CSE 2202 : Algorithm Analysis and Design Laboratory

Assignment no : 01

Topic : Finding K shortest paths from a source to a destination in a directed , positive weighted graph using Yen's Algorithm .

To:

Mr. Md. Mehrab Hossain Opi
Lecturer,

Department of Computer Science and Engineering,

Khulna University of Engineering and Technology,Khulna

From:

Dadhichi Sarker Shayon

B2

2207118

2nd Year 2nd term

24/10/2025

Overview of the problem:

Implementing a program to compute the K shortest simple paths (paths with no repeated nodes) from a source node s to a destination node t in a directed graph with non negative edge weights,for knowledge of shortest path algorithms and introduction to advanced graph traversal techniques for finding alternate shortest paths.

Given,
A directed weighted graph G = (V, E) with edge weights
A source node s
A destination node t
An integer k >= 1

Task

Computing and returning the K shortest simple paths from s to t , ordered by total path cost (ascending).Each path must be simple, meaning no node appears more than once in any path.

Implementation Requirements

Using Dijkstra's algorithm or any single source shortest path algorithm as a subroutine to find the shortest path in the graph.Returning the K paths in order of increasing total cost.Handling cases where fewer than K distinct paths exist.

Example Output:

Path 1: A-->B-->C-->D Cost: 8
Path 2: A --> C --> D Cost: 9
Path 3: A --> B --> D Cost: 11

Explanation of Yen's Algorithm:

Yen's Algorithm is a method used to find the K shortest simple paths between a source node and a destination node in a weighted, directed graph. Unlike Dijkstra's Algorithm, which finds only the single shortest path, Yen's Algorithm iteratively generates multiple alternative paths, ensuring that each path is simple contains no cycles or repeated nodes and ordered by total cost.
The idea of Yen's Algorithm is to build on the shortest path solution and systematically explore deviations from previously found paths to generate the next best paths. It does this through the concepts of root paths, spur paths, and graph modification.

Finding the First Shortest Path

Running Dijkstra's Algorithm or another single source shortest path algorithm on the original graph to find the shortest path from the source s to the destination t.This path is stored as the first shortest path P1 in a container A of discovered paths.

Initializing Candidate Paths

Maintaining a set B to store unique candidate paths along with their total cost.This set ensures that the next path chosen is always the one with the lowest total cost among all candidates.

Iteratively Finding the Next Shortest Path

For each upcoming path (from 2 to K) selecting a Previous Path for Deviation Taking the last path found and iterating through each node in the path (except the last one).Each node in this path is considered a spur node, which is the point where a new alternative path will diverge from the existing path.

Splitting into Root Path and Spur Path

Root Path: The segment of last path from the source up to the spur node.

Spur Path: The path from the spur node to the destination, which will be recalculated. Modifying a duplicate Graph to Avoid Repetition.

Temporarily removing edges that would recreate previously discovered paths with the same root path

This ensures that the new path is unique and loop free.Specifically Removing edges from the spur node that were used in prior paths sharing the same root path.Removing all nodes in the root path except the spur node to prevent cycles.

Finding the Spur Path

Running Dijkstra's Algorithm on the modified graph, starting from the spur node to the destination.If a valid spur path exists, combining it with the root path to form a new total path candidate.

Calculating its total cost

By summing the weights of its edges.Adding the Candidate to the set.Pushing this candidate path into the set B, keeps the set sorted by total cost.

Selecting the Next Shortest Path

Extracting the candidate path with the lowest total cost from the set B.Adding it to the list of discovered paths A as the next shortest path.

Repeating the steps until either K paths are discovered or no new candidates remain.If fewer than K paths exist, the algorithm stops after finding all available simple paths.

By generating all possible deviations at each node of previously found paths, Yen's Algorithm systematically explores alternate routes without revisiting the same paths. The use of a set guarantees that the paths are always selected in ascending order of total cost.The careful graph modification ensures that every new path is unique and loop free, which is essential for real world applications such as network routing or logistics planning.

Inputs and Outputs Used:

Input:

7 12
0 1 2
0 2 5
0 3 1
1 2 2
1 4 7
2 4 1
2 3 2
3 2 3
3 5 6
4 5 1
4 6 5
5 6 1
0 6 5

Output:

Path 1: 0 3 2 4 5 6 Cost: 7
Path 2: 0 1 2 4 5 6 Cost: 7
Path 3: 0 2 4 5 6 Cost: 8
Path 4: 0 3 5 6 Cost: 8
Path 5: 0 1 2 4 6 Cost: 10

Input :

5 5
0 1 4
0 2 5
1 2 2
1 3 4
2 3 4
0 3 5

Output:

Path 1: 0 1 3 Cost: 8
Path 2: 0 2 3 Cost: 9
Path 3: 0 1 2 3 Cost: 10