CS263 ASSIGNMENT 9

NAME:

ARCHIT AGRAWAL

ROLL NO. :

202051213

SECTION:

A

1. **Given a weighted directed graph with few negative edges, find the shortest path from the source to all other vertices in the given graph. Write the code and complexity of your algorithm.**

Algorithm

- Firstly, create an array that stores the minimum distance from source to all vertices. Update the distance of source to 0, and others to infinity.
- Run a loop for V – 1 times, where V is number of vertices, and relax each edge in every iteration i.e.

    For each edge (u, v), if distance of u from source plus the distance of edge (u,v) is lesser than the distance of v from source, then update distance of v from source with distance of u from source plus the distance of edge (u,v).

- Traverse through the edges one more time and relax the edges. If there is further relaxation in any of the edge, then there exists a negative cycle in the graph i.e a cycle whose total edges weight is negative. If no shorter path is present for any of the vertices, then we have obtained the shortest paths for each of the vertex from the source.

Note- Consider two iterations of outer loop in step 2 of the above algorithm. If the distance for any of the vertex is not updated in two consecutive iterations, then no further shorter path will occur for any vertex and we can bring our algorithm to a halt.

To implement this, we just need a counter variable inside the outer loop in step 2. This variable maintains the number of vertices that are relaxed in each iteration of outer loop. If, in a particular iteration, the counter variable is 0, we can end our algorithm as we will have shortest paths for each vertex from the source. This helps in reducing the iterations of outer loop in average cases. However, in the worst case, the complexity will remain the same.

Analysis

Since, the algorithm relaxes (in O(1) time) each edge V – 1 times, where, V is the number of vertices and E is the number of edges. The complexity of the algorithm is O(V.E).

- The first step of the algorithm takes O(V) time. Since, array of size V is initialised to infinity.
- The second step takes O(V.E) time as each edge is relaxed (V – 1) times.
- The third step takes O(E) time, as each edge is relaxed once more.

Hence,

$$T(V,E) = O(V) + O(V.E) + O(E)$$

$$T(V,E) = O(V.E)$$

In the worst case, the graph is complete and there are (V(V-1))/2 edges. Hence, the complexity of the algorithm will be:

$$T(V) = O(V^3)$$

Example