

Erin Alltop

3/18/17 – Winter 2017

CS261 – Assignment 7 Graphs

1. The graph in the provided code is stored as an adjacency matrix.
2. Graphs 2 and 3 are connected. You can tell this because when running the output using the provided testing code, you can see that those two graphs both have a path to all vertices in the graph.
3. As the current graphs we are working with are undirected, then the output would not change if they were run in the opposite direction. If the graphs were directed graphs then the output would likely change as some vertices would be able to access others via an edge in one direction but not in the other, which would change the path (or eliminate it entirely).
4. A possible pro of DFS is that it is possible that it could get lucky and find the path directly, while that is not possible for BFS as BFS searches as if it were a wave moving in several directions at once. This also means that BFS stores more memory in general than DFS. A con of DFS however is that it might take longer for DFS to find the path as it would have to regularly reverse its steps and find a new path if it reaches a dead end. One might use BFS over DFS if there is a smaller amount of data to search and the solution is more difficult to find. In this case BFS might be faster than DFS. Alternatively, given the memory requirements for BFS, DFS would make more sense if you need to search a very large amount of data or go very “deep” to find a path.
5. The big O execution time to determine if a vertex is reachable from another vertex is $O(E)$ where E is the total number of edges in the graph.