UNIT III

Dynamic Programming

Dynamic programming is a Algorithmic Designed method that can be used when the solution to the problem can be viewed as a result of sequence of decision.

Principle of optimality:

The principle of optimality states that an optimal sequence of decision has the property that whatever the initial state and the decision are, The remaining decision must constitude the optimal decision sequence with regard to the state resulting from the first decision

All Pair shortest path (Floyd's Algorithm)

Let $G = (V, E)$ be a directed graph with n vertices.

Let cost be a cost adjacency Matrix for G such that $cost[i, i] = 0$, $1 \leq i \leq n$

cost [i, j] is a length or cost of edge $\langle i, j \rangle$ if $\langle i, j \rangle \in E$ of G

cost [i, j] = ∞, if $\langle i, j \rangle$ does not belongs to ($\notin$) E of G

The All pair shortest path problem is to determine matrix A, A[i, j] is the length of shortest path from i to j

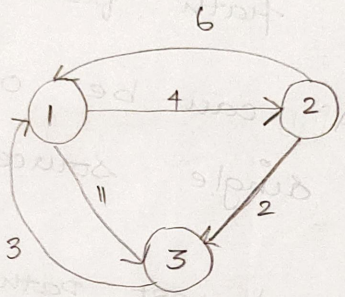
The matrix A[i,j] can be obtained by solving n single source problems.

Algorithm for All pair shortest path :
(Floyd's Algorithm)

```
Algorithm Allpaths (cost, A, n)
// cost [1:n, 1:n] - cost adjacency matrix of
// graph G with n vertices
// A[i, j] - cost of shortest path from vertex
// i to vertex j.          1 ≤ i ≤ n
// cost [i, i] = 0 ;
{
    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]);
}
```

consider the directed graph G



cost (i, j)	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

A[i,j] = cost [i, j]

A^0 (i, j)	1	2	3
1	0	4	11
2	6	0	2
3	3	∞	0

intermediate vertex	<i,j>	$A^0[i,j]$	A[i,k] + A[k,j]	min (A[i,j], A[i,k]+A[k,j])
	1,1	0	(1,1)+(1,1) = 0+0 = 0	0
	1,2	4	(1,1)+(1,2) = 0+4 = 4	4
	1,3	11	(1,1)+(1,3) = 0+11 = 11	11
	2,1	6	(2,1)+(1,1) = 6+0 = 6	6
	2,2	0	(2,1)+(1,2) = 6+4 = 10	10
K=1	2,3	2	(2,1)+(1,3) = 6+11 = 17	2
	3,1	3	(3,1)+(1,1) = 3+0 = 3	3
	3,2	∞	(3,1)+(1,2) = 3+4 = 7	7
	3,3	0	(3,1)+(1,3) = 3+11 = 14	0

$A^1[i,j]$	1	2	3
1	0	4	11
2	6	10	2
3	3	7	0

intermediate vertex	<i,j>	$A^1[i,j]$	A[i,k] + A[k,j]	min (A[i,j], A[i,k]+A[k,j])
	1,1	0	(1,2)+(2,1) = 4+6 = 10	0
	1,2	4	(1,2)+(2,2) = 4+0 = 4	4
	1,3	11	(1,2)+(2,3) = 4+2 = 6	6
	2,1	6	(2,2)+(2,1) = 0+6 = 6	6
	2,2	0	(2,2)+(2,2) = 0+0 = 0	0
k=2	2,3	2	(2,2)+(2,3) = 0+2 = 2	2
	3,1	3	(3,2)+(2,1) = 7+6 = 13	3
	3,2	7	(3,2)+(2,2) = 7+0 = 7	7
	3,3	0	(3,2)+(2,3) = 7+2 = 9	9

$A^2[i,j]$	1	2	3
1	0	4	6
2	6	0	2
3	3	7	0

Intermediate Vertex	$<i,j>$	$A[i,j]$	$A[i,k] + A[k,j]$	$\min(A[i,j], A[i,k]+A[k,j])$
	1, 1	0	(1,3) + (3,1) = 6+3 = 9	0
	1, 2	4	(1,3) + (3,2) = 6+7 = 13	4
	1, 3	6	(1,3) + (3,3) = 6+0 = 6	6
	2, 1	6	(2,3) + (3,1) = 2+3 = 5	5
K = 3	2, 2	0	(2,3) + (3,2) = 2+7 = 9	0
	2, 3	2	(2,3) + (3,3) = 2+0 = 2	2
	3, 1	3	(3,3) + (3,1) = 0+3 = 3	3
	3, 2	7	(3,3) + (3,2) = 0+7 = 7	7
	3, 3	0	(3,3) + (3,3) = 0+0 = 0	0

$A^3[i,j]$	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0

source vertex	destination Vertex	Shortest Path traveled	Shortest distance from vertex i to vertex j
{1}	{1}	-	0
{1}	{2}	1-2	4
{1}	{3}	1-2-3	6
{2}	{1}	2-3-1	5
{2}	{2}	-	0
{2}	{3}	2-3	2
{3}	{1}	3-1	3
{3}	{2}	3-1-2	7
{3}	{3}	-	0