

ICS 361 Assignment 3: State Space Representations and Search (100 pts)

Due: Monday, October 19

Your task here is to implement a general-purpose (i.e. not puzzle-specific) version of the A* search strategy, and apply it to three different puzzles. Create meaningful comments in all files with a description of the problem and descriptions of each of the variables and functions used in the file. You are welcome to start with existing code or strategies you find online, but you *must* cite your source, indicate clearly which code is yours and which is from someone else, and write all of the comments in your own words.

Part A (20 points)

Implement **heuristic search** (A*). Your search code must be **independent** of the application problem (i.e. you should use the same search code for all three problems in Part B, although you may use a depth limit on the 8-puzzle problem). Add printing functions to your searches so that the first node on the open list is printed in each iteration. Also add printing functions so that when the goal is reached, the solution path and the lengths of the open and closed lists are printed.

Part B (50 points)

In Assignment 2, you (hopefully!) created state space representations for three puzzles. Now, use them (or an improved version) with your A* implementation. For each puzzle, you will also have to implement a heuristic function, which estimates the distance from the given state to the goal state. The default heuristic for all problems is the **number of "items"** not in the same place as they are in the goal state.

- (15 points) Farmer, Wolf, Goat and Cabbage
- (15 points) Water Jugs
- (20 points) 8-puzzle. Provide transcripts for three start states: easy, difficult and impossible.

Part C (30 points)

For the 8-puzzle:

- (15 points) Implement the **Manhattan distance** out-of-place heuristic, and provide transcripts of its performance on the same three start states you used in Part B. How does it compare with the default heuristic in Part B? Is it more informed? Why or why not?
- (15 points) Specify another heuristic for the 8-puzzle (you could use the strategy you described in Assignment 2, or something else). Implement it, and provide transcripts of its performance on the same three start states you used in Part B. How does it compare with the default heuristic in Part B and with the Manhattan distance heuristic? Is it admissible? Why or why not?

What to submit

Turn in the following (all via Laulima):

- All code, appropriately commented.
- All transcripts: single start-to-goal runs for FWG&C and WJ, three runs using the default heuristic for 8-puzzle, three using the Manhattan distance heuristic, and three using your own heuristic.
- A text file with your response to Part C.