CS121 – Introduction to Artificial Intelligence – Winter 2011

Homework #1
(Search Problems)
Out: 1/10/11 — Due: 1/18/11 (at noon)

How to complete this HW: First copy this file; then type your answers in the file immediately below each question; start each question on a separate page, write your name on every page. Finally, print this file and return it in the drawer marked CS121 of a file cabinet located in the entryway of the Gates building next to office Gates 182 (see red arrow in http://ai.stanford.edu/~latombe/cs121/2010/map.jpg) no later than Tuesday 1/18 at noon.

Your name: ……………………………………………………………………………………

Your email address: ………………………………………………………………………………

Note on Honor Code: You must NOT look at previously published solutions of any of these problems in preparing your answers. You may discuss these problems with other students in the class (in fact, you are encouraged to do so) and/or look into other documents (books, web sites), with the exception of published solutions, without taking any written or electronic notes. If you have discussed any of the problems with other students, indicate their name(s) here:
……………………………………………………………………………………………………
Any intentional transgression of these rules will be considered an honor code violation.

General information: Justify your answers, but keep explanations short and to the point. Excessive verbosity will be penalized. If you have any doubt on how to interpret a question, tell us in advance, so that we can help you understand the question, or tell us how you understand it in your returned solution.

Grading:

Problem#	**Max. grade**	**Your grade**
I	25	
II	25	
III	25	
IV	25	
Total	100	

Name ……………………………………………………………………..

I. Basic Search Methods (25 points)

[*Warning: Facts and numbers used in the following problem are historically incorrect. Some are pure fiction. Stick to the problem as formulated below, not to actual history.*]

Sergio Lopez was a rich ship owner living in Spain in the 17^{th} century. In 1650, one of his ships was sunk in the Caribbean Sea with a huge cargo of gold. In 2008, an international company, Gold-Retriever, found the wreckage of the ship and brought back the gold worth $50 million. A month later, a Spanish citizen, named Alberto Lopez claimed that he was a direct descendent of Sergio and that, therefore, he was the owner of the gold. To prove his point he hired an attorney and asked him to collect all birth certificates showing that he is a direct descendant of Sergio.

We will assume that Spain has kept archives of all birth certificates during the last four centuries and that all Alberto's ancestors up to Sergio lived in Spain. We will also assume that the population of Spain between Sergio's birth and the present time has monotonically increased from about 6 million people to about 41 million today.

Let Parents(x) be the function that returns the two parents of an individual x (and the date of birth of x) and Children(y) the function that returns the children of an individual y (and their dates of birth) We will finally assume that a birth certificate can be retrieved as easily by either knowing the name of a child or the name of one of his/her parents. In other words, it will cost the attorney the same amount of time to evaluate either function Parents or function Children.

1. Formulate the Attorney's task to prove that Alberto is a descendant of Sergio as a search problem: What are the state space, the initial state, the successor function, and the goal state? Is there a single possible formulation? If your answer is 'no', then give another one.

2. What would be the easier way for the attorney to proceed: with an initial state of Alberto, or an initial state of Sergio? Why?

3. Which search technique – breadth-first, depth-first, bi-directional, iterative deepening – would you recommend to the attorney? Justify your answer. [*Note: There might not be a clear-cut answer*.]

Name ………………………………………………………………..

II. 8-Puzzle (25 points)

Consider the 8-puzzle game, a 3×3 board of 9 tiles (Figure 1). Eight tiles are numbered 1 through 8. The 9th tile is the empty tile. Each legal move consists of switching the positions of the empty tile and one of its adjacent tiles. For instance, in the state of the 8-puzzle shown in Figure 1, the empty tile can be switched with either tile 2, or tile 7.

8	2	
3	4	7
5	1	6

Figure 1: A state of the 8-puzzle game

1. Assume that the goal state *G* is as shown below, where all the tiles are sorted from left to right and top to bottom, with the empty tile at the bottom right:

1	2	3
4	5	6
7	8	

   What criterion can we use to determine whether a given initial state *I* can be transformed into *G* by a sequence of legal moves of the empty tile? [*Hint: Adapt the criterion of slides 17-19 of class #2*.]

2. Consider the search problem where the goal test GOAL? (see slide 11 of class #2) only requires the empty tile to be located at the center of the board. This goal test says nothing about the location of the other tiles.
   a. How many states of the 8-puzzle satisfy the goal test?
   b. Given an initial state of the 8-puzzle, how many goal states can be reached?

Name ………………………………………………………………..

III. Multi-Agent Planning (25 points)

Consider $k > 1$ agents (for instance, robots), numbered 1, 2, ..., k, moving in an $n \times n$ grid. Each square of the grid either is empty or is an obstacle.

Time is discretized into successive instants $i = 0, 1, 2, ...$ separated by one unit of time. An agent can only be in an empty square. No two agents can be in the same square at the same instant i. The agents must move step by step from an initial state where they occupy certain squares to a goal state where they occupy other squares. Each step starts at some instant i and ends at the next instant $i+1$. At each step, each agent can move one square up, down, left or right, or stay in the same square. Although two agents cannot be in the same square at the end of any step, they can swap positions during a step. The cost of each step is 1, independent of the number of agents that actually move.

This problem is illustrated by Figure 3, where $k = 4$ and $n = 10$: (a) shows an initial state, (b) and (c) show two possible successor states of the initial state, and (d) shows a goal state. To reach the state in (c) agents 1 and 2 swap positions.

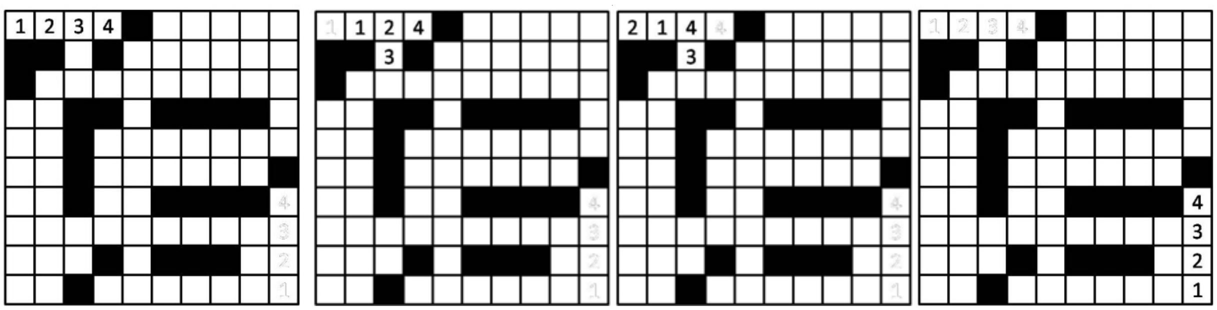

Figure 3: Illustration for the multi-agent planning problem

Figure 3, however, is only an illustration. **The following questions must be answered for the general case of $k > 1$ agents in an $n \times n$ grid with an arbitrary distribution of obstacles.**

We assume that a single problem solver decides how the agents move at each time step. So, this problem solver searches a state space, where each state is defined by the positions of the k agents.

1. What is the size of the state space? Try to give the smallest upper bound valid for any obstacle distribution. Show that this upper bound can be achieved.

2. What is the branching factor (expressed as a function of k and n) of this search problem? Again, try to give the smallest upper bound valid for any obstacle distribution. Show that this upper bound can be achieved.

Name ………………………………………………………………..

IV. *n*-Queen Problem (25 points)

The goal of the 8-queen problem is to place 8 queens on a chessboard, such that no two queens attack each other. The *n*-queen problem generalizes this problem to an arbitrary number, $n > 1$, of queens and an $n \times n$ board.

One way to formulate the *n*-queen problem as a search problem is the following (this corresponds to formulation #2 given in the slides shown in class):

- The states are all the arrangements of k queens in the k leftmost columns of the board with no two queens attacking each other, where $k = 0, 1, 2, ..., n$,
- The initial state is the state with 0 queens on the board,
- Each of the successors of a state is obtained by adding one queen in a square that is not attacked by any queen already in the board, in the leftmost empty column,
- The goal test is "all n queens are on the board".

1. Explain why the number of states in the state space is at least $(n!)^{1/3}$.

2. In class you were told that the exact number of states for the 8-queen problem is 2,057. Compare this number to $(8!)^{1/3}$. Are the two numbers very different? If yes, why?