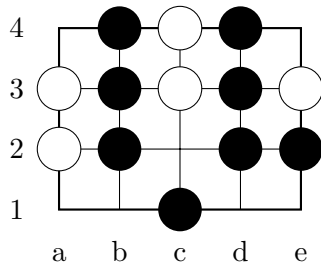---

1. Consider the following 4×5 Go position, with White stones at a2, a3, c3, c4, e3 and Black stones at b2, b3, b4, c1, d2, d3, d4, e2.


   (a) Compute the Tromp-Taylor Score for each player.

   (b) Can Black make a move that captures at least one White stone? Give coordinates or explain why not.

   (c) Can White make a move that captures at least one Black stone? Give coordinates or explain why not.

   (d) List all of Black's blocks.

   (e) List all of White's blocks.

2. Consider the following 4×4 Go position after 7 moves, with White stones at b1, b2, c1, d2 and Black stones at b3, b4, c2, c4, d3.
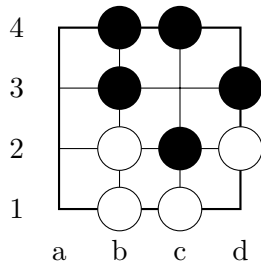

   (a) List the liberties for the block containing c4.

   (b) List the liberties for the block containing c2.

   (c) Suppose that White places a stone at c3. What stones, if any, does it capture?

   (d) Suppose that Black replies by playing a stone at c2. Is this a legal move? Why or why not?

   (e) Would it be legal for Black to reply by playing at d1 instead? Why or why not?

3. Consider the following 3×3 Hex board with White stones at a2, c2 and Black stones at a3, and b2.
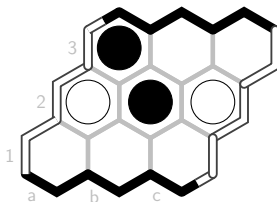
   (a) List every move Black could make to win the game.

   (b) Is there *any* sequence of play from this position in which White wins? If so, list the moves in the form "B[a3], W[c2],...". If not, explain why not.

4. Consider the following position on a 3×4 go board, created by the moves 1.B[a1], 2.W[b2], 3.B[c1], 4.W[d3], 5.B[b1], 6.W[c2]. Assume that neighbours are iterated over in the order above, right, below, left, and that union is implemented without the union-rank optimization.
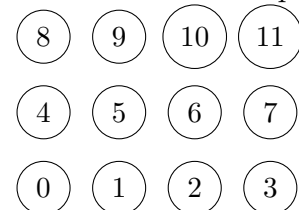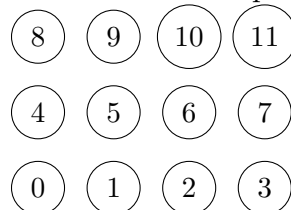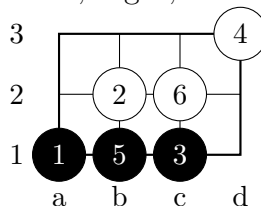
   (a) Draw the parent pointers for this position on the parent diagram.

   (b) Repeat the question, but assuming that the union-rank optimization is implemented

   (c) Why does the iteration order matter in (a)?

   (d) Does the iteration order matter in (b)? Why or why not?

5. Consider the following position on a 3×4 Hex board, created by the moves 1.B[b1], 2.W[c1], 3.B[c2], 4.W[d2], 5.B[b2], 6.W[a1], 7.B[a3]. Assume that neighbours are iterated over in the order above left, above right, right, below right, below left, left, and that union is implemented without the union-rank optimization.
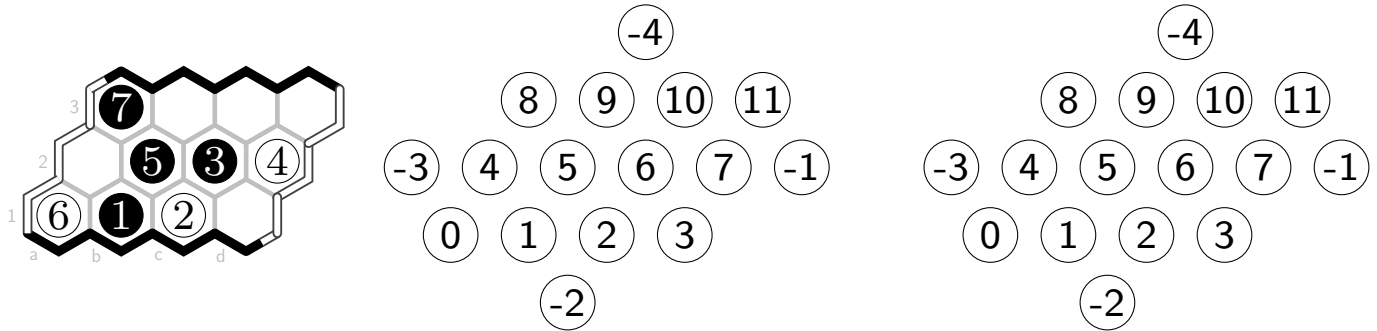
  (a) Draw the parent pointers for this position on the parent diagram.

  (b) Repeat the question, but assuming that the union-rank optimization is implemented.

6. The following line is from the hex board initialization in hexgo/stone_board.py:

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

  (a) Explain the purpose of this line.

  (b) Give the corresponding line for initializing a go board.

7. Explain the purpose of line 4 below:

    ```
    1   for r in range(self.r):
    2     for c in range(self.c):
    3       for (y,x) in self.nbr_offset:
    4         if r+y in r_range and c+x in c_range:
    5           self.nbrs[Pt.rc_point(r,c,self.c)].add(Pt.rc_point(r+y,c+x,self.c))
    ```

8. Explain how the rc_point function below works. You may be asked to reproduce it.

    ```
    def rc_point(row, col, num_cols):
      return col + row * num_cols
    ```

9. In hexgo/stone_board.py, explain the purpose of each line of merge_blocks:

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

10. Here is a portion of the `tromp_taylor_score` method in `go/go_helper.py`, with two expressions missing.

```
1     while (len(empty_points) > 0):
2       q = empty_points.pop()
3       for j in self.nbr_offsets:
4         x = j + q
5         b_nbr |= (self.brd[x] == BLACK)
6         w_nbr |= (self.brd[x] == WHITE)
7         if self.brd[x] == EMPTY and x not in empty_seen:
8           empty_seen.add(x)
9           empty_points.append(x)
10          territory += 1
11    if    _____expression_(A)_____:
12      bt += territory
13    elif _____expression_(B)_____:
14      wt += territory
```

(a) What should expression (A) and expression (B) be?
    The order of the clauses in each expression of course doesn't matter.

(b) Explain the purpose of lines 5 and 6.