Introduction

In this project exercise, we investigated numerous graph search algorithms to discover the best solution to the project problem. In the end, we settled on using a level order search technique known as breadth first search to map out possible flight itineraries from various airports because it has advantages over other alternatives such as depth first search. One of the selling factors for the adoption of breadth first search at the expense of depth first search was its guarantee to always generate a solution. This is a common shortcoming of the depth first approach.

Approach

The algorithm is implemented using object instances of a custom Airport class. These are then represented as graph nodes and enqueued onto the search frontier, which is ideally a queue/dequeue to capitalize on its FIFO nature. A link or route between airports is represented as a route object, which is utilized in the search to create successor airports (nodes). These nodes are arranged on the frontier based on the number of steps it took to expand each. In this iterative process, airports that have already been processed or have already been queued for processing are ignored when encountered to provide an added boost with regards to time and memory efficiency. At the end of the process, a node instance is returned and subsequently a custom class method is used to output and format the required flight information to a file.

Conclusion

This assignment taught me about the various types of search algorithms and how they fit different demands. When cost is immaterial to the search, breadth first search and depth first search are excellent graph search algorithms. However, when optimality by a cost component is necessary, greedy algorithms such as Dijkstra's and best first techniques like A* are undeniably

superior.

References

GeeksforGeeks. (2022, November 18). *Breadth First Search or BFS for a Graph*. https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/