COURSE NAME: ANALYSIS OF ALGORITHM

II YEAR/ IV SEMESTER

UNIT – III

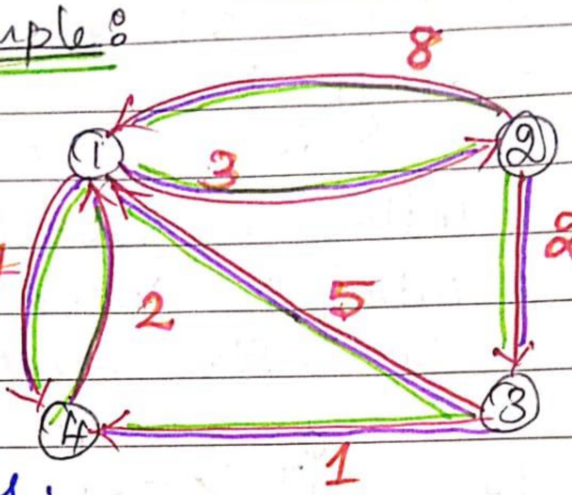
DYNAMIC PROGRAMMING

Topic

Dynamic Programming: Floyd Algorithm

UNIT - III Dynammic Programming
All Pairs Shortest path [Floyd-Warshall Algorithm] (ABDUL BARI Video)

Example:



Step 1:

$$A_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & \infty \\ 5 & \infty & 0 & 1 \\ 2 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

Starting from source vertex '1'.)

(i) Put '0' for all self loops

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

(ii) Put '∝' for no direct edges, to reach destination vertex.

(i) $1 \to 3$
(ii) $2 \to 4$
(iii) $3 \to 2$
(iv) $4 \to 2$
(v) $4 \to 3$

All are '∝' since no direct edges are there to reach destination vertex from source vertex.

Step 2: Now consider '1' to be intermediate between source & destination vertices

$$A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & \propto & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & \propto & 0 \end{array}$$

(i) put '0' for all self loops.

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

(ii) Fill the same values of previous graph for 1st row & 1st column, since we compute for A^1.

1st Row = | 0 | 3 | ∝ | 7 |

1st column = | 0 | 8 | 5 | 2 |

I Consider '1' as intermediate vertex

Compute 2,3 as 2-1 and 1-3

(iii) * $A^0[2,3] = A^0[2,1] + A^0[1,3]$

$2 < 8 + \infty$

$\boxed{\text{min is } 2}$ So $2 \to 3 = 2$

→ check in A^0 matrix

* $A^0[2,4] = A^0[2,1] + A^0[1,4]$

$\infty = 8 + 7$

$\infty > 15$

$\boxed{\text{min is } 15}$; So $2 \to 4 = 15$

Check in → A^0 matrix

* $A^0[3,2] = A^0[3,1] + A^0[1,2]$

$\infty = 5 + 3$

$\infty > 8$

$\boxed{\text{min is } 8}$; So $3 \to 2 = 8$

* $A^0[3,4] = A^0[3,1] + A^0[1,4]$

$1 = 5 + 7$

$1 < 12$

$\boxed{\text{min is } 1}$; So $3 \to 4 = 1$

* $A^0[4,2] = A^0[4,1] + A^0[1,2]$

$\infty = 2 + 3$

$\infty > 5$

$\boxed{\text{min is } 5}$; So $4 \to 2 = 5$

$A^0[4,3]$ = No path ; So remains ∞

II. Consider 2 as Intermediate vertex

Now fill 2nd row / 2nd column with same values in previous A_1 matrix. Fill self loop as '0' $1\text{-}1$, $2\text{-}2$, $3\text{-}3$, $4\text{-}4$

$$A_2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

(i) $A'[1,3] = A'[1,2] + A'[2,3]$

$\infty = 3 + 2$

$\infty > 5$ $\boxed{\min = 5}$; $1 \rightarrow 3 = 5$

(ii) $A'[1,4] = A'[1,2] + A'[2,4]$

$7 = 3 + 15$

$7 < 18$ $\boxed{\min = 7}$; $1 \rightarrow 4 = 7$

(iii) $A'[3,1] = A'[3,2] + A'[2,1]$

$5 = 8 + 8$

$5 < 16$ $\boxed{\min = 5}$; $3 \rightarrow 1 = 5$

(iv) $A'(3,4) = A'(3,2) + A'(2,4)$

$1 = 8 + 15$

$1 < 23$ $\boxed{min = 1}$; $3 \to 4 =$

(v) $A'(4,1) = A'(4,2) + A'(2,1)$

$2 = 5 + 8$

$2 < 13$ $\boxed{min = 2}$; $4 \to 1 =$

(vi) $A'(4,3) = A'(4,2) + A'(2,3)$

$\infty = 5 + 2$

$\infty > 7$ $\boxed{min = 7}$; $4 \to 3 =$

III Consider '3' as intermediate vertex

$$A_3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 6 \\ 2 & 7 & 0 & 2 & 3 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

Fill self loop as '0' & put the 3rd row & 3rd column as same value in A_2 matrix

'3' intermediate

check in A2 matrix →

(i) $A^2(1,2) = A[1,3] + A[3,2]$

$3 = 5 + 8$

$3 < 13$ $\boxed{min = 3}$; $1 \rightarrow 2 = 3$

(ii) $A^2(1,4) = A[1,3] + A[3,4]$

$7 = 5 + 1$

$7 > 6$ $\boxed{min = 6}$; $1 \rightarrow 4 = 6$

(iii) $A^2(2,1) = A[2,3] + A[3,1]$

$8 = 2 + 5$

$8 > 7$ $\boxed{min = 7}$; $2 \rightarrow 1 = 7$

(iv) $A^2(2,4) = A[2,3] + A[3,4]$

$15 = 2 + 1$

$15 > 3$ $\boxed{min = 3}$; $2 \rightarrow 4 = 3$

(v) $A^2(4,1) = A[4,3] + A[3,1]$

$2 = 7 + 5$

$2 < 12$ $\boxed{min = 2}$; $4 \rightarrow 1 = 2$

(vi) $A^2(4,2) = A[4,3] + A[3,2]$

$5 = 7 + 8$

← values in A^3 matrix $\boxed{5 < 15}$ $\boxed{min = 5}$; $4 \rightarrow 2 = 5$

IV. Consider '4' as intermediate Vertex.

$$A_4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 6 \\ 2 & 5 & 0 & 2 & 3 \\ 3 & 3 & 6 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

(1) $A^3[1,2] = A^3[1,4] + A^3[4,2]$
$3 = 6 + 5$
$3 < 11$ $\boxed{min = 3}$; $1 \to 2 = 3$

(2) $A^3[1,3] = A^3[1,4] + A^3[4,3]$
$5 = 6 + 7$
$5 \leq 13$ $\boxed{min\ 5}$; $1 \to 3 = 5$

(3) $A^3[2,1] = A^3[2,4] + A^3[4,1]$
$7 = 3 + 2$
$7 \geq 5$ $\boxed{min = 5}$; $2 \to 1 = 5$

(4) $A^3[2,3] = A^3[2,4] + A^3[4,3]$
$2 = 3 + 7$
$2 \leq 10$ $\boxed{min = 2}$; $2 \to 3 = 2$

(5) $A^3[3,1] = A^3[3,4] + A^3[4,1]$
$5 = 1 + 2$
$5 > 3$ $\boxed{min = 3}$; $3 \to 1 = 3$

Date
Page

$A^3[3,2] = A^3[3,4] + A^3[4,2]$

$8 = 1 + 5$

$8 > 6$ $\boxed{min = 6}$; $3 \to 2 = 6$

A^3 This is shortest path finally found.

Formula:

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

Algorithm floyd ( )

```
for(k=1, k<=n; k++) -> // generating matrix
{
for (i=1; i<=n; i++)      // In example 4 matrix
{
for(j=1; j<=n; j++)
{
A[i,j] = min (A[i,j], A[i,k]+A[k,j])
}
}
}
```

Time complexity

Analysis: 3 Nested for loops, we can write as.

$$C(n) = \sum_{k=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} 1$$

$$C(n) = \sum_{k=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} 1$$

By formula:

$$\sum_{i=1}^{u} 1 = (\text{upper limit } (u) - \text{lower limit } (l)$$

$$T(n) = \sum_{k=1}^{n} \sum_{i=1}^{n} (n - \cancel{1} + \cancel{1})$$

$$= \sum_{k=1}^{n} (n - \cancel{1} + \cancel{1}) * n$$

$$= \sum_{k=1}^{n} n * n$$

$$T(n) = \sum_{k=1}^{n} n^2$$

$$= (n - \cancel{1} + \cancel{1}) * n^2$$

$$\boxed{\therefore T(n) = n^3} \quad \boxed{T(n) = \Theta(n^3)}$$