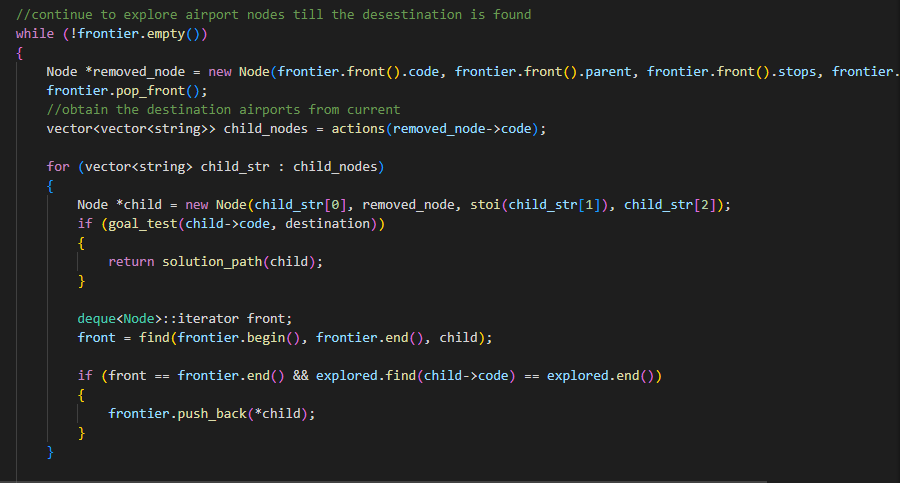
**Reflection on C++ Project**

In this program I tried to implement the ideas taught in class. From the basic Accessor methods for OOP to File I/O. The project I embarked on was primarily to help me understand and get accustomed to the C++ syntax. This project was initially written in Java; I therefore had to re-write the project using my initial understanding of the problem instead of converting the lines of code line by line. I worked on the project with the help of my colleague Bethel Choto C’24. I will therefore discuss more on the aspects I contributed and the challenges we faced as a team.

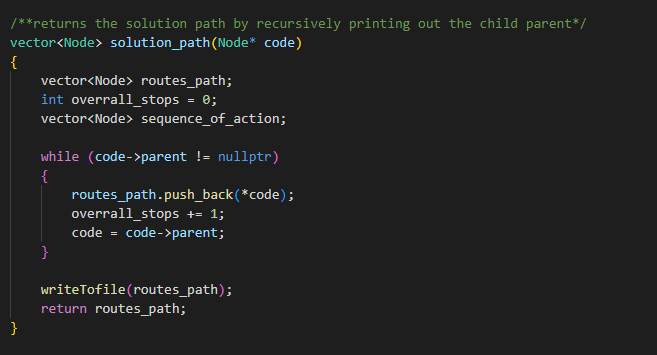
I implemented the search. This is the code fundamental to finding the best route from origin to the desired destination. With the given start airport code, I obtain all the destination airport codes which are directly connected to this airport by using function **actions()** which obtains the routes by use of a key in the unordered\_map.



The values obtained from from the actions class are each used to create child nodes, whose parent nodes are automatically set to the current node, before they are then added to the frontier for later expansion.

Before I added Nodes to the frontier I had to check the frontier first to avoid duplicating nodes which would later affect our code runtime. To check this I used the find() method that takes the start of the deque and end as well as the object to be searched for. it returns a null pointer in the event that it does not find the node. I had to override the operator==() in the Node class to perform more accurate object comparison.

We also later faced challenges in using the Node class. At a point, all the child nodes had their parent attribute point to the last obtained Node. This meant that during the while loop, the value stored at the location our parent node was pointing to was constantly bring altered, such that we lost values to our solution path. We had to change this approach by declaring a new Node in each loop.



Lastly, the solution path was obtained by recursively finding the parent of the current Node. These parent nodes were added to a vector<Node> variable which was then written to a file.

This so far, has been a very effective method in helping learn the **C++** syntax and logic fast. I will continue to experiment converting other projects I have written in Java and then move onto writing fresh projects in pure C++.