**Graph Traversal Write-Up**

One of my first focuses for this assignment was representing the graphs. While debating how I was going to go from edges and nodes to an adjacency matrix and list, I realized I had already overcomplicated it. None of the graphs contained nodes with no edges, so I can just use the collection of edges. I made a class to represent an edge storing the two nodes it relates to and whether the connection is bidirectional. Once I had my Edge class, the only difficulties with the list and matrix were indexing. The biggest issues with both of the traversal methods was handling choosing the least valued node when faced with options. This issue only came into effect when dealing with the AdjacencyList, so I simply sorted all of the lists before I started traversing which resolved all of my problems.

**An Update:** While comparing results with classmates, I realized there was one bug in my BreadthFirstTraversals where one node in the queue kept getting put off until the very end. It took lots of troubleshooting, print statements, and breakpoints, but I eventually realized my queue data structure was the problem. The priority queue was not the ideal choice for this situation. I switched it to a linkedlist and my problems were resolved. Updated output can be seen below.

A screenshot of a computer program

Description automatically generated with medium confidence