**Assignment 2–PS3-[SHIPMENT PROBLEM]**

## **Problem Statement:**

Amazon Warehouse in Bangalore was expecting a freight shipment at 10 am on a Monday. However, the company responsible for the shipment mistook the destination as Amazon Development Centre, Bangalore. In order to save the extra cost of transportation back to the warehouse, the warehouse manager wants to help them find the fastest route from the DC to warehouse. He has the location map available which is given below (the weightage given is the distance in km between each location).

Need to help manager with:

* shortest route to reach Warehouse from the DC
* expected time of arrival of the shipment at the warehouse

## **Algorithm for finding shortest path:**

To find the fastest route between two points in a map (graph provided), we essentially have to find the shortest weighed path between those 2 points. This can be achieved by BFS traversal of graph choosing the edge which gives lowest weight sum from source point (node) on each loop. We can use Dijkstra’s Algorithm to find the shortest path in a weighed graph.

Below are the steps used to find shortest path:

1. Initially Mark Distance from source to source as 0 and distance to all other nodes(connected or not connected) as infinity(or None/null in programmatic format)
2. Find the neighbours of current location(or source) and calculate the total distance from source to current location
3. Store neighbour location with current location from where we reach it and total distance from source
   * i.e. neighbour location info should contain information about current location and total distance from source
4. Add the current location(or node) to visited set and pick the node with smallest(shortest) distance(not null/None) as the next location to traverse
5. Repeat from step 2 taking next location that is not visited with shortest total distance form source as the new current location until current location matches with destination or all nodes visited
6. If all nodes are visited and yet current location and destination doesn’t match implies no path exists between them
7. Total distance will be equal to shortest distance found in the last loop of successful search
8. Shortest path can be traversed back from the stored neighbours information

**Pseudo Code**

MAP: Location map in the form of graph

ShortestPath(source, destination):

distances[source] <= (source, 0) // distance from source to source as 0

current <= source

visited<=[]

while current not = destination:

visited.add(current)

for each neighbour to current:

if neighbour not in distances:

distances[neighbour] = (current, distance from neighbour to current)

if neighbour not in distances:

if distance to source from this neighbour < distance to source from other neighbour:

distances[neighbour] = (current, distance from neighbour to current)

current<= node with minDistance(distances)