MD Shaherier 30132195

Complexity Analysis

Question 1

When a path begins and finishes at the same node, it is called a loop or cycle. Two circular loops with the same number of vertices starting from the same node and finishing at the same node can be a worst-case scenario. In this instance, BFS would take twice as long as DFS to detect the cycle. If we utilise a queue, a visited array, and maintain track of the ancestor node(s) to implement BFS, the recursion queue will go back and forth between the two loops, and the time complexity of this algorithm will be O(V + E). Even though the time complexity of a DFS algorithm is the same as that of a BFS method (i.e. O(V + E)), it will only take half the time. This is because DFS will concentrate on a single path until the search is completed, no mode edges are present, or the path is looped.

We can keep track of any back edges using a DFS traversal. An edge that loops from a node to itself or one of its ancestors is known as a back edge. In addition, we'll need a stack to hold the nodes in recursion and a visited array to keep the graph's visited nodes. If the nearby vertex/vertices from the current vertex have been visited at any point in time, we can check that stack to see if the vertex/vertices are still there. If that's the case, we've got a loop/cycle on our hands. The DFS implementation has a time complexity of O(V + E), where V denotes the number of vertices and E denotes the number of edges.

Question 2

DFS traversal performs substantially better than BFS based on the output files supplied by the application. DFS generates substantially shorter pathways than BFS, which corresponds to the time it took to locate the sink from the source. As a result, this displays how much memory the algorithm utilised to complete the operation. Because it must traverse more vertices to reach the target, BFS takes up more space than DFS. This is due to the fact that it must examine all neighbouring vertices before proceeding to the next level. Thus I think it is more clearer to use DFS.