CSCE476/876 **Fall 2021**

Homework 3

Assigned on: Friday, September 17, 2021.

Due: Friday, October 1, 2021.
Except for the programming questions (i.e., Exercices 1 and 7), which must be submitted with `webhandin` as probem#.*ext* (where *ext* corresponds to your languages extension), you may turn in your homework on paper or type it and submit it to `webhandin`.

Value: 80 points for ugrads and 85 points for grads.

Exercises: AIMA exercises are available online : ”https://aimacode.github.io/aima-exercises”

---

1 Implementing a simple-reflex agent. Total: 20 points

Bonus: Doing the programming in Common Lisp 5 points

- In the language of your choice, write a function that ‘models’ the simple-reflex agent for the vacuum-cleaner problem in an environment with two locations, as summarized on page 5 of the Intructor’s notes #4. The function should take as input the percepts of the agent as location of the agent and status of the room.
- Write a function that takes any of the 8 possible states of the vacuum-cleamer of Figure 3.2 of AIMA and runs the simple-reflect agent until the goal is reached.
- Design a performance measure that penalizes the agent for each step and each suck action.
- Write a function that runs your agent for each of the 8 possible states and displays the agent performance of each one of the above states. Record the agent performance for each one of these states. In addition to your program, turn in a file `readme.txt` that describes how to compile (if necessary) and run your program on CSE’s server so that the performance measures from above are displayed.

2 Chapter 3, Exercise 8, Source: AIMA online site. Total 10/15 points

Give a complete problem formulation for each of the following. Choose a formulation that is precise enough to be implemented.

- a: Using only four colors, you have to color a planar map in such a way that no two adjacent regions have the same color. (for ugrads and grads). 5 points
- b: A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot-high crates. (for ugrads and grads). 5 points
- d: You have three jugs, measuring 12 gallons, 8 gallons, and 3 gallons, and a water faucet. You can fill the jugs up or empty them out from one to another or onto the ground. You need to measure out exactly one gallon. (grads (bonus for ugrads)). 5 points

3 Chapter 3, Exercise 18, Source: AIMA online site Total: 10 points

Consider a state space where the start state is number 1 and each state k has two successors: numbers $2k$ and $2k + 1$.

1. Draw the portion of the state space for states 1 to 15.
2. Suppose the goal state is 11. List the order in which nodes will be visited for breadth-first search, depth-limited search with limit 3, and iterative deepening search.
3. How well would bidirectional search work on this problem? What is the branching factor in each direction of the bidirectional search?
4. Does the answer to part (3) suggest a reformulation of the problem that would allow you to solve the problem of getting from state 1 to a given goal state with almost no search?
5. Call the action going from k to $2k$ Left, and the action going to $2k+1$ Right. Can you find an algorithm that outputs the solution to this problem without any search at all?

4 Evaluation function. Total: 6 points

Adapted from AIMA, Edition 1.

With $g(n)$ being the path length,

1. Suppose that we run a greedy search algorithm with $h(n) = -g(n)$. What sort of search will the greedy search emulate?
   Explain. 3 points
2. Suppose that we run a search algorithm with $h(n) = g(n)$. What sort of search will the greedy search emulate?
   Explain. 3 points

5 Chapter 3, Exercise 25, Source: AIMA online site. Total: 9 points

Prove each of the following statements, or give a counterexample:

1. Breadth-first search is a special case of uniform-cost search.
2. Depth-first search is a special case of best-first tree search.
3. Uniform-cost search is a special case of A* search.

6 Chapter 3, Exercise 27, Source: AIMA online site. Total: 10 points

Trace the operation of A* search applied to the problem of getting to Bucharest from Lugoj using the straight-line distance heuristic. That is, show the sequence of nodes that the algorithm will consider and the f, g, and h score for each node.

7 Chapter 3, Exercise 38, Source: AIMA online site. Total: 15 points

Bonus: Doing the programming in Common Lisp 5 points

The traveling salesperson problem (TSP) can be solved with the minimum-spanning-tree (MST) heuristic, which estimates the cost of completing a tour, given that a partial tour has already been constructed. The MST cost of a set of cities is the smallest sum of the link costs of any tree that connects all the cities.

- Question a Show how this heuristic can be derived from a relaxed version of the TSP. 10 points
- Question b Show that the MST heuristic dominates straight-line distance. 5 points
- Question c: (Optional challenge) Write a problem generator for instances of the TSP where cities are represented by random points in the unit square. 15 bonus points
- Question d: (Optional challenge) Find an efficient algorithm in the literature for constructing the MST and use it with A* graph search to solve an instance of the TSP. 30 bonus points