355 fall 2022 assignment 5

For this assignment, post CLARIFYING QUESTIONS ONLY on eclass. Posting suggestions or your answers or hints on eclass or any other site – e.g. a discord server or anywhere else – is plagiarism.

You can work on this assignment in groups of up to 5: within your group, you can discuss any questions, but you cannot copy answers. Each student must submit their own assignment. Discussing or copying with any student outside your group is plagiarism.

We might ask you later to explain your answers: if you are unable to do so, we might deduct some or all marks and report this to the faculty of science.

For this assigment, each student's secret number is the 4th and 5th integer of their student id, interpreted as a 2-digit number. E.g. if your id is `***91**`, then your secret number is 91. Some questions ask you for m, your secret number mod 3. E.g. if your secret number is 91, your m is 1.

Submit each answer on eclass.

1. **[1 marks] If you do not answer this question we will not mark the assignment and your assignment score will be 0.**

   (a) In your own words, state that you accept the plagiarism policy above.

   (b) Give the names and ccids of all members of your discussion group (including yourself). Explain how you worked together: e.g. discussed every question, discussed only questions 1 and 3 with group members X and Z, etc.

2. [**5 marks**] Below is scrambled negamax code for tic-tac-toe. Rearrange the lines into the python3 procedure `negamax( )`, indent properly, and then insert the procedure (immediately after the comment that mentions negamax) into class github program `ttt/asn5.py`. Use asn5.py to answer the following questions.

```
calls += 1
calls += c
def negamax(calls, psn, ptm): # ptm: 1/0/-1 win/draw/loss
else:
for j in range(9):
if Cell.e not in psn.brd:
if psn.brd[j] == Cell.e:
if psn.has_win(ptm):
psn.brd = change_string(psn.brd, j, Cell.e)   # reset brd to original
psn.brd = change_string(psn.brd, j, ptm)
result, c = -1, 0 # result will be negated below
result, c = negamax(0, psn, opponent(ptm))
return 0, calls  # board full, no winner
return so_far, calls
so_far = -1  # best score so far
so_far = max(so_far, -result)
```

a) Use the program to find the minimax value of tic-tac-toe for the first player. Give the total number of calls.

b) Modify the program so that it stops the minimax search once it finds a win. (Hint. Insert this line somewhere `if so_far == 1: return 1, calls`) Now repeat question a) with your modified program.

c) Alt-ttt is this variant of tic-tac-toe: the game always continues until the board is full; if only one player has three-in-a-row, that player wins; if both players or neither player has three-in-a-row, the result is a draw. Explain briefly why the number of nodes in the tree of all continuations of alt-ttt is $1 + 9 + 9 \times 8 + 9 \times 8 \times 7 + \ldots + 9!$.

d) Give an arithmetic expression for the number of nodes in the directed-acyclic-graph (DAG) of all continuations of the game of alt-ttt, starting from the empty board. Hint. There is one node at the top level (level 0) of the dag. There are 9 nodes at level 1. There are 9×8 nodes at level 2. There are 9 choose 2 times 7 nodes at level 3, since there are 9 choose 2 ways to pick the 1st-player's two moves, and for each choice of those 2 moves, there are 7 ways to pick the 2nd player's move.

Answer like this:

$1 + 9 + 9\times8 + 9\times8\times7/2 + \ldots$

3. [**2 marks**] Solve nim(1 2 4) by hand, bottom-up. List all reachable positions in order by total number of stones. Then label each winning position W and each losing position L. E.g. here is the start of an answer for nim(1 2 3):

```
0 stones:
0 0 0  L

1 stone:
0 0 1  W

2 stones:
0 0 2  W
0 1 1  L

3 stones:
0 0 3  W
0 1 2  W
1 1 1  W

4 stones:
0 1 3  W
0 2 2  L
1 1 2  W

...
```

4. [**2 marks**] Draw the dag of all possible continuations for nim(1 2 4). Put an x under each losing position. Follow the format of the nim(2 2 2) dag drawing at the bottom of this page:

   https://webdocs.cs.ualberta.ca/~hayward/355/jem/nim.html#sol

5. [**2 marks**] Recall that a proof tree is a subtree of the minimax tree that is sufficient to prove an upper bound or lower bound on the minimax value. For x to play from your position P_m below, give a proof tree that shows that o can win.

```
P0   . . .          P1   . . .          P2     . . .
     x o x               x o x                x o x
     . o .               . o .                . o .
```

6. [**2 marks**] a) Three moves have been played in a tic-tac-toe game, leaving your tic-tac-toe position G_m below. Draw an xkcd-style diagram for tic-tac-toe showing each possible next move 4 for o and then (in each case) a minimax-best next move 5 for x. Follow this format: https://xkcd.com/832/

```
G0   x . .          G1   . . x          G2   . . .
     o x .               . x o               o x .
     . . .               . . .               x . .
```

b) Give the number of different correct answers to a). Explain briefly.