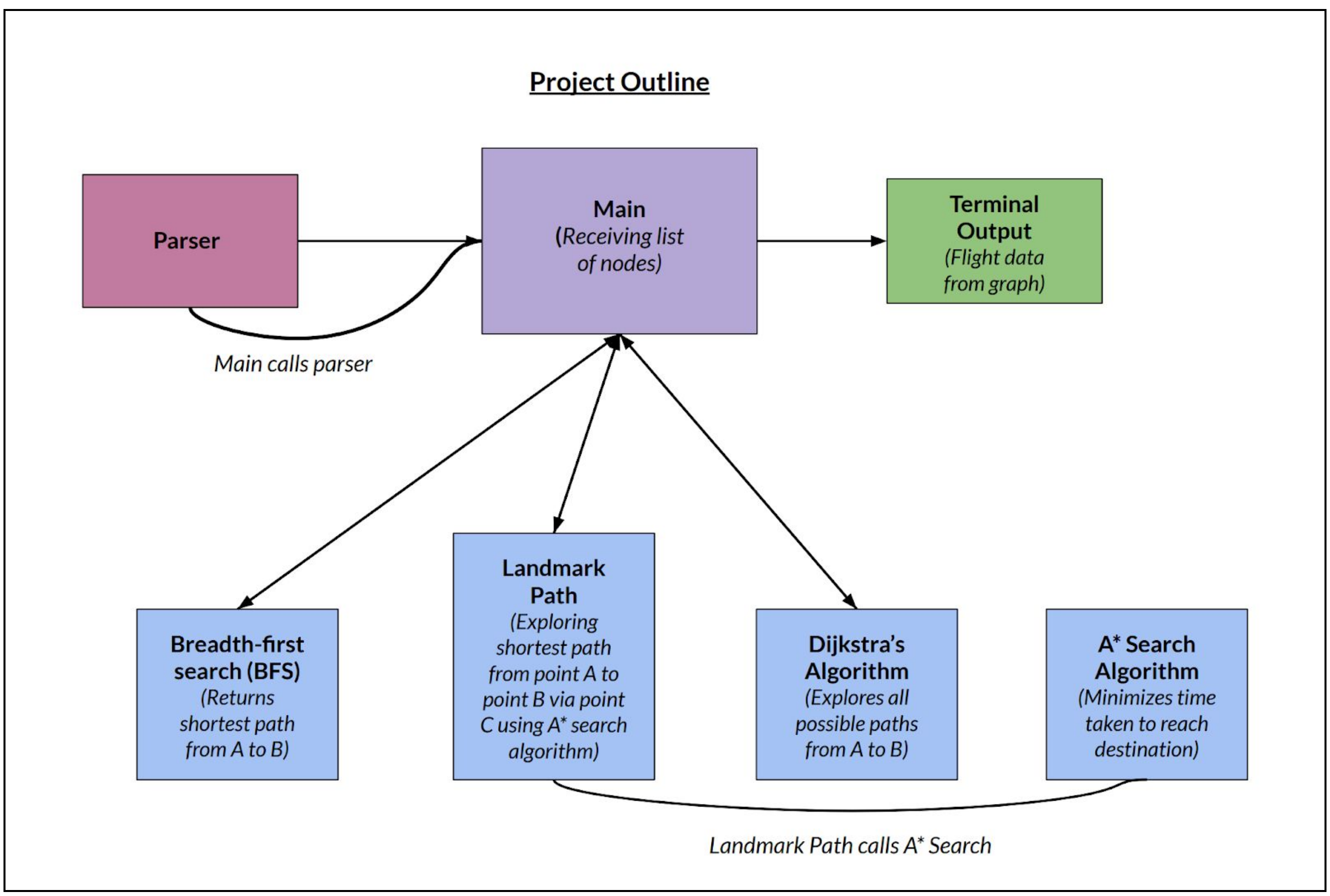
pmunshi2, rohanb4, anuraag3, abhayn2

CS 225 Final Project - Final Report (Results)

The primary goal for this project was to find the most distance-efficient route from a given city to a destination city with a flight dataset. We utilized the Harvard dataset and extracted all the flight data from 2008 into CSV files. The CSV dataset contains a variety of information including the distance between airports, departure time, arrival time, departure airport, arrival airport, flight time, scheduled timings, and frequency of every flight. We use a few of these parameters in our calculations: airport codes are used to form vertices of the graph, edges store flight information between airports such as distance and a string representing additional details.

We used three primary algorithms to design our project functionality. Our first major algorithm was Breadth-First Search (BFS) which finds the shortest path (least number of edges traversed) from a source vertex to a destination vertex. In our directional graph, the vertices represent airports and edges represent flight options between those airports. Thus, BFS allows us to find the shortest path from one airport to the other using the least number of flights (or stopovers).

The next algorithms used in the project were A* Search and Dijkstra's algorithm. These achieve the functionality of finding the path between airports with the least distance between them. Since our graph is weighted with distance for each edge, these search algorithms make use of the weights to determine the least distance-consuming path. We also utilized the Landmark Path algorithm as a further extension. It is used in order to get from point A to point B via point C. This utilizes the A* search algorithm at the backend to find the routes with minimal distance.



pmunshi2, rohanb4, anuraag3, abhayn2

All members of our group were undertaking something new with the A* and Dijkstra search algorithms, and they were quite interesting to learn about. There wasn't too much difference between these two approaches - both can be said to be cousins of each other, with the only difference being that A* utilizes a heuristics function, along with the general cost function like Dijkstra, to find the best, shortest path from Point A to B. The similarities and the differences between these two algorithms became very apparent to us when we proceeded to implement them in our project.

During the implementation process, almost all of us faced difficulties in integrating the pseudocode and theoretical implementations of the algorithms into the scope of our own project and dataset. Thus, we learned a lot about the practical applications of these (not just understanding how they work theoretically), which will surely be of huge help to us further down the line. We were able to resolve any issues any individual member had by doing extensive research and discussing during our regular meetings.

The motivation behind this project was to provide an efficient way for passengers to search for their optimal flights depending on their preferred factors such as number of stops or least distance. Although these searches are possible using basic imperative programming, the usage of graphs and the aforementioned search algorithms makes the program highly efficient in time and storage complexity requirements. In terms of future scope, similarly designed projects can be used for a variety of reservation and search systems for public transportation to use fewer computing resources and we hope this project motivates such endeavors.

The appendices below show the different kinds of output when the program is run, with each of the first three images showing the implementation of a single algorithm: Dijkstra's algorithm, Landmark path coupled with A* search, and the BFS traversal respectively. This is done by the user inputting a 1, 2 or 3 to demonstrate their choice of algorithm, which corresponds to the different purposes already discussed in the goals and development files. The images in the Appendix below all have the same initial source and destination airports, and show the possible paths for each algorithm given the constraints (the Landmark intermediary airport) when Option 2 (A*) is selected by the user. This is given along with their respective departure and arrival times, thereby plotting the entire flight travel for the user, as is the intended use of our program.

pmunshi2, rohanb4, anuraag3, abhayn2

Appendix:

Dijkstra’s Algorithm Output

```
pmunshi2@Olil3-PC:/mnt/d/225_final_project$ ./finalproj
Enter the location of the dataset, or type 'default' for default dataset (dataset/1987.csv): dataset/2008.csv
Successfully read file! (Errors printed to console if any)
/****************************************************************************************************************************/
Please choose one of the following:
Enter '1': Shortest Flight route from Point A to Point B w.r.t. distance. (Using Dijkstra's Algorithm)
Enter '2': Shortest Flight route from Point A to Point B through Point C w.r.t. distance. (Using Landmark Path and A* search)
Enter '3': Least number of stops from Point A to Point B (Using BFS - this may result in a longer route w.r.t. distance).
Choice: 1
Please enter the source airport: CMI
Please enter the destination airport: TEX
Printing Flight Data

CMI   ----------------------------------->  DFW   ----------------------------------->  PHX   ----------------------------------->  TEX
     MQ3450 -> Dep: 0635hrs - Arr: 0855hrs          US32 -> Dep: 1525hrs - Arr: 1659hrs        YV2951 -> Dep: 0945hrs - Arr: 1137hrs
      US550 -> Dep: 2020hrs - Arr: 2155hrs        YV2953 -> Dep: 1055hrs - Arr: 1246hrs
      US191 -> Dep: 1715hrs - Arr: 1859hrs        YV2882 -> Dep: 1055hrs - Arr: 1251hrs
      US627 -> Dep: 1825hrs - Arr: 2009hrs
      US476 -> Dep: 1155hrs - Arr: 1336hrs
      US308 -> Dep: 1525hrs - Arr: 1659hrs
      US641 -> Dep: 0555hrs - Arr: 0730hrs
      US541 -> Dep: 1830hrs - Arr: 2013hrs
     AA1123 -> Dep: 0720hrs - Arr: 0900hrs
      US450 -> Dep: 1155hrs - Arr: 1336hrs
     AA1205 -> Dep: 0955hrs - Arr: 1135hrs
      US520 -> Dep: 0555hrs - Arr: 0739hrs
      AA567 -> Dep: 1120hrs - Arr: 1255hrs
      US524 -> Dep: 1200hrs - Arr: 1335hrs
      US401 -> Dep: 2014hrs - Arr: 2049hrs
     AA1801 -> Dep: 1955hrs - Arr: 2130hrs
     AA1279 -> Dep: 1225hrs - Arr: 1400hrs
      US625 -> Dep: 1200hrs - Arr: 1335hrs
     AA1445 -> Dep: 1620hrs - Arr: 1655hrs
     AA1457 -> Dep: 1355hrs - Arr: 1530hrs
     AA1607 -> Dep: 1545hrs - Arr: 1720hrs
      US586 -> Dep: 0840hrs - Arr: 1021hrs
      AA707 -> Dep: 1110hrs - Arr: 1140hrs
     AA1695 -> Dep: 0835hrs - Arr: 1015hrs
     AA2285 -> Dep: 2235hrs - Arr: 2300hrs
     AA1955 -> Dep: 2200hrs - Arr: 2330hrs
     AA1875 -> Dep: 1740hrs - Arr: 1915hrs
       US38 -> Dep: 2014hrs - Arr: 2155hrs
      US639 -> Dep: 1825hrs - Arr: 2009hrs
       US29 -> Dep: 1155hrs - Arr: 1236hrs
      US324 -> Dep: 0825hrs - Arr: 0901hrs
     AA1031 -> Dep: 0955hrs - Arr: 1030hrs

Enter 'x' to try another search. Enter anything else to exit.
Choice:
```

pmunshi2, rohanb4, anuraag3, abhayn2

Landmark Path and A* Search Algorithm Output:

```
Please choose one of the following:
Enter '1': Shortest Flight route from Point A to Point B w.r.t. distance. (Using Dijkstra's Algorithm)
Enter '2': Shortest Flight route from Point A to Point B through Point C w.r.t. distance. (Using Landmark Path and A* search)
Enter '3': Least number of stops from Point A to Point B (Using BFS - this may result in a longer route w.r.t. distance).
Choice: 2
Please enter the source airport: CMI
Please enter the landmark airport: ATW
Please enter the destination airport: TEX
Printing Flight Data

CMI   ------------------------------------>   ORD   ------------------------------------>   ATW   ------------------------------------>   DSM   ------------------------------------>   PHX   ------------------------------------>   TEX
     MQ4373 -> Dep: 0910hrs - Arr: 1010hrs         YV7123 -> Dep: 1155hrs - Arr: 1249hrs         OO6777 -> Dep: 0724hrs - Arr: 0814hrs         YV2748 -> Dep: 0810hrs - Arr: 1020hrs         YV2951 -> Dep: 0945hrs - Arr: 1137hrs
     MQ4129 -> Dep: 1530hrs - Arr: 1630hrs         OO5855 -> Dep: 0800hrs - Arr: 0856hrs                                                       YV2929 -> Dep: 1755hrs - Arr: 2005hrs         YV2953 -> Dep: 1055hrs - Arr: 1246hrs
     MQ4218 -> Dep: 1815hrs - Arr: 1920hrs         YV7053 -> Dep: 2139hrs - Arr: 2231hrs                                                       YV2743 -> Dep: 1450hrs - Arr: 1659hrs         YV2882 -> Dep: 1055hrs - Arr: 1251hrs
     MQ3905 -> Dep: 1235hrs - Arr: 1329hrs         OO6195 -> Dep: 1155hrs - Arr: 1248hrs                                                       YV2722 -> Dep: 1450hrs - Arr: 1556hrs
     MQ4278 -> Dep: 0710hrs - Arr: 0810hrs         YV7085 -> Dep: 1647hrs - Arr: 1741hrs                                                       YV2944 -> Dep: 1450hrs - Arr: 1657hrs
     MQ4374 -> Dep: 0610hrs - Arr: 0705hrs         OO6153 -> Dep: 1645hrs - Arr: 1741hrs                                                       YV2814 -> Dep: 1450hrs - Arr: 1556hrs
     MQ4401 -> Dep: 0810hrs - Arr: 0910hrs         YV7136 -> Dep: 1445hrs - Arr: 1538hrs
     MQ4052 -> Dep: 1045hrs - Arr: 1140hrs         OO5873 -> Dep: 1015hrs - Arr: 1109hrs
     OO6175 -> Dep: 1007hrs - Arr: 1103hrs
     YV7423 -> Dep: 1155hrs - Arr: 1248hrs
     YV7137 -> Dep: 1955hrs - Arr: 2052hrs
     OO5845 -> Dep: 1155hrs - Arr: 1249hrs
     OO5832 -> Dep: 1000hrs - Arr: 1056hrs
     OO6095 -> Dep: 1955hrs - Arr: 2051hrs
     OO6315 -> Dep: 1445hrs - Arr: 1541hrs
     YV7101 -> Dep: 1004hrs - Arr: 1100hrs
     YV7143 -> Dep: 1445hrs - Arr: 1539hrs
     OO5937 -> Dep: 1445hrs - Arr: 1541hrs
     OO6075 -> Dep: 1955hrs - Arr: 2051hrs
     OO5941 -> Dep: 1955hrs - Arr: 2052hrs
     OO6300 -> Dep: 1445hrs - Arr: 1541hrs
     YV7104 -> Dep: 1642hrs - Arr: 1738hrs
     YV7191 -> Dep: 2130hrs - Arr: 2226hrs
     YV7351 -> Dep: 1955hrs - Arr: 2051hrs
     OO5939 -> Dep: 2130hrs - Arr: 2227hrs
     YV7184 -> Dep: 1007hrs - Arr: 1103hrs
     YV7290 -> Dep: 0952hrs - Arr: 1048hrs
     YV7321 -> Dep: 2140hrs - Arr: 2236hrs
     YV7300 -> Dep: 1445hrs - Arr: 1541hrs
     OO6005 -> Dep: 1155hrs - Arr: 1251hrs
     OO6459 -> Dep: 1155hrs - Arr: 1249hrs
     OO5960 -> Dep: 1645hrs - Arr: 1741hrs
     YV7112 -> Dep: 1955hrs - Arr: 2050hrs
     YV7109 -> Dep: 1155hrs - Arr: 1249hrs
     YV7061 -> Dep: 1445hrs - Arr: 1539hrs
     YV7253 -> Dep: 0800hrs - Arr: 0855hrs
     OO5843 -> Dep: 1545hrs - Arr: 1639hrs
     OO6079 -> Dep: 1004hrs - Arr: 1059hrs
     OO6790 -> Dep: 1955hrs - Arr: 2050hrs

Enter 'x' to try another search. Enter anything else to exit.
Choice:
```

pmunshi2, rohanb4, anuraag3, abhayn2

Breadth-First Search Traversal Output:

```
Please choose one of the following:
Enter '1': Shortest Flight route from Point A to Point B w.r.t. distance. (Using Dijkstra's Algorithm)
Enter '2': Shortest Flight route from Point A to Point B through Point C w.r.t. distance. (Using Landmark Path and A* search)
Enter '3': Least number of stops from Point A to Point B (Using BFS - this may result in a longer route w.r.t. distance).
Choice: 3
Please enter the source airport: CMI
Please enter the destination airport: TEX
Printing Flight Data

CMI  ----------------------------------->  ORD  ----------------------------------->  PHX  ----------------------------------->  TEX
     MQ4373 -> Dep: 0910hrs - Arr: 1010hrs      UA1477 -> Dep: 0820hrs - Arr: 1112hrs      YV2951 -> Dep: 0945hrs - Arr: 1137hrs
     MQ4129 -> Dep: 1530hrs - Arr: 1630hrs         US8 -> Dep: 1723hrs - Arr: 2015hrs      YV2953 -> Dep: 1055hrs - Arr: 1246hrs
     MQ4218 -> Dep: 1815hrs - Arr: 1920hrs      UA1479 -> Dep: 1155hrs - Arr: 1447hrs      YV2882 -> Dep: 1055hrs - Arr: 1251hrs
     MQ3905 -> Dep: 1235hrs - Arr: 1329hrs       US188 -> Dep: 1710hrs - Arr: 2004hrs
     MQ4278 -> Dep: 0710hrs - Arr: 0810hrs      UA1495 -> Dep: 1455hrs - Arr: 1747hrs
     MQ4374 -> Dep: 0610hrs - Arr: 0705hrs       UA499 -> Dep: 1200hrs - Arr: 1449hrs
     MQ4401 -> Dep: 0810hrs - Arr: 0910hrs      UA1497 -> Dep: 1740hrs - Arr: 2032hrs
     MQ4052 -> Dep: 1045hrs - Arr: 1140hrs      AA2435 -> Dep: 1220hrs - Arr: 1510hrs
       US38 -> Dep: 1850hrs - Arr: 2037hrs
      US294 -> Dep: 1405hrs - Arr: 1656hrs
       US92 -> Dep: 1405hrs - Arr: 1656hrs
        US4 -> Dep: 1040hrs - Arr: 1333hrs
       US10 -> Dep: 1900hrs - Arr: 2152hrs
     UA1499 -> Dep: 2015hrs - Arr: 2307hrs
     AA2455 -> Dep: 0715hrs - Arr: 1005hrs
      US472 -> Dep: 0725hrs - Arr: 1016hrs
      US476 -> Dep: 1040hrs - Arr: 1334hrs
      AA593 -> Dep: 1520hrs - Arr: 1805hrs
      US607 -> Dep: 0500hrs - Arr: 0740hrs
      US333 -> Dep: 1040hrs - Arr: 1334hrs
      US625 -> Dep: 1040hrs - Arr: 1333hrs
     UA1627 -> Dep: 2015hrs - Arr: 2307hrs
      US660 -> Dep: 1405hrs - Arr: 1656hrs
     AA1733 -> Dep: 1925hrs - Arr: 2210hrs
      US343 -> Dep: 0725hrs - Arr: 1020hrs
     AA2481 -> Dep: 0955hrs - Arr: 1245hrs
     AA2487 -> Dep: 1725hrs - Arr: 2020hrs
      US695 -> Dep: 0500hrs - Arr: 0751hrs
      US415 -> Dep: 1710hrs - Arr: 2004hrs
      AA309 -> Dep: 1000hrs - Arr: 1145hrs
     AA1969 -> Dep: 1625hrs - Arr: 1810hrs

Enter 'x' to try another search. Enter anything else to exit.
Choice: e

Successfully ending program. Thank you!
pmunshi2@Olil3-PC:/mnt/d/225_final_project$
```

Test Cases Passed:

```
pmunshi2@Olil3-PC:/mnt/d/225_final_project$ ./test
Testing test_vertices..
Passed test_vertices
Testing test_dijksta_v_bfs..

Passed test_dijksta_v_bfs
pmunshi2@Olil3-PC:/mnt/d/225_final_project$
```