**CSCI 311 Data Structures & Algorithms**

**LAB 4 Assignment**

**PROJECT DESIGN:-** The project is designed using 2 primary classes namely class Graph and class Vertex. The project implements BFS, DFS and Prim’s algorithms taking input from the text file.

The file graph.h/.cpp contains declarations/definitions of the members used to represent the graph and carry out the functionality. The design is as follows:-

**Class Graph:-** Includes the following:-

* Vertex \* getVertex(string &) :- function to create a new vertex, if not present already and insert into map.
* void minHeapify(int i,int heapsize):- implements minimum priority queue for execution of prims algorithm.
* void addEdge(string &,string &,int &):- Function to represent graph by adjacency list.
* void BFS(Vertex\*):- function to implement BFS algorithm.
* void DFS(vmap &):- function to implement DFS algorithm.
* void Prims(Vertex\*):- function to implement Prim’s algorithm.

**Class Vertex:- Includes the following:-**

* vector<pair<int,Vertex \*> > adj:- data member to represent adjacency list.
* void reset(map<string,Vertex \*>):- function to re-initialize the values of vertices.
* Data members:-wt,color,path to store the edge weights, vertex color and predecessor respectively.

**Class FileReader:- Includes the following:-**

* **Void readFile(char\*):-** Function to read file.

**Graph representation:-**  In the project, the graph is represented using a map and vector . Map contains list of all the vertices in the graph, while the vector pair stores the adjacent vertices along with the weight associated. While creating graph, the project uses map::find function which runs in logarithmic time. It also uses vector::push\_back function which runs in constant time. So the overall complexity for graph representation is log of size of the map.