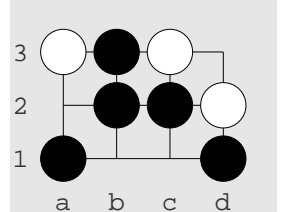


## cmput 355 2024 homework 1

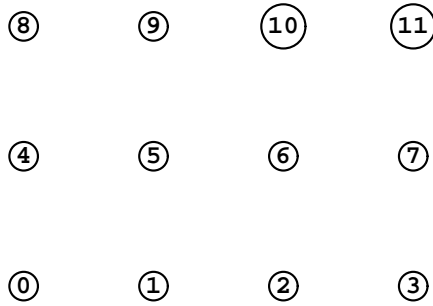
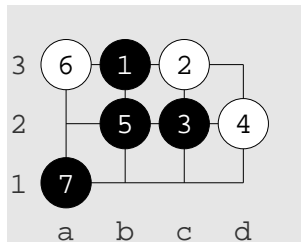
In this course, unless we say otherwise, in all go games, we follow the **logical rules with no self-capture**, and we assume that the komi is 0.

1. What is the main difference between the logical rules of go and the Chinese rules of go?

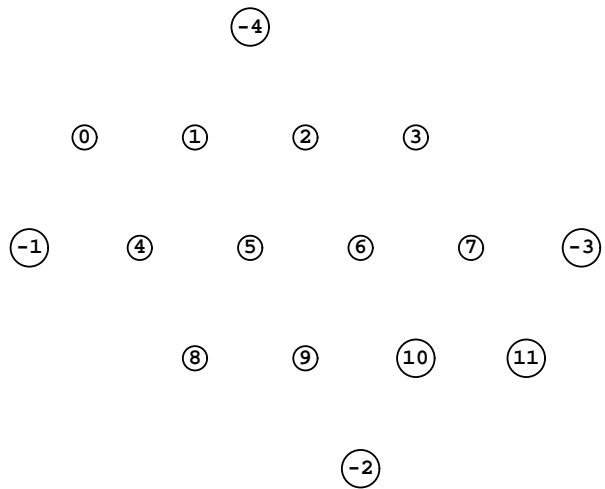
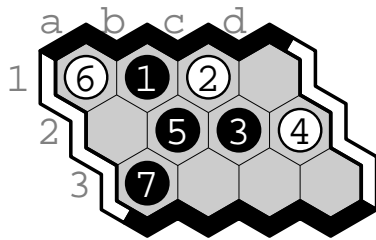
2. (a) For this go position, give all black blocks and all white blocks. Write each block as a set of points, e.g. {a1}. (b) For each block, give all liberties (again, as a set of points). (c) If this is the final position in the game, what is the final score? Answer like this: Black has 32, White has 17. (d) Repeat (c) if the komi is 3.5.



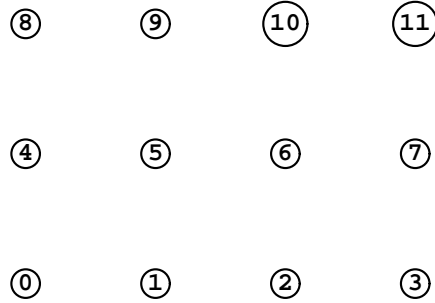
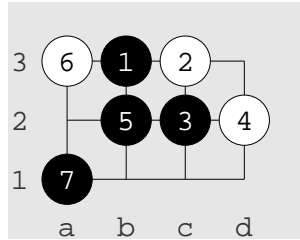
3. (a) For this go position, on the pointer diagram, draw the parent pointers. (b) Repeat the question if moves 3 and 5 are exchanged. (c) Why does the order of moves matter?



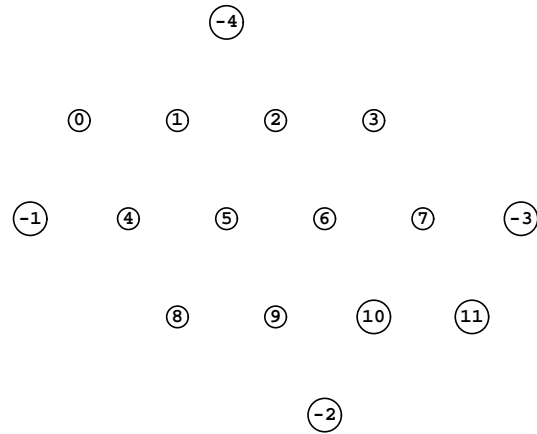
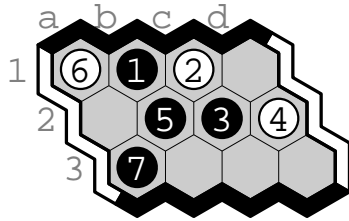
4. Repeat the previous question for this hex position. Don't forget to change rows  $1 \leftrightarrow 3$ .



5. (a) For github program `hexgo/hex.py`, after move 4, on the pointer diagram at each point that is a root of its block, give the set of liberties of that block. (b) Repeat (a) for 5.

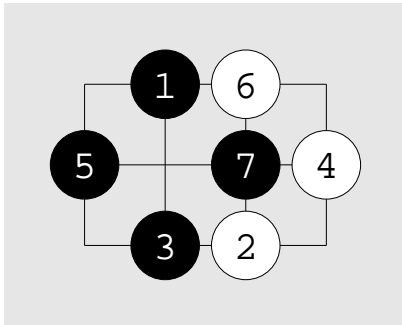


6. Repeat the previous question for this hex position.



7. (a) Explain why there are 3 possible 1x1 go positions. (b) Explain why only 1 of the positions in (a) is legal. (c) Explain why the only legal 1x1 go game is 1.B[pass] 2.W[pass]. (d) Explain why there are 81 possible 2x2 go positions. (e) Find the number of legal 2x2 go positions. Justify briefly.
8. In Hex, there is no pass move and the game ends as soon as a player has joined their two sides. Find the number of legal 2x2 hex games.

9. (a) In each of the following go games, give the first illegal move, and explain why it is illegal (occupied point, liberty violation, superko violation). Answer **all moves legal** if all moves are legal. (b) Repeat (a) if we allow self-capture.



- ... 8.W[b2] 9.B[c3] 10.W[d3]  
 ... 8.W[b2] 9.B[c3] 10.W[d3]  
 ... 8.W[b2] 9.B[a3] 10.W[a1]  
 ... 8.W[a1] 9.B[a3] 10.W[a1]  
 ... 8.W[b2] 9.B[d3] 10.W[a1]  
 ... 8.W[d1] 9.B[a1] 10.W[d3] 11.B[b2]

10. Explain the significance of each of these moves in the DeepMind challenge match between Lee Sedol and AlphaGo. (a) game 1: move 7 (b) game 3: move 37 (c) game 4: move 78

11. In file `stone_board.py` in directory `hexgo` in class github repo

`https://github.com/ryanbhayward/games-puzzles-algorithms` :

- (a) explain carefully the purpose of this line and how it is used:

```
self.nbr_offset = ((-1,0),(0,1),(1,0),(0,-1))
```

- (b) repeat (a) for this line

```
self.nbr_offset = ((-1,0),(-1,1),(0,1),(1,0),(1,-1),(0,-1))
```

- (c) explain carefully whether this line is necessary or can be omitted

```
if r+y in r_range and c+x in c_range:
```

12. From file `go_helper.py` in directory `go` in the class github repo:

(a) unscramble and indent the python code from the while loop inside the function that computes the score for each player:

```
b_nbr |= (self.brd[x] == BLACK)
empty_points.append(x)
empty_seen.add(x)
if self.brd[x] == EMPTY and x not in empty_seen:
for j in self.nbr_offsets:
w_nbr |= (self.brd[x] == WHITE)
while (len(empty_points) > 0):
q = empty_points.pop()
x = j + q
territory += 1
```

(b) for each line, give a brief comment explaining what is happening

13. In the class github repo, in directory `hexgo`, in file `stone_board.py`: (a) explain the purpose of each line of `merge_blocks()`

```
def merge_blocks(self, p, q):
    proot, qroot = UF.union(self.parents, p, q)
    self.blocks[proot].update(self.blocks[qroot])
    self.liberties[proot].update(self.liberties[qroot])
    self.liberties[proot] -= self.blocks[proot]
```

(b) explain the purpose of each line of `remove_liberties()` .

14. In the class github repo, in directory `hexgo`, execute `hex.py` . (a) List the hex moves that have been made in the game by `m33demo` (the first move is `1.B[b2]`). (b) Which of the four lines of output tells you that black has won the game? Explain carefully.