

Report

1. Water Jug Puzzle

- a. Best Strategy: Breadth-first search. Though the worst-case time complexity is worse than other search algorithms but the average case time and space complexity is the best amongst all algorithms. Also, we know it is complete and optimal.
- b. Heuristic Functions: I used two different heuristic functions:
 - i. $h(n)=0$ if $\text{current_state} == \text{goal_state}$ else 1
 - ii. $h(n) = |X - \text{goal_x}|$ in case of $\text{goal_state} = (\text{goal_x}, 0)$ and vice versa.

Both the heuristic functions are admissible as both of them don't over-estimate the distance to goal state.

- c. The surprising fact was the performance of BFS which was not supposed to perform the best amongst all algorithms. Also, the iterative deepening DFS had worst average time complexity because of repeated generation of nodes.

2. Path-Planning Puzzle

- a. Best Strategy: Greedy Search with Euclidean/Manhattan distance as heuristic function. Even though the space time complexity of BFS is marginally better but the time complexity of greedy overpowers all other algorithms significantly.
- b. Heuristic Functions: I used two different heuristic functions:
 - i. $h(n) = \text{euclidean_dist}(n, \text{goal_state})$
 - ii. $h(n) = \text{manhattan_dist}(n, \text{goal_state})$

Both the heuristic functions are admissible.

- c. Some interesting observations:
 - i. Greedy performed significantly better with a good heuristic function and the impact of heuristic function is clearly seen.
 - ii. A*, in contrast, shows minimal effect of heuristic function may be because of the cost function over-powering the heuristic function. However, IDA* does show significant improvement with change of heuristic function as with the case of greedy search.

3. Burnt Pancake Puzzle

- a. Best Strategy: I was not able to get the results for the given test case. However, for a smaller test case file with initial state (3, -1, 2, 5, 4), iterative deepening DFS performed the best. Since this puzzle has a lot of states, iddfs is the most efficient in terms of space complexity with comparable time complexity to other algorithms.
- b. Heuristic Functions: I chose a function which counts the number of elements which are misplaced from their position and hence needs to be placed in right position. However, I am not really sure if it is admissible as one flip of pancake can put more than one cakes in their respective positions which will make the heuristic function over-estimate the cost.