

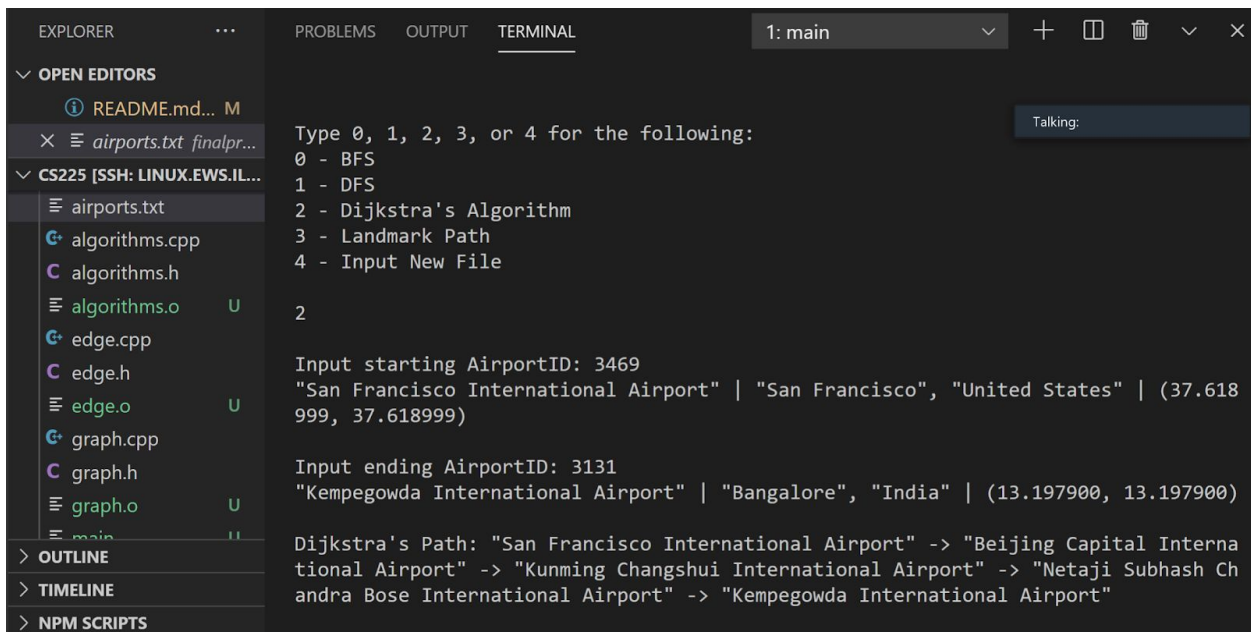
Open Flights Results

By Praneeth Mekapati pm14, Chidambara Anagani canaga2,
Alaa Shuaibi ashuai6, Rahul Vasanth rvasant2

- Outcomes:

We were able to successfully create a directed graph based on the data of the files “airports.txt” and “routes.txt”. All algorithms successfully traversed through the directed graph. The algorithms used for the graph were BFS, DFS, Dijkstra, and Landmark. A majority of the algorithms were responsible for finding a short and easy path. Dijkstra was able to easily give us the path of each user input starting and ending airports. It was able to successfully describe the location (city, country) and the name of the airport. It was successful for the BFS and DFS as well with outputting the airport details and the paths. BFS and DFS were able to finish in a minute and 20 seconds as Dijkstra and Landmark were able to finish in an instant.

main.cpp->OpenFlights->Algorithms->graph->edge
->Miscellanies->ReadFromFile



```
1: main
Type 0, 1, 2, 3, or 4 for the following:
0 - BFS
1 - DFS
2 - Dijkstra's Algorithm
3 - Landmark Path
4 - Input New File

2

Input starting AirportID: 3469
"San Francisco International Airport" | "San Francisco", "United States" | (37.618999, 37.618999)

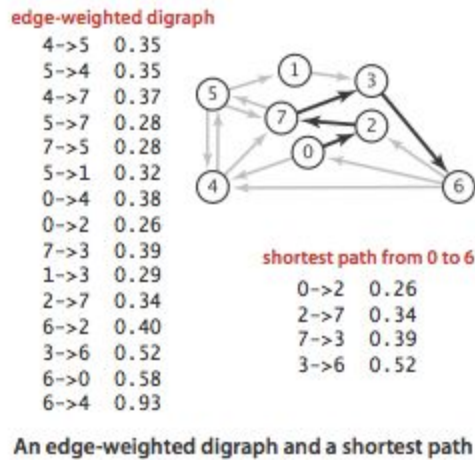
Input ending AirportID: 3131
"Kempegowda International Airport" | "Bangalore", "India" | (13.197900, 13.197900)

Dijkstra's Path: "San Francisco International Airport" -> "Beijing Capital International Airport" -> "Kunming Changshui International Airport" -> "Netaji Subhash Chandra Bose International Airport" -> "Kempegowda International Airport"
```

- What was accomplished?

We were able to quickly get the results of each algorithm. We were able to view the important details as well in the output for the paths traversed or taken. We also were able to successfully

verify the results by comparing them to the theoretical results such as comparing the Dijkstra drawn graph to the function in the program.



```

1:main
Type 0, 1, 2, 3, or 4 for the following:
0 - BFS
1 - DFS
2 - Dijkstra's Algorithm
3 - Landmark Path
4 - Input New File
2
Input starting AirportID: 3469
"San Francisco International Airport" | "San Francisco", "United States" | (37.618
999, 37.618999)
Input ending AirportID: 3131
"Kempegowda International Airport" | "Bangalore", "India" | (13.197900, 13.197900)
Dijkstra's Path: "San Francisco International Airport" -> "Beijing Capital Interna
tional Airport" -> "Kunming Changshui International Airport" -> "Netaji Subhash Ch
andra Bose International Airport" -> "Kempegowda International Airport"
  
```

- What we could have done differently?

We thought organizing the code was one thing we can improve upon as it was confusing to navigate through the functions at first. Also, with more time, we could have written better test cases for the file reading functions. We could have implemented the functions to convert the strings from the txt into int in a better way.

- Conclusion:

Overall, we were able to accomplish our major tasks for the graph of OpenFlights. We were able to effectively utilize each day to make sure our algorithms worked without any bugs. This overall result was very neat and organized and we were able to understand them. They were easily implemented.