**MILESTONE 1**

Graphs are very powerful tools that can be used to represent large amounts of data in a clear and concise matter. Any data can be represented as a graph, the possibilities are endless. With respect to this project, graphs are utilized in a map where we essentially have to traverse through different nodes. The basic objective of this assignment is to traverse through nodes with the shortest possible cost. We will be provided with an input source and a destination and then we will traverse through the weighted undirected graph in the shortest way. We will start off with the root that is set to zero and every other vertex will be set to infinity. These vertices do have weighted values which will be used later to determine routes and cost. As per the question we will be applying Djikstra’s Algorithm to get our solution.

Djikstra’s algorithm is widely applied in GPS systems to find the shortest possible route between the source and the destination provided. To explain a little more about the algorithm itself the first thing we need to start off with is that it utilizes a breadth first traversal system. All the distances from the root are set to infinity; this is basically the starting node/root. From here we check the distances to all the surrounding nodes and calculate the weighted sum of the vertices. If this weighted sum if less than the current weighted sum the two are replaced. The relaxation process is used to get the new weighted sums of all the vertices connected to the current one, in this way we traverse from node to node. This process has some preconditions associated with it and they are that the vertices connected should never be negative or zero and if an element is not found it should return with a failure error.

Moving on, the basic way that I will parse the data is that I will have two data structures associated with the input data file. These structures will be used to store the coordinates (x and y coordinates) and the adjacency matrix (matrix used to see what nodes are connected to each other). Looking at the input file we know that first line stores the vertices and the distances/edges. We will go through this array until we reach the last element after which the adjacency matrix will start reducing. Furthermore, if the value in the current set is larger than the one in the next set the two are replaced. In this way we will manipulate the data. Moreover, there is another function that accepts an input file and returns and stores the value that we searched for and it returns an error if it’s not present.

In the above way I will develop my algorithm so as to traverse with the smallest possible distance between nodes and knowing that Djikstra's algorithm is widely used the results will be very accurate.