

Results

BFS and Dijkstra:

Our goals were to implement BFS to traverse the graph, Dijkstra's to find the min distance between two points, and a graphical representation of the shortest path with start point, connecting flights, and destination.

Some runs of BFS and Dijkstra's can be shown below. Figure 1 is a source and destination with no stop in between. Figure 2 has one stop. In Figure 3, we have a test to see if BFS gets the right flights and in the right order. In Figure 4, we have a test for distance of Djikstra with 2 stops.

From these runs and tests we can see that our BFS method gives us all possible flight options from a source and our Dijkstra's method gives us the shortest distance.

Figure 1:

```
Please enter the number of the airport you want to fly from
3830
Please enter the number of your destination airport
3093

Your path will be represented with a green line.
Loading Flight Path...

All Flights from 3830: 3830 5726 4042 4019 5718 2170 3808 3602 2179 3673 4859 3875 4037 3690 3448 3820 3849 1382 3578
4015 4042 3466 3876 3759 4049 3719 1852 3488 4045 3637 4288 3520 3751 3670 11851 3729 3645 599 345 3659 4018 3494 4292
1555 3533 4089 4099 4041 3653 3685 3077 3389 4113 3550 3457 3585 3712 3797 3877 3484 4017 3697 507 3660 4044 1229 478
1788 3458 3878 3542 3473 1824 4050 3576 3717 4072 3675 3459 3858 3861 2279 3863 3454 3722 3720 3364 3752 3462 4046 35
70 3839 3401 1836 3626 3608 3807 3622 4048 3793 3731 3621 4014 3577 3469 4348 3748 1840 2890 3536 3867 3678 3753 3745
4066 3851 3878 3541 3679 3691 3678 3690 3679 3681 3684 3774 1892 1871 508 3939
1769 1926 3818 346 679 1678 3581 3515 3453 1613 737 680 302 1784 4355 4054 3457 4867 4286 3811 5495 3711 4001 3904 4112
3806 5732 3819 4255 3598 5735 3946 3843 2564 4088 4034 3728 3714 3508 3544 3543 4126 4011 5754 5762 1825 1953 3456 36
11 4117 3564 3641 3862 4087 4016 3613 4359 4318 1885 3817 4336 4067 4271 2883 2899 166

3830->3093 Distance: 12021 km
Total Flight Distance: 12021 km
```

Figure 2:

```
Your path will be represented with a green line.
Loading Flight Path...

All Flights from 2990: 2990 2965 2966 2968 4029 6156 2948 2975 1688 8076 2191 6149 2981 2983 2988 2985 2979 6147 1701
2992 2912 2913
All Flights from 2975: 2975 2990 6969 2972 4364 4078 6140 4363 2979 6146 2993 2969 2989 2992 7603 4377 421 2954 7563 8
876 2191 6159 2983 6141 2912 2910 2912 2913 1587 2939 3964 4029 3364 2965 2960 2948 2962 2964 2985 6147 1701 1382 2188
1555 3406 4374 1579 1197 346 4352 1590 3885 3179 1138 4057 2988 8428 4274 6138 2974 4373 4385 6139 6143 6137

2990->2975 Distance: 723 km
2975->3974 Distance: 1930 km
Total Flight Distance: 2653 km

Thank you for using our Shortest Flight Implementation.
```

Figure 3:

```
SECTION("Check visited correct airport first") {
    REQUIRE(path.at(0).getCode() == 1);
}

SECTION("Visited airports in correct order") {
    REQUIRE(path.at(0).getCode() == 1);
    REQUIRE(path.at(1).getCode() == 3);
    REQUIRE(path.at(2).getCode() == 4);
}
```

Figure 4:

```
SECTION("Correct Distance") {
    Vertex a = path[0]; //source
    Vertex b = path[1]; //stop
    Vertex c = path[2]; //stop
    Vertex d = path[3]; //dest
    REQUIRE(g.getDistance(a,b) == 106);
    REQUIRE(g.getDistance(b,c) == 179);
    REQUIRE(g.getDistance(c,d) == 281);
    REQUIRE(g.getDistance(c,d)+g.getDistance(b,c)+g.getDistance(a,b) == 566);
}
```

Graphical Implementation:

Below are example results for the graphical implementation. The inputs for Figure 5 were 3830 as the starting airport and 3093 as the destination airport. The inputs for Figure 6 were 2990 as the starting airport and 4374 as the destination airport. These inputs can be put in our program to replicate our results. Figure 5 displays a flight path with no connecting flights to show that our program works for this scenario. Figure 6 is a flight path with a connecting flight at Airport number 2975. The inputs for Figure 6 prove our flight path visualization works with connecting flights. Together Figure 5 and 6 are representations of the graphical representation working as intended.

Figure 5:

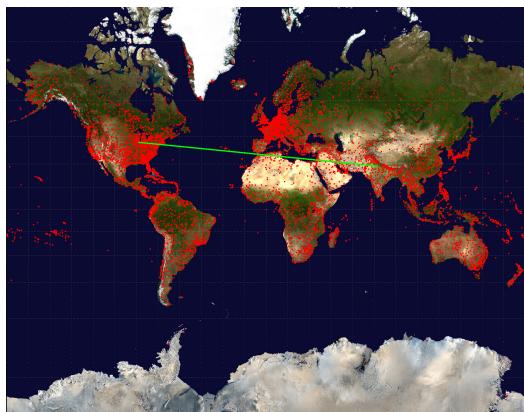
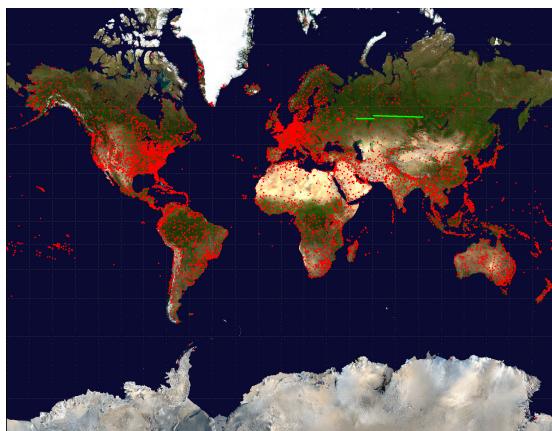


Figure 6:



Discoveries:

While working on this project we realized that our project had multiple applications. Originally, we thought that our program would be most applicable to a casual flyer who isn't used to navigating a large list of flights. However, as the project progressed, we realized that frequent flyers and airline employees could also benefit from knowing the shortest distance as it could help them minimize exposure to radiation. Before pursuing this project, since most of us don't travel often, we were not aware of radiation risks associated with air travel.