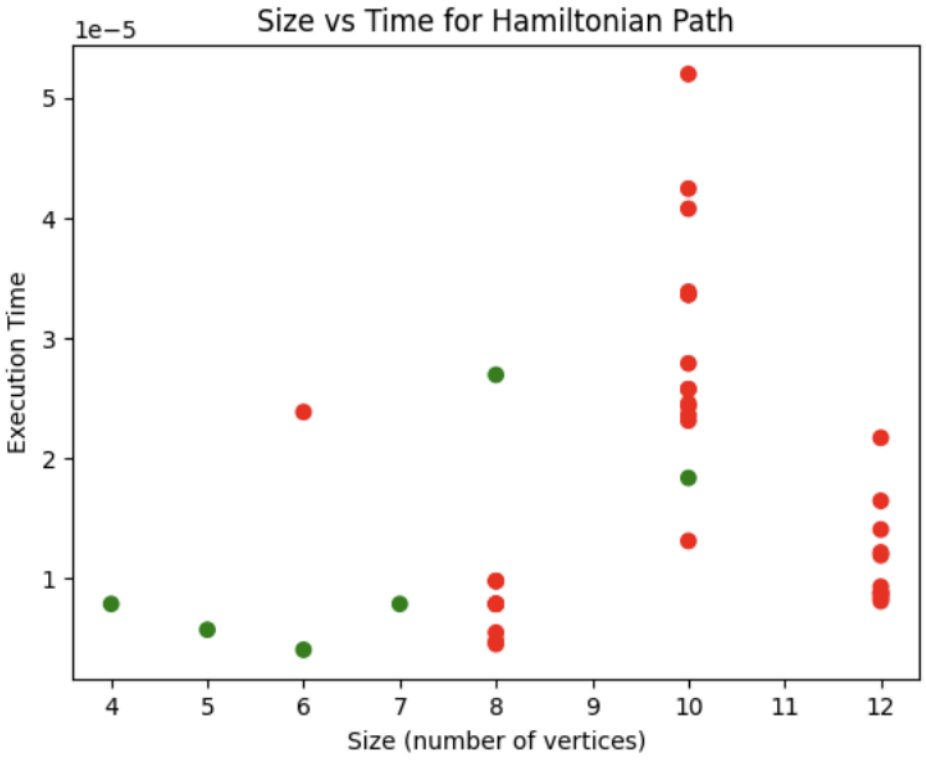
This is the output from the program data_ham_paths_MNO.py, which takes the file check_ham_paths_MNO.csv as its input and looks for instances of a Hamiltonian path while also measuring the execution time. It stops when it finds the first instance of a Hamiltonian path in a given graph. It then creates the size vs time graph for finding a Hamiltonian path.

```
TSP Instance 1:
Execution Time: 7.867813110351562e-06 seconds
Hamiltonian Path: ['1', '2', '4', '3']

TSP Instance 2:
Execution Time: 4.0531158447265625e-06 seconds
Hamiltonian Path: ['2', '1', '3', '6', '5', '4']

TSP Instance 3:
Execution Time: 5.7220458984375e-06 seconds
Hamiltonian Path: ['1', '2', '4', '5', '3']

TSP Instance 4:
Execution Time: 7.867813110351562e-06 seconds
Hamiltonian Path: ['1', '2', '4', '6', '3', '5', '7']

TSP Instance 5:
Execution Time: 4.5299530029296875e-06 seconds
No Hamiltonian Path Found

TSP Instance 6:
Execution Time: 3.361701965332031e-05 seconds
No Hamiltonian Path Found

TSP Instance 7:
Execution Time: 2.6941299438476562e-05 seconds
Hamiltonian Path: ['1', '6', '3', '8', '5', '2', '4', '7']

TSP Instance 8:
Execution Time: 1.8358230590820312e-05 seconds
Hamiltonian Path: ['5', '8', '1', '3', '6', '9', '2', '4', '7', '10']

TSP Instance 9:
Execution Time: 8.821487426757812e-06 seconds
No Hamiltonian Path Found
```



Red dot = Hamiltonian path not found
Green dot = Hamiltonian path found

DISCUSSION of plot:

The size vs time graph for Hamiltonian paths above indicates that as the number of vertices and number of edges within a graph increase, the time it takes to find a Hamiltonian path increases as well. There is a clear exponential increase in the graph above demonstrating this. Note that each red dot represents that a Hamiltonian path was not found, and there are a lot more of these than green dots because for each Hamiltonian path that was found, the program first checked a bunch of options that did not successfully find a Hamiltonian path. This accounts for some of the shorter execution times at large sizes that are represented by red dots. Looking at the trend of the green dots, however, the exponential growth in size vs time is clear.

In creating test cases for this graph, I found it interesting that the execution time is more strongly correlated with the number of edges in a graph than the number of vertices. This makes sense though, since more edges require the program to spend more time checking different paths, whereas adding more vertices does not necessarily mean there are more paths to check. For example, a graph with 20 vertices all in a straight line, each with only one edge connecting them would have a faster execution time than a graph where there are 20 vertices and 4 edges connected to each vertice because there are a lot more potential paths to check in the second graph. Overall, this project was successful in showing the exponential growth in execution time of finding a Hamiltonian path as the size of a graph is increased.