Abigail Animah Owusu

Solomon Kwabena Asante-Ansong

Oheneba Kwaku Addo Dade

CS222 Data Structures and Algorithms

Capstone Team Report

2nd May 2022

**Introduction**

Students, especially freshmen in Ashesi, have difficulty navigating around the various offices and classrooms on campus. Due to this, we have decided to create a solution that would aid students in identifying the multiple offices and obtain possible paths that would take them there. This problem is essential because it could reduce the number of students who make it to class late. We unanimously agreed that tackling an issue of this nature could help us be better citizens who are solving problems in the space they find themselves in.

Graphs would help us create nodes representing the various campus locations and map out the possible pathways that link these locations. Dijkstra's algorithms would enable the calculation of the shortest distance between two points, which would then be recommended to the students. Again, we would utilize a two-dimensional array to create a matrix to serve as the grid for the graph structure.

**Background**

"Graph data structure comprises of various nodes and multiple edges connecting these nodes" (Mehta et al., 2019). A couple of short path algorithms are associated with graphs that allow for finding the shortest path from one node to another. One of such algorithms which we employed is Dijkstra's algorithm. With a given source as the root, we create an SPT (shortest path tree). We keep two sets: one contains vertices already in the shortest-path tree, and the other includes vertices not yet in the shortest-path tree. We find a vertex in the other set (set of not yet included) that is the shortest distance from the source at each stage of the procedure (Geeks For Geeks, 2018).

Interestingly, these concepts are employed in numerous large-scale applications such as Google Maps.

**Methodology**

The underlying data structure of the project was a graph. Because the application was explicitly streamlined for use on the Ashesi Campus, a pre-determined number of vertices were employed with their corresponding locations. Each vertex was indicated with an integer, and a HashMap was used to effectively map each integer vertex to its corresponding string location to improve the user experience. The next stage was to implement Dijkstra's algorithm, which the help of pseudocode available in textbooks ensured we completed. We included methods that allowed the actual path to be displayed and not necessarily just the distance of the shortest route.

Finally, we decided to create a Graphical User Interface. After deliberation, we settled on using Java Swing. A couple of tutorial videos and the documentation proved beneficial in bringing our code to visual life.

**Results**

We used the different locations in Ashesi and calculated distances from other points. We used intuition to measure the distances. For example, we knew that we would have the most extended length and the heaviest weight on the graph from the school gate to the Ash pitch. After implementing the code, we asked some students to test by using our application. We realized that some of the distances were not so accurate, so it gave different paths other than what was expected. To correct this, we had two points (from the school gate to the inner gate) as our reference and ensured each distance we gave was selected in relation to the distance between these points. After all these corrections, our code worked just like how we expected.

**Conclusion**

In conclusion, our project, *Application of Graphs: Ashesi Campus Roadmap*, implements non-linear data structures such as graphs, HashMaps and, linear data structures, arrays with the help of Swing, a Java GUI API, which creates a simulation of Ashesi's campus allowing users to select their location and the destination they want to reach. With the help of Dijkstra's algorithm, the program outputs the shortest path the user should take to arrive at their destination. The key lessons we gained from this project were using the GUI Swing to create frames and buttons and combining the GUI with linear and non-linear structures to display the map to the user. We also gained more insights on shortest path problems and how algorithms such as Dijkstra's have contributed to solving real-world issues. Moving on, in the future, we could implement other data structures to increase the program's efficiency or also use the actual distances from google maps to display the map of Ashesi.

**References**

Mehta, H., Kanani, P., & Lande, P. (2019). Google Maps. International Journal of Computer Applications, 178, 41–46. <https://doi.org/10.5120/ijca2019918791>

‌Geeks For Geeks (2018). Dijkstra's *algorithm*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/.

‌