**CSCI 651 APPLICATIONS OF GRAPH THEORY TERM PROJECT SPRING 2022**

AIR TRANSPORT NETWORK ANALYSIS

**Final Report**

**Team Members**

Mounika Thudi - z1948596

Akhila Priyanka Thirunagaru - z1916001

Gopi Sai Krishna Vellanki - z1944991

# 

# Content

|  |  |
| --- | --- |
| Introduction …………………………………………………………………………………... | 2 |
| Goals ………………………………………………………………………………………….. | 2 |
| Data …………………………………………………………………………………………... | 2 |
| Methods Implemented ……………………………………………………………………... | 3 |
| Programming Languages ……………………………………………………………………... | 4 |
| Results ………………………………………………………………………………………... | 6 |
| Discussions…………………………………………………………………………………… | 8 |
| Future Work …………………………………………………………………………………... | 8 |

Introduction

Graph theory is used to identify communities within networks; the graphs are used as a device for modeling and describing network systems of real-world systems. It is an easy way to represent the relationship between two objects in the real world. In this project, graphs are used to construct the relationship between airports and the flights between one and the other.

Flight Network is very important in terms of traveling considering factors like time and cost. In recent times, there has been significant growth for traveling through flights.

By the end of the project, following questions are answered:

1. Minimum spanning tree using Kruskal’s algorithm &
2. Shortest path between two vertices (i.e., airports) using Dijkstra’s algorithm

Goals

Project goals are:

* Constructing minimum spanning tree using Kruskal’s algorithm connecting all airports
* Finding shortest path between two airports using Dijkstra’s algorithm

Data

Data has been extracted from Kaggle which has information about 2018 USA airports and flights. It is in CSV format which has 9M+ rows and 14 columns.

<https://www.kaggle.com/datasets/zernach/2018-airplane-flights>

The dataset has below columns -

1. Unnamed: drop this column (it's a duplicate index column)

2-3. ItinID & MktID: vaguely demonstrates the order in which tickets were ordered (lower ID #'s being ordered first)

4. MktCoupons: the number of coupons in the market for that flight

5. Quarter: 1, 2, 3, or 4, all of which are in 2018

6. Origin: the city out of which the flight begins

7. OriginWac: USA State/Territory World Area Code

8. Dest: the city out of which the flight begins

9. DestWac: USA State/Territory World Area Code

10. Miles: the number of miles traveled

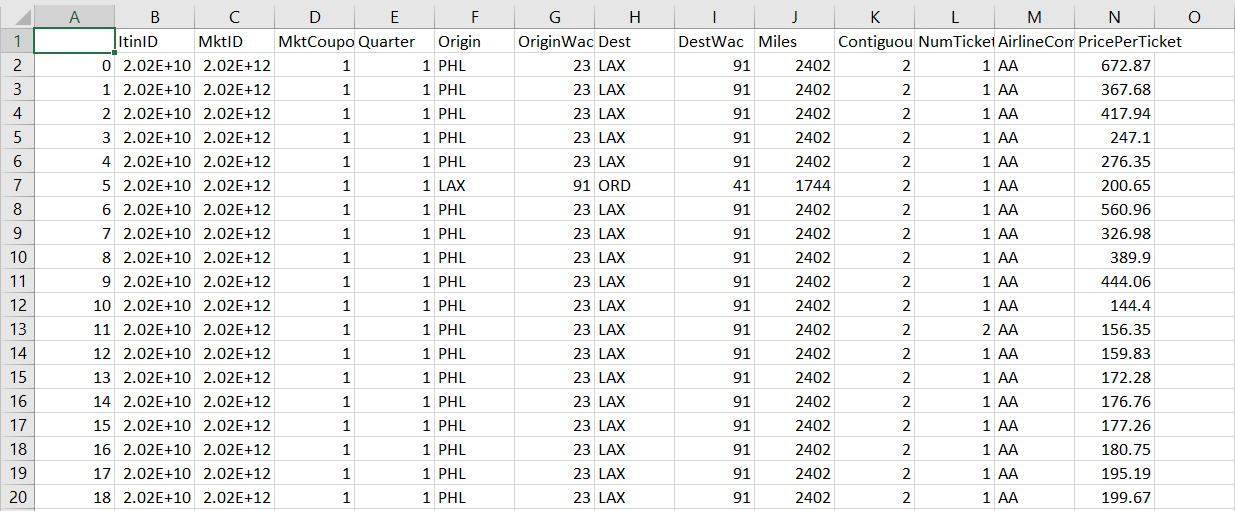
11. ContiguousUSA: binary column -- (2) meaning flight is in the contiguous (48) USA states, and (1) meaning it is not (i.e.: Hawaii, Alaska, off-shore territories)

12. NumTicketsOrdered: number of tickets that were purchased by the user

13. Airline Company: the two-letter airline company code that the user used from start to finish (key codes below)

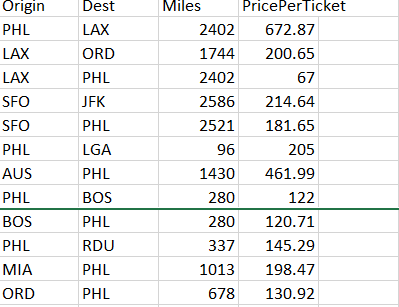
14. PricePerTicket: target prediction column

The following is the snapshot of dataset with few rows.



Only relevant columns like origin, destination, ticket price, distance etc., are used and remaining columns are dropped.

The following is the snapshot of filtered data



Methods Implemented

A graph is defined as the finite number of points (also known as vertices) connected by lines (also known as edge weights). Weighted graphs are being constructed in which airports act as vertices and Ticket Price as edge weight.

Steps involved in the project:

1. **Importing all the necessary libraries**
2. **Filtering Dataset**
   1. Duplicates are dropped from the dataset
   2. Origin, Dest, Miles & PricePerTicket are the only columns used
3. **Constructing minimum spanning tree using Kruskal’s algorithm**

Given a connected and undirected graph, a spanning tree of a graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. A minimum spanning tree or minimum weight spanning tree for a weighted, connected, undirected graph is a spanning tree with a weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree.

Below are the steps for finding minimum spanning tree using Kruskal Algorithm:

* Sort all the edges in non-decreasing order of their weight.
* Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.
* Repeat second step until there are (v-1) edges in the spanning tree. (Where v is number of vertices)

1. **Finding shortest path between two airports given by the user**

There will be different path options to reach from airport X to airport Y, but the aim is to find the shortest path with a minimum cost. Firstly, we start from airport X, which is chosen as a permanent airport. Analyzing the distances of the neighborhood airports from airport X, the shortest path to airport Y will be identified. Afterwards, airport Y will be considered as a permanent airport and follows the same procedure to find the shortest path from that airport, and the same continues until the destination airport has been reached. By this, we find the shortest path between two airports i.e., source and destination.

The network vertex (airport or city) at which a flight takes off is called the origin while the network vertex at which flight lands is called the destination. Network vertices are referenced as cities or airports. If network vertex is called a city, it is assumed that there is an airport in that city.

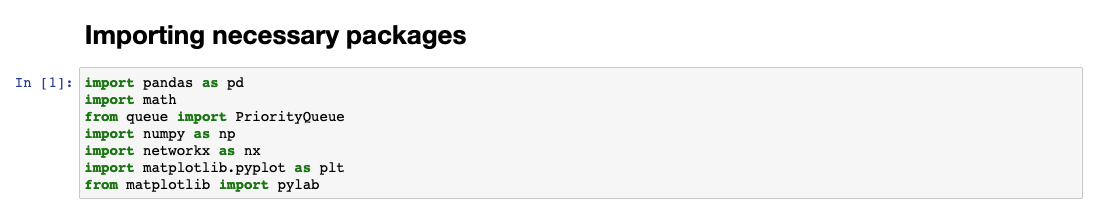
Programming Languages

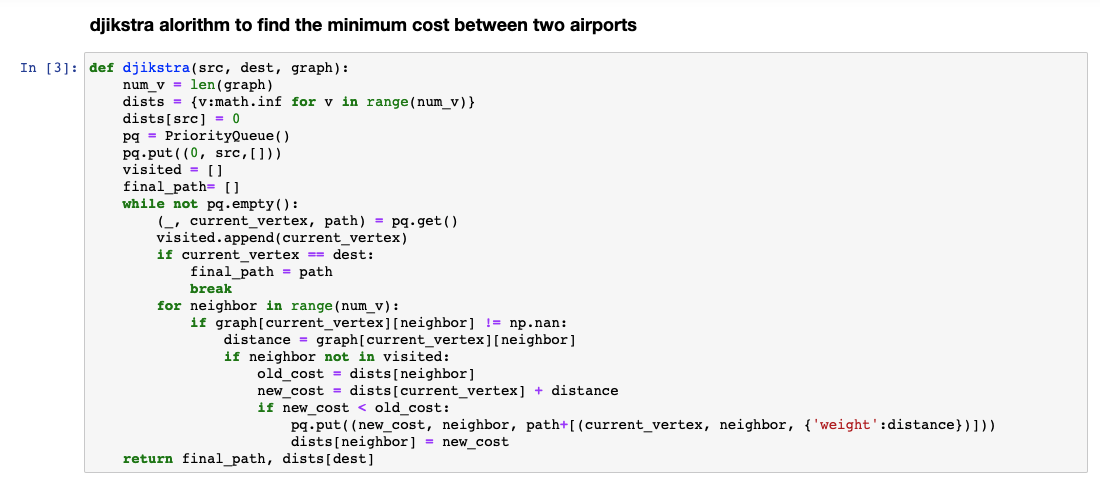
**Programming language:** We have used python language to develop this term project.

We have coded whole project in single file using different cells.

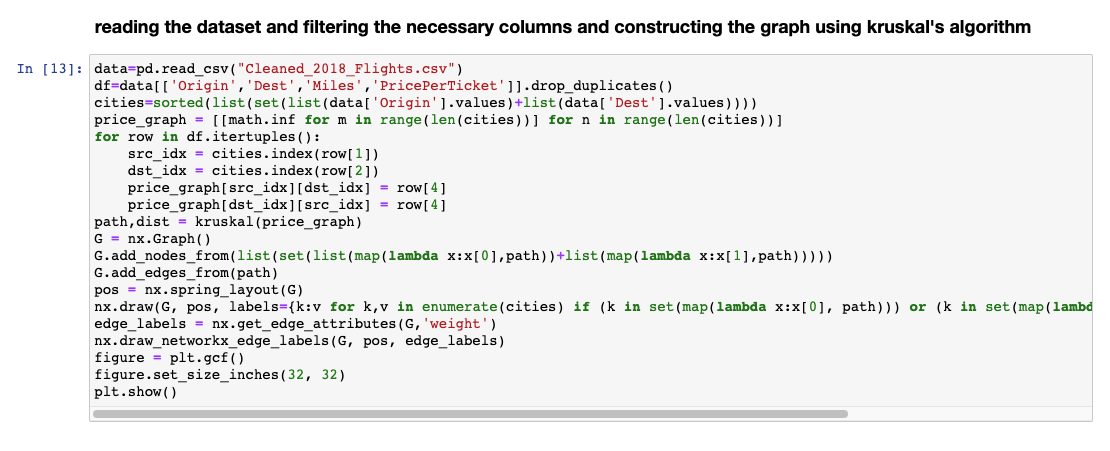
**Libraries used in project:** matplotlib (plots and displaying data, statistics), pandas (organizing and better handling the data frame), Networkx (the creation and manipulation of graphs and networks), math for calculations of distance and cost, queue, numpy & basemap (visualization).

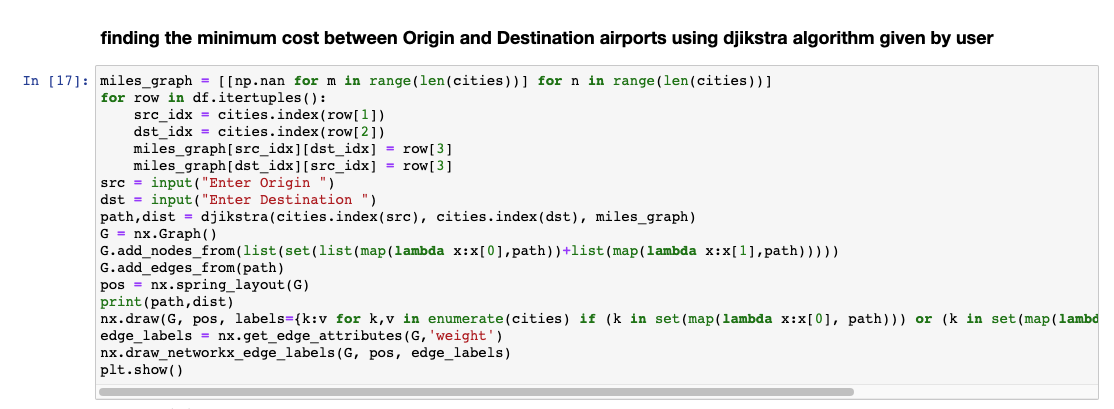
The following are the snapshots of code









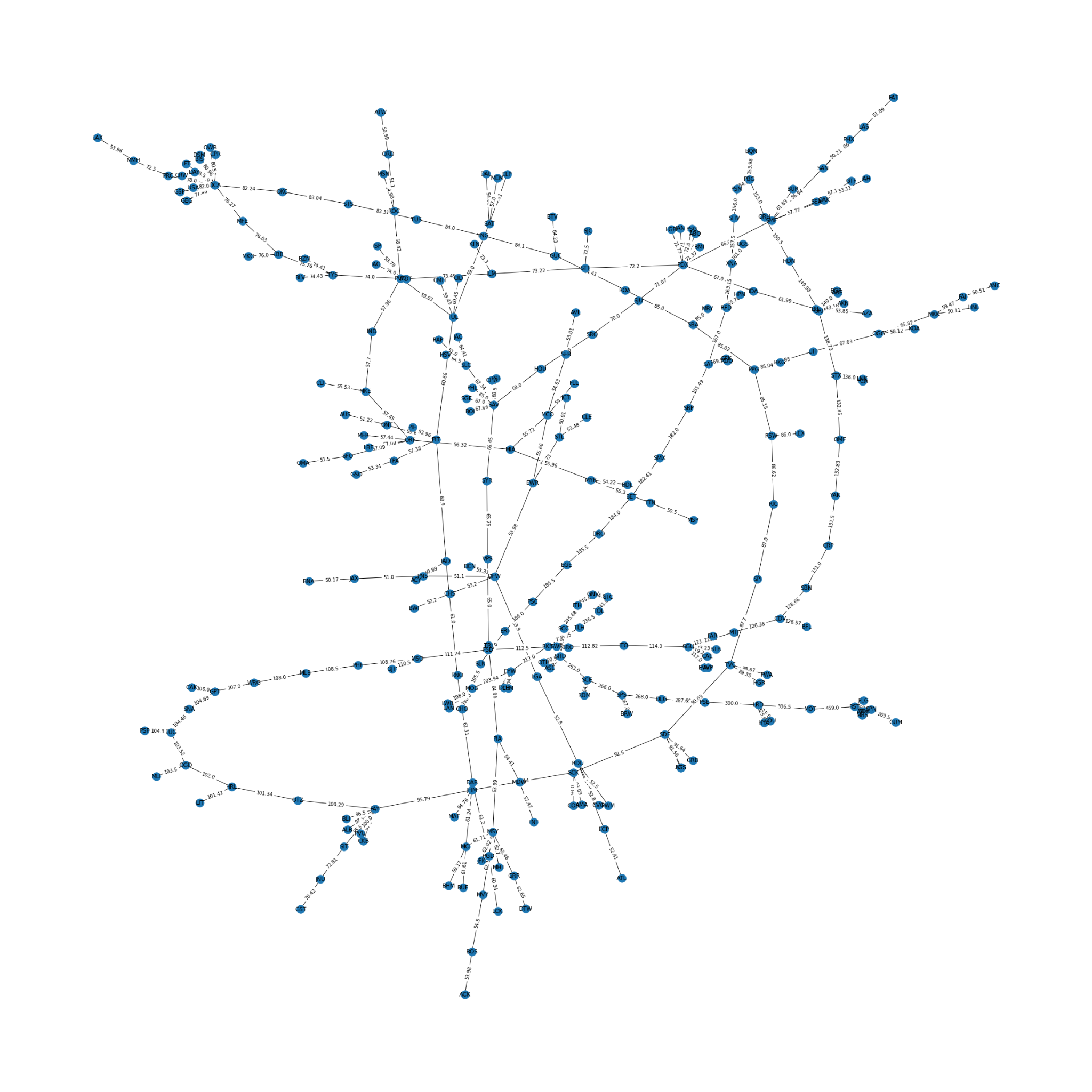


Results

**Minimum spanning tree using Kruskal’s algorithm is below:**

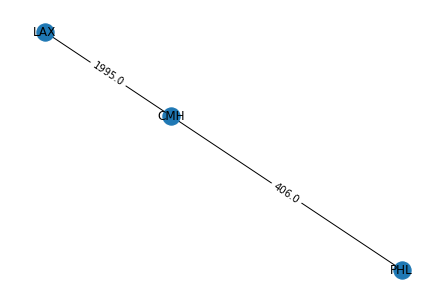
Statistics of constructed graph are:

* Number of vertices in the graph constructed are about 262
* Number of edges of minimum spanning tree is 261(i.e., (v-1) edges, where v is number of vertices in a graph)
* Maximum degree of one of the vertices in a graph is 10
* Degree of airport (i.e., vertex) is the number of airports connected to that particular airport.
* The number of vertices with odd degrees are even in this graph.
* The sum of the vertex degree is twice the number of edges, that is 588 in this case.



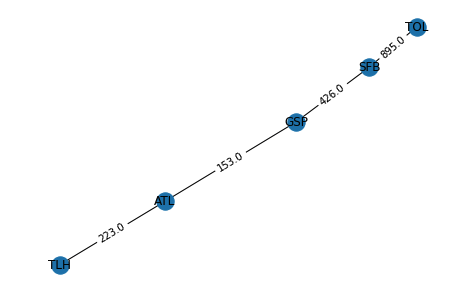
**Shortest path between two airports**

Considering LAX as origin and PHL as destination, the fastest route is through CMH



Total distance between LAX & PHL is LAX to CMH + CMH to PHL i.e., 1995 + 406 = 2401, which is less than the actual distance between PHL and LAX.

Similarly, by taking TLH, TOL as source and destination,

Below is the shortest path obtained.

Discussions

The Dijkstra algorithm will find the fastest route between two points. The Kruskal algorithm will find the minimum tree that connects all the given vertices. The results that are obtained for example show that Dijkstra algorithm is a very effective tool to find the way with the lowest cost from any airport. I also tried to find the worst-case scenario to reach from one airport to another airport which is about 40-50% more expensive.

Future Work

In project we have taken USA 2018 flights details to work. Likewise, we would implement air transportation network analysis of another country. As part of future work, we can continue to find eccentricity, radius of network.