Around the world

Analysis of Algorithms

CS-232

Assignment 3

Submitted to:

Prof. Dr. Juniad Akhar

Submitted by:

Kiran Zafar   (BSCS-2019- 36)

In this era, airlines play an important role in people's lives, thousands of people each and every day travel from one city to another, or one country to another for their work, business purpose, for education, for visiting, using different airlines. Everyone wants to travel, to get exposure, to meet different people, to explore a different culture, etc. the handling of finding the shortest path between various locations is seems to be a major problem in airlines network which is low cost. In this internet world, many people also raise this issue and try to sort out and present the best solution. We also contribute to getting the best solution for this, we used Dijkstra's Algorithm.

Modern problems need modern solutions; the main problem which is faced by everyone is which route is best, low cost and highly effective for their journey. The smallest way to find the path between the two nodes on a given graph, so that the sum of the weights on its edges is minimum. Dijkstra's algorithm, we design to solve the single source for the shortest path problem. We have to give the first source node and then this algorithm performs and calculates the shortest path from the whole network. The distance is the weight between two nodes is evaluated in advance to get preciseness result.

For finding the best way to travel between the two paths, the shortest path algorithm was created. Dijkstra algorithm is used to solve this problem which follows on that side where the weight of non-negative edges is lower; it's a graph search algorithm that solves one source shortest path problem. It stops when finding the best route for a destination; it finds the S finding cost of the shortest path from the source vertex to the final vertex. For example, if vertices represent cities and the edges represent the path cost, then Dijkstra's algorithm is used to find the shortest path between one source city to another city or all other cities.

Dijkstra's algorithm is a finding the shortest path algorithm, it performs on non-negative edges weights and also on a directed graph. We can also use this for undirected graphs for that we used a breadth-first search algorithm, as we know of handle unweight and undirected graph so for weight and directed graph we have to do some changes in BFS algorithm. For an arbitrary directed graph with non-negative weight, we will come to know that **Dijkstra's algorithm is asymptotically the fastest algorithm for finding the shortest path.**

The graph is made up of two sets of nodes (the collection of all vertices) and the Edges (the collection of all edges). It also represents as

G=(V, E)

V is the set of all vertices from V_0 to V_{n-1} and E is the edges from E_1 to E_n.

 We have to solve the problem of traveling through airlines and finding the find the shortest and less costly path between the source city and the other required city or all other cities. So the node represents the city and edges represent the distance value in terms of algorithm weight between two edges. **Dijkstra's algorithm will give some initial distance values** and also modified them step by step after each iteration until to get the best route. For the first iteration, the distance will be zero as it's a starting point. After this which is directly connected to the first node which is unvisited the distance is updated between the unvisited and starting node.

Update the distance to every other node which is directly connected to our given node. And also calculate the sum of the distance between the unvisited intersection and the current intersection value, if the value is less than the current value then give that value to unvisited edges. In this way, we have the shortest path than the previously known path. And also give all other paths which are pointing to it because now we have the shortest path. Now mark the current node as a visited node and move to an unvisited intersection with the lowest distance, do this again and again until finding the shortest path between the source node and the required node.

In short, we just do 3, 4 steps for finding the shortest path.

1. Source node as a visited node or a permanent node and assign 0 costs because its starting node and distance is zero

2. Now check all nodes which are directly connected to previously visited node mean permanent node.

3. Calculate the sum of the cost of each unvisited directly connected node.

4. Check the last unvisited nodes which have the lowest cost and shortest weight then other visited nodes selected that one and make it a permanent node.

5. Repeat this until all nodes are visited and we get the shortest path.

Interesting Question:

- **Find the shortest path to visit all the cities which are given?**

- **Find the longest shortest path to visit all the cities which are given?**

- **Find the route and distance between the two cities with the minimum cost?**