DSA Lab 15
23 Jan 2017

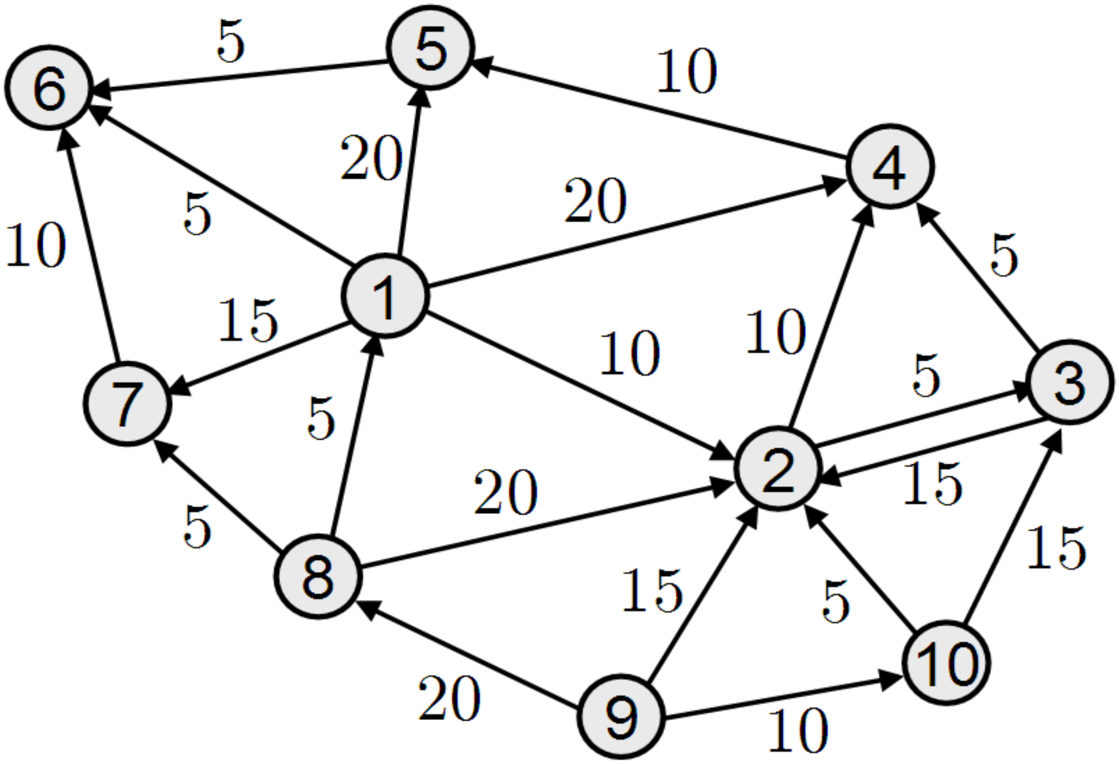
Implementation of Graph:

Implement following functions using Adjency Matric. [25]
Graph(int n); // Constructor for graph having N vertices [5]
bool isConnected(int u, int v); // return there is an edge from U to V [5]
void addEdge(int u, int v); // function to add an edge from U to V [5]
void BFS(int s); // prints BFS traversal from a given source s [10]

Implement following functions using Adjency List.[25]
Graph(int n); // Constructor for graph having N vertices [5]
bool isConnected(int u, int v); // return there is an edge from U to V [5]
void addEdge(int u, int v); // function to add an edge from U to V [5]
void BFS(int s); // prints BFS traversal from a given source s [10]

Implement following function using adjency list assuming graph is weighted and directed.[40]
void addEdge(String from, int to, String cost); // function to add an edge from U to V with cost C.
void showNeighbours(String u);// display neighbours or adjacent node for U
bool isPath(String from, String to);// return is there exist a path from source to destination.
void shortestpath(String from, String to);// find the shortest path from source to destination.\

Use the following graph for adding edges, BFS. and Shortest path


5
5
6
10
20
4
20
10
5
5
1
15
10
10
5
3
7
5
2
15
5
20
15
5
8
15
5
10
20
9
10