CPSC 319 Assignment 4 Questions Ethan Reed - 30123128

Q1: Describe the algorithm based on the graph traversal to find if there is a loop (a path of

directed edges of the graph that starts and ends at the same node). Which traversal

technique (breadth-first or depth-first) is more suitable? What is the worst-case

complexity of this algorithm?

A1: To find a cycle in a directed graph, complete a depth-first search while keeping a list of visited nodes. Mark each node as visited when moving on or past it, and then mark it as unvisited when backtracking through it. If at any point the program attempts to visit a node marked as visited, return true. Depth-first traversal is used because breadth-first traversal is not reliable for finding loops in directed graphs. With breadth-first traversal, even if a node is marked as visited, it might have been visited from a separate path. Therefore visiting an already visited node does not necessarily indicate a loop. The worst-case complexity of this algorithm is O(V x E), where V is the number of vertices, and E is the number of edges. All operations within the recursive depth-first search function are performed in constant time, and in the worst-case, the algorithm will visit every node once and check every edge once.

Q2: For a given query file, which traversal technique (breadth-first or depth-first) is more

efficient in determining if the path exists between the given nodes? Use the provided

input file and query file to help answer this question.

A2: Breadth-first traversal more efficiently finds paths. If more than one path exists between the start node and end node, breadth-first traversal will always find the shortest path first (unless the paths are weighted). Because depth-first search chooses its next edge randomly, it could by chance find the longest path from start to end before finding any other paths.