In theory, let's analyze the Uniform Cost Search (UCS) and Iterative Deepening Search (IDS) algorithms to determine which one is more optimal for a given problem based on their characteristics:

Uniform Cost Search (UCS):

- UCS explores the search space by selecting nodes with the lowest path cost first. It uses a priority queue to achieve this.

- UCS is guaranteed to find the shortest path to the goal state if the edge costs are non-negative. This makes it an optimal algorithm in terms of finding the lowest cost path.

- The time complexity of UCS is exponential in the worst case, mainly because it may expand a large number of nodes if the cost varies significantly across the search space.

Iterative Deepening Search (IDS):

- IDS performs depth-limited searches starting from a depth of 0 and incrementally increasing the depth until the goal state is found.

- IDS is complete, meaning that it will always find a solution if one exists. However, it may not always find the shortest path to the goal because it performs a depth-first search within each depth limit.

- The time complexity of IDS is also exponential in the worst case, but it has the advantage of being memory-efficient since it only keeps track of nodes within the current depth limit.

In summary, UCS is optimal in finding the lowest cost path but may have a high time complexity for some problems. IDS, on the other hand, is complete and memory-efficient but may not always find the shortest path. The choice between the two algorithms depends on the specific problem and trade-offs between optimality and computational resources.

If you are looking for the shortest path with a guarantee of optimality and can handle the potentially higher computational cost, UCS is a suitable choice. However, if memory usage is a concern, or if you need to find a solution within a reasonable time frame, IDS is a good option, even though it might not guarantee the shortest path.