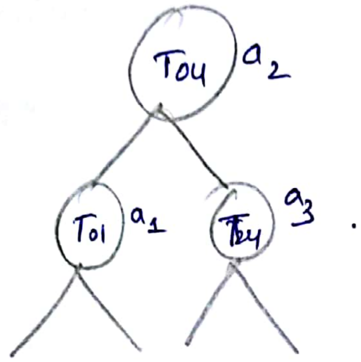
From root node $k=2$
$j=4$

$R_{01} = 1$ i.e a_1 . $R_{24} = 3$ i.e a_3









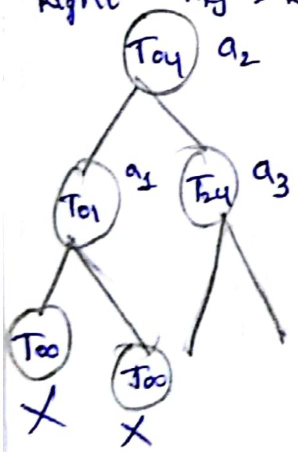
From root node T_{01}

$i=0, j=1, k=1$

$R_{00} = 0 = a_0$

But a_0 is not their.

Right — $R_{kj} = R_{11} = 0$.

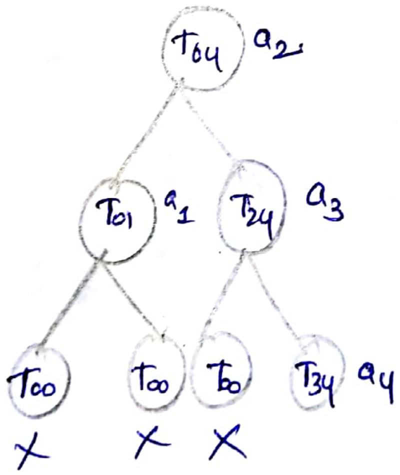


From root node T_{24}

$i=2, j=4, k=3$.

$R_{22} = 0$ (discard .

$R_{kj} = R_{34} = 4 = a_4$.



∴ The optimal binary search tree is.