

CS4750/7750 HW #2 (20 points)

Fall 2016

(Due 9/15)

In this programming assignment, you are asked to implement

- a) iterative deepening tree search (IDS),
- b) depth-first graph search (DFGS), and
- c) A*,

to solve two instances of a variant of the vacuum-cleaner world problem discussed in the lecture.

- The environment is a 2-D grid, 4x4 or 5x6.
- Actions: The agent can choose to move left, right, up, down, or suck up the dirt. Clean rooms stay clean. The agent remains where it is when an action would take it outside the environment.
- Step cost: 1 for left or right; 1.3 for up or down; 0 for suck.
- The goal is that all squares are clean.

You may use any programming language in your implementation. The algorithms should be implemented according to their pseudocodes in the lecture notes. During search, a tie is broken based on increasing order of the agent's location coordinates, i.e., increasing order in the first dimension, then increasing order in the second dimension. For example, assume starting the agent at (3,2). Its next location can be (2,2), (4,2), (3,1), (3,2), or (3,3). Based on the tie-breaking rule, the order of the corresponding search nodes in the fringe should be (2,2), (3,1), (3,2), (3,3), (4,2).

For A*, design and present your own admissible heuristic function.

Instance #1: 4x4 grid. Initial state: (3,2). Dirty squares: (1,2), (1,4), (2,2), (2,3), (3,1), (4,2), (4,4).

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	...	
...			

Instance #2: 5x6 grid. Initial state: (3,2). Dirty squares: (1,2), (1,4), (1,6) (2,1), (2,3), (2,4) (3,1), (3,5), (3,6), (4,2), (4,4), (5,3), (5,4), (5,6).

Your program should print out the following for each algorithm and each instance:

- a) List the first 10 nodes in the order they would be expanded.
- b) Give the solution found, i.e., the sequence of actions to the goal.
- c) Give the path cost of the solution.
- d) Give the CPU execution time in milliseconds.

Your submission should be a single pdf file with file name containing your name and assignment number. For example, *firstnameInitial_lastname_hw2.pdf* for HW2. The file should contain three parts:

- 1) A brief description of your algorithm, implementation, and execution result.
Discuss your finding. Cite any existing software program you use.
- 2) Printout of your programs in solving the two problem instances.
- 3) Your code with appropriate comments.