Problem Description

In this section, describe the problem you are solving and the functionalities of your program to solve the problem.

The main purpose of our program is to solve the problem of finding the shortest route between two predetermined points. A user will enter the two letter abbreviations of two cities and the program will calculate the shortest path between those two cities. The program will display an ordered list of the cities a person will need to visit to reach their intended destination in the shortest possible path. The program has an additional function of displaying fun facts about the starting city and destination city like population size and elevation. There is also a secret function to where the program will tell the user they entered an invalid city ID, and the program will even mention which of the two entered IDs was the invalid one.

Program Design

Describe the overall design of your program.

What algorithm did you choose for implementing the shortest path problem? Why is this algorithm applicable here? What classes do you use or add? Why do you choose to use those classes?

The algorithm we used to find the shortest path is Dijkstra's Algorithm. This algorithm was applicable because it finds the shortest path for every city based on a starting city. Since the user enters a starting and ending city, this algorithm can be used to find the shortest path from the starting city to every other city. The classes we used are a Graph class, Vertex class, and Heap class. The Graph class is used to represent the cities and the connections between cities. It consists of vertices which are cities and edges which are the roads connecting cities. The Vertex class is used to store information about a city including its index in the graph, name, city code, population, elevation, distance from a starting city, what the previous city in the shortest path is, and whether it has been visited in Dijkstra's algorithm. The Vertex class is used in both the Graph class and Heap class. In the Graph class, the Vertex class is used to represent cities. In the Heap class, the Vertex class is the element being stored inside the Heap. The Heap class is a minimum Heap and is used to create a priority queue of cities based on their distance from a starting city. The Heap class is used in Dijkstra's algorithm as part of the implementation.

System Implementation

Describe the details of your implementations. Did you run into problems in your implementation? How did you overcome those problems?

A major problem with making this assignment came from implementing the Dijkstra Algorithm. We were testing our search algorithm by comparing it to the provided example output but for some strange reason we would always get a different number of cities that needed to be passed through to reach the destination. We decided to draw the whole graph on draw.io so we can manually see where and why our search algorithm was making wrong turns. We pulled up the roads.txt file and found out there were 78 roads between the 20 cities. After drawing 5 roads we saw that the road from Anaheim to Bakersfield was 732 but the road from Bakersfield to Anaheim was 66. This meant that our algorithm was written for an undirected graph and was being applied to a directed graph. The reason our output was so different from the example output was because our search algorithm was going the wrong way on a one-way road. We had to quickly retrofit the entire search algorithm to pay attention to which roads it is allowed to go down at any given time. After the retrofitting was done, we started getting the same output as the example document.

Results

Did your results match the output in the “sample\_results.txt” file?

Our output did match the "sample\_results.txt" file output.

What are your answers to the following questions:

The shortest distance and path from FI to GG:

The shortest distance from IRWIN to GRPVE is 24.

The route is through IRWIN->PARKER->GRPVE.

The shortest distance and path from PD to PM:

The shortest distance from PARKER to POMONA is 133.

The route is through PARKER->BOSSTOWN->TORRANCE->POMONA.

The shortest distance and path from PM to PD:

The shortest distance from POMONA to PARKER is 357.

The route is through POMONA->EDWIN->ANAHEIM->VICTORVILLE->CHINO->GRPVE->IRWIN->PARKER.

The shortest distance and path from SB to PR:

The shortest distance from BERNADINO to RIVERA is 152.

The route is through BERNADINO->ISABELLA->BREA->CHINO->RIVERA

Conclusion

Give the conclusion and summary of your project. Did it solve the intended problem successfully? What functionalities does your program have? What lessons have you learned from this assignment?

Our program was able to successfully answer any direction-based questions about the graph. The program will state which city IDs were valid and will display some fun facts about each of the two cities provided in the command line. The most important lesson we learned in this assignment is to never make assumptions about input files because one bad assumption can completely ruin any written code. We assumed that we would be working with an undirected graph, but we were working with a directed graph the whole time. We should have opened up both of the input files and read them to see what kind of graph would be formed from their provided information. We ended up wasting a whole day just to fix a major mistake that would never have happened in the first place if we did our due diligence. We will make sure that will not happen again next time.