COP3530 – Data Structures and Algorithms
Project 3 Report : Road Trip

Team name

The Desperate Three

Team members

Brandon Corbin

Zachary Webb

Francisco Sanchez

GitHub URL : https://github.com/BCorbin825/Roadtrip

YouTube URL : https://youtu.be/I0QvtV_3S68

PROPOSAL

Problem

The goal of this project was to emulate a global positioning system (GPS) that gives the shortest path and the immediate path possible from city to city in the United States of America. In order to emulate a GPS for the U.S, Depth First Search and Breadth First Search will be used to create an adjacency list that represents the cities in the U.S. Since both DFS and BFS finds the immediate path, we can compare the path and performance of these algorithms. To compare the algorithms, we will weigh the time taken for each algorithm to find a path from one city to another.

Motivation

The motivation to complete this project was to grasp graph algorithms and learn how to apply them to solve real world problems. Once we are in the career field, we will have to solve similar problems using data structures and algorithms that have been discussed in this class. We believe that simulating a similar problem we will encounter in the future will refine both our programming and problem-solving skills.

Features

The main feature implemented in the GPS simulation was the search window used to find the distance and path from one desired city to another. From this search window we are also able to switch between algorithms that will be used to find the paths. Another important feature is the mapping of the U.S and having the cities, that are in the path from the two cities, accurately represented graphically as black dots on the map.

Data

Our data set are cities all across the U.S. Vertices are cities, and edges connect cities near each other and are weighted by the distance in miles. This data will be supported by the free database from https://simplemaps.com/data/us-cities. It has every incorporated city in the US. It tracks their longitude and latitude, which we can use to calculate the distance between them as well as the coordinates on the window.

Tools

The tools used to complete this project was C++ programming language, Geographic/UTM Coordinate converter library, and the SFML graphical library.

Graph Algorithms

The graph algorithms implemented in the project was Depth First Search and Breadth First Search. Both the Depth First Search and Breadth First Search is used to find the first path between the two desired cities. This is accomplished by finding the closest cities of the city of origin and finding the closest cities of those cities and so on, until it finds the desired city.

Additional Data Structures

The other data structure used to accomplish the GPS simulation was unordered maps. The unordered maps are used to contain the cities and a vector of the adjacent cities to the city of origin.

Distribution of Responsibility

Brandon Corbin - handle the graphical portion of the project which includes the interface and all components that come with it.

Francisco Sanchez - DFS and BFS algorithms.

Zachary Webb - manage the dataset and convert it into an adjacency list graph.

ANALYSIS

The only change made to the project was changing the originally planned Dijkstra's algorithm for a Depth First Search. The reason for this change was due to the amount of time Dijkstra's algorithm took to execute was not practical for the scenario that was created in this project.

Complexity Analysis

InsertEdge

- Both the average and worst-case time complexity for this function is O(1). This is due to the easily accessible unordered map data structure used to store the cities.

InsertCity

- Both the average and worst-case time complexity for this function is O(1). This is due to the easily accessible unordered map data structure used that allows us to insert a city freely.

GetWeight

- Both the average and worst-case time complexity for this function is O(1). This is due to the easily accessible unordered map data structure used to store the cities.

GetAdjacent

- The average-case time complexity for this function is O(log(n)). This is due to not having to go through every city stored in the unordered map if there is a large data set. However, the worst-case time complexity is O(n) and this is if there is a small data set where the desired city is adjacent to every city in the unordered map.

PrintGraph

- Both the average and worst-case time complexity for this function is O(n). This is due to having to traverse through the entire unordered map to find all the cities to print their name.

Depth First Search Algorithm

- The time complexity for this algorithm is O(V + E).

Breadth First Search Algorithm

- The time complexity for this algorithm is O(V + E).

REFLECTION

The overall experience for this project was exciting and educational, while also frustrating at times. The strategy originally put in place for this project did come to fruition smoothly even though it proved to have its challenges. One of the tasks that proved to be challenging was finding a strategy to get accurate positioning of the cities on the U.S map. To accomplish this task, we devised a way to use the longitude and latitude positioning of the U.S cities to create x and y coordinates in SFML. Even after this approach initially functioned as intended, problems emerged as we continued to program the project that made us go back in progress in order to fully resolve the task.

If we were to start over again as a group, there would not be much we would change. If allotted more time to dedicate solely to the project, we would have set up meetings more frequently to allow us to work more cohesively as group as opposed to working on different tasks for the project individually. Additionally, if having more time given, we would have preferred to add a zoom in/out method so that the cities were not grouped up undistinguishably in some areas.

Member comments

Zach Webb : My biggest takeaway was how to work with large datasets and coming up with ways to optimize the process of dealing with tens of thousands of entries. I also learned the process of coding in a group since it is not something I have not done before. Finally, I learned a significant number of projections used to show the geography of a flat surface and the benefits and disadvantage of each type of projection.

Francisco Sanchez : My biggest takeaway from this project was working and coordinating with a group to complete a project since normally academic projects have been assigned individually. I also learned how to take large datasets with meaningful data and learning how to apply it to solve a real-world problem. Lastly, using two different algorithms to accomplish the same goal, I learned that certain graph algorithms are only optimal depending on its application.

Brandon Corbin : The biggest takeaway from this project was learning to work as a team and completing each step separately and merging every individual piece to for the finished project. Another thing that is always a learning process with every project I do, is error debugging. Undoubtedly, errors arise and learning how to debug better improves my coding abilities tremendously.

REFERENCES

Geographic Coordinate Converter - https://alephnull.net/software/gis/UTM_WGS84_C_plus_plus.shtml

- The strategy placed to get the coordinates of the U.S cities to x and y-coordinates for the graphical display was executed using this C++ library.

SFML - https://www.sfml-dev.org/

- The graphical interface used to implement the GPS map and city points, as well as search for the cities.

U.S Cities Database - https://simplemaps.com/data/us-cities

- To obtain all U.S cities to implement in this project, we obtained the information from this database.

Depth First Search - Canvas COP3530 Fall 2021 Files/ PowerPoints/ Aman Presentations/ Module 8 : Graphs/ 8b - Graph Traversals and Algorithms.pdf/ slide 29

- Depth First Search implemented in the project was templated from the class presentation slides from module 8.

Breadth First Search – Canvas COP3530 Fall 2021 Files/ PowerPoints/ Aman Presentations/ Module 8 : Graphs/ 8b - Graph Traversals and Algorithms.pdf/ slide 29

- Breadth First Search implemented in the project was templated from the class presentation slides from module 8. However, the algorithm in the slide provided was for a Depth First Search. This was modified slightly to convert it to a Breadth First Search to create a slightly unique solution for the project.