Assignment No.4

Analysis of Algorithm

M.Aashan Hussain(43914)

Problem Selection: Graph Traversal

Algorithm A: Breadth-First Search (BFS)

Strengths:

- Guaranteed to find the shortest path in an unweighted graph.
- Efficient for finding all nodes at a given distance from the source.

Weaknesses:

- Can be memory-intensive for large graphs.
- Less efficient for finding the shortest path in a weighted graph.

Algorithm B: Depth-First Search (DFS)

Strengths:

- Efficient in terms of space usage.
- Can be used for topological sorting and cycle detection.

Weaknesses:

- Doesn't guarantee finding the shortest path.
- May get stuck in deep branches of the graph.

Hybrid Design: Informed Depth-First Search (IDFS)

Idea:

Combine the space efficiency of DFS with the ability to find shorter paths, inspired by BFS.

Use a depth limit to prevent excessive exploration of deep branches.

Gradually increase the depth limit until the target node is found or the entire graph is explored.

Algorithm:

1. Initialize:

Set the initial depth limit to a small value.

2. Iterative Deepening:

While the target node is not found:

Perform a depth-limited DFS.

If the target node is not found, increase the depth limit.

3.Return:

If the target node is found, return the path.

Otherwise, return "Not found."

Performance Analysis:

Space Complexity: Similar to DFS, as it only needs to store the current path.

Time Complexity:

In the worst case, it may explore the entire graph, but it's often more efficient than BFS, especially for deep graphs.

The time complexity depends on the branching factor of the graph and the depth of the target node.

Advantages of IDFS:

Efficient Space Usage: Like DFS, it avoids the memory overhead of BFS.

Better Path Finding: It's more likely to find shorter paths than a simple DFS.

Flexibility:The depth limit can be adjusted based on problem requirements.

Potential Applications:

Game AI: For exploring game trees and finding optimal moves.

Web Crawling: For efficient crawling of large websites.

Network Routing: For finding shortest paths in networks.

By combining the strengths of BFS and DFS, IDFS offers a more efficient and effective approach to graph traversal in various applications.