Strengths

- My code passed the test case, and the optional debugging print statements confirmed it worked
- The code is well-commented, and the structure is easy to follow.
- The use of functions for adding edges and running DFS breaks up the code functionality nicely
- Handles edge cases, has no memory leaks

Improvements

- I think my implementation of the adjacency lists is a bit hard to follow. I had difficulty grasping it and used trial and error until it worked.
- I should have designed more custom test cases. Unfortunately, I was pressed for time, so I could not achieve this.
- I could have read the graph with a separate function as well to make the code more well-structured.

Thoughts

- This assignment felt much more difficult than the first two and a big step up. I realized I didn't even know how to write the make file initially, but I got around to it since I had to. The most difficult aspect for me was to recreate the graph in adjacency list format using the structures specified. I got really confused by exactly how to represent each adjacency list while following the requirements, as I would initially have liked to create a matrix of predetermined length and let it have empty entries. But I suppose this harder way of dynamically allocating memory for variable length arrays is more efficient, and I figured out how to use it. Freeing this memory was also tricky.