Project Write-Up: Finding Busiest Airports Using Degree Distribution

Introduction:

Degree distribution describes the connectivity of vertices in a graph through the frequency distribution of degrees that are connected by edges across the network. This project analyzes a graph that is composed of airports and routes, where the nodes are represented by airports and edges are represented by routes. The degree of an airport is determined by the amount of direct connection it has to other airports. Based on the routes provided by the full_routes.csv (Full_Merge_of_All_Unique_Routes.csv) dataset, an airport is busier if its degree is higher. Statistics of minimum, maximum, mean, medium, and the percentiles of airport degrees are outputted, as well as a written file (Ranked Busiest Airports in the World by Degrees.csv) of the airports listed from the most to least degrees. The analysis reveals statistics about degrees at a distance of one neighbor, and a distance at two neighbors. At a distance of two neighbors, the concept of second-degree connections can reveal further insight of how busy an airport is in terms of how busy their neighbor airports are. Furthermore, the user is also prompted to input two airport IDs to find the shortest amount of switches, or edge connections, through the use of a Breadth-First-Search algorithm.

First Part of the Output:

Statistics of Airports for neighbors of distance 1 from full_routes.csv
Minimum Degree: 0
Maximum Degree: 1937
Mean Degree: 20.42
Median Degree: 0

Percentiles of Airports for neighbors of distance 1 from full_routes.csv
Percent of airports with < 100 degrees: 95.56%
Percent of airports with < 250 degrees: 2.36%
Percent of airports with < 500 degrees: 1.22%
Percent of airports with < 750 degrees: 0.48%
Percent of airports with < 1000 degrees: 0.23%
Percent of airports with < 1250 degrees: 0.11%
Percent of airports with < 1500 degrees: 0.02%
Percent of airports with < 1750 degrees: 0.00%
Percent of airports with < 2000 degrees: 0.01%

Statistics of Airports for neighbors of distance 2 from full_routes.csv
Minimum Degree: 0
Maximum Degree: 2260
Mean Degree: 108.46
Median Degree: 0

Percentiles of Airports for neighbors of distance 2 from full_routes.csv
Percent of airports with < 100 degrees: 75.60%
Percent of airports with < 250 degrees: 8.83%
Percent of airports with < 500 degrees: 9.20%
Percent of airports with < 750 degrees: 3.49%
Percent of airports with < 1000 degrees: 1.30%
Percent of airports with < 1250 degrees: 0.69%
Percent of airports with < 1500 degrees: 0.36%
Percent of airports with < 1750 degrees: 0.28%
Percent of airports with < 2000 degrees: 0.11%

The first part of the code's output describes the statistics of airports for neighbors at distance 1 and 2. By analyzing the airport degrees at distance 1, we find that a vertex with higher degrees is in a much smaller percentile than compared with lower degree vertices. For example, 95% of the vertices have 100 or less degrees, while only 0.01% of vertices have 1750-2000 vertices. A similar distribution is followed when you increase the distance to 2 neighbors. However, increasing the distance creates a more evenly spread distribution, revealing a more connective network.

Analyzing the Distribution:

The distribution of degrees for the airports in Full_Merge_of_All_Unique_Routes.csv looks to be like a power-law distribution, as the higher the degree an airport is, the more rare the node is compared to other airports.

Second Part of the Output:

Calculate the shortest path from input departure airport to destination airport
Please enter departure airport ID:
YUL
Please enter destination airport ID:
HKG
YUL -> HKG: 2 switches

The program then prompts the user to enter two airport IDs, one for a departure airport and another for a destination airport. It then returns the shortest amount of switches needed to get from the departure airport to the destination airport. The purpose of this function is to understand the connectivity of busiest or non-busiest airports from other specific airports.

Modules:

main.rs reads the data from the CSV file, full_routes.csv (Full_Merge_of_All_Unique_Routes.csv) and full_airports.csv (Full_Merge_of_All_Unique.csv). main.rs calls read_airports to construct a hash map where each key-value pair corresponds to an airport and its properties. Then, it calls read_routes so only airports from full_routes.csv are used, but full_airports.csv is utilized to find the names of each airport from full_routes.csv. main.rs calls update_degrees to iterate through each route to update the degree of each airport. It creates a file named “Ranked Busiest Airports in the World by Degrees.csv" that lists the airports from most to least degrees and calculates the statistics of the program. Finally, main.rs prompts the user for two airports and calls calculate_switches to return the shortest amount of switches to get from the two input airports.

lib.rs contains the functionalities and data structures used in the program:

The struct 'Airport’ stores each of the airport’s data, including the name, airport ID, degree at distance 1, and degree at distance 2.

The struct ‘Route’ contains the links between two airports, including the departure and destination airport IDs.

The function ‘read_airports’ reads the airport data from the csv file “full_airports.csv.”

The function ‘read_routes’ reads the route data from the csv file “full_routes.csv.”

The function ‘update degrees’ updates the degree of each airport based on the route data.

The function ‘calculate_statistics’ computes statistical metrics for the degrees.

graph.rs contains the ‘AirportGraph’ struct that implements graph-related functionalities, such as calculating the shortest path distances using Breadth-First Search (BFS).

Additional Notes:

Although the project technically only requires full_routes.csv for connectivity analysis, I decided to combine airports from full_airports.csv as well to match each ID to a corresponding airport. Although not all airports in full_airports.csv are used by routes, I believe it further emphasizes the distribution of how some airports are not as utilized as other airports.