Assignment Three: Depth vs Breadth First Search

Brandon Trinkle

Arizona State University

Course Number: IFT 360

Professor Durgesh Sharma

9/1/24

**Question 1**

Breadth First Traversal is defined as “which the root node is expanded first, then all the successors of

the root node are expanded next, then their successors are expanded, and so-on,” (Artificial Intellligence: A Modern Approach, 2021, p. 77). Therefore, the graph shown below will be analyzed as followed:  
A diagram of a network

Description automatically generated

1. A
2. B
3. C
4. D
5. E
6. F
7. G
8. H
9. I
10. J
11. K
12. L

Depth First Traversal “always expands the deepest node in the frontier first.” (Artificial Intellligence: A Modern Approach, 2021, p. 78). Based on this definition the nodes are analyzed in the following order:

1. A
2. B
3. D
4. E
5. I
6. J
7. F
8. C
9. G
10. H
11. K
12. L

**Question 2**

Breadth-First Search will always return the most cost-optimal path to a target node when all edges have the same cost because it explores nodes level by level, ensuring the shortest path (in terms of edge count) is found first. This characteristic makes Breadth-First Search optimal for uniform-cost problems. Depth-First Search does not guarantee the most cost-optimal path because it explores one full path to its depth before backtracking and exploring other paths. This approach can lead Depth-First Search to miss shorter paths that are discovered later, as it does not prioritize shorter or cheaper paths. (Artificial Intellligence: A Modern Approach, 2021, pp. 78-87)

**Question 3**

Breadth-First Search requires significantly more memory because it needs to store all nodes at the current level of the tree before moving on to the next level. As trees grow, the number of nodes at each level increases exponentially, leading to high memory consumption. This is particularly challenging for large trees where the breadth can become unmanageable. Depth-First Search requires much less memory compared to Breadth-First Search because it only needs to store the nodes along the current path from the root to the deepest node. Once a path is fully explored, it backtracks and does not retain memory for other paths simultaneously, making it more memory-efficient for deep or large trees (Artificial Intellligence: A Modern Approach, 2021, pp. 78-89).

# References

Artificial Intellligence: A Modern Approach. (2021). In S. Russell, & P. Norvig. Hoboken, NJ: Pearson.