

## CS4750/7750 HW #4 (20 points + 4 bonus points)

Fall 2018

Implement the minimax algorithm to play a two-player, four-in-a-row game, which is a variation of tic-tac-toe: two players, *X* and *O*, take turns marking the spaces in a 6×6 grid. The player who succeeds in placing 4 of their marks consecutively in a horizontal, vertical, or diagonal row wins the game.

You may use any programming language in your implementation.

- 1) Player 1: Places the first piece in the center of the board, if making the first move, and runs the minimax algorithm on a 2-ply game tree, i.e., looking ahead 2 moves (one move by the player and one move by the opponent) and then apply the following evaluation function on the resulting board. The heuristic evaluation function for cutoff nodes (non-terminals) is

$$\begin{aligned} h(n) = & 5 * [\# \text{ of two-side-open-3-in-a-row for me}] \\ & - 10 * [\# \text{ of two-side-open-3-in-a-row for opponent}] \\ & + 3 * [\# \text{ of one-side-open-3-in-a-row for me}] \\ & - 6 * [\# \text{ of one-side-open-3-in-a-row for opponent}] \\ & + [\# \text{ of open-2-in-a-row for me}] - [\# \text{ of open-2-in-a-row for opponent}] \end{aligned}$$

- “one-side-open-3-in-a-row”: there is a blank space at one end of a 3-in-a-row.
- “two-side-open-3-in-a-row”: there are blank spaces at both ends of a 3-in-a-row.
- “open-2-in-a-row”: there are blank spaces at one or both ends of a 2-in-a-row.

For example, for player ‘x’, the value of the following state is

$$h = 5*0 - 10*1 + 3*1 - 6*0 + 1 - 2 = -8$$

	O	X			
	O	O	X		
O	X	X	O		
	X				

For states with the same  $h$  values, the search breaks tie by placing a piece toward the center of the board. Breaking the remaining ties randomly.

- 2) Player 2: same as Play 1, but runs the minimax algorithm on a 4-ply game tree, i.e., looking ahead 4 moves (two moves by the player and two moves by the opponent).
- 3) Report every step of a game played between Player 1 (plays first) and Player 2, by showing the board after every move. In addition, for every move by either Player 1 or 2, report the # of nodes generated by the search algorithm and the CPU execution time in milliseconds.
- 4) (bonus 4 points).
  - a. Report the result of 100 games played between Player 1 (plays first) and Player 2, in terms of (# Player 1 win, # Player 2 win, # tie).
  - b. Same as a), but Player 2 plays first.

Submission:

- a) A brief description of your implementation and experiments in terms of hardware, software, and existing code used.
- b) Report the result of item 3) above.
- c) Report the result of item 4) above.
- d) A zip file containing your code with appropriate comments.

You may form a team of up to three people. The team may be different from that for HW #2. Each team needs one submission on Canvas.