CSE 4301/5290 Homework 2
Due: Sep 30, Wed, 5pm
Submit Server: class = ai, assignment = hw2

For programming problems (lisp/java/c/c++/python):

- Submit:
  - all files that are needed to compile and run (and executable)
  - README.txt with compilation and run instructions

- Your program should compile and run on code.fit.edu (Linux, remote access via ssh)

1. Q3.6b, p113, 3Ed (Q3.7b, p90, 2Ed). Consider a 10x10x8 (Wx Dx H) grid/discrete world. (a) Define states as a set of variables and their possible values. (b) Derive the number of possible states. (c) Successor function: for each action, discuss the input/parent and output/child state. (d) Define goal test with respect to states in (a).

2. Consider Deep Blue can evaluate 200 million positions a second. Assume at each move, a pawn can go to 2 possible positions, a rook 14, a knight 8, a bishop 14, and a king 8. Each side has 8 pawns, 2 rooks, 2 knights, 2 bishops, a queen and a king. Under standard regulations, each side makes 40 moves within the first 2 hours (or 3 minutes a move on the average).

(a) Using the breadth-first search algorithm, how many levels can Deep Blue evaluate (visit) before each move (in 3 minutes)?
(b) To examine 20 levels in 3 minutes, how many positions Deep Blue needs to evaluate (visit) in a second?

3. Simplified from Q3.6d, p114, 3Ed (Q3.7d, p90, 2Ed). Consider Deep Blue can evaluate 200 million positions a second. Assume at each move, a pawn can go to 2 possible positions, a rook 14, a knight 8, a bishop 14, and a king 8. Each side has 8 pawns, 2 rooks, 2 knights, 2 bishops, a queen and a king. Under standard regulations, each side makes 40 moves within the first 2 hours (or 3 minutes a move on the average).

(a) Using the breadth-first search algorithm, how many levels can Deep Blue evaluate (visit) before each move (in 3 minutes)?
(b) To examine 20 levels in 3 minutes, how many positions Deep Blue needs to evaluate (visit) in a second?

4. Explain and derive the number of *reachable* states in Problem 3 with both jugs empty initially.

5. Q3.9, p115, 3Ed (Q3.9, p90, 2Ed). Missionaries and Cannibals (MC) problem. Name the banks of the river as left and right, and the missionaries and cannibals are initially on the left bank.

(a) Q3.9a. Cost: one unit to move a missionary and two units to move a cannibal on the boat from one bank to the other.
(b) Programming: Using the same general-search in Problem 3d, add mc- functions that can solve the MC problem using UCS and A* search. For A*, describe why your heuristic function is admissible in the comments. For testing different initial states, provide in Lisp:

```lisp
(defun test-jug (jug-init-state alg)
  (cond
    ((equal alg 'ucs)
     (general-search #'ucs-insert jug-init-state #'jug-goal-test
                    #'jug-successor #'jug-step-cost nil))
    ((equal alg 'astar)
     (general-search #'astar-insert jug-init-state #'jug-goal-test
                    #'jug-successor #'jug-step-cost #'jug-heuristic))
    (t (format t "Error: unknown algorithm ~a~%" alg))
  )
)
```

or in java/c/c++/python 3 command-line arguments: small-start, large-start, alg; for example: java TestJug 0 0 astar and print the solution path returned by general-search.