# CSE 4301/5290 Homework 3

Due: Oct 23, Wed, 5pm; Submit Server: class = ai , assignment = hw3

For programming problems (Lisp, Java, C, or C++):

- Submit:
  - all files that are needed to compile and run
  - README.txt with compilation and run instructions
- Your program should compile and run on `code.fit.edu` (Linux, remote access via ssh) or `hopper.cs.fit.edu` (Windows, remote access via Remote Desktop).

1. Q5.9, p197, 3Ed (Q6.1, p189, 2Ed)
2. Programming:
   - **CSE 4301 only** Connect 4 has a rack with 7 columns, each column has a depth of 6. Each player in turn drops a token into one of the 7 columns. The first player achieving 4 in a row/column/diagonal wins. Each move is specified by a column number.

```
| | | | | | | |
| | | | | | | |
| | | | |X| | |
| | |O|X|O| | |
| | |X|O|O| | |
| |X|O|X|X| | |
---------------
 0 1 2 3 4 5 6
```

   - **CSE 5290 only** 4x4x4 3D tic-tac-toe. Each player in turn marks a cell, the first player achieving 4 in a row/column/diagonal wins. Each move is specified by level, row, and column numbers (in that order).

```
   Level 0     Level 1     Level 2     Level 3

   0 1 2 3     0 1 2 3     0 1 2 3     0 1 2 3
  ---------   ---------   ---------   ---------
0 |X| | | |   | | | | |   | | | | |   | | | | |
  ---------   ---------   ---------   ---------
1 | | | |O|   | |X|O| |   | | | | |   |O| | | |
  ---------   ---------   ---------   ---------
2 | | | | |   | | | | |   | | |X| |   | | | | |
  ---------   ---------   ---------   ---------
3 | | | | |   | | | | |   | | | | |   | | | |X|
  ---------   ---------   ---------   ---------
```

### Program operations:

(a) From keyboard input, assign your program to be player $X$ or player $O$ (the opponent respectively becomes player $O$ or $X$).

(b) Player $X$ always starts.

(c) Display the initial empty configuration

(d) Display a move or enter a move from the keyboard (connect4: column; 3D tic-tac-toe: level, row and column)

(e) Check if the move is legal, **ask human** for confirmation if illegal

(f) Make the move and display the board

(g) Repeat steps (d) to (f) until there is a winner

(h) Declare winner

### Program requirements:

(a) Functions (stated in LISP) including:

```
; perform search with alpha beta pruning, return an action
; *all* actions must be determined by alpha-beta-search
; (you can vary parameters at different states)
(defun alpha-beta-search (state ...)  ...)

; return the "quality" of the state
; description of your evaluation function...
(defun eval-state (state)  ...)
```

(b) Each move should not take more than one minute. Hence, you might want to have a parameter(s) that sets the limit(s) of your search.

### LISP details

- `compile-file` prepares a compiled version of your program. You need to `load` the compiled version. Running the compiled version will be faster than interpreting the source.
- `get-universal-time` returns the number of seconds since Jan 1, 1970. `get-internal-real-time` returns the number of time units based on `internal-time-units-per-second` (a constant)—might need to handle the "wrap-around" issue: end-time < start-time.

### Tournament rules:

(a) Oct 23, Wed, 5-6:15pm

(b) CSE 4301 and CSE 5290 students compete separately.

(c) Your program will play against two other programs, which are randomly assigned.

(d) Your program starts in one game; your opponent starts in the other game.

(e) You can get up to 10 points, which constitute 10% of your hw3 grade. (win: 5 points; tie: 3.5; lose: 2; non-functioning: 0)

(f) Your program wins if it functions but your opponent's doesn't.

(g) Each move cannot take more than one minute. Your program is considered non-functioning if it takes too long.

(h) Your program is considered non-functioning if it suggests or allows illegal moves.

(i) If a game takes more than half an hour, the game may be declared as a tie.

---

### CSE 5290 only

3. Q5.16, p200, 3Ed