## Course Code: CT2302 Course Name: Lab-CN

## Name: Kartik Yeole Semester/Section: V-A

## Roll No: 51 Enrolment No: 19010773

# PRACTICAL NO. 4

Aim: To implement Dijkstra’s Routing algorithm using backtracking approach.

Theory: The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on. If the subnet uses datagrams internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time. If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the previously-established route.

Dijkstra’s algorithm, given by a Dutch scientist Edsger Dijkstra in 1959, is used to find the shortest path tree. This algorithm is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First). Given a graph G and a source node A, the algorithm is used to find the shortest path (one having the lowest cost) between A (source node) and every other node. Moreover, Dijkstra’s algorithm is also used for finding the costs of the shortest paths from a source node to a destination node.

Algorithm: 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.

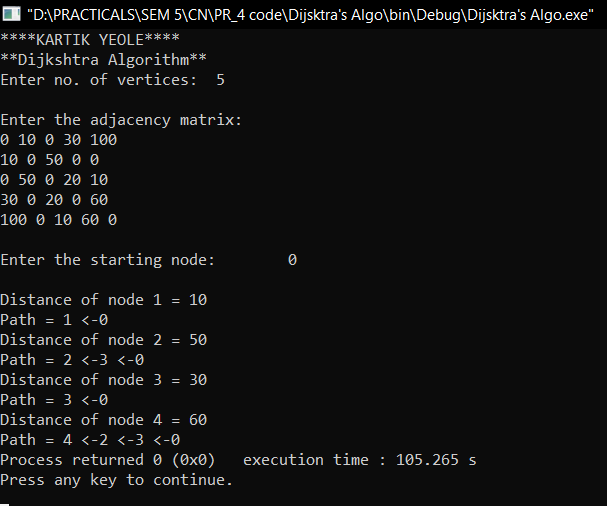
1. Select the source node also called the initial node
2. Define an empty set N that will be used to hold nodes to which a shortest path has been found.
3. Label the initial node with, and insert it into N.
4. Repeat Steps 5 to 7 until the destination node is in Nor there are no more labelled nodes in N.
5. Consider each node that is not in N and is connected by an edge from the newly inserted node.
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.

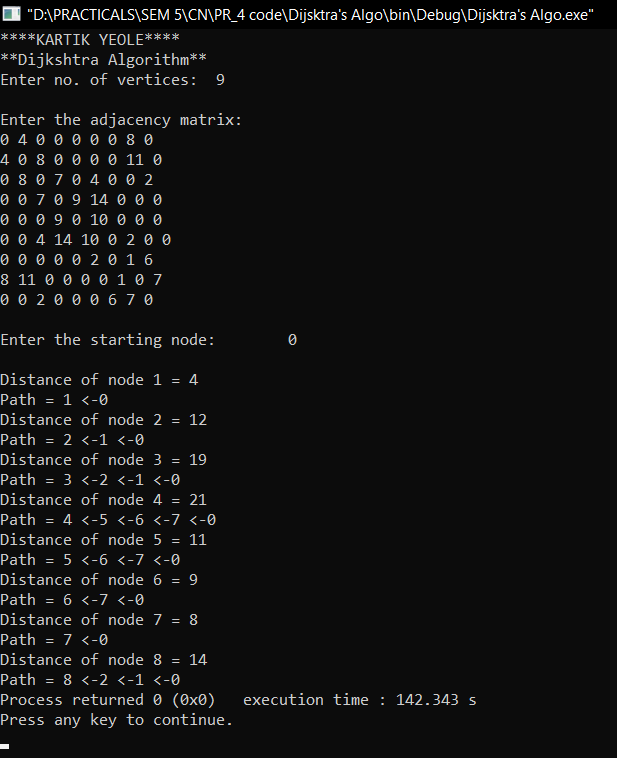
(b) Else if the node that is not in N was already labelled, then SET its new label = minimum (label of newly inserted vertex + length of edge, old label).

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

Code:

Screenshots:





Conclusion: Hence, we learned about Dijkstra’s Algorithm and applied the algorithm in a C program which accepts a Adjacency Matrix and tells us the Shortest path to all nodes from a given starting point.