Design and Analysis of Algorithms

Lecture-36

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

Floyd’s Warshall Algorithm for shortest path

- This algorithm is used to solve the all-pairs shortest-paths problem on a directed graph G =(V,E).
- This algorithm is based on the dynamic programming approach.
- Let $d^{(k)}_{ij}$ be the weight of a shortest path from vertex i to vertex j for which all intermediate vertices are in the set {1, 2, ………, k}.
- When k = 0, a path from vertex i to vertex j with no intermediate vertex numbered higher than 0 has no intermediate vertices at all. Such a path has at most one edge, and hence $d^{(0)}_{ij} = w_{ij}$.

Floyd's Warshall Algorithm for shortest path

Recursive formula to compute $d^{(k)}_{ij}$

We define $d^{(k)}_{ij}$ as the following:-

$$d^{(k)}_{ij} = w_{ij} \quad \text{if } k = 0$$

$$= \min\{ d^{(k-1)}_{ij} , d^{(k-1)}_{ik} + d^{(k-1)}_{kj} \} \quad \text{if } k \geq 1.$$

- Because for any path, all intermediate vertices are in the set {1, 2, ..., n}, therefore the matrix $D^{(n)} = (d^{(n)}_{ij})$ gives the final answer.

Floyd's Warshall Algorithm for shortest path

Constructing a shortest path

- We define $\pi^{(k)}_{ij}$ as the predecessor of vertex j on a shortest path from vertex i with all intermediate vertices in the set {1, 2, ….., k}.
- When **k = 0**, a shortest path from i to j has no intermediate vertices at all. Therefore,

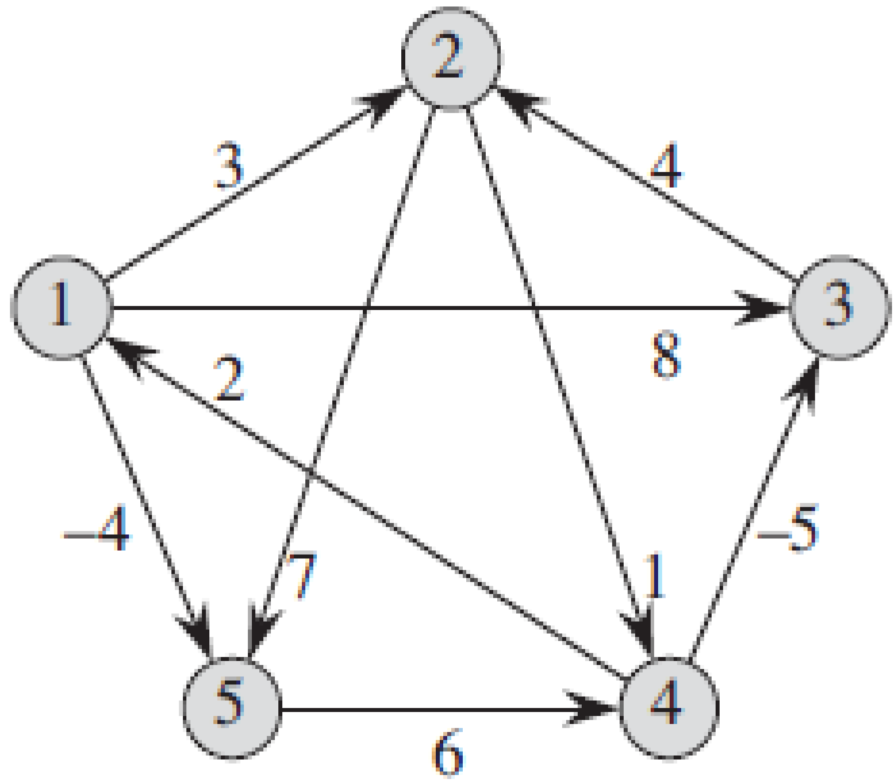
$$\pi^{(0)}_{ij} = \text{NIL} \qquad \text{if } i=j \text{ or } w_{ij} = \infty$$
$$= i \qquad \text{if } i \neq j \text{ and } w_{ij} < \infty.$$

- **For k ≥ 1.**

$$\pi^{(k)}_{ij} = \pi^{(k-1)}_{ij}, \quad \text{if } d^{(k-1)}_{ij} \leq d^{(k-1)}_{ik} + d^{(k-1)}_{kj}$$
$$= \pi^{(k-1)}_{kj}, \quad \text{if } d^{(k-1)}_{ij} > d^{(k-1)}_{ik} + d^{(k-1)}_{kj}$$

Floyd's Warshall Algorithm for shortest path

Example: Apply the Floyd's Warshall algorithm in the following graph :-



Floyd's Warshall Algorithm for shortest path

Solution: Weighted matrix of this graph is the following:-

$$W = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

Therefore,

$$D^{(0)} = W = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(0)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

Floyd's Warshall Algorithm for shortest path

Here, we have to calculate $D^{(5)}$ and $\Pi^{(5)}$. It is calculated as following:-

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(1)} = \begin{pmatrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & 1 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad \Pi^{(2)} = \begin{pmatrix} \text{NIL} & 1 & 1 & 2 & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & 2 & 2 \\ 4 & 1 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

Floyd's Warshall Algorithm for shortest path

•

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \Pi^{(3)} = \begin{pmatrix} \text{NIL} & 1 & 1 & 2 & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & 2 & 2 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \qquad \Pi^{(4)} = \begin{pmatrix} \text{NIL} & 1 & 4 & 2 & 1 \\ 4 & \text{NIL} & 4 & 2 & 1 \\ 4 & 3 & \text{NIL} & 2 & 1 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ 4 & 3 & 4 & 5 & \text{NIL} \end{pmatrix}$$

Floyd's Warshall Algorithm for shortest path

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \qquad \Pi^{(5)} = \begin{pmatrix} \text{NIL} & 3 & 4 & 5 & 1 \\ 4 & \text{NIL} & 4 & 2 & 1 \\ 4 & 3 & \text{NIL} & 2 & 1 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ 4 & 3 & 4 & 5 & \text{NIL} \end{pmatrix}$$

- Matrix $D^{(5)}$ stores the shortest distance between each pair of vertices.
- Matrix $\Pi^{(5)}$ is used to find shortest path between each pair of vertices.

Floyd's Warshall Algorithm for shortest path

```
Floyd-Warshall(W, n)
    D(0) = W
    for i←1 to n
        for j← 1 to n
            if (i=j or w_ij=∞)
                π(0)_ij = Nil
            if (i≠j and w_ij< ∞)
                π(0)_ij = i
    for k←1 to n
        for i←1 to n
            for j← 1 to n
                if (d(k-1)_ij ≤ d(k-1)_ik + d(k-1)_kj )
                    d(k)_ij = d(k-1)_ij
                    π(k)_ij = π(k-1)_ij
                else
                    d(k)_ij = d(k-1)_ik + d(k-1)_kj
                    π(k)_ij = π(k-1)_kj
```

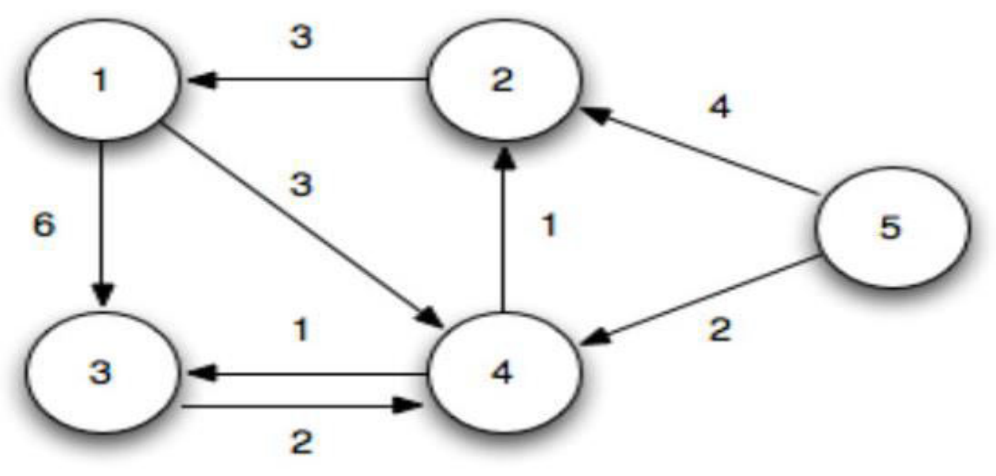
Time complexity of this algorithm is $\theta(n^3)$.

AKTU Examination Questions

1. Define principal of optimality. When and how dynamic programming is applicable.
2. Give Floyd Warshall algorithm to find the shortest path for all pairs of vertices in a graph. Give the complexity of the algorithm. Explain with example.
3. Difference between Greedy Technique and Dynamic programming.
4. Write down an algorithm to compute Longest Common Subsequence (LCS) of two given strings and analyze its time complexity.
5. What is dynamic programming? How is this approach different from recursion? Explain with example.
6. Solve the following 0/1 knapsack problem using dynamic programming. P=[11,21,31,33] w=[2,11,22,15] c=40, n=4.

AKTU Examination Questions

7. Define Floyd Warshall Algorithm for all pair shortest path and apply the same on following graph:



7. Define the terms—LCS, Matrix Chain multiplication & Bellman-Ford algorithm.
8. Explain the Floyd Warshall algorithm with an example.
9. Find an optimal parenthesization of a matrix chain product whose sequence of dimensions is {10, 5, 3, 12, 6}.