Project I: Push the Box

Yehyun Suh

Question 1: Basic search

Question 1.1 (1pts): Shall we choose DFS or BFS for the task? Why?

We must use BFS in this task. It is because DFS can return results that are not optimal for the search problem.

Question 1.2 (1pts for description + 3pts for code): Following code template Q1.py, finish the search of your answer in Q1.1. Briefly describe how you design your programs.

1. Read the input file and return box, player, and star position
2. From the starting point, move to up, down, right, left if the spot is moveable
3. Save all the visited position in (player, box) format, so that the search problem doesn't reach the same state again - in this process, I have used set instead of list to reduce searching time
4. When it reaches a new state, save it to the queue and move to the next state in the queue
5. Loop until the search problem reaches the goal state

Question 2: Heuristic search

Question 2.1 (1pts): Design an admissible heuristic function for the task.

$h(n)$ is the Euclidean distance of state and target. $h(n)$ is an admissible heuristic function because for every node, $h(n) \leq h^*(n)$. To be specific, the minimum moves from state to target cannot exceed the Euclidean distance of state and target in any cases.

Question 2.2 (1pts for description + 3pts for code): Following code template Q2.py, complete an A* algorithm with the heuristic of your choice in Q2.1. Briefly describe how you design your programs.

1. Read the input file and return box, player, and star position
2. From the starting point, move to up, down, right, left if the spot is moveable
3. Save all the visited position in (player, box) format, so that the search problem doesn't reach the same state again - in this process, I have used set instead of list to reduce searching time
4. When it reaches a new state, save it to the priority queue
5. Move to the next state which has the minimum $f(n)$ value, which is
   $f(n) = g(n) + h(n)$
   where $g(n)$ stands for the total cost from start state to goal and $h(n)$ is the Euclidean distance of state and target.
6. Loop until the search problem reaches the goal state