**Route Finding**

**1. Introduction**

The topic of this project is route finding. The aim is to develop an application to compute the best path between 2 points on a map. The best path is defined as the shortest route between 2 points on a map in this project.

**2. Modular structure**

There are three source-code files and three header files in this application:

• source-code files:

main.cpp

buildMap.cpp

routeFinding.cpp

• header files:

mapStructure.h

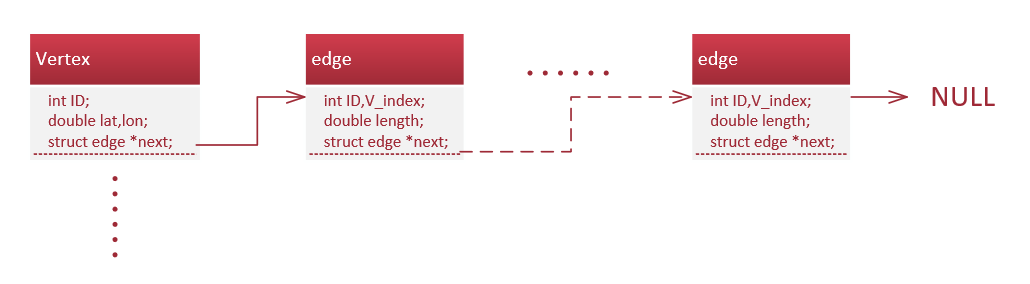
buildMap.h

routeFinding.h

First of all, this application is based on a map structure: adjacency list which is defined in the header file ‘mapStructure.h’.

There is a struct Vertex to represent Vertexes in the map and a struct edge to represent edges between two vertexes. The struct Vertex stores ID(int), its position: latitude(double) and longitude(double) and a pointer pointing to an edge from this Vertex. The struct edge stores ID(int), the index of vertex pointed by this edge, the length(double) and a pointer pointing to the next edge. Hence, the whole map can be represented by an array of Vertexes in the map. This application uses a vector<Vertex> myMap to represent the map.

The map structure can be represented as followed:



The application has three modules in it:

a. Build map module

b. Route finding module

c. User interface

**3. Design strategy**

**4. Test**

The program will work in both Windows and Linux system. So, it will be tested in both two systems.

For map initializing, changing and printing modules, there will be three tests. First, let the program read the date file and then output its map date into a given file. The expected output is that the date outputting is the same as the date inputting. Second, let the program read the date file and add one vertex and one edge into the map. And then the program prints the map. The expected output is that the program prints the right map. Third, let the program read the date file and delete one vertex and one edge from the map. The expected output is that the program prints the right map.

For the route finding and user interface modules, there will be two tests. First, input two unavailable points which don’t exist in the map. The expected output is an error message. Second, input two available points in the map. The expected output is the shortest route between 2 given points.

**5. Reflection**

In the first week of this project, the map initializing, changing and printing modules will be developed and tested. The developed code and test results will be updated to Github. And the route finding and user interface modules will be designed.

In the second week of this project the route finding and user interface modules will be developed and tested. The developed code and test results will also be updated to Github.