CS 225 Project Goals

Project Idea

Our goal for this project is to utilize the OpenFlights airports data set to perform a BFS traversal and shortest path finding. For shortest path finding, we will use the coordinates of each airport to find the shortest route of two given points using Dijkstra's Shortest Path Algorithm. We also plan on using the Landmark Path algorithm to find the shortest path between two given points that goes through another point.

Data Set

For our project, we will be using OpenFlights airport dataset which contains entries for over 10,000 airports, train stations, and ferry ports.

- **(SOURCE LOCATION)** https://openflights.org/data.html
- **(FORMAT)** Each entry contains the following information:
    - Airport ID - unique OpenFlights identifier
    - Name - name of airport
    - City - where airport is located
    - Country - where airport is located
    - IATA Code
    - ICAO Code
    - Latitude - decimal degrees to six sig figs (negative = South, positive = North)
    - Longitude - decimal degrees to six sig figs (negative = West, positive = East)
    - Altitude - in ft.
    - Timezone - offset from UTC
    - DST - daylight savings time
    - Tz Database Timezone - in tz format
    - Type - airport type (airport, train station, ferry port, unknown)
    - Source - data source
- Sample entry:
    - `507,"London Heathrow Airport","London","United Kingdom","LHR","EGLL",51.4706,-0.461941,83,0,"E","Europe/London","airport","OurAirports"`

Traversal

- BFS (Breadth First Search Traversal)
    - A traversal visits every node once in a specific order.

- A BFS traversal starts with the current or initial node, then visits all of its neighboring nodes on the current level, then visits all neighbor nodes on the next level, and so on until all nodes have been visited once.

Algorithms

- Shortest Path using Dijkstra's Algorithm
  - Given a graph and source vertex in the graph, Dijkstra's algorithm finds the shortest paths from the source to all vertices in the given graph.
    - Generate an SPT (shortest path tree) with a given source as the root.
    - Maintain two sets; one set contains vertices included in SPT, other set contains vertices not yet included in SPT.
    - At every step of the algorithm, we find a vertex in the other set that has a minimum distance from the source.
  - Time complexity of this algorithm is O(E log (V)) where V is the number of vertices and E is the total number of edges

- Landmark Path (complex/uncovered)
  - Finds approximate shortest paths in weighted graphs
    - Landmarks are placed at vertices throughout the graph, and each landmark stores the shortest path from itself to all other vertices
    - Searches the intersection points along the landmark paths to approximate a path between two points

Schedule

First Week Deliverables:

- Submit Team Contract (11/18)
- Submit GOALS (11/18)
- Project Goals Confirmation
- Update DEVELOPMENT (11/20)

Second Week Deliverables:

- Design Project
- Begin Coding (Finish ½ Project)
- Update DEVELOPMENT (11/26)

Third Week Deliverables:

- Finish Coding (12/11)
- Begin Final Project Presentation/Written Report
- Mid-project Check-in Meeting (12/4)
- Update DEVELOPMENT (12/4)

Fourth Week:

- Final Project Presentation (12/11)
- Written Report (12/11)
- Update DEVELOPMENT (12/11)

CALENDAR

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
11/16	11/17	11/18 Submit team contract & goals	11/19	11/20	11/21	11/22 Bi-weekly meeting
11/23	11/24 Bi-weekly meeting	11/25	11/26	11/27	11/28	11/29 Bi-weekly meeting
11/30	12/1 Bi-weekly meeting	12/2	12/3	12/4 Final day for mid-project check-in meetings	12/5	12/6 Bi-weekly meeting
12/7	12/8 Bi-weekly meeting	12/9	12/10	12/11 **Final project due**		