James Palmer
Assignment 7 answers

1. How is the graph stored in the provided code -- adjacency matrix or edge list?

An edge list, as the adjacency matrix would store all of the data in an array. The memory usage for an edge list is much more memory-friendly because the size is based on the total number of vertices/edges but for an adjacency matrix it's squared that since it's a full matrix with the x and y values being the various vertices.

2. Which of the graphs are connected? How can you tell?

From using the recursive DFS function of this week's code, all graphs are connected except for graph 3.

3. Imagine that we ran each search in the other direction (from destination to source, instead of source to destination) -- would the output change at all? What if the graphs were directed graphs?

The output should remain the same when our graph is not directed. This is because all of the vertices are continually pointing to one another in the same connection that they prior. However, if a directed graph’s output is change, as the vertices could become unreachable, from the other way around (or in reverse per say).

4. What are a few pros and cons of DFS vs. BFS?

DFS normally has less memory necessities because you don't necessarily have to store all of the child pointers at each level. Whereas BFS, however, finds and report a path with the minimum number of edges between two given vertices

5. What's the Big O execution time to determine if a node is reachable from another node?

The Big O execution time is O(n) where n is the number of vertices + the number of edges.