Artificial Intelligence: Homework 1 (HW1)

Instructor: Dr. Shengquan Wang

Due Time: 10PM, 9/27/2020

Instructions

- No teamwork is allowed for all homework assignment.
- `VeriCite` option has been enabled on Canvas to compare your submission with other submissions from this semester and past semesters and Internet database.
- Your solution should be written in a ".docx", ".doc", or ".pdf" format and submitted to the assignment folder HW1 on Canvas.

Questions

1. Consider the following task environments, show their corresponding i) `PEAS` and ii) `properties` of task environments as shown in the Introduction slides:
   (a) Playing football
   (b) Mowing the lawn
   (c) Translating English to German
   (d) Performing a high jump
2. We consider the maze under windy situation as shown in Figure 1, where 3b is the goal state and 4c is the start state and all black squares are obstacles. We assume that the wind comes from the north

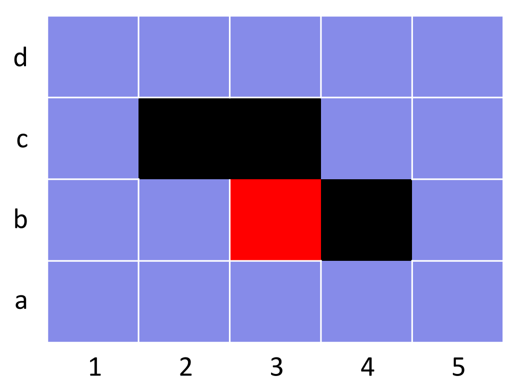

Figure 1: Maze

and the cost of one step for the agent is defined as follows: 1 for moving southward; 2 for moving westward or eastward; 3 for moving northward.

For all best-first search algorithms (GBFS and A*), we use a modified Manhattan distance used in class as the heuristic function $h(n)$ by considering the windy situation. For example, for the start node, the agent has to move at least 1 step southward and 1 step westward in order to reach the goal. Therefore, we have $h(n) = 1 * 1 + 2 * 1 = 3$ at the start node.

We use a label we did in class to indicates the order of choosing the corresponding unlabeled square and adding it to the frontier. To break tier for unlabeled squares (expanding children nodes), use this order: first westward; then northward; then eastward; then southward. To break tier for labeled squares (picking one child node to expand), the smallest label is picked first.

Follow the same way as done in the class to show the search steps with labels (plus subscript numbers if needed) inside circles for the following search algorithms:

(a) `BFS`

(b) `UCS`

(c) `DFS`

(d) `GBFS`

(e) `A*`

3. Please visit `http://tristanpenman.com/demos/n-puzzle/` to play the 8-puzzel problem. Choose the following configuration: A* search for search algorithm; Titles out-of-place for the heuristic function; single step model for the control mode. Pay attention to how the search tree is built and how each node in the tree is structured. We will use the same format to draw the search tree for the following variant of the 5-puzzle problem with obstacles (left: start state; right: goal state; black square: obstacle).

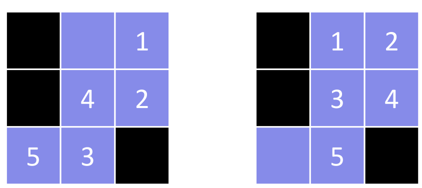

Figure 2: 5-Puzzle

4. We consider a Sudoku puzzle as shown in this webpage: `https://www.websudoku.com/?level=2&set_id=7029409198`. Make sure the ID number is 7029409198. It has 52 open squares. Solve the puzzle with the constraint propagation through inference. Show two screenshots in your report:

   (a) One is an intermediate result with the number of unfilled open squares between 15 and 30;

   (b) The other is the final result with no open squares remaining.

   Make sure that your screenshots have your username and the date in the background. They work as a proof that you did it.

5. We consider a variant of 5-Queen problem. We have 3 white queens and 3 black queens. Attacking can only occur between white queens and black queens. The same-color queens don't attack their own kind. Place all these 6 queens on the chessboard so that no attackings occur. Show all possible solutions if there are no more than 10 solutions. Otherwise, show 10 of them.

6. We play a Tic-Tac-Toe game. Go to this webpage: `https://www.google.com/search?q=tic+tac+toe` and play a couple of games for warmup. We assume you start with `X` and Table 1 shows its current state of your game play, where next turn is you as `X`. Draw a game tree starting from the current state with `X` as `MAX` and `O` as `MIN`. Show the values of the `MAX` and `MIN` nodes. We assume that a win is 1 point; a draw is 0 point; a loss is -1 point.

X	O	O
	X	
	X	O

Table 1: A Tic-Tac-Toe game

7. Consider the game tree in Figure 3.

   (a) What are the values of the `MAX` and `MIN` nodes? No pruning is used here.

   (b) We consider the alpha-beta pruning method and we assume that the right child is added to the frontier first and the left child is expanded or returns utility (if terminal node) first. Cross out the pruned branches and also cross out the node(s) whose exact value(s) the alpha-beta pruning method never determines.

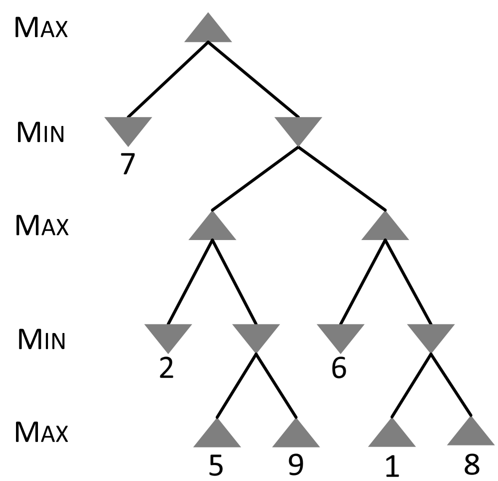

Figure 3: Game Tree

8. Consider the game tree for a stochastic coin-flipping game in Figure 3. This is an uneven coin-flipping: for each coin flipping, head (H) will show up with 25% while tail (T) showing up with 75%. Obtain the value at each node.

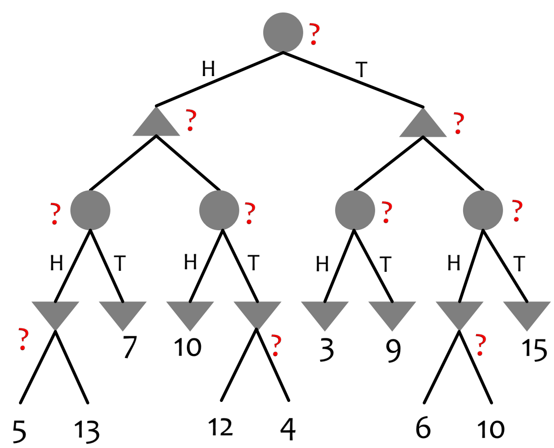

Figure 4: Stochastic Game Tree