

# CS 121, Summer 2009

## Homework #1

---

**Out:** June 24, 2009

**Due:** July 6, 2009

---

**How to complete this Homework:** Your answers can be typed or carefully hand-written. Please begin each problem on a new page and make sure your name is written on each page of your assignment. Print out this cover sheet and fill in your name and email address, as well as any people you collaborated with. Staple all of your work together and turn in at the beginning of class, July 6, 2009. If you will not be in class you can turn it in prior to class under the door of Gates 132 (with the time of submission written on the homework). Late homeworks will not be accepted.

**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, so long as your final submission is prepared on your own without referring to any notes taken during such collaboration. 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
1	15	
2	20	
3	20	
4	20	
5	25	
Total	100	

---

**1. PEAS and environment types [15 points]**

For each of the following agents:

- Develop a PEAS description of the task environment (see Section 2.3)
  - Characterize the environment according to the properties given in Section 2.3
- (a) [5 points] A computer poker playing agent
- (b) [5 points] A security robot that monitors a buildings hallways during the night
- (c) [5 points] A scheduling agent that coordinates employee schedules and coordinates staff meetings

**2. Jug problem [20 points]**

Express the following problem as a search state problem:

*Three jugs are filled with water. There are no measuring markers on any of them. Each can be fully or partially emptied in a drain or another jug. The jugs respectively measure 15, 7, and 3 gallons. The task is to measure out exactly 2 gallons.*

Give:

- (a) [7 points] the general description of a state
- (b) [3 points] the initial state
- (c) [4 points] the goal test
- (d) [6 points] the successor function. For this it is not necessary that you write every possible “state  $\rightarrow$  state” combination, but you should make it clear how one would compute the successor states of any given state.

Note that we do not ask you to describe a solution of the problem!

**3. Basic Search Methods [20 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 history]*

Sergio Lopez was a rich ship owner living in Spain in the 17th 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 Albertos ancestors up to Sergio lived in Spain. We will also assume that the population of Spain between Sergios birth and the present time has monotonically increased from about 6 million people to about 41 million today.

Let  $\text{Parents}(x)$  be the function that returns the two parents of an individual  $x$  (and the date of birth of  $x$ ) and  $\text{Children}(y)$  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  $\text{Parents}()$  or function  $\text{Children}()$ .

- (a) [14 points] Formulate the Attorneys 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 ‘yes’, then explain why there is only one. If your answer is no, then give another one.
- (b) [3 points] What would be the easier way for the attorney to proceed: with an initial state of Alberto, or an initial state of Sergio? Why?
- (c) [3 points] Which search technique – breadth-first, depth-first, bi-directional, iterative deepening – would you recommend to the attorney? Justify your answer. [There might not be a clear-cut answer.]

#### 4. Four-peg Towers of Hanoi Puzzle [20 points]

The four-peg version of the Towers-of-Hanoi puzzle (Figure 1) is as follows. Four pegs  $A$ ,  $B$ ,  $C$ , and  $D$  can each hold  $n$  rings at a time. There are  $n$  rings  $R_1, R_2, \dots, R_n$ , such that  $R_i$  is smaller than  $R_j$  for any  $i < j$ . A legal placement of the rings on the pegs requires that

- whenever any two rings appear on the same peg, the smaller one is above the larger one, and
- all  $n$  rings must be on pegs.

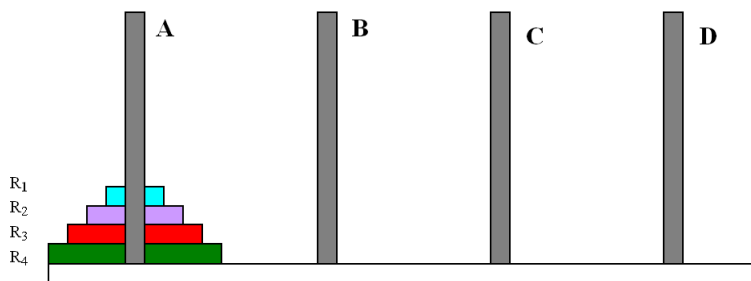


Figure 1: Towers of Hanoi puzzle start placement (4 pegs)

In the start placement, all rings are on peg  $A$  (Figure 1 shows the start state for  $n = 4$ ). In the goal placement, all rings are on peg  $D$ . A move consists of moving a single ring from one peg to another. It is legal if it produces a legal placement. The goal of the puzzle is to find a sequence of legal moves that achieves the goal placement from the start placement.

- (a) [10 points] Formulate this problem as a search problem, that is:
- [3 points] Define a representation for a state. [We don't ask you to list all states! Only define a representation that is general enough to describe every state.]
  - [3 points] Give the initial and goal states in this representation.
  - [4 points] Define the successor function. [Here, we ask you to describe how one can compute the successors of a state.]
- (b) [5 points] What is the total number of legal states?
- (c) [5 points] How many successors can a state have at most?

5. **Blind search on a grid [25 points]**

The state space for a search problem is given in Figure 2. The remainder of the search problem is defined as follows:

- **Initial state:** State  $a$
- **Goal state:** State  $o$
- **Successor function:** The successor function applied to any state  $s$  gives the states connected to  $s$  by an edge. The successors are generated in *alphabetical* order. They are added to the **FRINGE** in the same order.
- **Edge cost:** The edge costs are all 1

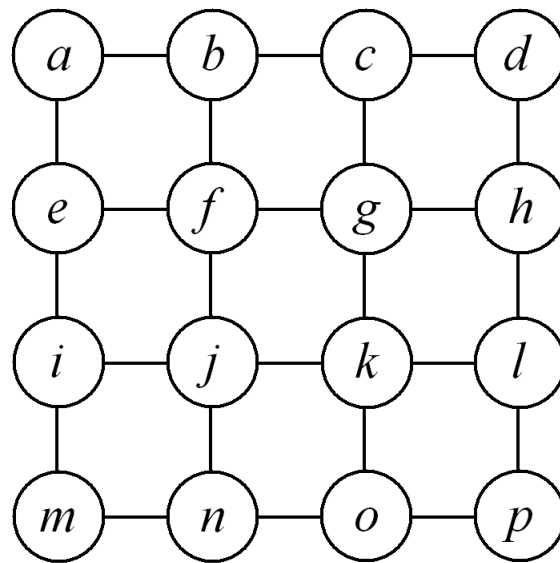


Figure 2: Grid state space for question 5

For each of the following blind search algorithms:

- List the states in the order they are **expanded** by the search algorithm
- Give the path from the initial state to the goal state that is returned by the algorithm.

Assume that the successor function does not generate states that have already been added to the **FRINGE**. Also, only apply the goal test to a state when it is expanded.

- [8 points]** Breadth-first search
- [8 points]** Depth-first search
- [9 points]** Bidirectional search – Use breadth first starting from both the start and the goal states, expanding one state at a time in each search, alternating searches after each expansion. Begin with the start state. When a node is going to be expanded by one search, check to see if it is already in the **FRINGE** of the other search. If it is, then the search concludes. Assume that predecessors are also generated and added to the **FRINGE** in alphabetical order.