CS4700 Fall 2011: Foundations of Artificial Intelligence

Homework #1

Due Date: Friday Sep 16 on CMS (pdf) and in class (hardcopy)

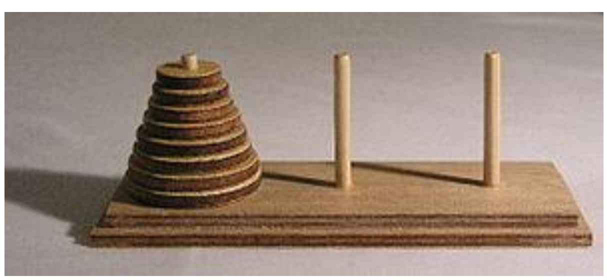
Late submission: 10% off each 24H period late until grades are posted (or use slack days). Collect your homework from Upson 360. Submit paper copies at the beginning of class or to TA directly. Please include your **name**, **NetID on top of each page** and staple all pages together**. Start new page for response to each question.** On first page also state the **CMS submission date and time**, **slack days** used and remaining. **Hardcopy must match exactly the PDF** uploaded to CMS. Your submission should include the answers and an appendix with code printout. Print code in 8pt monospace font (e.g. Courier or Consolas) and single line spacing with function names **highlighted**. Your assignments should reflect your individual work – ok to discuss strategies but you must write your own code and responses yourself. Cite any external sources used.

Question 1: Analysis (30 Points)

Suppose you want to model each provided scenario as a search problem. No need to program, just explain the following for **Problem A and B** in your own words

1. State representation: How would you represent the solution state?
2. Initial state: The starting state
3. Goal test: A Boolean function that tells you whether a given state is the goal
4. Successor function: Lists all valid actions from a given state and the new states they create
5. Cost function: Such that the cost of a path to a given state can be calculated
6. Estimate or bound on the branching factor of this formulation
7. Estimate or bound on the solution depth of this formulation
8. A suggestion as to which search algorithm to use for this problem

<u>Problem A — Tower of Ezra</u>



The Tower of Hanoi puzzle was invented in 1883 which is a very interesting puzzle. It's been well studied, so naturally most Cornellians are quickly bored of it and prefer a more difficult version: The Tower of Ezra. Instead of the normal three rods, you are given four rods configured in a square and numbered 1, 2, 3, 4 in clockwise order. N disks are initially stacked in decreasing size on rod 1. The object of the puzzle is to recreate the stack on the rod 2. However, there are four restrictions:

1. Only one disk may be moved at a time.

2. Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks that may already be present on that rod.
3. No disk may be placed on top of a smaller disk.
4. You may only move a disk one rod or two rods in the clockwise direction. That is, you cannot move a disk from rod 4 to 3, 3 to 2, 2 to 1, or 1 to 4.

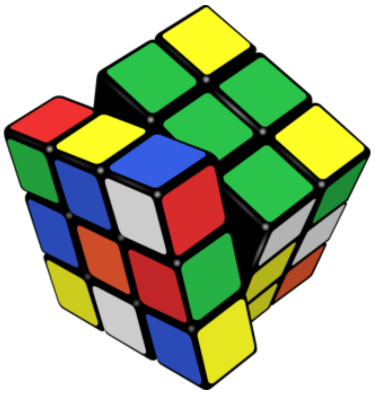
Problem B — Chinese Checkers

Chinese Checkers is a traditional board game for two to six players. Each player has ten marbles of one color at one starting point of the star. The objective of the game is to occupy the star point directly opposite. The first player to accomplish this is the winner. Of course if in the same round more than one player reaches the end point all of them are winners.

Each player moves in turn, (most of the time clockwise). A move consists of placing a marble into one of the adjacent holes or jumping over any other one marble into a hole beyond. The direction of the move has to be one of the following: left/right, forward left/right, backward left/right in respect with the player's position on the board. Only one move is made at a time, except when jumping. Then the player can make any number of jumps in any direction as long as all the moves consist of jumps. Players may jump their own or their opponents' marbles. No marbles are removed from the board during the play.

Question 2: Synthesis (50 Points)

Rubik's Cube, named after its inventor, Hungarian professor Ernő Rubik, has been a popular toy for nearly 40 years. In this problem, we will write a program that solves an easier three color version of a Rubik's Cube.



Standard Rubik's cube

Three color Rubik's cube (all blue faces are colored green, orange faces red, and yellow faces white)

A single move of a Rubik's cube consists of turning one of the six faces by 90° in either the clockwise (CW) or counter-clockwise (CCW) directions. Moves are generally denoted by a single capital letter for the face turned — **U**p, **D**own, **L**eft, **R**ight, **F**ront, **B**ack — and an extra "prime" mark if the face was turned CCW instead of CW.

For example, the following sequence of moves

U R R' U'

means to turn the Upper face clockwise, Right face clockwise, Right face CCW, Upper face CCW. In this case, the second two moves happen to reverse the first two moves, returning the cube to its initial configuration.

As you may know, a cube is solved when all six faces are a single color. A strategic note: turning any face moves all *cubelets* on that face *except* the center square, which never moves. Thus, the six center cubelets are completely immovable, and as a corollary, all solved cubes have the colors in the same order. That is, no combination of turning will ever result in, for example, the red face ending up next to the orange face.

One may construct a simpler three color version of the cube by starting with a normal cube and then painting all the blue faces green (blue and green being colors opposite each other in a normal cube), painting all the orange faces red, and all the yellow faces white. The solution to the three color version used in this problem will always have the two white faces opposite each other, the two red opposite each other, and the two green opposite each other.

Provide the following as a solution to this problem:

- State representation: How did you represent the solution state?
- Successor function: Lists all valid actions from a given state and the new states they create
- Estimate or bound on the branching factor of this formulation, if possible
- Estimate or bound on the solution depth of this formulation, if possible
- Implementation: Description of how you implemented this problem and why
- Solution: Set of sequences, for each of the three starting states below.

- Code: Documented code in any language. Implement everything from scratch.
- Performance: How many states did your algorithm visit to solve these three problems. Approximately how long did your algorithm take to run for each?

Hint: When searching the solution space, you'll probably want to hash each configuration you visit so that the second (third, fourth, …) time you see that configuration, you don't bother expanding the (identical) search tree below it. A completely naïve approach may take orders of magnitude longer to complete.

In the initial states below, the cube has been flattened for easy digestion in our favorite number of dimensions in which to print homework: 2. The face colors are represented by 0, 1, and 2 (corresponding, perhaps, to red, white, and green). The center of the cross shape is the Up face, and the one immediately below it is the Front face.

Initial state A (easy)	Initial state B (medium)	Initial state C (hard)
1 1 1 1 1 1 2 0 0 2 2 0 1 2 2 1 0 0 2 2 0 1 0 0 2 2 2 2 2 1 0 0 0 2 2 2 2 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 2 2	2 0 2 1 1 1 2 0 2 1 2 1 0 2 0 1 2 1 0 2 0 1 0 1 0 2 0 1 2 1 0 2 0 1 2 1 2 0 2 1 1 1 2 0 2 0 2 0 1 0 1 0 2 0	0 0 0 1 1 1 0 1 0 2 2 2 1 0 1 2 0 2 1 2 1 0 0 0 2 2 0 2 1 2 0 0 1 2 1 2 1 2 0 2 1 2 1 1 0 0 0 1 2 0 2 1 2 1

Extra Credit (20 pts): Solve the full Rubik's cube problem using all six colors for initial state C:

```
      r r r
      y y y
      o w o

g b g w o y g o g
y b y r r o b g r
b w b r o y b w b

      w g o
      g w b
      y w o

      r r w
      g o b
      w g y
```

Question 3: Critique (20 Points)

Read the following paper and write a short critique concerning its main issues. The purpose of a critique is to discuss the merit of two or three points from the paper; question the results, the assumptions, the applications, etc. There are helpful guidelines for critiques below. And remember, your critique should not just be a summary of the paper!

Allen Newell and Herbert A. Simon (1976) ***Computer science as empirical inquiry: symbols and search,*** *Communications of the ACM, Volume 19 Issue 3, March 1976*

Guidelines for Paper Critiques

Each critique should be no more than one page long. Less than a page is OK. The purpose of a critique is not to summarize the paper; rather you should choose two or three points about the work that you found interesting.

Examples of questions that you might address are:

- What is the main claim of the paper? Do you agree?
- What have we learned since? Were predictions/assumptions correct?
- What problem does this paper solve, and what are the strengths and limitations of its approach?
- Is the evaluation fair? Does it achieve it support the stated goals of the paper?
- Does the method described seem mature enough to use in real applications? Why or why not? What applications seem particularly amenable to this approach?
- What good ideas does the problem formulation, the solution, the approach or the research method contain that could be applied elsewhere?
- What would be good follow-on projects? Why?
- Are the paper's underlying assumptions valid?
- Which important issues in the field does this paper illuminate and how?
- Did the paper provide a clear enough and detailed enough description of the proposed methods for you to be able to implement them? If not, where is additional clarification or detail needed?

Your critique should be typed (single space) and should list the title of the paper and its authors at the top, along with your name. Avoid unsupported value judgments, like "I liked..." or "I disagreed with..." If you make judgments of this sort, explain why you liked or disagreed with the point you describe.

Be sure to distinguish comments about the writing of the paper from comment about the technical content of the work.