Difference in Space and Time Complexities for Different Search Algorithms

Peyton Roehnelt, Tejaswi Cherukuri, Shane Kennedy
21 October 2021

Purpose

To analyze the differences in space and time complexities of three different search algorithms and compare which would perform best on the mazes provided for Project 1.

Depth-First Search (DFS)

This search algorithm expands on the deepest node. However, the tree search implementation is susceptible to infinite loops because it does not keep track of previously visited nodes. It is also not an optimal search algorithm because it returns the first solution found, regardless of cost. The graph search version is complete, meaning it is not susceptible to infinite loops because it keeps track of visited nodes. Time complexity is O(bm), where b is the branching factor and m is the max depth. The space complexity is O(bm) for either implementation of the algorithm.

Breadth-First Search (BFS)

This specific search algorithm uses a list ordering implemented as a queue to find the solution. It expands on all nodes at a specific depth before any nodes below them are expanded. This algorithm is also complete and optimal, meaning there are no infinite loops, and it returns the optimal solution, not just the first solution found. Space complexity is O(bd) where b is the branching factor, and d is the depth of solution.

Greedy Best-First Search (GBFS)

This algorithm is not complete, as it can get stuck in infinite loops, unless searching through an already explored list. Inaccurate heuristics, or information, can lead to a non-optimal solution, so we cannot claim that this algorithm is always optimal. The time complexity is $O(b^m)$ where b is the branching factor and m is the maximum depth of the searching space. The space complexity is also the same as time complexity.

Overall Comparison

Comparing DFS to BFS, in the worst-case time complexity we can say that BFS is always better. Sometimes, on average DFS is better if there are many goals, no loops, and no infinite paths. BFS is worse memory wise than DFS, however DFS is in linear space and BFS may store the whole search space. In general, BFS is better if the goal is not deep, no infinite paths or loops, and the search space is small. DFS is better in terms of memory. Overall, GBFS algorithm is like DFS because it does not consider cost or optimality. We can say that all three of these algorithms will produce equivalent results in terms of cost for most mazes.