

1. (10 points) Come up with at least two proposals for your course presentation, and indicate who will be doing the presentation with you. Appended are instructions from the course syllabus. You are free to stray from the syllabus, but must get advance permission to do so, and we have to agree in advance on how you will be assessed. (For example, you may wish to give a presentation on a philosophical or political topic rather than on an operational program. Or you may wish to propose working in a group of three students.)

Be sure to choose topics which allow you to succeed with all three components of your presentation described in the syllabus.

You may wish to click on links (both journal articles, and AI web-sites) in the MCS-385 homepage for ideas. The textbook's bibliography may also have leads. Don't neglect to make a visit to the library (and ask for guidance if needed.) You should write both possibilities on a sheet of paper to hand in Friday. For each of the two proposals,

- (a) Locate and list three useful references (one should be a journal article)
 - (b) Address briefly, in writing, how well you can fulfill each of the three elements of the presentation given your topic and give the references you've located so far.
 - (c) Be prepared to tell the class something interesting about the proposal.
2. (60 = 5 + 10 + 10 + 35 points) (Refer to AIMA problems 2.5-2.11)
 - (a) AIMA 2.7, 2.8 and 2.10. For 2.10, you'll need to make a few reasonable assumptions. It may help to refer to 2.6.
 - (b) Implement a slightly interesting environment for the vacuum world. You may wish to change the shape of the room, add furniture, change the distribution of dirt and/or include some Murphy's-law characteristics. Although this isn't part of the contest, please endeavor to be creative in order to enhance the contest. Limit the room size to less three digits or less (i.e., fewer than 1000 squares.)
 - (c) Design a vacuum agent with internal state. Your agent will run for a maximum of 3000 steps in every person's world, including your own.

I will hold a contest as follows: Each agent will be run on every environment. On each environment, your agent scores 1 point for each of your competitors' agents your agent outperforms. If there are n competitors, you will score a maximum of $n - 1$ points for each of the n environments, so the maximum is $n(n - 1)$ points. (Tie-break rules will be determined by me at the last minute.)

We'll have three additional competitors:

- (a) I will be one of the competitors.
- (b) One vacuum agent which shuts off immediately.
- (c) One vacuum agent which sucks if there is dirt underneath it; otherwise it will shut off.

If either of the last two (trivial) competitors win, all others are disqualified.

The top performer will get an A on the problem set. The second and third performers will get an A on this problem. Every other student will be graded normally for clarity and style (and so might also get an A). If there is a tie because two programs seem to use the same strategy, both are disqualified from the contest.

Be sure that your agent and your environment have the same interface as the AIMA code. Be sure to test your environment on AIMA's agents and to test your agent on AIMA's environment.

A few more details on the rules. Each individual should have an agent and world named according to your login name. Mine will be called `wolfe-vacuum-world` and `wolfe-vacuum-agent`. Your agent *should not* print out any debugging information as it runs. If I call `(make-wolfe-vacuum-agent)` twice, the second agent should have fresh state. The first lines of your program should be:

```
(provide 'wolfe)
(defpackage wolfe)
(in-package wolfe)
(export '(make-wolfe-vacuum-agent make-wolfe-vacuum-world))
(use-package 'user)
```

This will guarantee that functions you define won't interfere with other agents. As you write your agent, you'll won't want these incantations. Three files are in the directory `~wolfe/public/385/wolfe-vacuum/` to help get you started:

`wolfe-work.lisp` — A typical file which I used while I worked on my agent. This agent has minimal state so you can see how you could use a `let` to create state.

`wolfe.lisp` — The file I would submit for the competition which has appropriate package commands

`run-agent.lisp` — A file which tests that `wolfe.lisp` loads and runs properly.

You can grab these files and global replaced `wolfe` with your e-mail address to get started.

3. (15 points) AIMA 3.5 and 3.17abf. For part 3.17b, write only a sentence or two for each algorithm. Feel free to make more assumptions if it makes it easier to talk about an algorithm.
4. (15 points) You may work in groups of up to three students on the following problem. Each group should complete one problem and present the results to the rest of the class. Submit a written group report describing your results.

Do one of the following three problems. Please choose randomly so that different groups get different problems.

(a) AIMA 4.9

- i. Euclidean distance
- ii. n -MaxSwap: the number of swaps of two (not necessarily adjacent) tiles needed to reach the goal.
- iii. $X - Y$: Like Manhattan distance, but separate out the X and Y components.

(b) AIMA 4.12

(c) AIMA 4.17

Note that almost all of the code is provided in `aima/search/`, but it does take some effort to figure out how to use it.

Your task is to design and implement appropriate experiment(s) which collect data to address your question. Be sure to *both* collect the data *and* interpret the data for the class!