**CS225 Final Project Result**

Our CS225 final project fully accomplished our goals: process the OpenFlight dataset as a graph structure and run the following algorithms on it: DFS traversals for traversing the whole graph, Floyd-Warshall algorithm for calculating shortest path between two points, airports and airline routes visualization on a world map. This document will briefly illustrate the outcome of this project and introduce our program.

After compelling and running the program, there will be specific instructions for choosing datasets and algorithms. Our program supports reading other data files formatted the same way as the airports and routes data in OpenFlight dataset. Due to the runtime of Floyd-Warshall algorithm, running the program on OpenFlight dataset can take up to 1.5 hours, so we also provided a subset for grading and testing purpose.

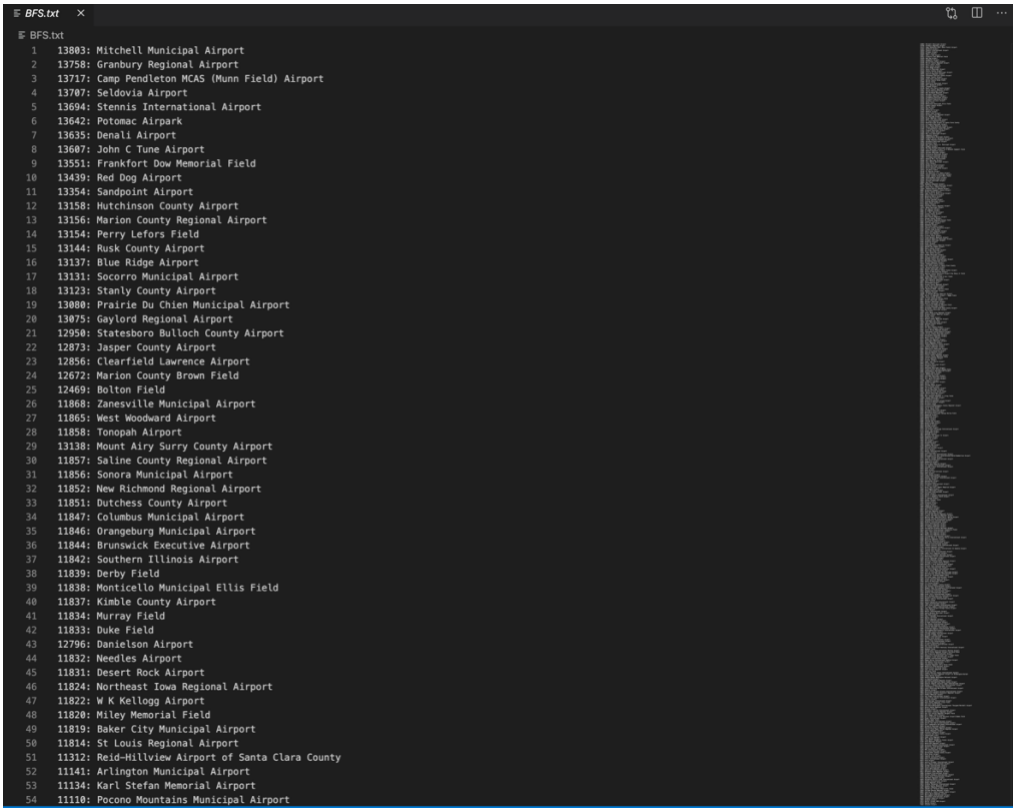
After choosing the dataset, users will be able to choose which algorithm to run. By choosing BFS or DFS, the program will respectively run each traversal algorithm and save the result to the same folder as “BFS.txt” or “DFS.txt” with a list of vertices based on the traversal order; By choosing shortest path, the program will calculate the distance matrix and the path matrix using Floyd-Warshall Algorithm, then it will demonstrate the shortest path as a list between the source and destination airports chosen by users. Since Floyd-Warshall Algorithm computes shortest distance between all pairs of vertices, the program supports multiple searches; By choosing visualization, program will result two PNGs: “world_map_with_airports.png”, and “world_map_with_airports_and_routes.png”, as two pictures with airports as nodes on the world map as well as the routes as edges.

We did have discoveries while proceeding our final project. In particular, we found drawing pixel lines between two specific points surprisingly challenging. We thought this can be

solved using a simple for loop but we failed after many attempts. After some researches, we found out that there are many different cases we need to consider and computer scientists have developed many algorithms for this problem. We chose the Bresenham's Line Algorithm. Moreover, we found many disconnected airports. This is because the airport and route data we used were collected in different years. We modified several our data processing functions to avoid problems caused by this situation.

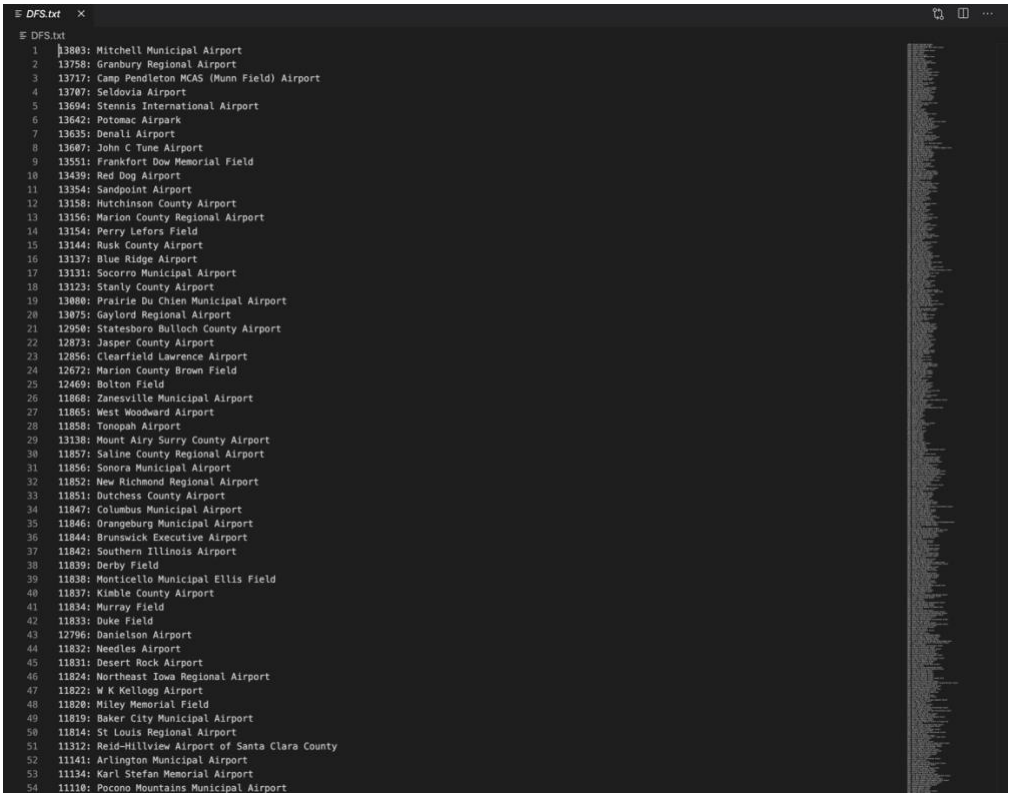
Screenshots of our outputs are included in next page

Traversal Result:



BFS.txt

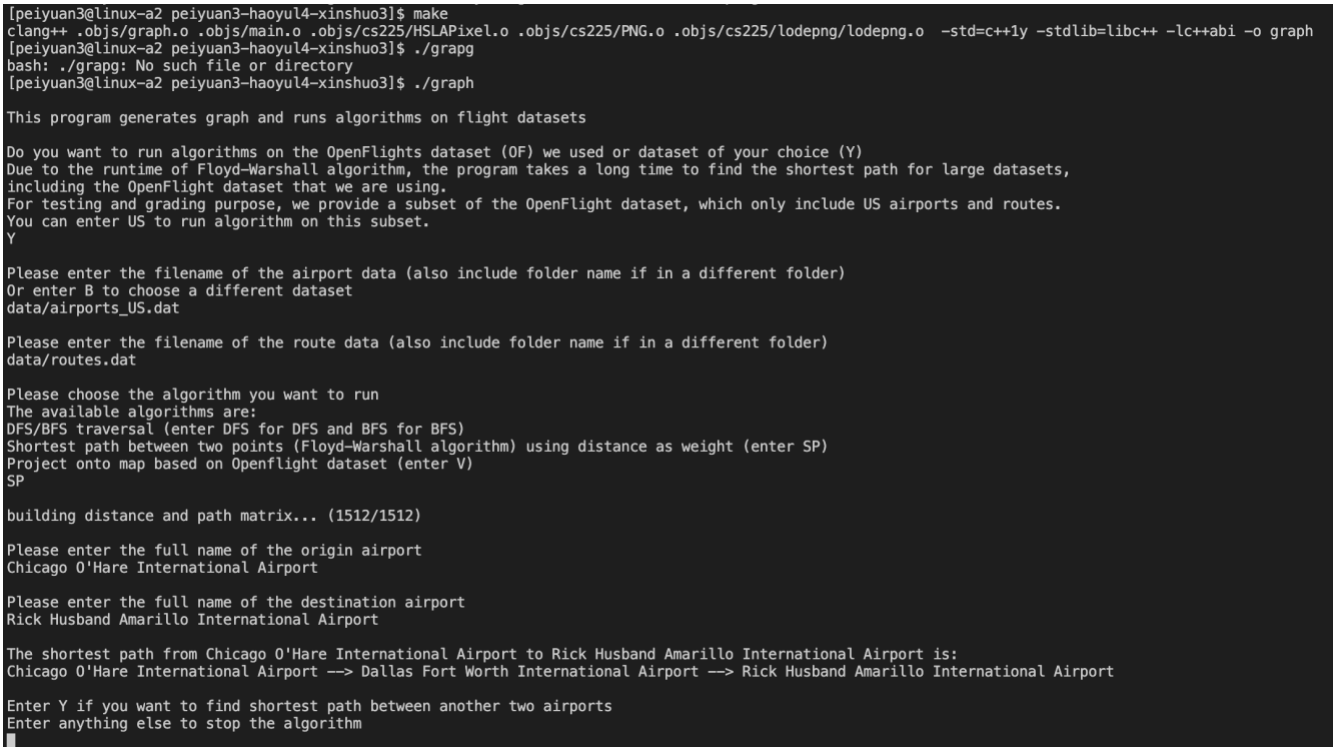
```
13803: Mitchell Municipal Airport
13758: Granbury Regional Airport
13717: Camp Pendleton MCAS (Munn Field) Airport
13707: Seldovia Airport
13694: Stennis International Airport
13642: Potomac Airpark
13635: Denali Airport
13607: John C Tune Airport
13551: Frankfort Dow Memorial Field
13439: Red Dog Airport
13354: Sandpoint Airport
13158: Hutchinson County Airport
13156: Marion County Regional Airport
13154: Perry Lefors Field
13144: Rusk County Airport
13137: Blue Ridge Airport
13131: Socorro Municipal Airport
13123: Stanly County Airport
13080: Prairie Du Chien Municipal Airport
13075: Gaylord Regional Airport
12950: Statesboro Bulloch County Airport
12873: Jasper County Airport
12856: Clearfield Lawrence Airport
12672: Marion County Brown Field
12469: Bolton Field
11868: Zanesville Municipal Airport
11865: West Woodward Airport
11858: Tonopah Airport
13138: Mount Airy Surry County Airport
11857: Saline County Regional Airport
11856: Sonora Municipal Airport
11852: New Richmond Regional Airport
11851: Dutchess County Airport
11847: Columbus Municipal Airport
11846: Orangeburg Municipal Airport
11844: Brunswick Executive Airport
11842: Southern Illinois Airport
11839: Derby Field
11838: Monticello Municipal Ellis Field
11837: Kimble County Airport
11834: Murray Field
11833: Duke Field
12796: Danielson Airport
11832: Needles Airport
11831: Desert Rock Airport
11824: Northeast Iowa Regional Airport
11822: W K Kellogg Airport
11820: Miley Memorial Field
11819: Baker City Municipal Airport
11814: St Louis Regional Airport
11312: Reid-Hillview Airport of Santa Clara County
11141: Arlington Municipal Airport
11134: Karl Stefan Memorial Airport
11110: Pocono Mountains Municipal Airport
```



DFS.txt

```
13803: Mitchell Municipal Airport
13758: Granbury Regional Airport
13717: Camp Pendleton MCAS (Munn Field) Airport
13707: Seldovia Airport
13694: Stennis International Airport
13642: Potomac Airpark
13635: Denali Airport
13607: John C Tune Airport
13551: Frankfort Dow Memorial Field
13439: Red Dog Airport
13354: Sandpoint Airport
13158: Hutchinson County Airport
13156: Marion County Regional Airport
13154: Perry Lefors Field
13144: Rusk County Airport
13137: Blue Ridge Airport
13131: Socorro Municipal Airport
13123: Stanly County Airport
13080: Prairie Du Chien Municipal Airport
13075: Gaylord Regional Airport
12950: Statesboro Bulloch County Airport
12873: Jasper County Airport
12856: Clearfield Lawrence Airport
12672: Marion County Brown Field
12469: Bolton Field
11868: Zanesville Municipal Airport
11865: West Woodward Airport
11858: Tonopah Airport
13138: Mount Airy Surry County Airport
11857: Saline County Regional Airport
11856: Sonora Municipal Airport
11852: New Richmond Regional Airport
11851: Dutchess County Airport
11847: Columbus Municipal Airport
11846: Orangeburg Municipal Airport
11844: Brunswick Executive Airport
11842: Southern Illinois Airport
11839: Derby Field
11838: Monticello Municipal Ellis Field
11837: Kimble County Airport
11834: Murray Field
11833: Duke Field
12796: Danielson Airport
11832: Needles Airport
11831: Desert Rock Airport
11824: Northeast Iowa Regional Airport
11822: W K Kellogg Airport
11820: Miley Memorial Field
11819: Baker City Municipal Airport
11814: St Louis Regional Airport
11312: Reid-Hillview Airport of Santa Clara County
11141: Arlington Municipal Airport
11134: Karl Stefan Memorial Airport
11110: Pocono Mountains Municipal Airport
```

Shorted path example result:



Visualization result: