cmput 355 2025 practice questions 4

1. For each tic-tac-toe opening move, give a **sub-optimal** reply. Check answers with `ttt/tt.py`.

```
. . .          x . .          . x .
. x .          . . .          . . .
. . .          . . .          . . .
```

2. Consider this 2-player alternate-turn win/loss/draw game, played on a 3x3 board: *on a turn, a player colors any empty cell; once the board is full, the win/loss/draw condition (which we are not telling you) is used to determine the winner.*

   a) How does this game differ from tic-tac-toe?

   b) How many nodes are in the tree of all continuations of the game?

   c) Prove that the number of nodes in the graph of all continuations of the game is less than your answer to b).

3. a) When you solve tic-tac-toe from the empty board using `ttt/tt.py`, how many nodes are in the search tree? Is this about what you expected? Explain.

   b) Repeat a) using the transposition table.

   c) Repeat a) using isomorphisms.

   d) Repeat a) using isomorphisms and transposition table.

   e) In `ttt/tt.py`, in function `negamax()`, uncomment this line

   ```
   if so_far == 1: break   # improvement: return once win found
   ```

   and repeat each of the previous questions.

   f) Based on your answer to e), how could you further improve `ttt/tt.py` (so that the number of nodes searched is further reduced)?

4. Convert decimal 29 into binary. Convert binary `1 0 1 1 1 0 1 1` into decimal.

5. For a nim position with pile sizes 15, 27, 14, 25, 7, find all winning moves.

```
a    15       1 1 1 1
b    27       1 1 0 1 1
c    14       1 1 1 0
d    25       1 1 0 0 1
e     7       1 1 1
```

6. Find a 3-pile nim position with exactly 2 winning moves, or explain why there is no such position.

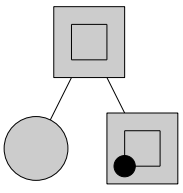
7. a) Draw the graph (dag) of all continuations of the game for nim(2 2 2). Group nodes by their multiset of non-zero pile sizes, e.g. group all 6 permutations of (1 0 2) as multiset {1, 2}, group all 3 permutations of (1 0 1) as multiset {1, 1}, etc. The root of your dag will be the node {2, 2, 2}, its children will be {1, 2, 2} and {2, 2}, and the lowest node in the dag will be {}, the position with all piles empty. Circle all losing nodes.

   b) Draw the top two levels (the root and all children) of the tree of all continuations (TOAC) of the game for nim(2 2 2). Also, for each multiset that appears as a node in the TOAC (so {}, {1}, {1,1}, {2}, {1,1,1}, {1,2}, {1,1,2}, {2,2}, {1,2,2}, {2,2,2}), give the number of nodes in the subtree of that multiset in the TOAC.

8. Here is output from recursive nim solver `nimsimp.py`. Give the missing number of states and messages: False, False dict, True dict, or True losing child.

```
(2, 1)
  (0, 2)
    (0, 0) __________
  (0, 2) __________
  (0, 1)
    (0, 0) __________
  (0, 1) __________
  (1, 1)
    (0, 1) __________
    (0, 1) __________
  (1, 1) __________
(1, 2) __________
___ states
```

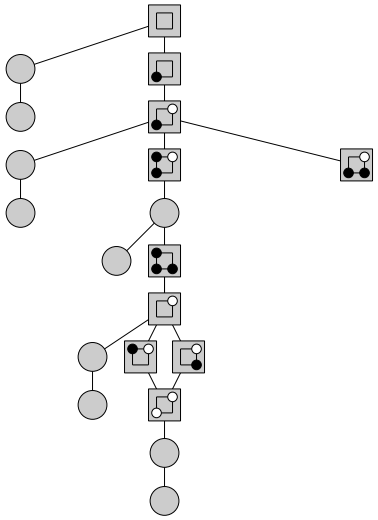
9. Starting from below, draw the next two levels of a 2×2 go second-player (White) strategy tree that shows how White can lose by at most 1. Draw each pass move as an empty circle. For the children of a game state at which no capturing has yet occurred, you can prune symmetric moves.



10. a) For a given game state S and player-to-move P, define a strategy tree.

    b) The diagram shows part of a White strategy for 2×2 go. Give the number of nodes in the complete strategy tree. Explain.

    c) Let m be the minimax score for Black for 2×2 go. Based on the diagram, what can you conclude about m? Answer like this: $m = 1$, or $m \geq -1$, or $m \leq 4$, etc. Explain.



11. What's wrong with the following argument? For each statement: state whether True or False; if False, explain.

    - For tic-tac-toe, the number of nodes in the tree of all continuations is about 500 000.
    - So ttt can be solved in a few seconds in python using minimax.
    - For 3×3 go, the number of legal positions is exactly 3^9.
    - For tic-tac-toe, the number of legal positions is exactly 3^9.
    - So we expect that 3×3 go can be solved in a few seconds in python using minimax.

12. a) According to John Tromp, what is the number of legal 3×3 go games?

    b) What is the relationship between the number of legal 3×3 go games and the number of nodes in the TOAC.

    c) Why is this number so much bigger than the number of nodes in the TOAC for tic-tac-toe?

13. Trigo is go (with positional superko and no self-capture) played on a triangle.

    Below, from the move 4 game state, draw the next 2 levels of the tree of all continuations of the game. Label each leaf node with its black minimax value (−3, 0 or 3).

14. Here is output from `go/tromp.py`, which solves 2×2 go with alphabeta. Give the missing lines. Check your answer by running the program.

```
0 (-4,4)
 . .
 . .
1 (-4,4) pass
 -----
 -----
2 (-4,0)
 o .
 . .
3 (-4,0) pass
 o .
 . .  CUT -4 -4
```

```
3 (-4,0)
 -----
 -----
4 (-4,0) pass
 o *
 . .  CUT -----
4 (-4,0)
 -----
 -----
5 (-4,0) pass
 -----
 -----
```

```
6 (-4,-1)
 o .
 o o
7 (-4,-1) pass
 o .
 o o CUT -----
7 (-4,-1)
 -----
 -----
```

15. a) In `go/tromp.c` and `go/tromp.py`, how would you change the move ordering so that the pass move is considered last instead of first?

    b) With this change, is 2x2 go solved faster or slower? Why?

    c) In `go/tromp.c`, why does Tromp write code that depends on bit manipulation, instead of writing code that is easier to understand?

    d) What changes would you have to make to `go/tromp.py` so that it solves 3x3 go instead of 2x2 go?

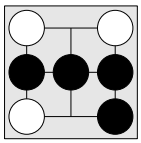
16. a) Give the minimax value of 1×2 go.

    b) Give the minimax value of 2×2 go.

    c) Give the minimax value of 2×2 go after the move 1.B[pass].

    d) For the game of go (positional superko, no self-capture, komi 0) played on any graph, prove that the minimax value is at least 0 (first player Black can always at least draw).

17. For 3×3 Go, a player has a *middle-3* position if they have 3 stones on the middle row and on at least one top-row point they don't have a stone, and on at least one bottom-row point they don't have a stone.



    Prove that once a player reaches a middle-3 position they can win by 9.

hints

- see `https://webdocs.cs.ualberta.ca/~hayward/355/ssgo.pdf`