MATH2221 MATLAB project description

Project highlights

The project I am working on is about applying my knowledge on graphs to simulate shortest paths in CUHK. I created my own graph in which the nodes describe places in CUHK, and edges represents different paths between places. My objective is to find the shortest walking paths for users to travel between two points in CUHK. This program is specifically designed for people who are not familiar with places in CUHK, which aims to point directions for them to travel around CUHK.

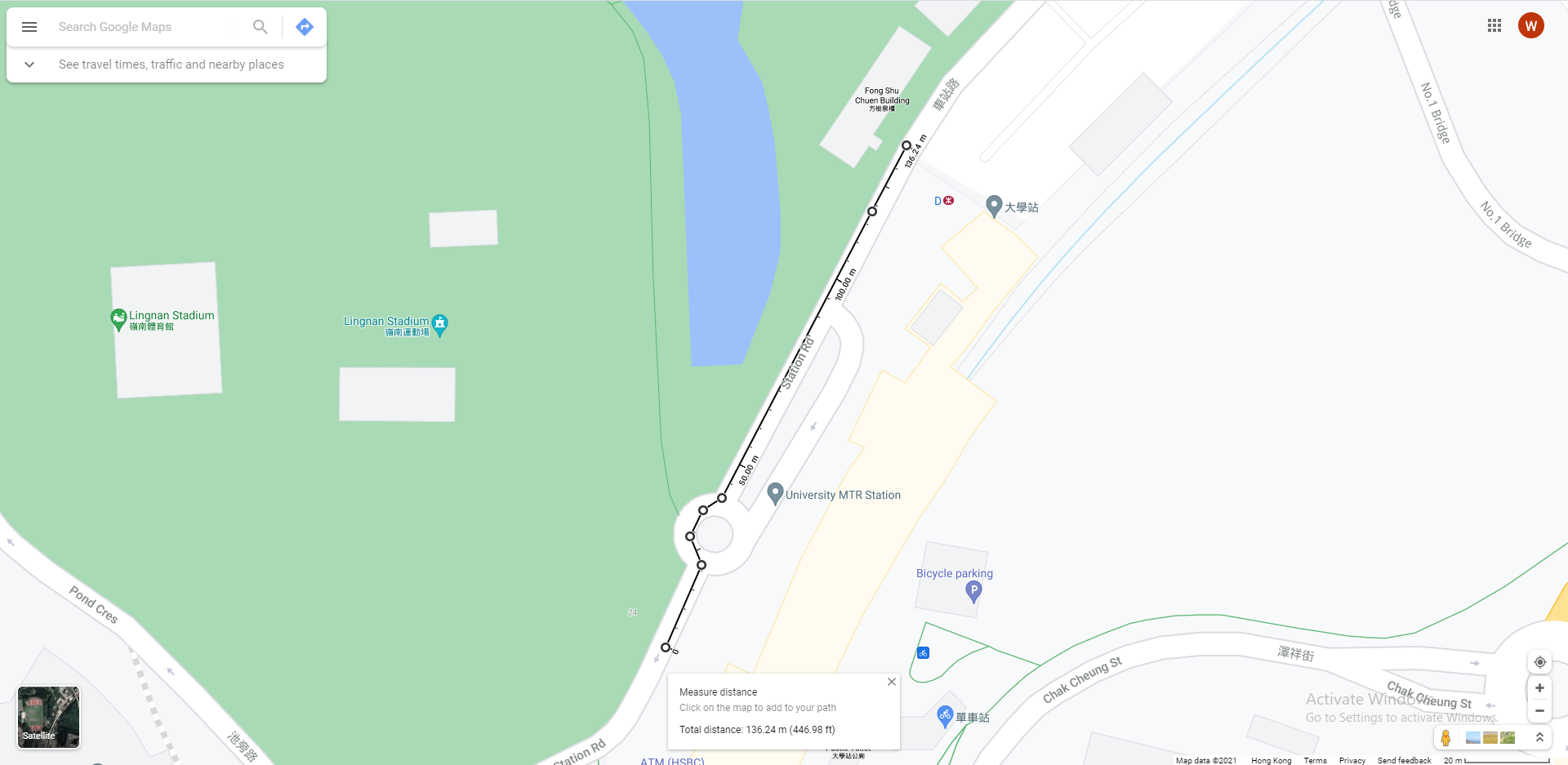
Methodology

There are three main parts in my program, which includes creating the CUHK graph, performing the shortest path calculation, and the user interface using MATLAB. When creating the CUHK map graph, I use the help of Google Map and Draw.io to obtain my data:



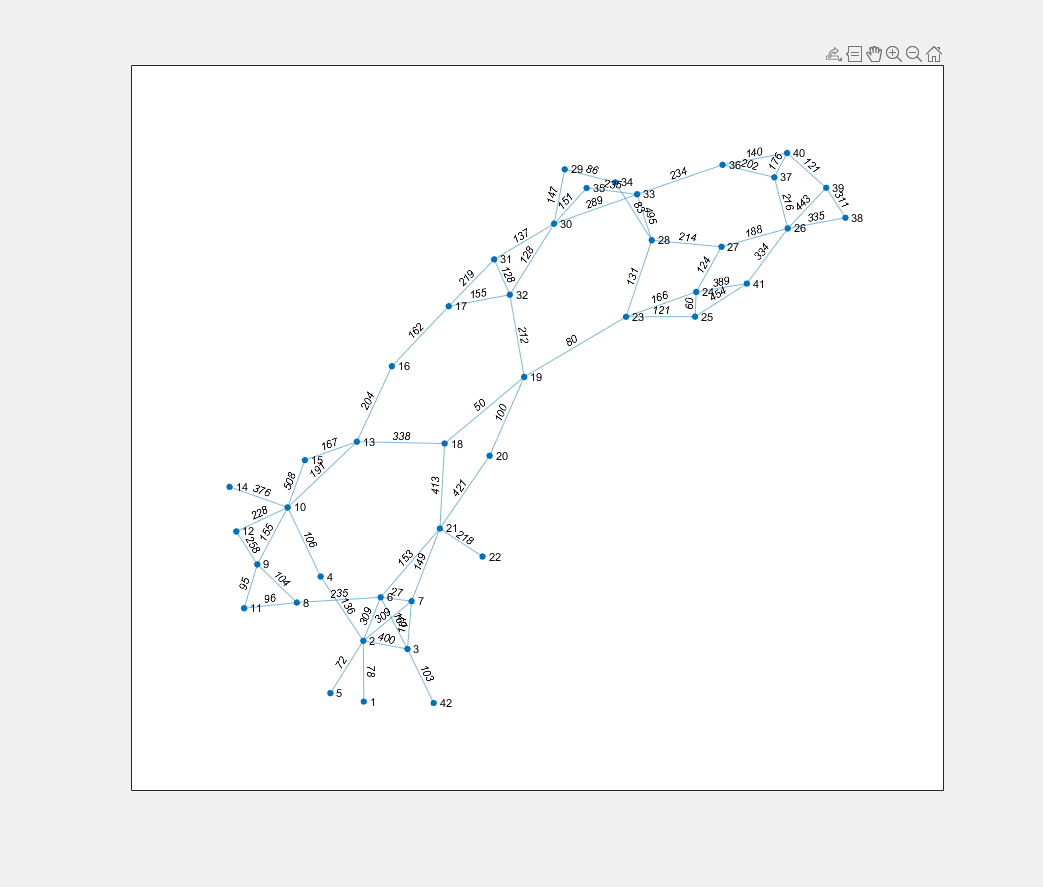
*Figure 1 The CUHK map with nodes and edges labelled with numbers and weight*

This map is from the CUHK official website, which includes all the content I want. I labelled all the significant places around CUHK with the lower nodes having a lower number. All the nodes represents the closest pathway to go from one node to another. The number on the edge (which will be referred as weight later on) represents the distance of the shortest path between these two nodes. They are calculated by the following method:



*Figure 2 A demonstration of how weight is measured on Google Map (node 2 to 4)*

Due to lack of time, the map only extends to Wu Yee Sun and Lee Woo Sing College. All of these data will be written in MATLAB using the “graph” function. If we plot the graph out without the map, it looks something like the following:



*Figure 3 The undirected weighted graph of CUHK map (using MATLAB)*

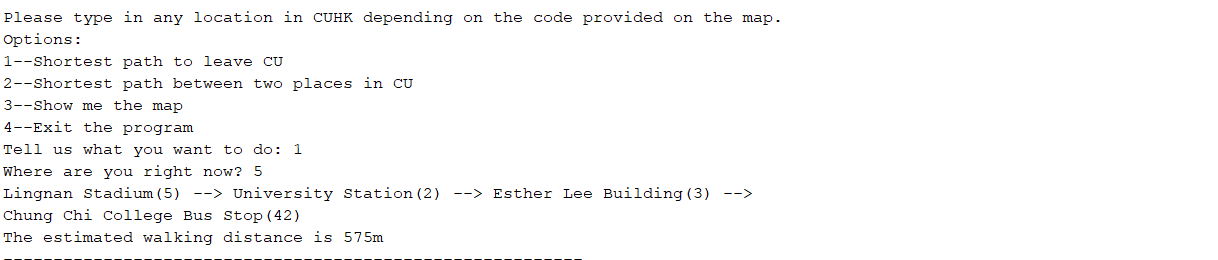
Although this is not the perfect graph (as I might be missing some of the faster paths, and I did not take in account of slope), it gives an approximate idea of how each places are connected with several nodes.

The second part is about the shortest path calculation. We use Dijkstra’s algorithm in this case due to its simplicity. Dijkstra’s algorithm is a solution to the shortest path problem where given any starting point, the algorithm can calculate its shortest possible path to every single node. Below are steps of performing the Dijkstra’s algorithm:

1. Create the graph and specify which node we are starting from. Also create a table that assigns a distance value to every single node according to its distance from source. If no path is yet found between source and the node, then distance = infinity. (For the first iteration only the nodes adjacent to the source has finite distance values)
2. Pick the unvisited node with minimum distance, check if there are any updates to distance values to any nodes.
3. During the algorithm, a visited node means the shortest path from source to this node has been found already.
4. The algorithm stops until it traversed the whole graph.

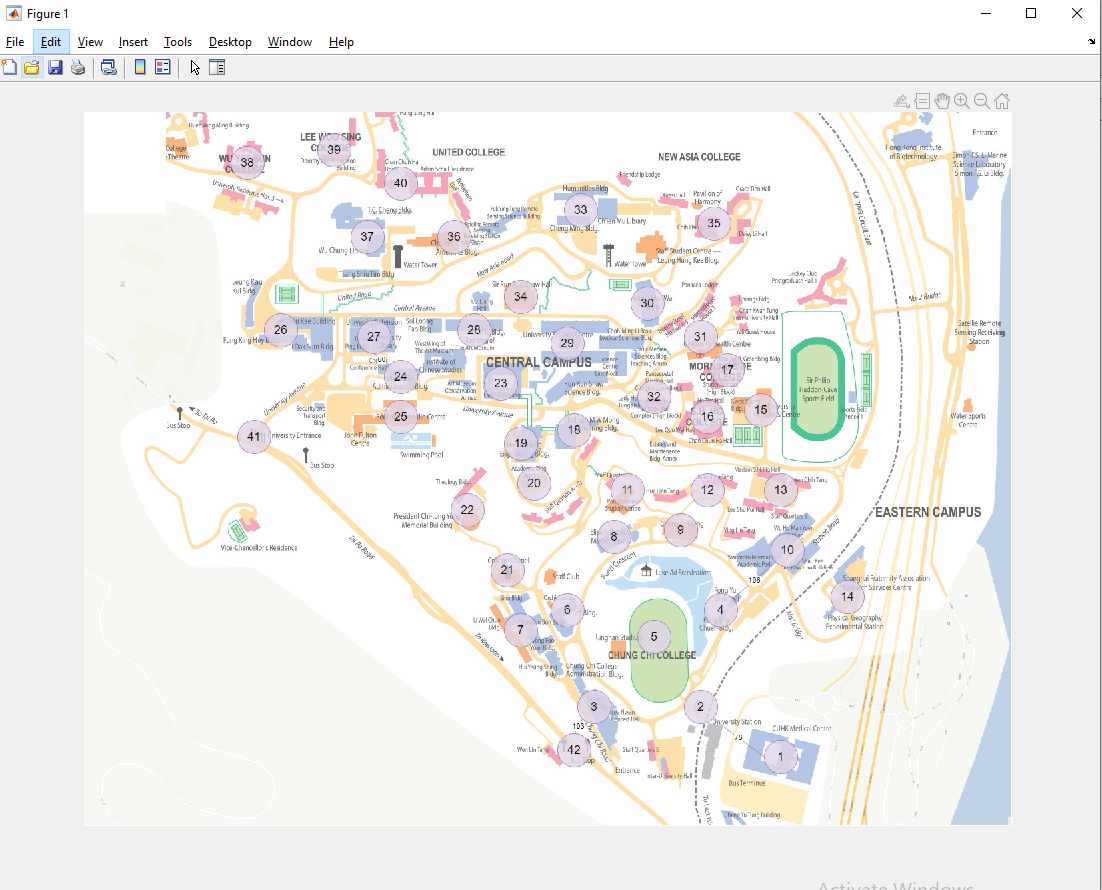
However, there is a function in MATLAB “shortestpath” that finds the shortest path between two nodes of the graph. By specifying the method to be ‘positive’, Dijkstra algorithm is applied during this function call. Note that Dijkstra only works when all edges have positive weight.

The final part is about the user interface. There are several options for users to choose from. Option 1 prompts user for its location and find the shortest path for them to leave CUHK. Option 2 prompts starting and final location from user and return the shortest path between these two points. Both of these options will show output below:



*Figure 4 Expected output for showing shortest path to leave*

Option 3 will show the map to the user again, where all the places are labelled with a node.



*Figure 5 The map shown to user*

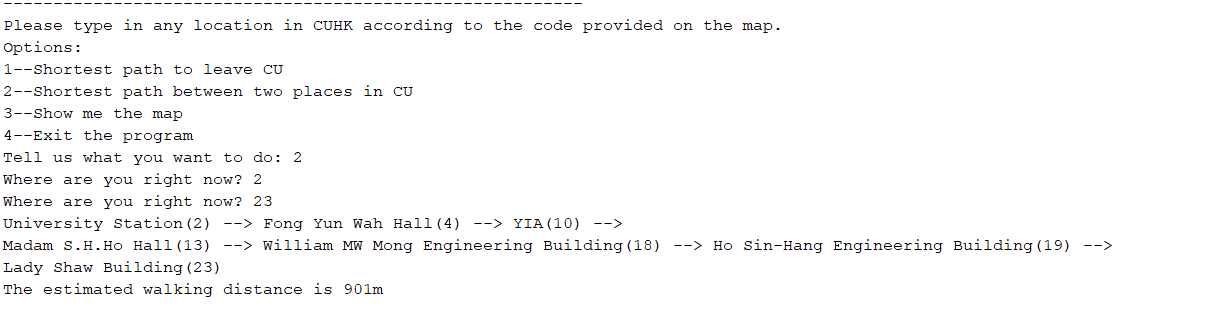
Option 4 allow users to exit the program, and the process will be terminated.

File structure

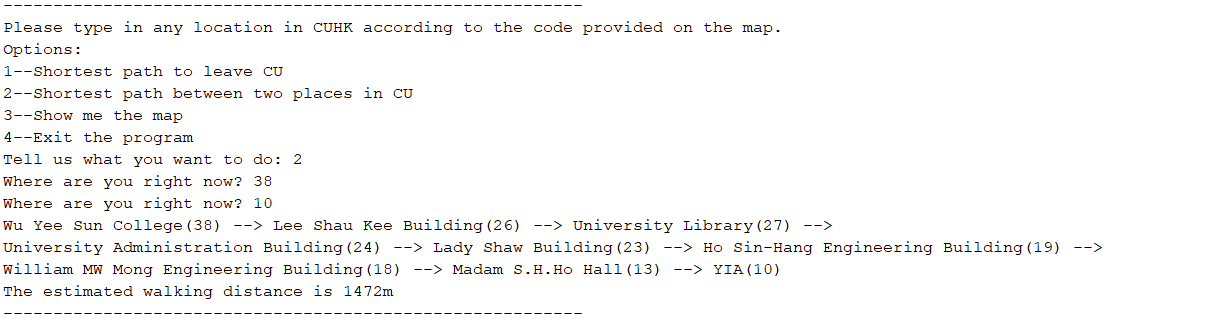
There are only two files in the project, where the one named “MATLAB\_project” is the file that contains the construction of CUHK map, the interface and mostly all codes. The other file named “translate” is just a file that converts the numbers of nodes to its corresponding name of places.

Example outputs

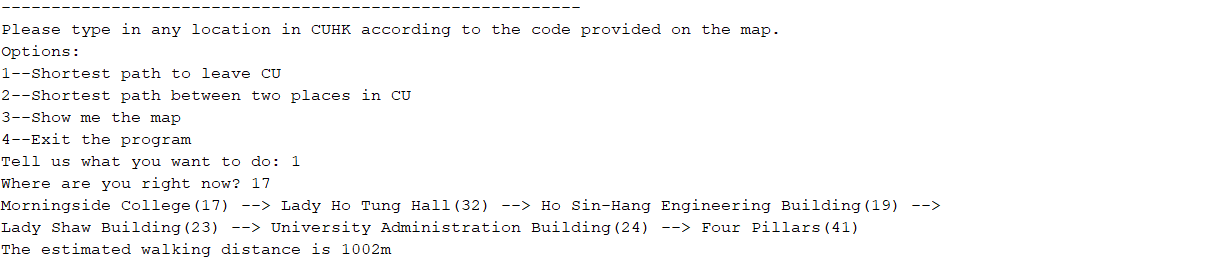
Example 1: Starting from University station to Lady Shaw Building



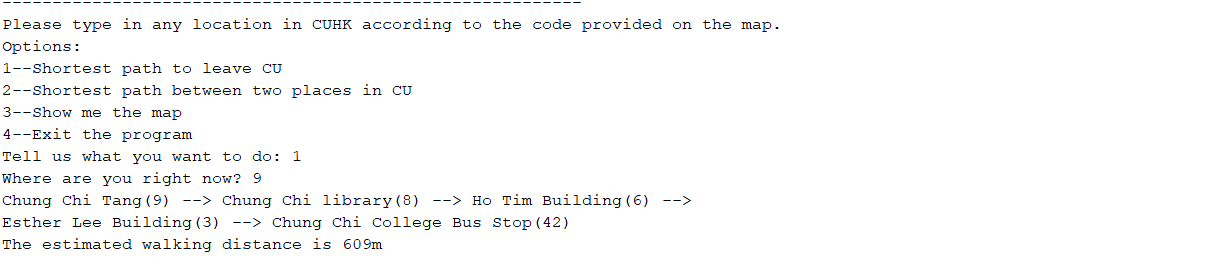
Example 2: Starting from Wu Yee Sun College to YIA



Example 3: Leaving CUHK from Morningside College



Example 4: Leaving CUHK from Chung Chi Tang



Example 5: Invalid inputs

