9/1/24 3. Dynamic Programming:

## Dynamic Programming:

→ It is also one of the optimization technique which is used for finding optimal solution to any problem.

→ The difference between Greedy and Dynamic Programming is, in greedy method we are using a common approach to find an optimal solution whereas in dynamic programming it finds all the possible solutions and take optimal solution out of them.

→ In Dynamic programming it uses 2 approaches finding optimal solution

1) Memorization technique
2) Tabulation method

Memorization technique uses recursive approach whereas tabulation method uses iterative approach.

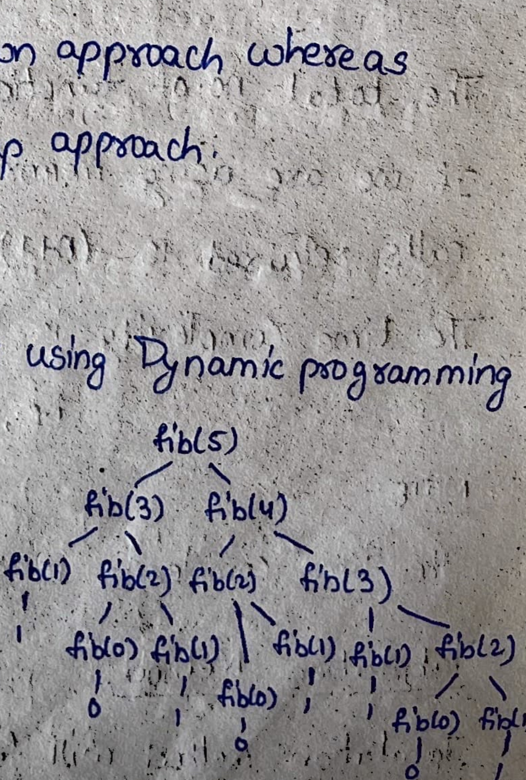
→ In Dynamic programming, memorization uses Top down approach whereas Tabulation method uses Bottom up approach.

### General Method:

1. Find ~~Factorial~~ Fib of a given number using Dynamic programming

Psuedo code

```
fib(int n)
{
   if (n≤1)
      return n;
   fib(n-2)+fib(n-1)
}
```




The no.of calls reluire if we are using General approach is 15.

The Time Complexity is $2 \times T(n-1) + 1$

i.e., $O(2^n)$

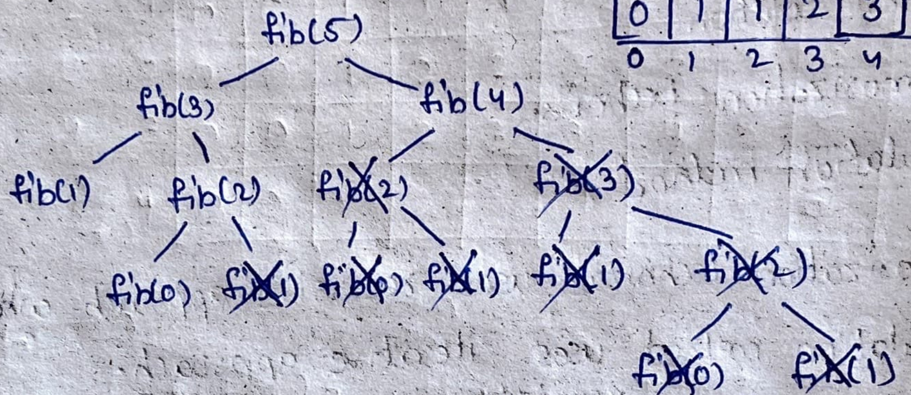
## Memorization Technique:

If we are implementing memorization technique we are using an array and initially contains -1 at every index.

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | -1 |

Whenever value is generated it is stored in array for reducing no.of function calls.

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 3 | 5 |




The total no.of function calls required = 6

If we are using memorization technique the total function calls required is $(n+1)$

The time Complexity is $O(n+1)$

i.e., $O(n)$

**NOTE!**

The Time complexity is reduced from $2^n$ to $n$

## Tabulation Method:

Tabulation method will follows an iterative approach

Psuedo code

```
fib(int n)
{
  if(n≤1)
    return n;
  F[0]=0, F[1]=1;
  for (int i=2; i<n; i++)
  {
     F[i]= F[i-2]+F[i-1];
     return F[i];
  }
}
```

| 0 | 1 | 1 | 2 | 3 | 5 |
|---|---|---|---|---|---|
| F[0] | F[1] | F[2] | F[3] | F[4] | F[5] |

## Applications of Dynamic Programming:

Some of the applications of Dynamic

1. 0/1 Knapsack Problem (0/1)
2. Single source shortest Path - All Destinations (Bellman ford)
3. All pairs shortest path - Floyds algorithm
4. Travelling Salesperson problem
5. Multi stage graphs
6. Optimal Binary Search trees

### 1. 0/1 Knapsack Problem:

Knapsack Problem using dynamic programming can be solved in two ways

1) By using Tabulation method
2) By using sets method

In this problem, if the object is included into bag its value is 1. Otherwise 0.

No partial inclusions of an object into bag is not allowed.

0/1 Knapsack problem can produce optimal solution as

Round number and not a fraction value

In this problem, we have to find all the solutions and out of those solutions we are selecting one solution as optimal solution

<u>0/1 Knapsack problem using Tabulation method:</u>

1) Find optimal solution for the following objects using 0/1 Knapsack Problem where $m=8$

$W = \{2, 3, 4, 5\}$

$P = \{1, 2, 3, 4\}$

Sol)

| $P_i$ | $w_i$ | $x_i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 2 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 3 | 2 | 0 | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 |
| 3 | 4 | 3 | 0 | 0 | 1 | 2 | 3 | 3 | 4 | 5 | 5 |
| 4 | 5 | 4 | 0 | 0 | 1 | 2 | 3 | 4 | 4 | 5 | 6 |

<u>Formula:</u>

$V[i, w] = \max\{V[i-1, w], V[i-1, w-w_i] + P_i\}$

$V[4, 1] = \max\{V[3, 1], V[3, 1-5] + 4\} = \max\{V[3, 1], V[3, -4] + 4\}$

$= 0$

$V[4, 2] = \max\{V[3, 2], V[3, 2-5] + 4\}$

$= 1$

$V[4, 3] = \max\{V[3, 3], V[3, 3-5] + 4\}$

$= 2$

$V[4, 4] = \max\{V[3, 4], V[3, 4-5] + 4\} = 3$

$V[4, 5] = \max\{V[3, 5], V[3, 0] + 4\} = \max\{3, 0+4\} = 4$

$V[4, 6] = \max\{V[3, 6], V[3, 6-5] + 4\} = \max\{V[3, 6], V[3, 1] + 4\}$

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

$V[4, 7] = \max\{V[3, 7], V[3, 7-5] + 4\}$

$= max\{5, 1+4\} = 5$

$v[4,8] = max\{v[3,8], v[3,8-5]+4\}$

$= max\{5, 2+4\} = 6$

| $x_4$ | $x_3$ | $x_2$ | $x_1$ |
|---|---|---|---|
| 1 | 0 | 1 | 0 |

2) $m=6$

$W = \{1, 2, 3, 4\}$

$P = \{4, 5, 6, 7\}$

| $x_4$ | $x_3$ | $x_2$ | $x_1$ |
|---|---|---|---|
| 0 | 1 | 1 | 1 |

Sol)

| $P_i$ | $w_i$ | $x_i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 1 | 0 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 2 | 2 | 0 | 4 | 5 | 9 | 9 | 9 | 9 |
| 6 | 3 | 3 | 0 | 4 | 5 | 6 | 10 | 11 | 15 |
| 7 | 4 | 4 | 0 | 4 | 5 | 6 | 7 | 11 | 12 |

Highest = 15 So Keep 1 at $x_3$

Now $15 - P_i = 15 - 6 = 9$ So keep 1 at $x_2$

Now $9 - P_i = 9 - 5 = 4$ So keep 1 at $x_1$

11/1/24

2. Single Source Shortest path : using Dynamic Programming

→ Single source shortest Path using Dynamic Programming is working for negative weighted edge also.

→ For implementing Single Source shortest path we're using Bellmenford algorithm.

→ In Bellmenford algorithm we are having relaxation for vertex cost.

→ In this, if we are having n vertices, relaxation can be done up to (n-1) times.

if $c(u) + d(u,v) < c(v)$

then $c(v) = c(u) + d(u,v)$

→ In this algorithm, the following steps are used for finding an optimal solution.

**step-1:** Consider any vertex as a source vertex and update cost to every vertex.

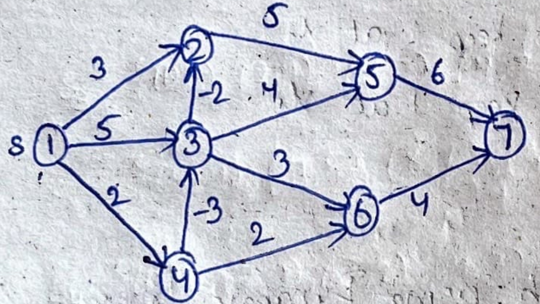
Cost to reach source vertex is 0 and all the remaining vertices is infinite.

**step-2:** Perform relaxation from source vertex to all the remaining vertices until (n-1) times.

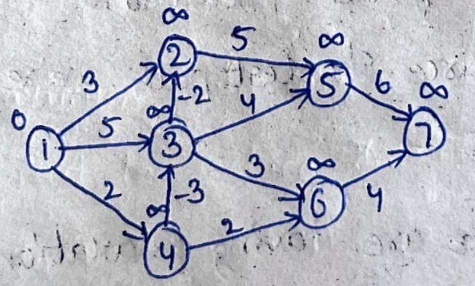
At any stage if no cost is updated then we can stop the procedure.

**Example:**

Find optimal solution for the following Negative weighted directed graph

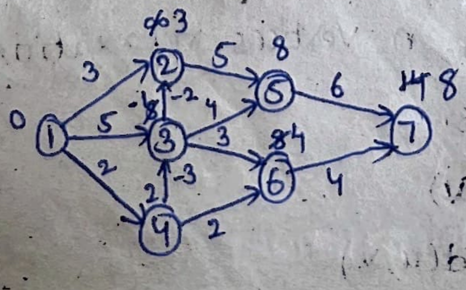


**step-1!** Update cost of source vertex with 0 and all the remaining vertices to ∞

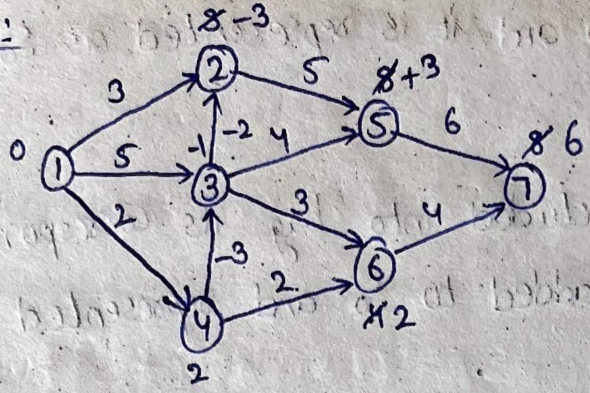


Edges → (1, 2), (1, 3), (1, 4), (2,5), (3,2), (3,5), (3,6), (4,3), (4,6), (5, 7), (6, 7)

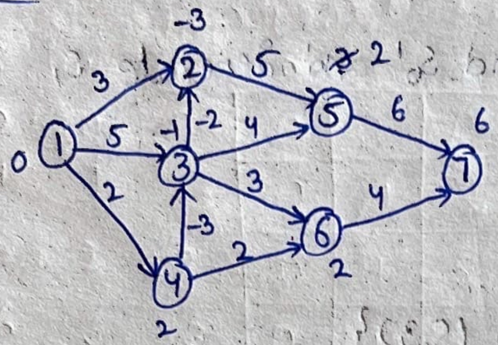
**step-2:**



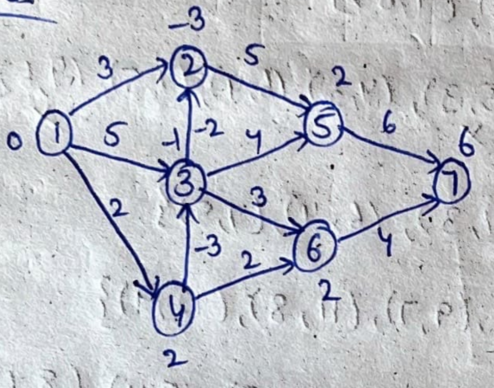
**step-3:**



**step-4:**



**step-5:**



Costs are

1 - 0

2 - -3

3 - -1

4 - 2

5 - 2

6 - 2

7 - 6

1, **0/1 Knapsack Problem using Sets method:**

Find Optimal solution for the following objects using Sets method where $m=8$, $w=\{2,3,4,5\}$

$p=\{1,2,3,4\}$

Sol. Initially no object is included into bag and its.

Corresponding profit is 0 and it is represented as $S_0$

$S_0 = (0,0)$

Now, First object is included into bag its corresponding weight and profit is added to $S_0$ and represented as $S_0^1$.

$$S_0^1 = (0+2, 0+1) = (2,1)$$

Now we are merging $S_0$ and $S_0^1$ and generate $S_1$

$$S_1 = \{(0,0), (2,1)\}$$

$$S_1^1 = \{(3,2), (5,3)\}$$

$$S_2 = \{(0,0), (2,1), (3,2), (5,3)\}$$

$$S_2^1 = \{(4,3), (6,4), (7,5), (9,6)\}$$

$$S_3 = \{(0,0), (2,1), (3,2), \cancel{(5,3)}, (4,3), (6,4), (7,5), \cancel{(9,6)}\}$$

$$S_3 = \{(0,0), (2,1), (3,2), (4,3), (6,4), (7,5)\}$$

$$S_3^1 = \{(5,4), (7,5), (8,6), (9,7), (11,8), (12,9)\}$$

$$S_4 = \{(0,0), (2,1), (3,2), (4,3), \cancel{(6,4)}, (7,5), (5,4), (8,6), \cancel{(9,7)}, \cancel{(11,8)}, \cancel{(12,9)}\}$$

$$S_4 = \{(0,0), (2,1), (3,2), (4,3), (5,4), (7,5), (8,6)\}$$

The maximum profit pair is (8,6) which is present in $S_4$ only.

So we need to include $x_4$ into bag. (i.e., $x_4 = 1$)

$$x_4 = 1 \ \& \ (8-5, 6-4) = (3,2)$$

The Pair 1st (3,2) is present in $S_4$, $S_3$, $S_2$. So we need to include $x_2$ into bag.

$$\text{i.e., } x_2 = 1 \ \& \ (3-3, 2-2) = (0,0)$$

No other object is included into bag

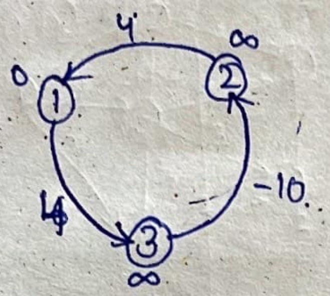
The optimal solution is

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

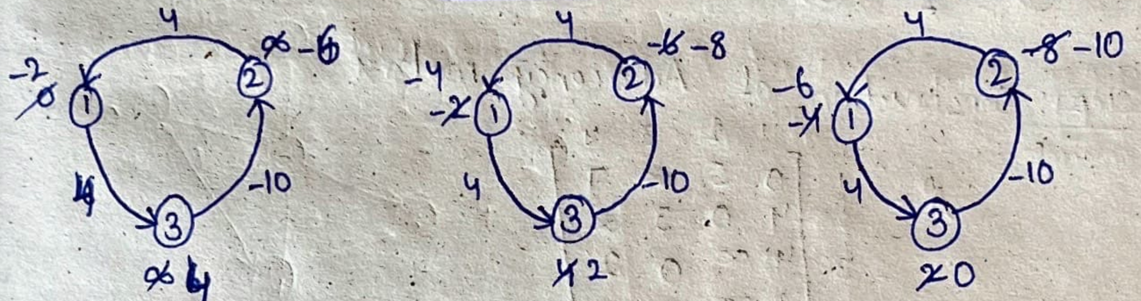
## Drawback with Bellmen ford Algorithm:

In Bellmenford algorithm, if a graph contains a cycle with negative weighted edge after $(n-1)^{th}$ iteration also the cost of vertex is going to be relaxed.

eg:




Iteration-1:    Iteration-2:    Iteration-3:




## 3. All Pairs Shortest path! – Floyds / Warshals

In this algorithm we have to find minimum cost path from any vertex to remaining all other vertices, in this algorithm every vertex will be considered as source vertex.

The following steps are used for finding an optimal solution

**Step-1:** Construct an adjacency matrix for the given graph represented as $A^0$

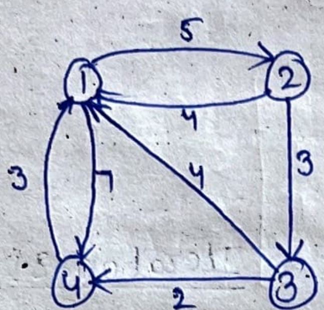
**Step-2:** Construct $A^1$ matrix without changing $1^{st}$ row, $1^{st}$ column and diagonal values from $A^0$

**Step-3:** Repeat step-2 upto $A^K$ matrix, where $K$ is no. of vertices

The following formula is used for constructing $A^K$ matrix.

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

**eg:** Find optimal solution for the following graph using All Pair shortest path or Floyds.




**Step-1:** Construction of Adjacency matrix

$$A^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & \infty & 7 \\ 4 & 0 & 3 & \infty \\ 4 & \infty & 0 & 2 \\ 3 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

**Step-2:** Now construct $A^1$ matrix without changing 1st row, 1st column & diagonal values from $A^0$

$$A^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & \infty & 7 \\ 4 & 0 & & \\ 4 & & 0 & \\ 3 & & & 0 \end{bmatrix} \end{array}$$

we have to calculate $A^1(2,3)$, $A^1(2,4)$, $A^1(3,2)$, $A^1(3,4)$, $A^1(4,2)$, $A^1(4,3)$

Formula is

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

Now,

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

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

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

$= \min\{\infty, 4+7\} = \min(\infty, 11) = 11$

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

$= \min\{\infty, 4+5\} = \min(\infty, 9) = 9$

$A^1(3,4) = \min\{A^0(3,4), A^0(3,1) + A^0(1,4)\}$

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

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

$= \min\{\infty, 3+5\} = 8$

$A^1(4,3) = \min\{A^0(4,3), A^0(4,1) + A^0(1,3)\}$

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

Now,

$$A^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & \infty & 7 \\ 4 & 0 & 3 & 11 \\ 4 & 9 & 0 & 2 \\ 3 & 8 & \infty & 0 \end{bmatrix} \end{array}$$

Step-3: Now construct $A^2$ matrix without changing 2nd row, 2nd column & diagonal values from $A^1$

$$A^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & & \\ 4 & 0 & 3 & 11 \\ & 9 & 0 & \\ & 8 & & 0 \end{bmatrix} \end{array}$$

we have to calculate $A^2(1,3)$, $A^2(1,4)$, $A^2(3,1)$, $A^2(3,4)$, $A^2(4,1)$, $A^2(4,3)$

Now,

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

$= \min\{\infty, 5+3\} = 8$

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

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

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

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

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

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

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

$= \min\{3,\ 8+4\} = 3$

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

$= \min\{\infty,\ 8+3\} = 11$

$$\therefore A^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 8 & 7 \\ 4 & 0 & 3 & 11 \\ 4 & 9 & 0 & 2 \\ 3 & 8 & 11 & 0 \end{bmatrix} \end{array}$$

Step-4: Now construct $A^3$ matrix without changing 3rd row & 3rd column & diagonal values from $A^2$

$$A^3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & & 8 & \\ & 0 & 3 & \\ 4 & 9 & 0 & 2 \\ & & 11 & 0 \end{bmatrix} \end{array}$$

we have to calculate $A^3(1,2)$, $A^3(1,4)$, $A^3(2,1)$, $A^3(2,4)$, $A^3(4,1)$, $A^3(4,2)$

Now,

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

$= \min\{5,\ 8+9\} = 5$

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

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

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

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

$A^3(2,4) = \min\{A^2(2,4),\ A^2(2,3) + A^2(3,4)\}$

$= \min\{11,\ 3+2\} = 5$

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

$= \min\{3,\ 11+4\} = 3$

$A^3(4,2) = \min\{A^2(4,2),\ A^2(4,3) + A^2(3,2)\}$.

$= \min\{8,\ 11+9\} = 8$

$$\therefore A^3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 8 & 7 \\ 4 & 0 & 3 & 5 \\ 4 & 9 & 0 & 2 \\ 3 & 8 & 11 & 0 \end{bmatrix} \end{array}$$

<u>step-5</u>: Now construct $A^4$ matrix without changing $4^{th}$ row & $4^{th}$ column and diagonal values from $A^3$.

$$A^4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & & & 7 \\ & 0 & & 5 \\ & & 0 & 2 \\ 3 & 8 & 11 & 0 \end{bmatrix} \end{array}$$

we have to calculate $A^4(1,2)$, $A^4(1,3)$, $A^4(2,1)$, $A^4(2,3)$, $A^4(3,1)$, $A^4(3,2)$

Now,

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

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

$A^4(1,3) = \min\{A^3(1,3),\ A^3(1,4) + A^3(4,3)\}$

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

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

$= \min\{4,\ 5+3\} = 4$

$A^4(2,3) = \min\{A^3(2,3),\ A^3(2,4) + A^3(4,3)\}$

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

$A^4(3,1) = \min\{A^3(3,1),\ A^3(3,4) + A^3(4,1)\}$

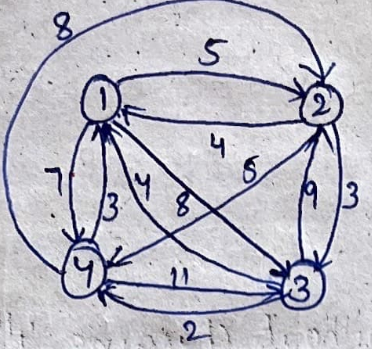
$= \min\{4,\ 2+3\} = 4$

$A^4(3,2) = \min\{A^3(3,2),\ A^3(3,4) + A^3(4,2)\}$

$= \min\{9,\ 2+8\} = 9$

$$\therefore A^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 8 & 7 \\ 2 & 4 & 0 & 3 & 5 \\ 3 & 4 & 9 & 0 & 2 \\ 4 & 3 & 8 & 11 & 0 \end{array}$$

Graph:




22/1/24

4. ~~Travelling Salesperson Problem~~

Job Sequencing with Dead Lines (Greedy Method)

→ In this problem we have to execute jobs within the deadline by getting a total profit should be maximized.

→ The Jobs are executed within its dead line by assigning time sequences for the job.

→ The Optimal solution for this problem is getting whenever we select the job which is having maximum profit.

eg:1. Find optimal solution for the following jobs using with its deadlines.

| Jobs | $J_1$ | $J_2$ | $J_3$ | $J_4$ |
|---|---|---|---|---|
| deadlines | 2 | 3 | 1 | 2 |
| Profit | 15 | 10 | 5 | 20 |

Sol) maximum time slot = 3hrs

0 $\underline{\quad J_1 \quad}$ 1 $\underline{\quad J_4 \quad}$ 2 $\underline{\quad J_2 \quad}$ 3

| Selected Job | Sequence | Series | Profit |
|---|---|---|---|
| $J_4$ | [1, 2] | $J_4$ | 20 |
| $J_1$ | [0, 1] [1, 2] | $J_1, J_4$ | 15+20 |
| $J_2$ | [0, 1] [1, 2], [2, 3] | $J_1, J_4, J_2$ | 15+20+10 |

~~$J_3$~~ $[0,1][1,2][2,3]$ $J_1, J_4, J_2$ 45

1. The optimal solution for executing jobs using deadline is 45 and the Job sequence is $J_1, J_4, J_2$

2. Find optimal solution for the following jobs using Job sequencing with deadline

| Jobs | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | $J_6$ |
|---|---|---|---|---|---|---|
| deadline | 2 | 3 | 4 | 1 | 2 | 3 |
| Profit | 10 | 5 | 10 | 15 | 20 | 20 |

Sol) maximum time slot = 4hrs

0 $\underline{J_4}$ 1 $\underline{J_5}$ 2 $\underline{J_6}$ 3 $\underline{J_3}$ 4

| Selected Job | Sequence | Series | Profit |
|---|---|---|---|
| $J_6$ | [2,3] | $J_6$ | 20 |
| $J_5$ | [1,2],[2,3] | $J_5, J_6$ | 40 |
| $J_4$ | [0,1],[1,2],[2,3] | $J_4, J_5, J_6$ | 55 |
| $J_3$ | [0,1],[1,2],[2,3],[3,4] | $J_4, J_5, J_6, J_3$ | 65 |
| ✗ $J_1$ | [0,1],[1,2],[2,3],[3,4] | $J_4, J_5, J_6, J_3$ | 65 |
| ✗ $J_2$ | [0,1],[1,2],[2,3],[3,4] | $J_4, J_5, J_6, J_3$ | 65 |

∴ The optimal solution for executing jobs using deadline is 65. and the Job sequence is $J_4, J_5, J_6, J_3$

5/2/24

4. Travelling Salesperson Problem:

→ In Travelling Salesperson problem, Salesperson starts from one vertex consider it as a starting vertex, Salesperson have to visit all the remaining vertices in a graph with min cost and reach to starting vertex.

→ In Travelling Salesperson problem set 'S' contains the no.of vertices we have to visit, if there is no vertex to visit then $|S|=0$, at Previous level $|S|=1$ and at first level $|S|=n-1$.

→ For solving Travelling Salesperson problem we are using.

$$G(i, S) = \min\{C_{ij} + G(j, \{S\} - j)\}$$

Example: Find optimal solution for the following matrix using Travelling salesperson.

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 5 & 10 & 9 \\ 2 & 13 & 0 & 8 & 12 \\ 3 & 9 & 6 & 0 & 14 \\ 4 & 11 & 15 & 5 & 0 \end{array}$$

Sol) Step-1: $|S|=0$

At last level there is no vertex to visit, we have to find $G(2,\emptyset)$, $G(3,\emptyset)$ and $G(4,\emptyset)$.

$G(2,\emptyset) = c(2,1) = 13$

$G(3,\emptyset) = c(3,1) = 9$

$G(4,\emptyset) = c(4,1) = 11$

Step-2: $|S|=1$

we have to find $G(2,3)$, $G(2,4)$, $G(3,2)$, $G(3,4)$, $G(4,2)$, $G(4,3)$

$G(2,3) = \min\{C_{ij} + G(j, \{S\} - j\}$

$= \min\{C_{23} + G(3, \{3\} - 3)\} = \min\{8 + G(3,\emptyset)\} = \min\{8+9\} = 17$

$G(2,4) = \min\{C_{24} + G(4, \{4\} - 4)\}$

$= \min\{12 + G(4,\emptyset)\} = \min\{12+11\} = 23$

$G(3,2) = \min\{C_{32} + G(2, \{2\} - 2)\}$

$= \min\{6 + G(2,\emptyset)\} = \min\{6+13\} = 19$

$G(3,4) = \min\{C_{34} + G(4, \{4\} - 4)\}$

$= \min\{14 + G(4,\emptyset)\} = \min(14+11) = 25$

$G(4,2) = \min\{C_{42} + G(2, \{2\} - 2)\}$

$= \min\{15 + G(2, \phi)\} = \min\{15 + 13\} = 28$

$G(4,3) = \min\{C_{43} + G(3, \{3\} - 3)\}$

$= \min\{5 + G(3, \phi)\} = 14$

Step-3: $|S| = 2$

we have to find $G(2,\{3,4\})$, $G(3, \{2,4\})$, $G(4, \{2,3\})$

$G(2, \{3,4\}) = \min\{C_{ij} + G(j, \{S\} - j)\}$

$= \min\begin{Bmatrix} C_{23} + G(3, \{3,4\} - 3) \\ C_{24} + G(4, \{3,4\} - 4) \end{Bmatrix}$

$= \min\{8 + G(3,4), 12 + G(4,3)\}$

$= \min\{8+25, 12+14\}$

$= \min\{33, 26\} = 26$

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

$= \min\{6 + G(2,4), 14 + G(4,2)\}$

$= \min\{6+23, 14+28\} = 29$

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

$= \min\{\cancel{15} + G(2,3), \cancel{0}5 + G(3,2)\}$

$= \min\{\cancel{15} + 17, 5 + 19\} = 24$

Step-4: $|S| = 3$

we have to find $G(1, \{2,3,4\})$

$G(1, \{2,3,4\}) = \min\{C_{ij} + G(j, \{S\} - j)\}$

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

$= \min\{5 + G(2, \{3,4\}), 10 + G(3, \{2,4\}), 9 + G(4, \{2,3\})\}$

$= \min\{5+26, 10+29, 9+24\}$

$= 31$

Path is 1-2-4-3-1

The salesperson travelled in the above path for getting an optimal solution.

14/2/24

5. Multi stage Graphs

→ This problem can be solved by using Dynamic Programming - Tabulation method.

→ In this problem we are having stages and at every stage we have to calculate cost of every vertex, the vertex which produces the minimum value will consider as a visited vertex.

→ This algorithm will works based on backward method because the cost of last vertex is produced first

→ Cost of vertices is finding by using

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

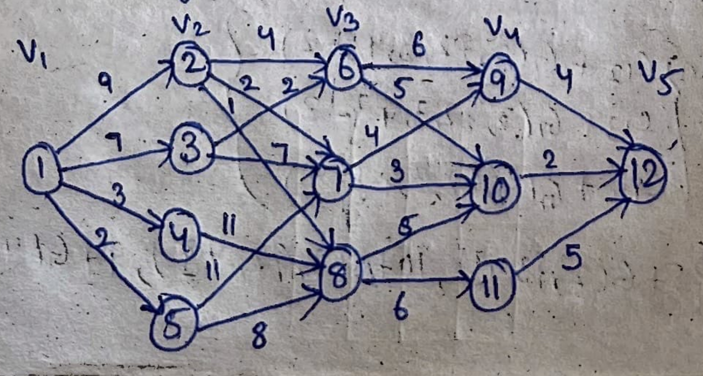
where, $i$ is stage number

$j$ is vertex

$c(j, l)$ is edge cost

$\text{cost}(i+1, l)$ - Cost of vertex in table

eg: Find Optimal solution for the following directed graph using Multistage.



Step-1: The cost of ending vertex is 0 and there is no vertices to visit.

i.e., $cost(5, 12) = 0$

| vertex | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cost | | 7 | 9 | 18 | | 7 | 5 | 7 | 4 | 2 | 5 | 0 |
| d | | 7 | 6 | 8 | | 10 | 10 | 10 | 12 | 12 | 12 | - |

stage-4

$cost(4,9) = \min\{c(9,12) + cost(4+1, 12)\} = \min\{4+0\} = 4$

$cost(4,10) = \min\{c(10,12) + cost(4+1, 12)\} = \min\{2+0\} = 2$

$cost(4,11) = \min\{c(11,12) + cost(4+1, 12)\} = \min\{5+0\} = 5$

stage-3

$cost(3,6) = \min\{c(6,9) + cost(4,9), c(6,10) + cost(4,10)\}$

$= \min\{6+4, 5+2\} = 7$

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

$= \min\{4+4, 3+2\} = 5$

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

$= \min\{5+2, 6+5\} = 7$

stage-2

$cost(2,2) = \min\{c(2,6) + cost(3,6), c(2,7) + cost(3,7),$

$c(2,8) + cost(3,8)\}$

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

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

$= \min\{2+7, 7+5\} = 9$

$cost(2,4) = \min\{c(4,8) + cost(3,8)\} = \min\{11+7\} = 18$