CS 225 Final Project

Qianmeng Chen, Peilin He, Yikai Teng, Guangxun Zhai

University of Illinois at Urbana-Champaign

May 10, 2021

Our goals

Leading Question

What is the most important airport around the world?

Our Approach

The world's airline system can be viewed as a graph, with

- Airports acting as nodes of the graph.
- Airlines (flights) acting as edges of the graph, and length of the flights acting as weights of edges.

Assumptions

- The graph is undirected, since any flight has a corresponding return flight.
- The graph is "irreflexive", meaning there is no flight taking off and landing at the same airport.
- The graph is simple, since two same fights can be seen as one.

Accomplishments

Accomplishments

Within this project, we successfully accomplished the following:

- Read .dat files with given format as a graph.
- Traverse in the sense of breadth first search.
- Find a shortest path via Dijkstra Algorithm.
- Find a minimal spanning tree with Kruskal's Algorithm.
- Use page rank algorithm to determine which airport is the most inportant.
- Be able to visualize our result by rendering the graph to a "world map" PNG file.

Airport, Airline, and the World

The Airport Class

An airport consists of an 3 or 4 letter abbreviation, and its coordinates on Earth.

The Airline Class

An airline consists of two pointers of airports, representing the source and destination, and a way to calculate the spherical distance between the two airports.

The world- the graph class

We use **Adjacent Matrices** to implement the graph because we do not need to change the number of airports a lot once we read the files.

Processing .dat files

Overview

We need two .dat files, one for airports and the other for airlines.
The data we use for this project comes from "openflights.org".
In interpreting the data, if any broken data is observed, such as the latitude and longitude are NOT viable numbers, the corresponding entry will be neglected.

A file for airports

The 3-letter, 4-letter abbr, latitudes, and longitudes are required to be in the 5th, 6th, 7th, 8th columns respectively.

A file for airlines

The 3-letter or 4-letter abbr for the departing and arriving airports are required to be in the 3rd and 5th columns respectively.

Traversal and Shortest Path

Traversal

BFS algorithm:
We built a void BFS function and a function that can return a vector saving all the traversing airports pointers. The overall codes for BFS followed the lectures: creating a list to record the status of whether being visited and a queue saving to deal with nodes and adjacent nodes.

Shortest Path (Dijkstra's algorithm)

1. Update the neighbor's tentative distance value if we found a closer path.
2. Put the neighbor into the priority queue.
3. Put the current node into the visited set.

MST

Kruskal's Minimum Spanning Tree

We built a Minimum Spanning tree by inserting edges in sorted order and keep the tree acyclic without any cycle. With Disjoint sets, we can keep track of nodes that are in the same connected components. This function returns a vector that contains all the airline pointer that has been traversed.

Kruskal's Algorithm

1. Sort the edges in increasing order of weights
2. Initialize a separate disjoint set for each vertex
3. For each edge uv, determine if they are in different sets
4. If they are, add them to solution vector and union the sets

Page Rank

Helper Functions

- Construct a matrix to store the value of our adjacent matrix.
- Have a function to conduct matrix multiplication.
- Normalize the vector to prevent it from growing unlimited large.

Page Rank

We built a page rank function that can return a vector saving the results of our Markov Chain model. We count the number and quality of airline to a airport to determine a rough estimate of how important the airport is. This helps us to determine the most and top three important airports.

Running Time

BFS

$$O(n).$$

ShortestPath (Dijkstra)

$$O(|E| + |V|log(|V|)).$$

MST (Kruskal)

$$O(m * log(m)).$$

Tests conducted

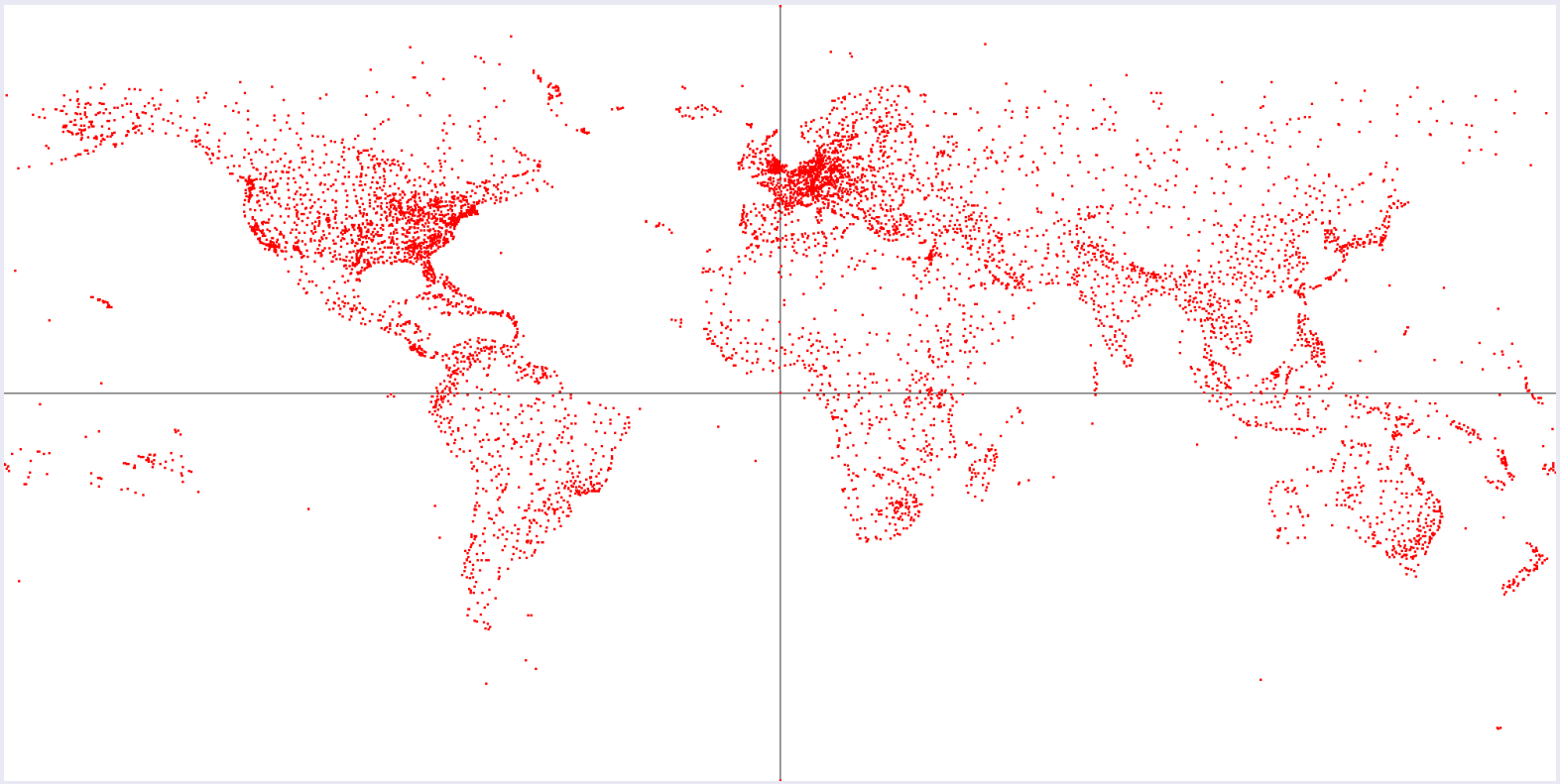
Error estimate via Richardson Extrapolation

We can use Richard extrapolation to achieve an error estimation. By choosing $n =, ,,$ we have that the order of the errors is

Some Visualizations

World Airports

Here is a plot of world airports (nodes of the graph). This also acts as a proof of the readFromFile works.



Some Visualizations, cont.

World Airlines

Here is a plot of world airlines (edges of the graph). This also acts as a proof of the readFromFile works.

Answer to the leading question

Running time analysis

By running Page Rank algorithm, with distance between airports as weights, we have a ranking of all recorded airports in the world. The results are:

- The moat important airport in the world is **FRA**, Frankfurt Airport in Germany.
- The second most important airport is **CDG**, Charles de Gaulle Airport in France.
- The third most important airport is **DXB**, Dubai International Airport in United Arab Emirates.