Design - Assignment 4 - The Perambulations of Denver Long

The objective of this assignment is to provide the functionality of a stack and adjacency, and with it, return the shortest path to reach every point in the graph once. The travelling salesman problem, solved with a depth-first search and a little bit of guess and check.

For the purposes of this project, there will be two stacks involved at any given point in time. One will store the current, most efficient path, and the other stack will keep track of the path currently being traversed.

```
Fastest path recorded:
Dynamically allocate an array of n length
Is it a hamiltonian cycle? Is it shorter than this?
    Replace this path with that one
```

```
Current path being stored:
Dynamically allocate an array of n length
If the current path has no more neighbors:
    Go back and take a different one
If the current path is shorter than the shortest path:
    Replace the shortest path
If you're out of possible paths
    You've got the fastest path stored already
```

And now, the problem of finding the quickest path, in its entirety, will be solved by the method of depth-first search. The process is simple:

```
Start from a node
Go from that node to a neighbor
Keep doing that until you no longer have anywhere to go
Is your current path Hamiltonian? If so, can you reach the beginning from the end?
    Now test if it's shorter than the shortest recorded path
        If so, replace it
```

Then, recursively loop back to an earlier, unvisited node, and pick a different neighbor
Do this until you've explored every possible combination of nodes

This method of finding the shortest path isn't entirely intuitive, and has a computational complexity of *O(n!)*. In addition, other methods of finding the shortest path do not improve much, as in an adjacency matrix, there are few ways to outlier certain paths. Your best possible option, in a situation in which you're unaware of the nature of the graph, is to preemptively abort any path you've ventured on that is already longer than the shortest recorded path. Perhaps that is how the travelling salesman problem has earned its name.

Note: The credit for the pseudocode and its respective concepts go to Professor Long, and any supporting writers, in the assignment documentation.