**DSA**

**Assignment 01**

**Name:** Aamir Ansari  **Batch:** A **Roll no:** 01

**Assignment on Dijkstra Algorithm to find shortest path**

Dijkstra’s algorithm is used to find the length of an optimal path between two nodes in a graph. The term optimal can mean anything, shortest, cheapest, or fastest. If we start the algorithm with an initial node, then the distance of a node Y can be given as the distance from the initial node to that node.

**Algorithm for Dijkstra’s algorithm:**

Step 1: Select the source node also called the initial node

Step 2: Define an empty set N that will be used to hold nodes to which a shortest path has been found.

Step 3: Label the initial node with, and insert it into N.

Step 4: Repeat Steps 5 to 7 until the destination node is in N or there are no more labelled nodes in N.

Step 5: Consider each node that is not in N and is connected by an edge from the newly inserted node.

Step 6: (a) If the node that is not in N has no label then SET the label of the node = the label of the

newly inserted node + the length of the edge.

(b) Else if the node that is not in N was already labelled, then SET its new label = minimum

(label of newly inserted vertex + length of edge, old label)

Step 7: Pick a node not in N that has the smallest label assigned to it and add it to N.

**Example:**

Consider the graph G given alongside. Taking D as

the initial node, we execute the Dijkstra’s

algorithm.

Step 1: Set the label of D = 0 and N = {D}.

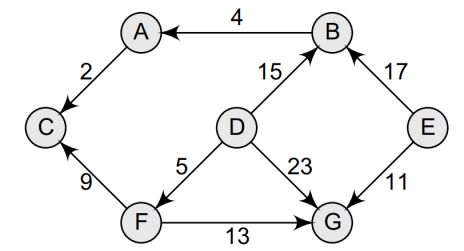
Step 2: Label of D = 0, B = 15, G = 23, and F = 5.

Therefore, N = {D, F}.

Step 3: Label of D = 0, B = 15, G has been relabelled 18 because minimum (5 + 13, 23) = 18, C has been relabeled 14 (5 + 9). Therefore, N = {D, F, C}.

Step 4: Label of D = 0, B = 15, G = 18. Therefore, N = {D, F, C, B}.

Step 5: Label of D = 0, B = 15, G = 18 and A = 19 (15 + 4). Therefore, N = {D, F, C, B, G}.

Step 6: Label of D = 0 and A = 19. Therefore, N = {D, F, C, B, G, A}