Assignment 2: A Better Vacuum Robot Planner
CS 730/830, Spring 2025
Due at **11pm on Mon, Feb 3**

Overview

You will extend your vacuum robot planner with more advanced search algorithms that should allow it to tackle much larger problems.

Input

Your program should now accept up to two command-line arguments:

algorithm one of `depth-first`, `depth-first-id` (depth-first iterative deepening), `uniform-cost`, or `a-star`

heuristic if the algorithm is `a-star`, this second argument will specify either `h0` ($h(n) = 0$), `h1` (a heuristic you design), or `h2` (an even better heuristic you design). Your heuristics must be admissible. The reference solution actually has three heuristics (`h1`–`h3`).

Output

Same as before.

Submission

Electronically submit your solution using the instructions on the course web page, including your source code as well as a transcript of your program running with the validator.

Also electronically submit a brief write-up answering the following questions:

1. Describe any implementation choices you made that you felt were important. Clearly explain any aspects of your program that aren't working. Mention anything else that we should know when evaluating your work.
2. Explain each heuristic function you devised and prove that each is admissible.
3. What is the time and space complexity of each algorithm you implemented? Which algorithms are admissible?
4. Provide empirical results confirming your answers to the previous question.
5. What suggestions do you have for improving this assignment in the future?

Graduate Extensions

Those in 830 must extend the base assignment in two ways. First, your planner should support two additional algorithms `ida-star` (iterative-deepening A*) and `greedy` (greedy best-first search). Second, you should implement an additional admissible heuristic function `h3` that takes the battery into account somehow. (The reference solution's `h4` and `h5` take the battery into account but are unfortunately inadmissible.)

Evaluation

Rough guide to grading:

0 nothing

1 something but basically nothing

2 write-up is correct but no code works

3 DFID works but A* doesn't. write-up has no discussion of heuristic anything.

6 Multiple significant problems.

7 A* returns non-admissible solutions.

8 Significant problem.

9 Code works fine, but is not super-awesome. Write-up is fine but not super-awesome.

10 Everything runs smoothly and correctly. The implementation roughly on par with the reference solution even for large problems. Write-up is clear and convincing, with no errors.