Time Complexities of Functions in graph.h and graph.cpp

createMap: FIXME

O(1) worst case time complexity because this function creates a flight object and inserting into graph with matrix representation. The worst case time complexity for inserting into a graph with a matrix representation is O(1).

cheapestDirect:

The worst case time complexity of our function cheapestDirect is O(1) because it runs a for-loop a fixed number of times (100) since we know our matrix is 100x100. Accessing elements in our graph also has a worst case time complexity of O(1) since matrix representations allow for random access and thus O(1) access time. Furthermore, the nested if statements checking if the current flight is not nullptr and that the current flight price is lower than the minimum is also a O(1) operation time. Thus our overall time complexity would be O(1\*1\*1) = O(1).

cheapestIntl:

The worst case time complexity of this function is O(1) because it runs a for-loop a fixed number of times (100) since we know our matrix is 100x100. Accessing elements in our graph also has a worst case time complexity of O(1) since matrix representations allow for random access and thus O(1) access time. Furthermore, the nested if statements checking if the current flight is not nullptr and that the current flight price is lower than the minimum is also a O(1) operation time. Thus our overall time complexity would be O(1\*1\*1) = O(1).

directExists:

The worst case time complexity for the directExists function is O(1) because accessing elements in our graph has a worst case time complexity of O(1) since matrix representations allow for random access and thus O(1) access time.

flightMonth:

The worst case time complexity of the flightMonth function is O(1) because it runs a for-loop a fixed number of times (100) since we know our matrix is 100x100. Accessing elements in our graph also has a worst case time complexity of O(1) since matrix representations allow for random access and thus O(1) access time. Furthermore, the nested if statements checking if the current flight is not nullptr and that the current flight price is lower than the minimum and is the right month is also a O(1) operation time. Thus our overall time complexity would be O(1\*1\*1) = O(1).

international:

minCity:

path:

getBest:

readCSVFile:

n: