

# ITCS 6114

## PROJECT PROPOSAL

### Group Members:

Nisarg Antony, UNCC #800989573

Sarthak Dash Mohapatra, UNCC #800972832

### Problem Selection:

We have gone through all the problems that are posted, and finally decided to work on 'Problem Two' of the 'More project problems' document, i.e., the problem of "**Weighted Undirected Graph**".

### Implementation:

We are going to implement the above selected problem in two different programming languages, C and JAVA.

- 1) Adjacency: The program will print the Adjacency matrix of the graph based on the provided inputs.
- 2) Connectivity: In order to determine the connectivity of the graph, we propose the following two steps. First, create a Minimum Spanning Tree (MST) for the particular graph employing all the popular MST algorithms as well as the greedy strategy. Second, compare the number of nodes in the MST with the number of vertices in the original graph to determine if it's connected or not.
- 3) Minimum Spanning Tree and Minimum Spanning Forest: In order to generate the minimum spanning forest for the graph dataset, we plan on undertaking the following method using two steps. First, run multiple MST algorithms (ex. Prim's algorithm, Kruskal's algorithm, Greedy algorithm etc.) with every component of the graph as the root node. Second, generate and print the minimum spanning forest (if any).

### Expected Outcomes:

As we generate minimum spanning trees using various MST algorithms, we expect to see variations in run-time complexity. Depending on the particular graph, we may or may not get multiple MSTs. We expect to get better insight into the problem after analyzing the complexity of every algorithm and classifying it on the basis of run-time. Of all the algorithms we plan to implement, in order to find the MST for a graph, we expect Prim's algorithm to be faster, where the number of edges exceed the number of vertices.