CS225 Final Project Proposal:

anwesha6, mniwas2, rohanjs3, sofials3

1. *Leading Question*

In our final project we aim to optimize daily travel in New York City by using the graph algorithms of Breadth First Search (BFS), Dijkstra's Algorithm and Page Rank Algorithm. Our project will create a tool that finds the shortest route between two different bus stops. We will be using data from (New York City Bus Data | Kaggle) that will provide information on the starting and ending locations of bus routes.Using our algorithms we will be able to aid people in their daily commutes.

2. *Dataset Acquisition, Processing and Storage*

In this project, bus locations and routes will be represented in a graph. We are pulling our first dataset from *Kaggle* where we have located 4 CSV files that provide detailed information regarding New York City's bus routes. The current dataset contains values that represent direction reference, origin location (name, longitude, and latitude), and respective information for the destination as well. If any of the data is incorrect or does not have a proper weightage, therefore being invalid, would not be considered in our project as we do not want it to affect the result. For data storage, we will use an adjacency list to store all of our values that we gather from the implementation of the PageRank algorithm. Our top picked values will then be connected to the most traveled locations in New York City.

3. *Graph Algorithms: (BFS, Page Rank, and Dijkstra's Algorithm)*

a. Function Inputs + Outputs
    i. Our dataset contains origin latitude and longitude as well as destination latitude and longitude for each row. All nodes will be defined by the origin latitude and longitudes, and they will be directed toward the nodes corresponding to the destination latitude and longitude. In this way, the edges are routes between bus stops.
    ii. "Find" will accept a longitude and latitude. Then, we will use a BFS algorithm to find the bus stop closest to the coordinates given. This can be used to solve the problem of searching for the nearest bus stop to a given location. The user can call this function to solve the smaller problem of finding a bus stop nearby, but this function will also be used when finding the shortest path between two bus stops and tourist locations.
        i. The PageRank is essentially used to rank a set of values against each other and in our given scenario, we will use it to figure out all of the important

tourist locations in NYC and how the bus routes are positioned accordingly. We will provide 10 locations that are scattered across New York City to be used in the PageRank algorithm.

ii. Dijkstra's Algorithm will take in consideration the starting latitude and longitude as well as the destination latitude and longitude. We will return a list of bus stops on the shortest route between the starting point and end point.

b. Function Efficiency

Our BFS algorithm will have a run time of O(n) by traversing the graph and keeping track of all visited nodes until a target node is reached. Dijkstra's algorithm will have a priority queue that has a runtime of O(Edges + Vertices*Log(Vertices)).

4. *Timeline*

a. Week 11/5-11/11:
   i. Set up workspace (github, vscode environment)
   ii. Install dependencies (datasets, any libraries needed)
   iii. Data formatting

b. Week 11/12-11/18: Midpoint Check-in at end of week
   i. Data correction
   ii. Data Storage
   iii. Implement Find

c. Week 11/19-11/25 (Thanksgiving Break):
   i. Implement Landmark algorithm
   ii. Write tests for the Landmark algorithm
   iii. Implement Dijkstra's algorithm
   iv. Write tests for the Dijkstra's algorithm

d. Week 11/26-12/2:
   i. Fix errors in algorithm and test edge cases
   ii. Clean up code and add any missing comments

e. Week 12/3-12/8: Finish Project (Due 12/8):
   i. Finish README
   ii. Write results.md
   iii. Prepare final presentation video