RESULTS

Our Dijkstra.cpp makes use of the Dijkstra's algorithm, which can output a vector containing all the airport names along the path. At the same time, our shortest_path function can calculate the shortest distance from the source airport to the destination airport. As can be seen in figure 2, we tried to find the shortest path from Guangzhou to Chicago. To compare, we used an alternative path from Guangzhou to Shanghai to Chicago. The distance of the alternative path is slightly larger than the shortest path distance, indicating the correctness of our result. We also tried to find the shortest path between Los Angeles International Airport (LAX) and Chicago O'Hare International Airport (ORD). Interestingly, the shortest path is not the direct route from LAX to ORD, instead it has to go through Colorado. This made sense though. If we drew a straight line from LAX to ORD in Google Maps, we can actually see that Colorado is in the middle. However, the difference with the direct route is so tiny. This might be ascribed to some rounding errors when calculating the distances. The direct route should always be the shortest path in our implementation since we assume that every airline follows the straight line from one airport to the other.

In addition, we found some interesting differences after comparing Disjoint sets and BFS / DFS. Disjoint set is used to find if there are disconnected parts. After removing all the airports that do not have incoming and outgoing routes, we find that there are 6 disconnected parts in total. However, the largest connected part found by disjoint sets and BFS / DFS are different. This is probably due to the fact that some parts are connected only in one direction, which explains the smaller number found by BFS / DFS

The pagerank.cpp and utils.cpp implement the pagerank algorithm, which simulates and outputs the most popular airport as the destination. By operating the loading of label rank, we create the transition matrix and keep left multiplication to get

the final prediction. As shown in Figure 4 below, the ten most “popular” airports are generated. A more accurate way to define “popular” is how likely the airport would be the destination of international travel. We also get a group of useful data by loading weight, which stands for airports of the longest routes. This group of data would be useful for analyzing the characteristics of different routines in future study.

Moreover, we also create the pathDrawer.cpp to draw graph of airlines to show the data visually.

For showing the output results more concise, we create the getXXXReport for each method to generate the conclusive output reports.

Attached Figures:

```
(base) joeye@JoeyedeMacBook-Air 225project % ./main
The shortest path distance is 4468 kilometers.
KZN
SVX
NBC
GYD
(base) joeye@JoeyedeMacBook-Air 225project %
```

Figure 1

```
(base) joeye@JoeyedeMacBook-Air 225project % ./main
an alternative route from Guangzhou to Chicago: Guangzhou -> Shanghai -> Chicago with a distance of 12536 km.
The shortest path distance is 12443 kilometers.
Beijing Capital International Airport
Chicago O'Hare International Airport
(base) joeye@JoeyedeMacBook-Air 225project %
```

Figure 2

```
(base) joeye@JoeyedeMacBook-Air 225project % ./main
direct route from LA to Chicago with a distance of 2802 km.
The shortest path distance is 2801 kilometers.
City of Colorado Springs Municipal Airport
Chicago O'Hare International Airport
(base) joeye@JoeyedeMacBook-Air 225project %
```

Figure 3

```
1: 3682      Hartsfield Jackson Atlanta International Airport
2: 3830      Chicago O'Hare International Airport
3: 3484      Los Angeles International Airport
4: 3670      Dallas Fort Worth International Airport
5: 1382      Charles de Gaulle International Airport
6: 507       London Heathrow Airport
7: 3364      Beijing Capital International Airport
8: 3316      Singapore Changi Airport
9: 3751      Denver International Airport
10: 340      Frankfurt am Main Airport
```

Figure 4