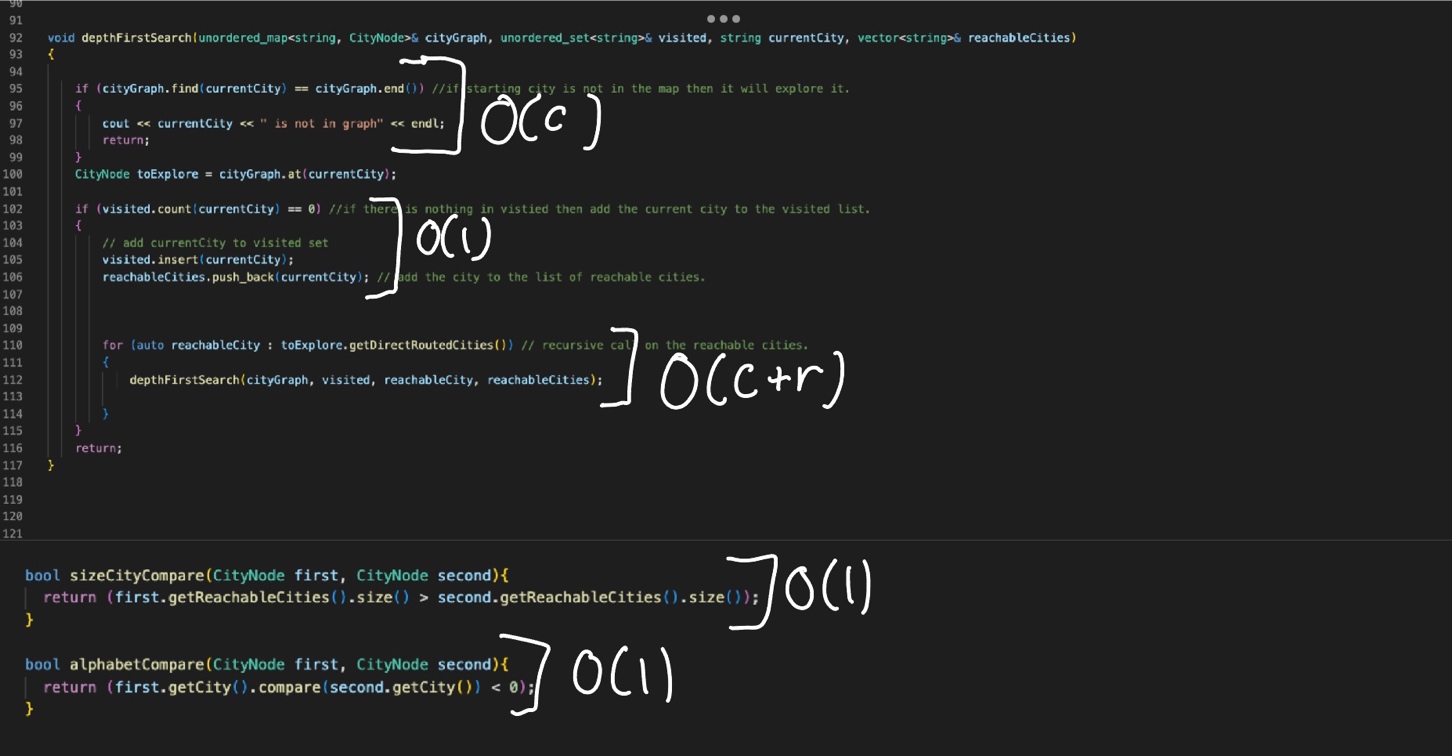
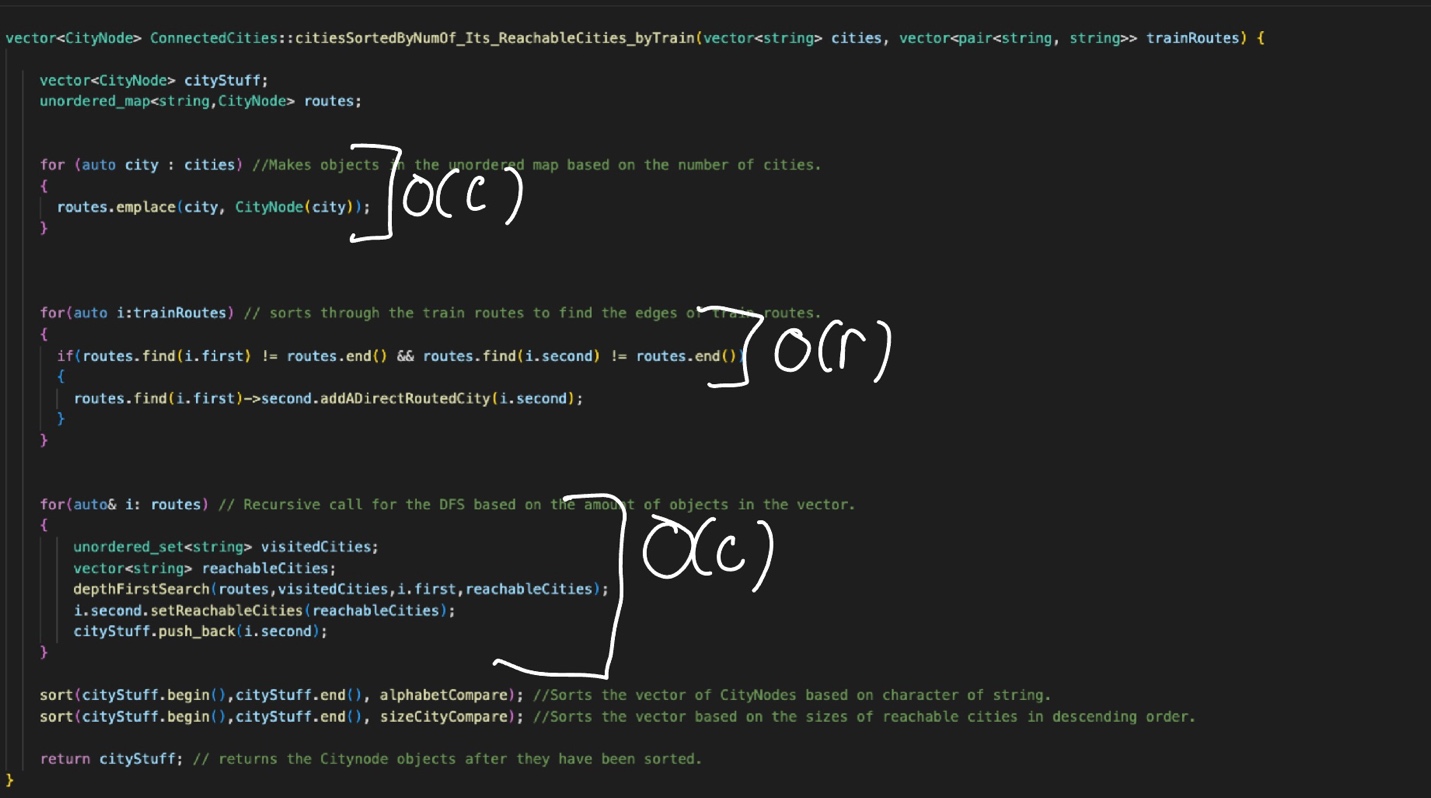
Assignment Four Complexity Analysis



Recursive call is reliant on number of the cities and the number of edges because it will need to compare against them all to provide an output of reachable cities.

This method is O(1) given that it is an if statement.

Same as the if below it.

Recursive call comes down to the number of cities because every city must be visited in the DFS

This for loop will need to search and identify all the edges so it is reliant on the number of edges (r)

With this loop we would need to iterate based on the number of items passed in by cities

**Time Complexity explanation:**

For the complexity of my code I believe it is running at O(c^2 + r), with the first statement that will need to run for the amount of cities O(c) to create my unordered map. The second for loop is dependent on the number of edges which will bring it to O(r). The third call is the recursive call which calls on the DFS function. With the recursive call function it need to search the number of cities and the number of the nodes so the O notation will need to add the two together, resulting in O(C + R). The sort functions that are called both have O(nlogN) complexity but given that the overall time space complexity is C^2 + R, I believe that overtakes the complexity of nLogn.

C + C + R = O(C^2 + R) for time space complexity given the worst case scenario.