crafting a hex/go environment © **2024**

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

programs we will see in this course

* **environments**

* **players (agents)**

* **solvers**

crafting a hex/go environment ?

language?

* **python3 (beginner) `C++` (experienced)**

program design choice ?

* **algorithms = algorithms + data structures**

hexgo algs/datastructs ? board ?

- **lists (implement: easy, perf slow)**
- **strings (imp'n: medium, perf: fast)**
- **sets (imp'n: easy/medium, perf: fast)**

hexgo algs/datastructs ? blocks ?

- **lists: search (imp'n: easy, perf: slow)**
- **strings: search (imp'n: easy/med, perf: fast)**
- **sets: union-find (imp'n: easy, perf: fast)**

`gpa/go/go_helper.py`

board: strings (guarded board)

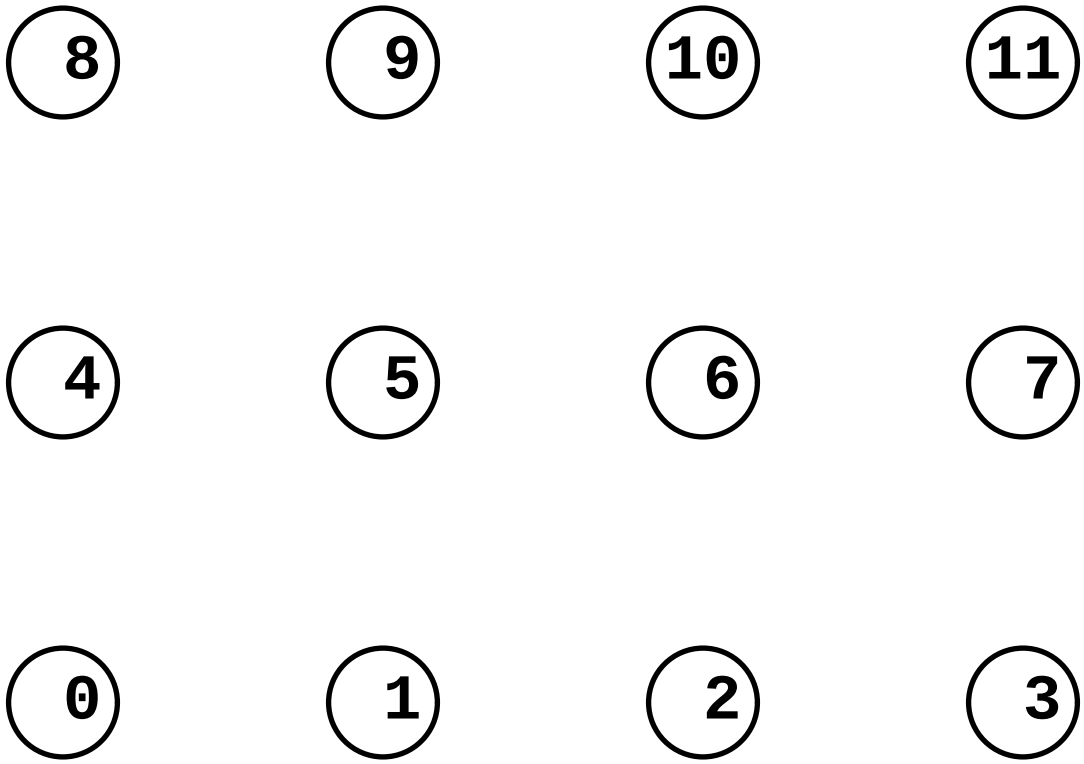
`gpa/hexgo/go.py`

board: sets (union-find operations)

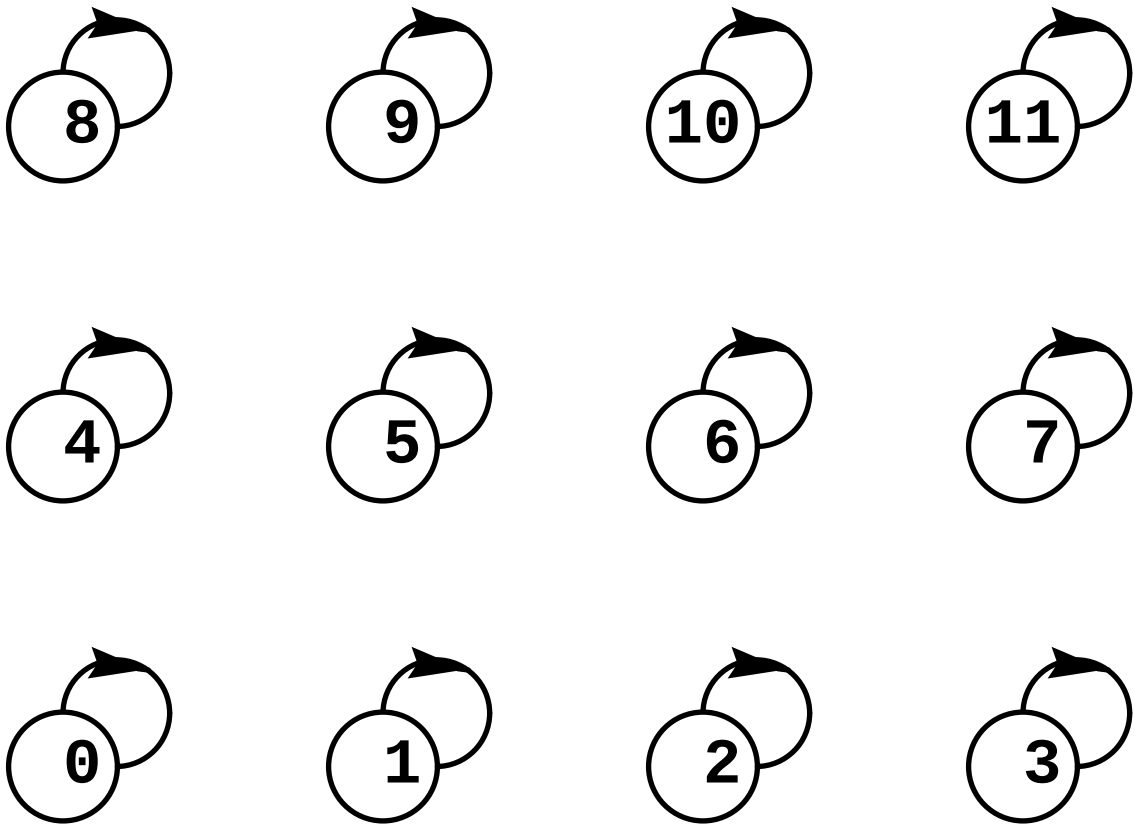
union-find data struct (used in min'm spanning tree algs)

```
#hexgo/stone_board.py   Class Stone_brd   def __init__()
for point in self.p_range:
    self.parents[point] = point

def union(parents, x, y):
    x = UF.find(parents, x)
    y = UF.find(parents, y)
    parents[y] = x # x is root of merged trees
    return x, y

  def find(parents, x):
    while x != parents[x]: x = parents[x]
    return x
```

3x4 go: parent pointers

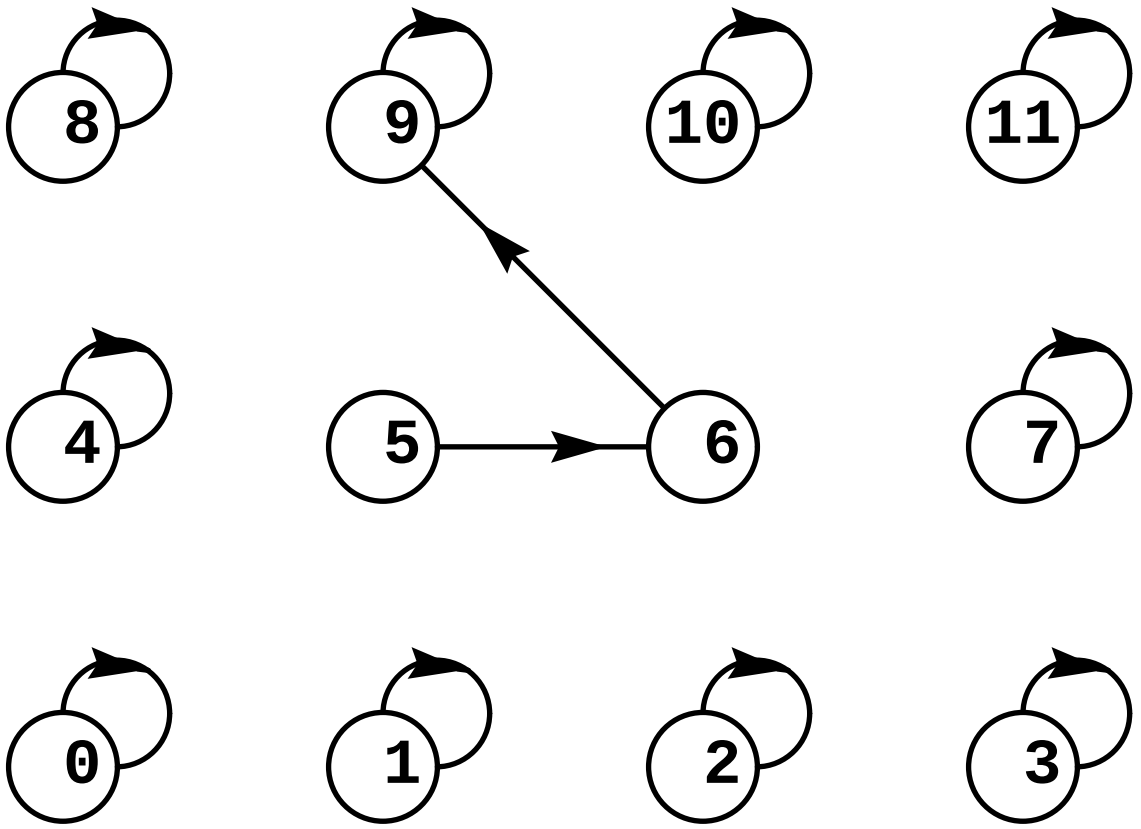


3x4 go: after `__init__()`



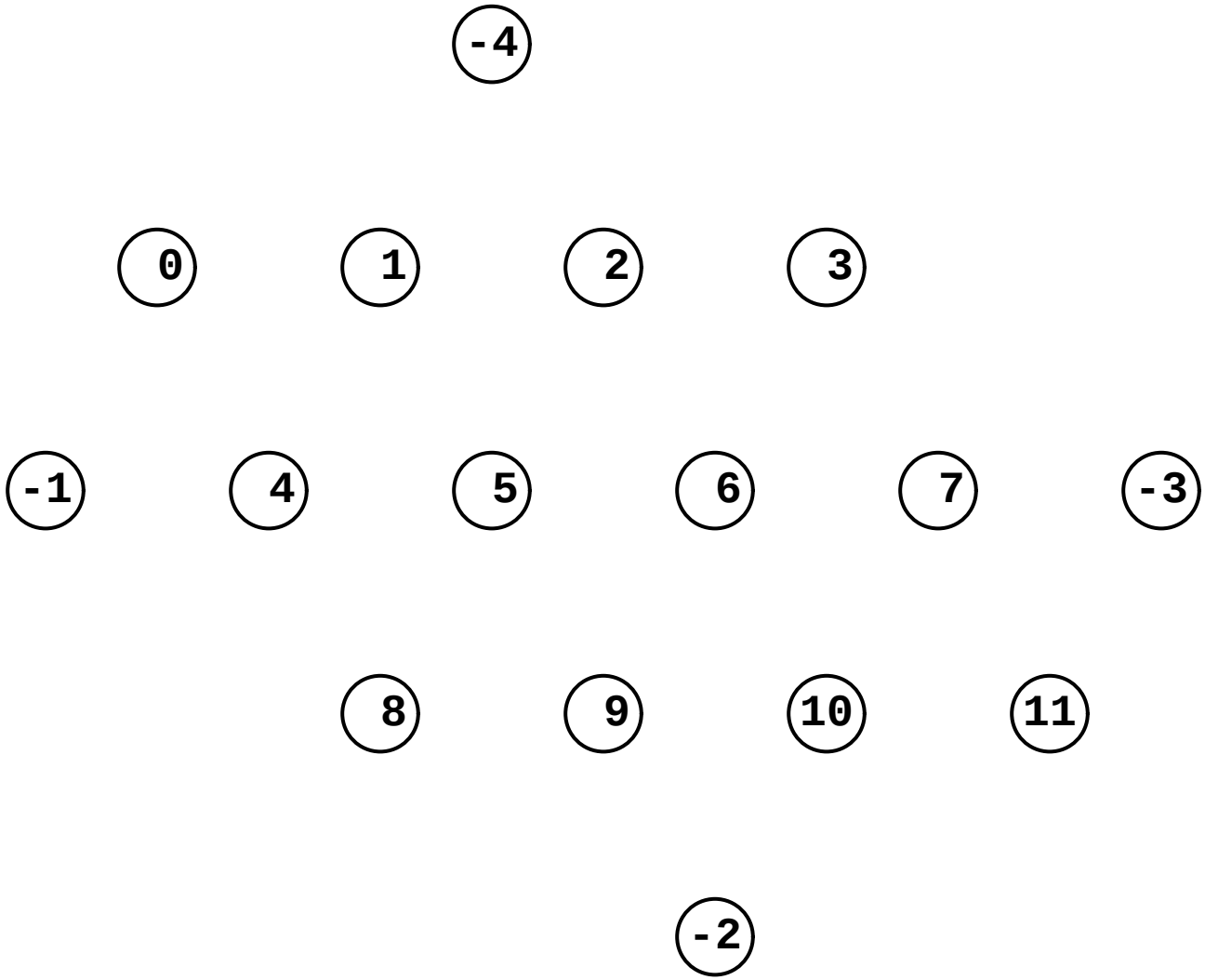
```
add_stone * 9   # 1.B[b3]       ( * 9 means Black 9 )

add_stone * 6   # 3.B[c2]

add_stone * 5 (union 6 5 union 9 5)

  root(5) now points to root(6): 5 -> 6

  root(5) now points to root(9): 6 -> 9

* blocks 9 {9, 5, 6}   lib's 9 {1, 2, 4, 8}

@ blocks 7 {7} 10 {10} lib's 7 {11, 3} 10 {11}

parents 5 6  6 9
```

after `add_stone * 9, add_stone * 6, add_stone * 5`

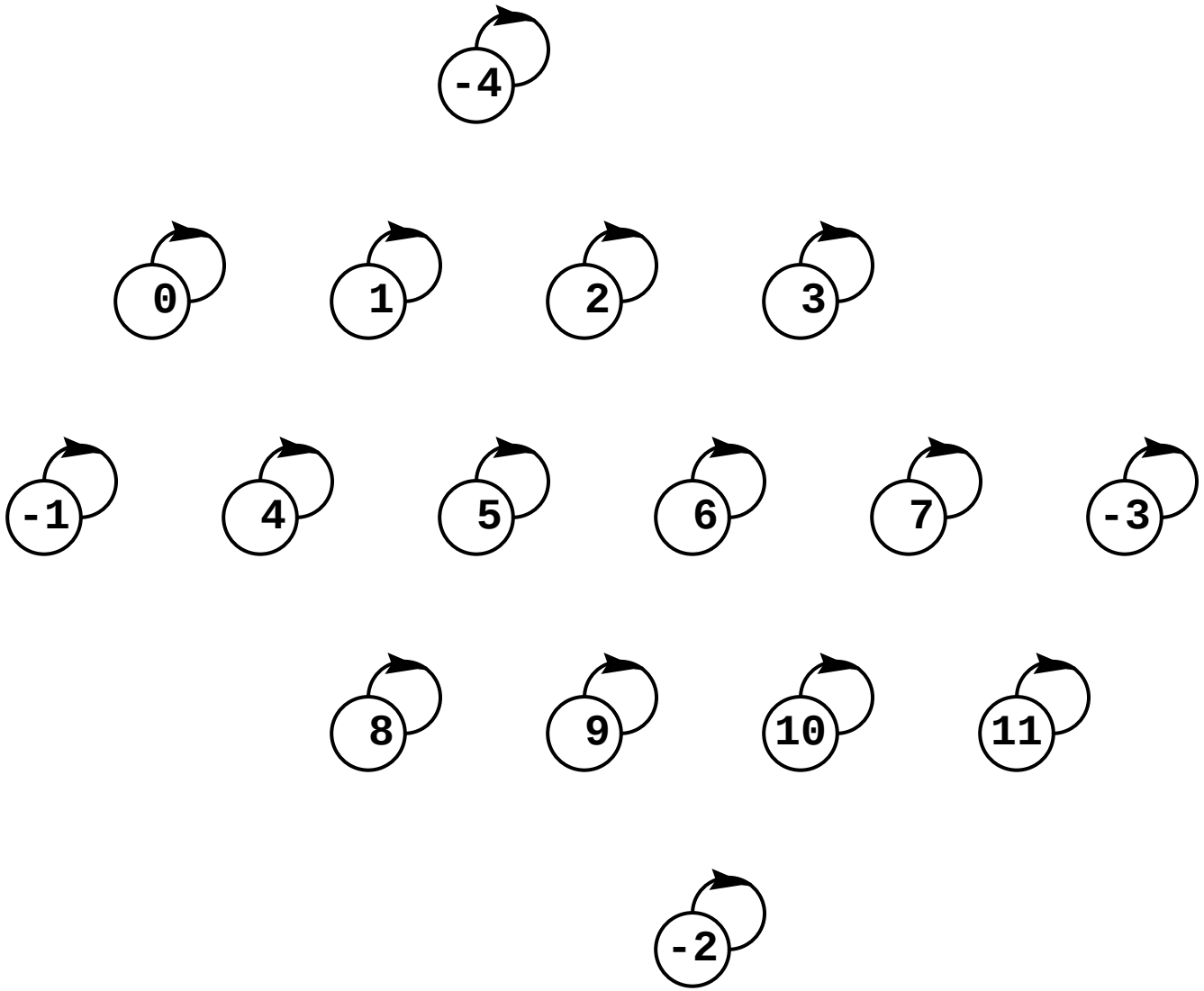


```
from /hexgo/stone_board.py

def __init__(self, gt, rows, cols):
    self.game_type = gt
    self.r, self.c, self.n = rows, cols, rows * cols
    if gt: # hex game
      self.top, self.rgt, self.btm, self.lft = -4, -3, -2, -1
      self.border = range(self.top, 0) # -4, -3, -2, -1
      self.p_range = range(self.top, self.n) # -4, ..., rows*cols-1
      self.nbr_offset = ((-1,0),(-1,1),(0,1),(1,0),(1,-1),(0,-1))
      #   0 1
      #  5 . 2
      #   4 3
    else: # go game
      self.p_range = range(self.n)
      self.nbr_offset = ((-1,0),(0,1),(1,0),(0,-1))
      #    0
      #  3 . 1
      #    2
    self.stones = [set(), set()]  # start with empty board
```

3x4 hex: parent pointers



3x4 hex: after `__init__()`



```
add_stone * -4
add_stone * -2
add_stone @ -1
add_stone @ -3
add_stone * 1 union -4 1 parents 1 -4

add_stone @ 2
add_stone * 6
add_stone @ 7 union -3 7 parents 7 -3

add_stone * 5 union 1 5 union 6 5
  parents -4 6  1 -4  5 -4  7 -3

add_stone @ 0 union -1 0 parents 0 -1

add_stone * 8 union 5 8 union -2 8
  parents -4 6  0 -1  1 -4  5 -4  6 -2  7 -3  8 6
```

after `add_stone * -4 ... add_stone * 8`