

Project Goals

Summary of dataset : Using openflights dataset

This dataset contains information about the world airline network, where each vertex represents an airport and each edge represents a directed flight between two airports. This dataset also includes the airport enplanement for 447 airports located in the United States.

Reference: <https://www.cs.cornell.edu/~arb/data/spatial-OpenFlights/>

We decide to use the open flights dataset. <https://openflights.org/data.html> This dataset has a format of

Airport ID	Unique OpenFlights identifier for this airport.
Name	Name of airport. May or may not contain the City name.
City	Main city served by airport. May be spelled differently from Name.
Country	Country or territory where airport is located. See <u>Countries</u> to cross-reference to ISO 3166-1 codes.
IATA	3-letter IATA code. Null if not assigned/unknown.
ICAO	4-letter ICAO code. Null if not assigned.
Latitude	Decimal degrees, usually to six significant digits. Negative is South, positive is North.
Longitude	Decimal degrees, usually to six significant digits. Negative is West, positive is East.
Altitude	In feet.
Timezone	Hours offset from UTC. Fractional hours are expressed as decimals, eg. India is 5.5.
DST	Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown). See also: <u>Help: Time</u>
Tz database time zone	Timezone in <u>"tz" (Olson) format</u> , eg. "America/Los_Angeles".
Type	Type of the airport. Value "airport" for air terminals, "station" for train stations, "port" for ferry terminals and "unknown" if not known. In airports.csv, only type=airport is included.
Source	Source of this data. "OurAirports" for data sourced from <u>OurAirports</u> , "Legacy" for old data not matched to OurAirports (mostly DAFIF),

"User" for unverified user contributions. In airports.csv, only source=OurAirports is included.

Traversals:

DFS

Covered Algorithms:

Shortest Path: Dijkstra's Algorithm

Uncovered Options:

Landmark Path

We decided to implement the algorithm above to find the shortest path between cities. Some conditions might be given. For example, the path must go through some city other than the starting point and the ending point.