Introduction to AI, Winter 2011
Assignment 1: Search and CSP

Out: November 9
Due: November 29, 12:15pm

NOTE: The maximum score for this homework is 110. If you submit the homework late, 10% of your score will be taken off per day. (10% for a single day late, 20% for two days late, and so on.)

Problem 1 - Search Formulation (20 points)

Part A: Towers of Hanoi

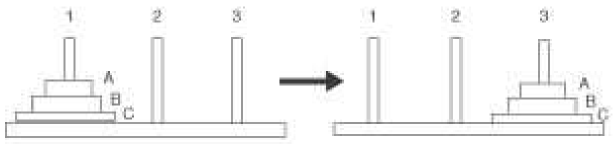


Figure 1: Q1: Towers of Hanoi

You are requested to phrase the 'Tower of Hanoi' puzzle as a search problem. In Towers of Hanoi, disks of various sizes are mounted on pegs, where a disk must always be placed on top of a larger disk. We will assume three pegs and three disks, as shown in Figure 1. In the initial state, all disks are mounted on the leftmost peg (see left part of Figure 1). The goal is to move all pegs to the rightmost peg (see right part of Figure 1). It is only allowed to move one disk at a time.

1. Define a state representation. A state should correspond to a specific position of disks. A good representation can be easily converted to some data structure in code.
2. Specify the initial and goal states representations.
3. What are the possible states after one move from the initial state? after 2 moves? after 3 moves? describe it as a search tree. (Use the same state representation.)

Part B

Three Westerners and three cannibals are on one side of a river, where one boat is available. The boat can carry one or two people. (The boat can not be moved to the other side of the river without people in it). The goal is to get everyone to the other side. Importantly, there must not be more cannibals then Westerners on either side of the rivers, otherwise the Westerners will be (complete yourself :).

Again, you are to formulate this puzzle as a search problem.

1. Define a state representation.
2. Give the initial and goal states in this representation.
3. What are the possible states after 1 move? 2 moves? after 3 moves? describe it as a search tree.
4. What is a suitable cost function in this case?

Problem 2 - Search (15 points)

All numbers can be reached from the number 1 using a sequences of the operations: $2n$ and $2n+1$.

1. Draw the corresponding state tree for states 1 to 15. As described above, given a state of value n, there are two successor functions to the state, of values $2n$ and $2n+1$. The initial state is 1.
2. Suppose the goal state is 11. List the exact order in which nodes will be visited for: breadth first search, depth first search, depth-limited search with limit 3, and iterative deepening search.
3. Describe how one could apply bidirectional search to this problem (starting from both the initial and goal states). Why is it appropriate?
4. What is the branching factor in each direction of the bidirectional search?
5. Suggest a more efficient search strategy to solve this specific problem.

Problem 3 - CSP (25 points)

You are given the crossword in Figure 2.

```
      1   2   3   4   5
    +---+---+---+---+---+
  1 | 1 |   | 2 |   | 3 |
    +---+---+---+---+---+
  2 | # | # |   | # |   |
    +---+---+---+---+---+
  3 | # | 4 |   | 5 |   |
    +---+---+---+---+---+
  4 | 6 | # | 7 |   |   |
    +---+---+---+---+---+
  5 | 8 |   |   |   |   |
    +---+---+---+---+---+
  6 |   | # | # |   | # |
    +---+---+---+---+---+
```

Figure 2: Crossword

This crossword should be filled with (a subset of) the following words:

ART, ASSET, EGG, FAN, HEEL, HONOR, KNOT, LEO, LIE, LIMB, MINE, MEET, NAILS, ROBOT, SHEET

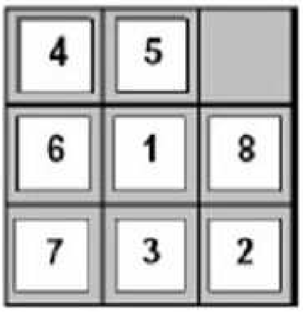
The numbers 1,2,3,4,5,6,7,8 in the crossword puzzle correspond to the words that will start at those locations.

- This crossword problem can be formulated (and solved) as a constraint satisfaction problem. What are the constraints in this case? Detail all constraints, and their domains.
- Show the constraint graph.
- One way to solve the puzzle is to check all possible assignments until a solution is found. This method is very slow. Its efficiency can be improved by variable ordering. Give an ordered list of variables to be assigned, and explain.
- Solve the puzzle (give a correct assignment), and detail the solution steps.

Problem 4 - 8-puzzle (50 points)

Implement both the best-first and the A* search algorithms to search for a solution to the 8-puzzle, as described in the textbook and lecture. The input to your program should be a configuration of the 8 puzzle, e.g.,

(4 5 b 6 1 8 7 3 2) (where b is a blank) corresponds to:



The goal state is (1 2 3 4 5 6 7 8 b).

Implement three different heuristics: the two described in the textbook (hamming distance and Manhattan distance) plus one of your own for evaluating states in the state space. Run your search algorithms using each of these three heuristics on each of three different initial states, specified in the files example1.txt, example2.txt and example3.txt. Your program should output the solution path it found (if one was found), for example,

(4 5 b 6 1 8 7 3 2) $\rightarrow$ (4 b 5 6 1 8 7 3 2) $\rightarrow$ $\rightarrow$ (1 2 3 4 5 6 7 8 b).

An algorithm should stop after some maximum number of steps if a solution is not found.

In your write-up, for each of the methods and heuristics, give the solution path discovered for each initial state (Overall, there are 2 methods, 3 heuristics and 3 initial states, so there should be 18 paths.); for each solution path, it should be which method and heuristic were used to generate it. In addition, you should detail the average number of steps per method and heuristic (averaging over the three examples).

Write a paragraph describing the conclusions you draw from your results. You should refer to optimality and running time, given the method used and the quality of the heuristic.

Logistics

Your code should accept command-line arguments specifying the puzzle name and the search type:
search example.txt dfs
or
search example.txt astar

Three example files are given in the Homework folder in the HighLearn system.

You should submit:

- A description of the heuristic you used for the A* search. Explain why the heuristic is admissible.
- Your source code (the code must include a class called *search* (that accepts the above arguments), compiled and tested.)
- Include the output of your code for the three example files, using DFS and A*. What are your conclusions?