Larissa Hahn

CS 261 – Programming Assignment 7

answers.pdf

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

   Edge List

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

   The code in main.c iterates through all the possible combinations of starting and ending nodes and informs the user if they are reachable. I know the graph is connected when the program informs me that all of these possible combinations are reachable. A graph is connected when there is a path from one node through every other node in the graph.

   Connected graphs: 1, 2, 4, 5 – all reachable!

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?

   For the undirected graphs, the output would not change if we ran the search in the other direction. It would be the same because the nodes are all connected regardless of which direction you trace through it.

   If the graphs were directed graphs, the output wouldn't be the same as you can't go in reverse on a directed graph so the nodes would not be reachable.

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

   DFS can take an unfortunate route and have to backtrack a long way multiple times. However, DFS can get very lucky and find the solution very quickly. If it finds the path on the first attempt, the running time is very fast. On the other hand, BFS might not find the path as quickly, but it will always find the solution. A pro is that BFS checks all paths of length 1 first, then length 2, and so on, so it's guaranteed to find a path containing the least amount of steps from start to finish (if it exists). There is the slight chance that DFS will not find a solution. If there's only one infinite path and DFS happens to go down that path, it will get stuck and NEVER find the solution; however, a pro of BFS is that BFS will go down that same infinite path and not get stuck.

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

The Big O execution time to determine if a node is reachable is Big O(E).