# Computer Networks Lab Report

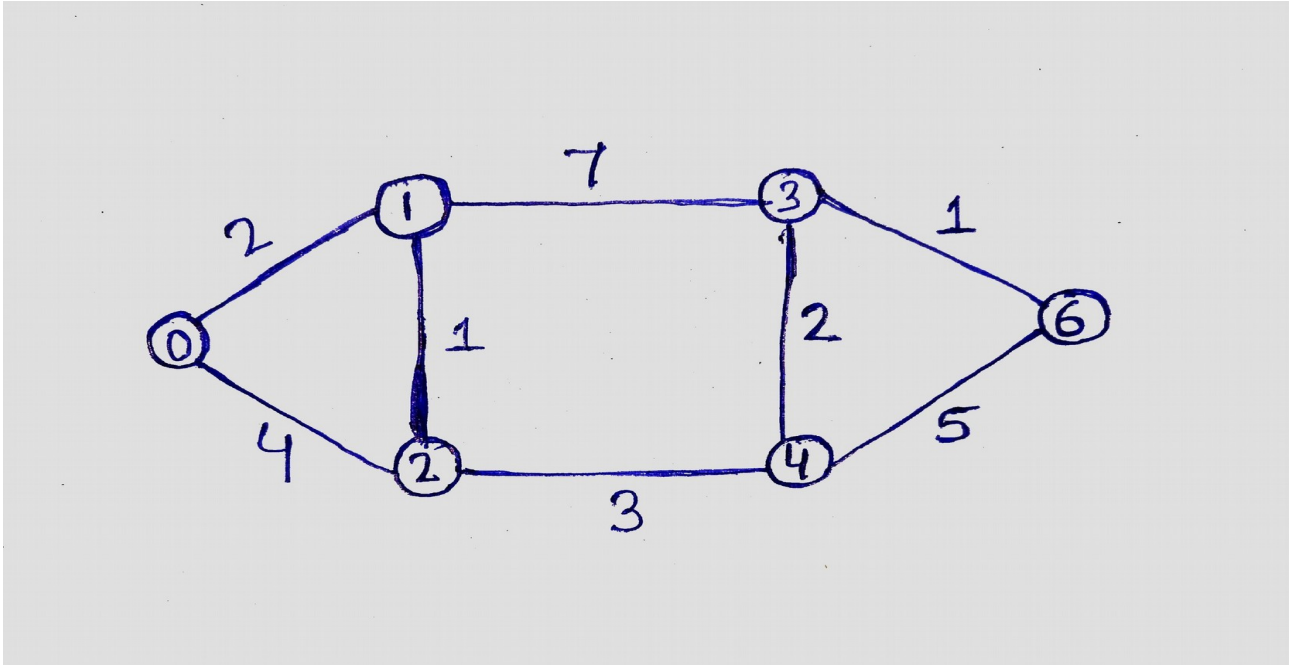
111701019 Md Talha Yaseen Khan

## Lab – 4 (Routing Algorithm)

### 1. Dijkstra's Algorithm

- **class** Graph represents Graph for which Dijkstra's Algorithm is implementing and whose cost matrix is inside input file.
- The method minDist(self, dist, Visited) finds the vertex from the set of unvisited vertex with minimum distance value.
- The method dijkstra(self) implements Dijkstra's Algorithm for a graph represented using adjacency matrix representation, this function returns a list dist containing minimum distance of all vertex from starting vertex.
- The main() function is taking 3 inputs n, src and filename from user ( n = Number of vertex, src = starting vertex and input file is containing matrix which includes cost of every edges of graph)
- We are taking 3 input (n, src and filename) from a file named compile and storing the output in file named out.
- In .zip format I attached two input files (compile0 and compile3), corresponding two output files (out0 and out3) ('0' refers starting vertex is 0 and '3' refers starting vertex is 3) and a cost.txt file which have matrix that have cost of every edges of graph.

**Below is the attached graph which I have taken as input:**




**2. Distance Vector Routing Algorithm**

- **class Graph** represents graph for which Distance Vector Routing Algorithm is implementing.
- **class RoutingTable** represents Routing Table of graph.
- The method DVR(self, node, table) is used to update the Routing Table of graph. In this initially I am finding the adjacent vertex of every given nodes and storing the adjacent vertex link cost and procedding towards the minimum cost nodes. And updating the Routing Table using adjacent node Distance Vector.
- The method table(self) is used to create the initial Routing Table with the help of cost matrix availe in the file cost.txt.
- The main() function is taking 2 inputs n and filename from user ( n = Number of vertex and input file is containing matrix which includes cost of every edges of graph). In this finction I am doing simulation and shuffling the nodes to get the cerrect final routing table.
- In .zip format I attachedninput files (compile), corresponding two output files (out.txt) and a cost.txt file which have matrix that have cost of every edges of graph.

**Below is the attached graph which I have taken as input:**