

Answers to Written Questions

1. How is the graph stored in the provided code? Is it represented as an adjacency matrix or list?

It is represented as an edge list.

2. Which of the 3 graphs are connected? How can you tell?

A graph is connected if there exists a path from each vertex to every other vertex. In this case, graphs 2 and 3 are connected but 1 is not.

3. Imagine that we ran each depth-first and breadth-first searches in the other direction (from destination to source). Would the output change at all? Would the output change if the graphs were directed graphs?

There would be no change in output since we have an undirected graph. If the graphs were directed, there would be a change in output as the starting location would determine which nodes are reachable.

4. What are some pros and cons of DFS vs BFS? When would you use one over the other?

Pros of BFS:

- BFS will not get stuck in an infinite path
- BFS is guaranteed to find a path containing the least steps from start to goal (if it exists)

Cons of BFS:

- BFS may take a longer time to find a path
- BFS may take up more space because it looks at all paths of a specific length at once

Pros of DFS:

- DFS can get lucky and find the solution very quickly
- DFS uses less memory

Cons of DFS:

- DFS can take an unfortunate route and have to backtrack a long way, and multiple times
- DFS can get stuck in an infinite path

When to use

BFS: if solution is close to root (quicker), if solution is rare and may be deep (quicker)

DFS: if solutions are frequent but deep in tree (less memory used)

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

$O(V + E)$