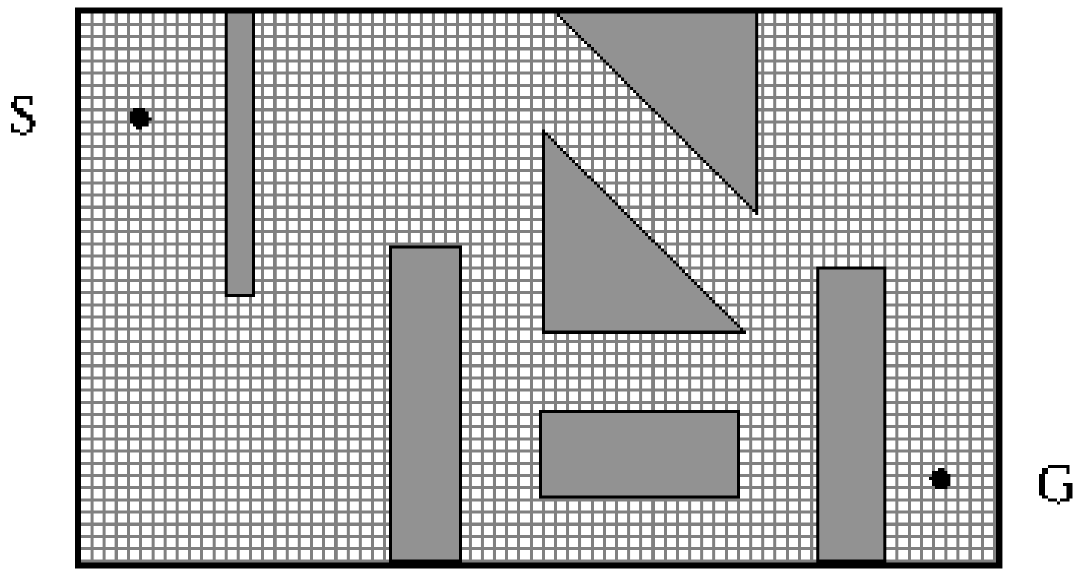
COMP3411/9414/9814 Artificial Intelligence
Session 1, 2015

Assignment 2 - Heuristics and Search

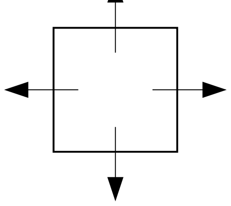
Due: Sunday 3 May, 11:59pm
Marks: 8% of final assessment

Question 1 – Maze Search Heuristics

Consider the problem of an agent moving around in a 2-dimensional grid world, trying to get from its current position (x, y) to the Goal position (x_G, y_G) in as few moves as possible, avoiding obstacles along the way.



(a) Assume that at each time step, the agent can move one unit either up, down, left or right, to the centre of an adjacent grid square:



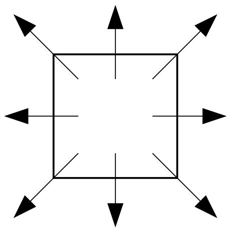
One admissible heuristic for this problem is the Straight-Line-Distance heuristic:

$$h_{\text{SLD}}(x, y, x_G, y_G) = \sqrt{(x - x_G)^2 + (y - y_G)^2}$$

However, this is not the best heuristic. Name another admissible heuristic which dominates the Straight-Line-Distance heuristic, and write the formula for it in the format:

$$h(x, y, x_G, y_G) = \quad \dots$$

(b) Now assume that at each time step, the agent can take one step either up, down, left, right or **diagonally**. When it moves diagonally, it travels to the centre of a diagonally neighboring grid square, but a diagonal step is still considered to have the same "cost" (i.e. one "move") as a horizontal or vertical step (like a King move in Chess).



(i) in terms of estimating the number of moves to reach the goal, is the Straight-Line-Distance heuristic still admissible? Explain why.
(ii) is your heuristic from part (a) still admissible? Explain why.
(iii) devise an admissible heuristic for this problem, and write a formula for it in the format:

$$h(x, y, x_G, y_G) = \quad \dots$$

Question 2 – Comparing BFS and A* for the 8-Puzzle

In this question you will construct a table showing the number of states expanded when the 8-puzzle is solved, from various starting positions, using four different searches:

a. Breadth First Search, excluding repeated states
b. Breadth First Search, including repeated states
c. A*Search, with Total Manhattan Distance heuristic
d. A*Search, with Count Misplaced Tiles heurisitic

(a) Copy the files from `prolog_search` into your own directory by typing

```
mkdir prolog_search
cd prolog_search
cp /home/cs9414/public_html/15s1/prolog_search/* .
```

(alternatively, you could click "`prolog_search`" on the Course Web site, and use "Download linked file as...")

Start `prolog` and load `puzzle8.pl` and `breadthfirst.pl` by typing

```
[puzzle8].
[breadthfirst].
```

Then invoke the search for the specified `start1` position by typing `start1(Pos),solve(Pos,Sol,N),showsol(Sol).`

(Note that the total number of states expanded during the search is returned as `N`). Repeat the above search for `start2`, `start3`, `start4` and `start5` and put these numbers into a table.

(b) Now comment out the line in `breadthfirst.pl` that says "exclude repeated states", and re-run the searches from part (a) for `start1` and `start2` (the others might take too long, or run out of memory).
Add these results to your table.

(c) Now switch from Breadth First Search to A*Search (by loading `astar.pl` instead of `breadthfirst.pl`) and repeat the searches from part (a).
Add the results to your table.

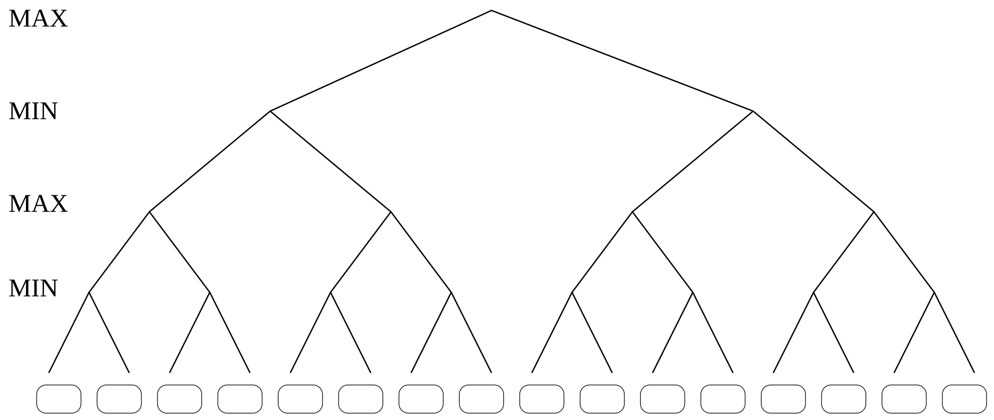
(d) Modify the code for `puzzle8.pl` so it uses the Count Misplaced Tiles heuristic instead of the Total Manhattan Distance heuristic (briefly show which section of code was changed, and the replacement code – you are free to use "cut" if you wish). Repeat the searches from part (c) and add the results to your table.

(e) Based on your results, rank the four searches in order of slowest to fastest.

(f) Invent three new starting positions – requiring 20, 23 and 26 moves, respectively. Run the fastest of the four searches on these states, and add the results to your table.

Question 3 - Game Trees and Pruning

(a) Consider a game tree of depth 4, where each internal node has exactly **two** children (shown below). Fill in the leaves of this game tree with all of the values from 0 to 15, in such a way that the alpha-beta algorithm prunes as many nodes as possible. Hint: make sure that, at each branch of the tree, all the leaves in the left subtree are preferable to all the leaves in the right subtree (for the player whose turn it is to move).



(b) Trace through the alpha-beta search algorithm on your tree. How many of the original 16 leaves are evaluated?

(c) Now consider another game tree of depth 4, but where each internal node has exactly **three** children. Assume that the leaves have been assigned in such a way that the alpha-beta algorithm prunes as many nodes as possible. Draw the shape of the pruned tree. How many of the original 81 leaves will be evaluated?

Hint: If you look closely at the pruned tree from part (b) you will see a pattern. Some nodes explore all of their children; other nodes explore only their leftmost child and prune the other children. The path down the extreme left side of the tree is called the line of best play or Principal Variation (PV). Nodes along this path are called PV-nodes. PV-nodes explore all of their children. If we follow a path starting from a PV-node but proceeding through non-PV nodes, we see an alternation between nodes which explore all of their children, and those which explore only one child. By reproducing this pattern for the tree in part (c), you should be able to draw the shape of the pruned tree (without actually assigning values to the leaves or tracing through the alpha-beta algorithm).

(d) What is the time complexity of alpha-beta search, if the best move is always examined first (at every branch of the tree)? Explain why.

Submission

This assignment must be submitted electronically.
COMP9414/9814 students should submit by typing

```
give cs9414 hw2 ...
```

COMP3411 students should submit by typing

```
give cs3411 hw2 ...
```

The give script will accept `*.pdf *.txt *.doc *.rtf`
If you prefer some other format, let me know.
Late submissions will incur a penalty of 15% per day, applied to the maximum mark.

Group submissions will not be allowed. By all means, discuss the assignment with your fellow students. But you must write (or type) your answers individually. Do NOT copy anyone else's assignment, or send your assignment to any other student.

Good luck!