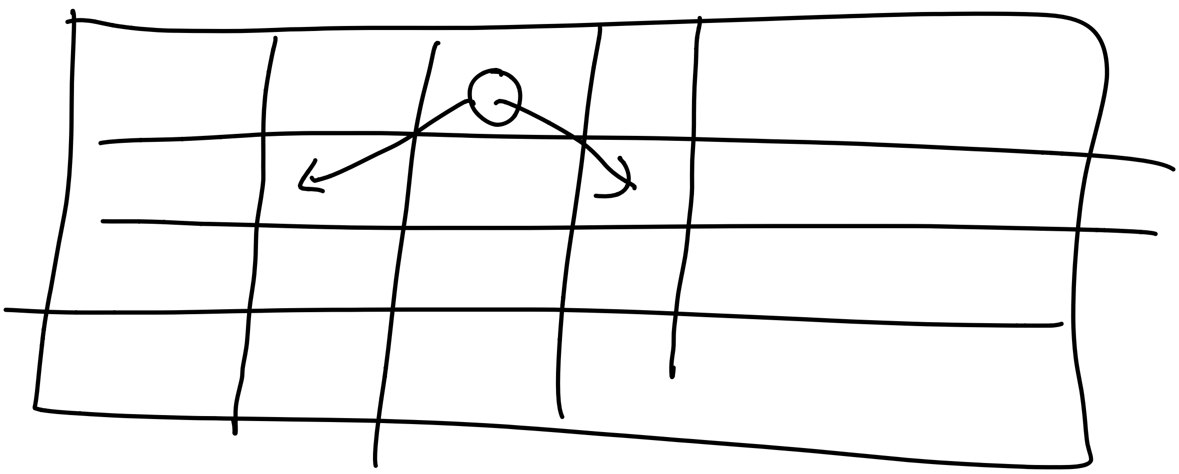
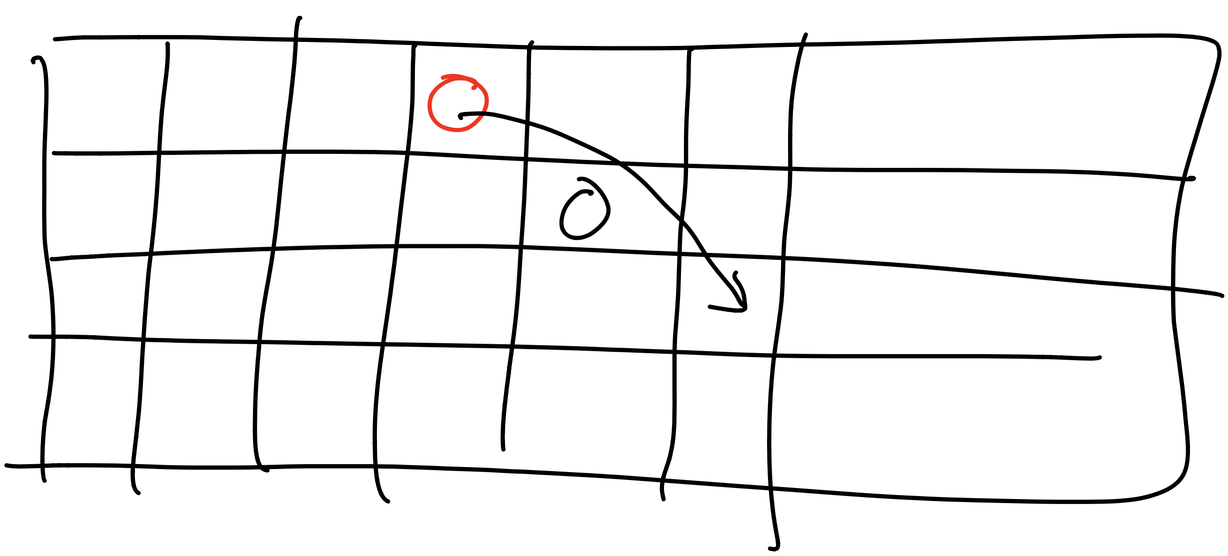
Office Hours

HW 9

1.

valid-move?





Two kinds of valid move

Input to the valid-move
- Checkers $\quad$ loc

↑
game state

game-state-clicked-piece

Want if clicked-piece
can be move to loc

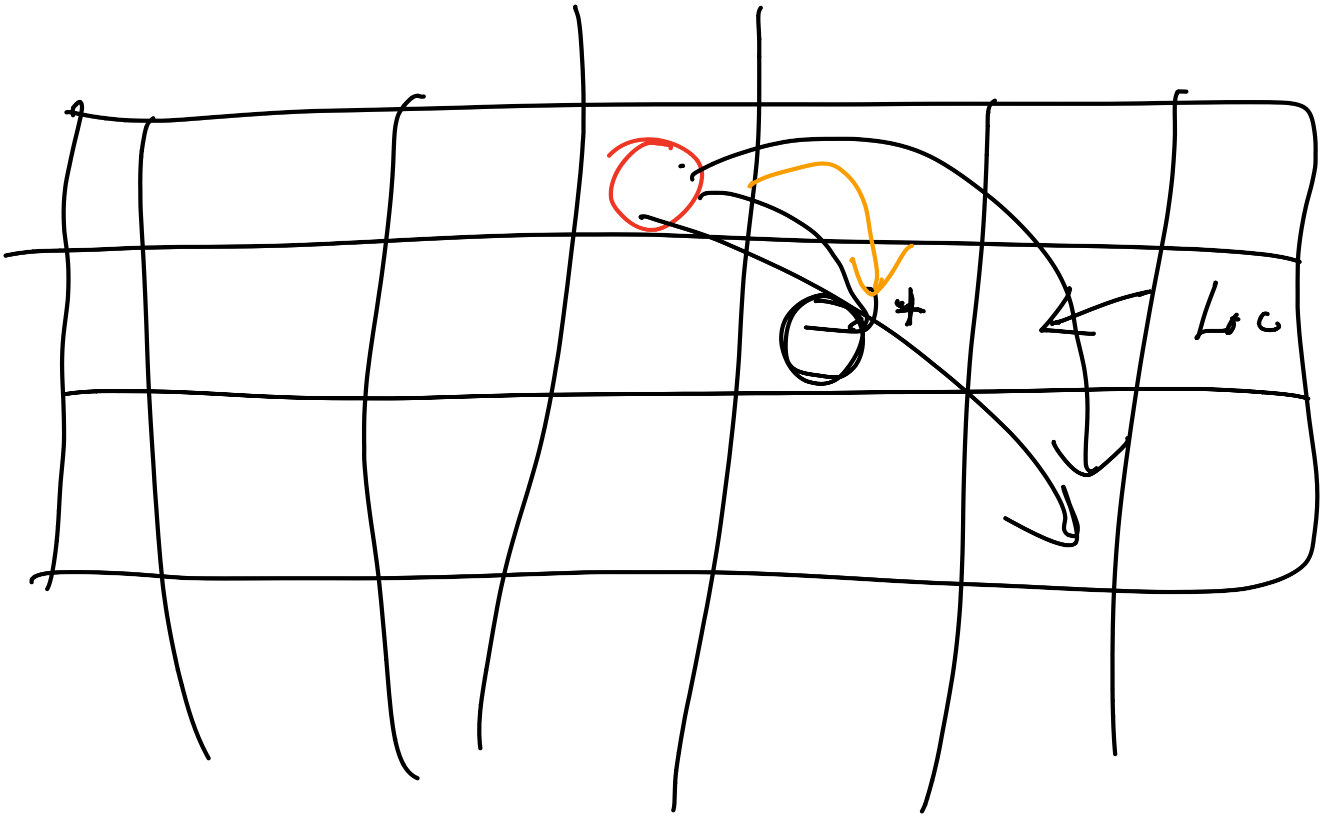
Check if loc is a valid for
clicked-piece.

Step 1:
Extract clicked-piece and all posible locations for its valid moves local

Step 2:
If the second loc is equal to one of the valid location.

match statement to check this

Clicked-piece can have four possible valid moves.



Your second input location is on the right side of the clicked

Clicked-piece can have two valid locations.

First compute this two valid locations

```
(: valid-move? : Checkers Loc -> Boolean)
(define (valid-move? game loc)
 (match* (game loc)

[((Checkers pieces turn (Piece color
      (Loc cur-r cur-c))) (Loc r c))
  ( and
   (symbol=? turn color)
   ( local
    {
     (: sign : Integer)
```

```
(define sign (if (symbol=? color
                 'red) 1 -1))


(: move-sign : Integer)
(define move-sign (if (>= c curr-c)
                  1 -1))
```

check if there is a piece at first valid location

```
(: one-way : (U 'none piece))
(define one-way (get-piece-or-none

  (Loc (+ curr-r (* sign 1))
       (+ curr-c (* move-sign 1)))
  game)
```

check if there is a piece at second valid location

```
(: two-away : (U 'none Piece))
```

```
(define two-away (get-piece-or-none

        (Loc (+ curr-r (* sign 2) )
             (+ curr-c (* move-sign 2))))
```

← First valid location

```
(: one-valid : Loc)
 (define one-valid (Loc
                    (+ curr-r (* sign 1) )
                    (+ curr-c (* move-sign 1)))
```

← second valid location

```
(: two-valid : Loc)
 (define two-valid (Loc
                    (+ curr-r (* sign 2))
                 (+ curr-c (* move-sign 2)))
 }
```

(or (and (loc=? (Loc r c) one-valid)
(symbol? one-way))

if there is a piece at this location or not.

(check the second valid move.)

Check 3 things -
check the location
check there is a piece at two-valid
Check there is a piece of flipped color at one-way
flipped color means
if chicked piece is red
the color of the other should black and vice-versa.

Write a flip function

which takes symbols
an input.
'red → 'black
'black — 'red.

3. game-over?

1. if there are no blackes piecey
or red pieces

2. clicked-piece field is none

HW 8

1

a.

```
(: count-old : (Listof Grocery) Integer
               -> Integer)

(define (count-old items date)
  (cond                      1 op        1 op
    [(empty? items) 0]
    [else (+                        1 op     T(n-1)
       (count-old (rest items) date)
             1 op           1 op
       (if (> (Grocery-daysold (first      1 op
              items) date) 1 0))])) 1 op
```

if size of items = n.

size of (rest items) = n-1

Suppose the time complexity of Step 0:

$T(n)$

$n \rightarrow$ size of the input list.

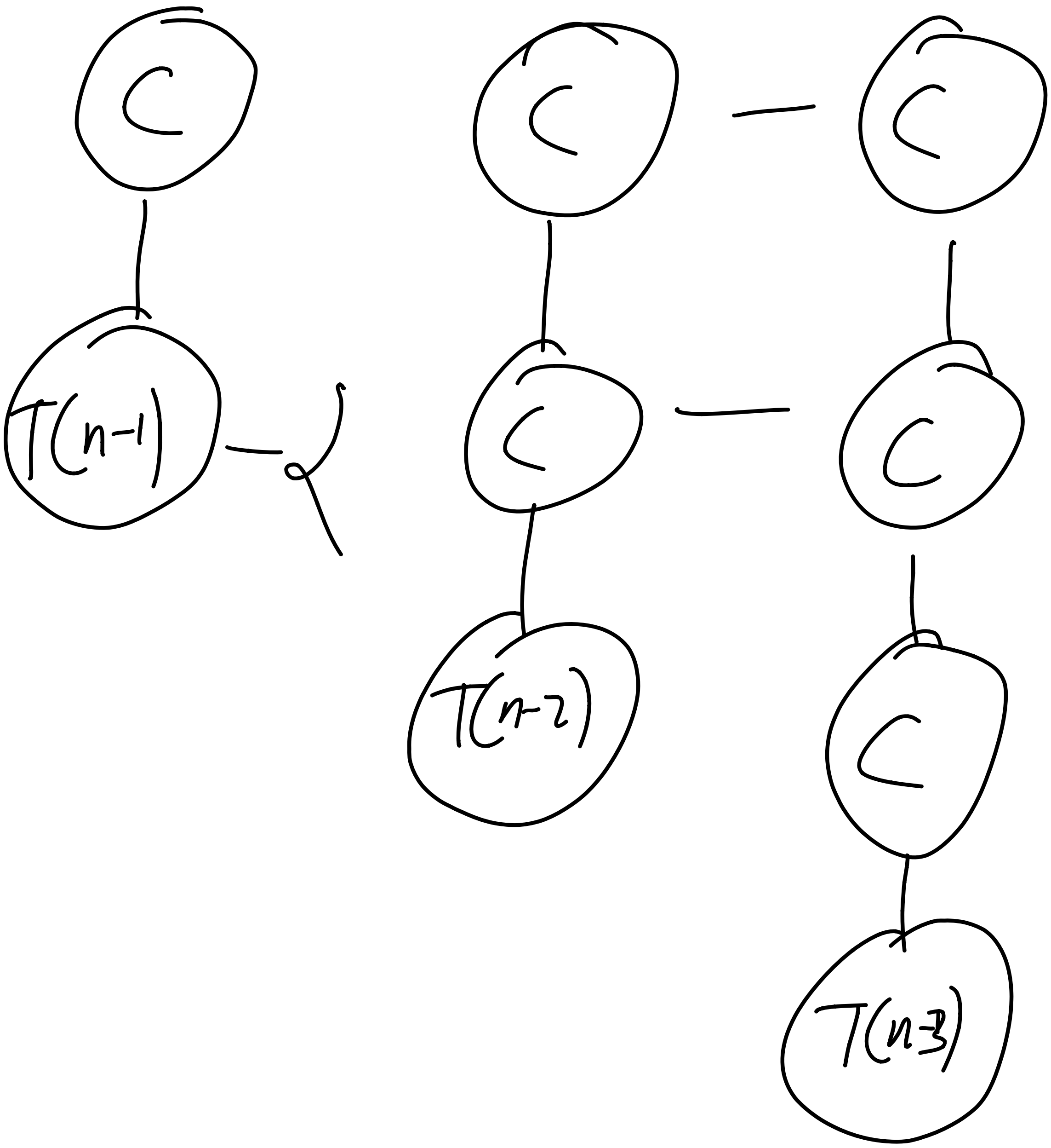
Step 1:

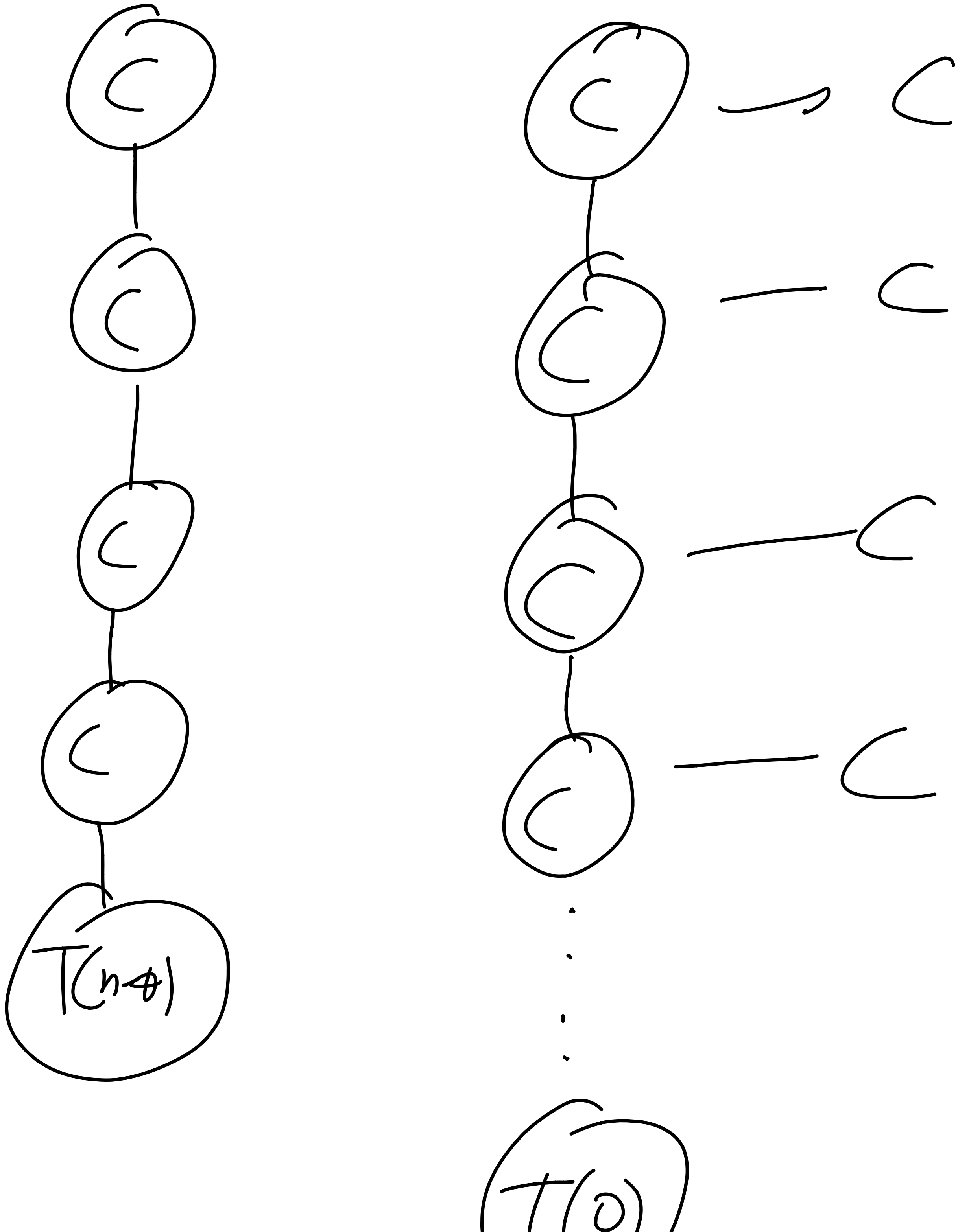
$$T(n) = T(n-1) + 3 + 4$$

$$T(n) = T(n-1) + 7$$

$$T(n) = T(n-1) + \underbrace{C}_{\text{constant}}$$

Step 2:
Solve this





$n, n-1, n-2, n-3, n-4 \ldots\ 0.$

Height of this tree: $n+1$

$$\underbrace{C + C + C + \quad C}_{(n+1)\text{ times}}$$

$$= \underline{(n+1) \cdot C}$$

$$T(n) = (n+1) \cdot C$$

$$= Cn + C$$

$$= \underline{O(n)}$$

$$T(n) = O(\underline{n}).$$

$$2n^0 + 1 = O(n)$$

$$2n^2 + n + 1000 = O(n^2)$$

$$100\, n^{100} + n^{80}_{\;0} = O(n^{100})$$

$$n^{100} >> n^{50}$$

$$f(n) \in O(g(n))$$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \text{ or constant}$$

$f(n) =$ $g(n) = n.$

$$2n+1 \in O(n).$$

$$\lim_{n\to\infty} \frac{2n+1}{n} = \frac{2+1/n}{1}$$

$$\frac{\lim_{n\to\infty} 2 + \lim_{n\to\infty} 1/n}{\lim_{n\to\infty} 1} = 2$$

$$n^3+n^2+1 \in O(n^3)$$

$$\lim_{n\to\infty} \frac{n^3+n^2+1}{n^3} =$$

$$\lim_{n\to\infty} \left(1 + \frac{1}{n} + \frac{1}{n^3}\right)$$

$$= \lim_{n \to \infty} 1 + \lim_{n \to \infty} \left(\frac{1}{n}\right) + \lim_{n \to \infty} \left(\frac{1}{n^3}\right)$$

$$= 1.$$

$$\lim_{n \to \infty} \frac{n^3 + n^2 + 1}{n^3} = 1$$

$$n^3 + n^2 + 1 \in O(n^3)$$

$$n^3 \notin O(n^2)$$

$$\lim_{n \to \infty} \frac{n^3}{n^2} = \lim_{n \to \infty} n = \infty$$

∞ is not 0 or constant

$n^3 \notin O(n^2)$

$\log n \in O(n)$ ✓

$$\lim_{n \to \infty} \frac{\log n}{n} = 0.$$

$cn \in O(n^2)$ True

$cn \in O(n) \Rightarrow$ Better

$\boxed{T(n) \in O(n^2)}$ ✓

$\boxed{T(n) \in O(n)}$

$n^2 \in \Theta(n^2)$

$2n^2 + 1 \in \underline{\Theta(n^2)}$

$n \notin \Theta(n^2)$

$n \in \underline{\underline{\Theta(n)}}$

$n \in O(n^2)$

$\underline{\underline{\leq}}$ $\underline{n \leq n^2}$

$\underline{\underline{2n^2+1}} \overset{\Theta}{\equiv} \underline{n^2}$