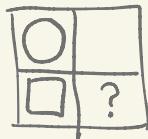


moone (5)

Goal :

$$1+2+3+\dots+n$$

moony-1:
(?)



$$= \frac{n}{2} (n+1)$$

$$= \cancel{\frac{n^2}{2}}$$

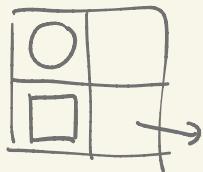
$$= O(n^2)$$

Wishful thinking

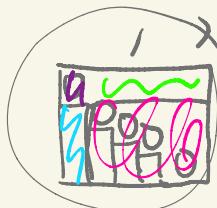
↳ base case

↳ inductive step:
 $f(n-1) \rightarrow f(n)$

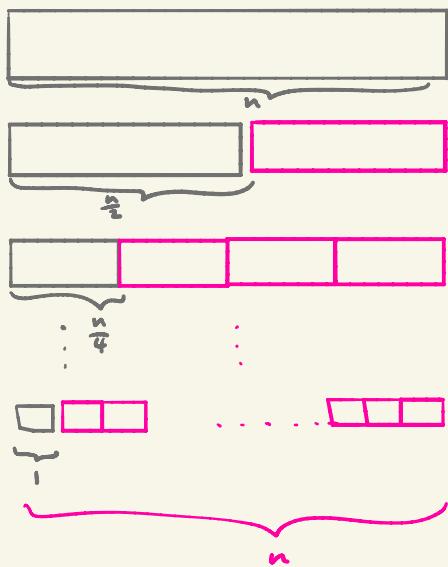
moony-2:
(n)



moony-3:



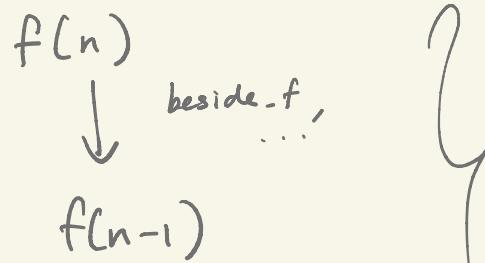
sf($\frac{1}{n}$, b.f($\frac{1}{n}$, 0, □),
b.f($\frac{1}{n}$, □, moony-2(n-1)))



```

fn_ cone(n, rune) {
    fn_ cone_helper(n, rune, max) {
        ↳ n == 1
        ? rune
            : overlay_f(  $\frac{1}{n}$ ,  $1 - \frac{n-1}{max}$ 
                scale(n, rune), max
                cone_helper(n-1, rune));
        }
    ↳ cone_helper(n, rune, n);
}

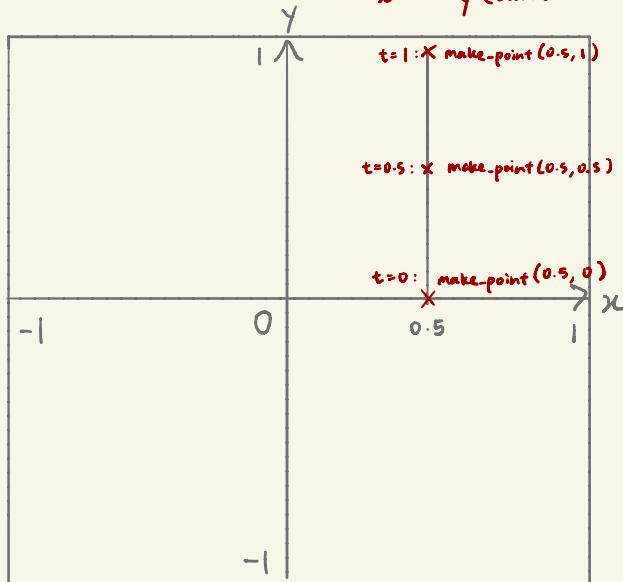
```



- ① Base case ($n=1$)
- ② Inductive step

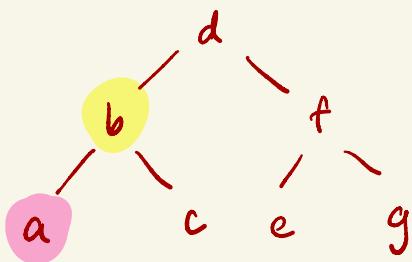
$\text{cone}(n)$ {
 ↳ $n == 1$
 ? O
 : $\text{cone}(n-1)$

$t \Rightarrow \text{make-point}(0.5, t)$



$p(b, p(b, p(b, p(b, p(b, 0))))))$

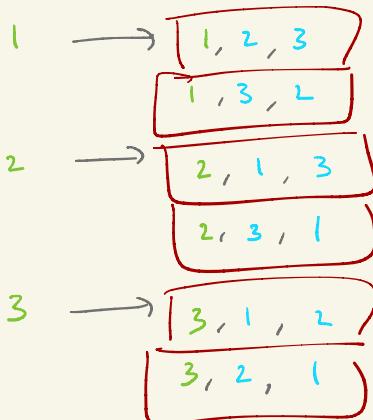
$$T(n) = n T$$



permutations (list)

list (1, 2, 3):

- ① Wishful thinking. For each element,
assume we have a solution
for the list minus that element
remove (x , list)

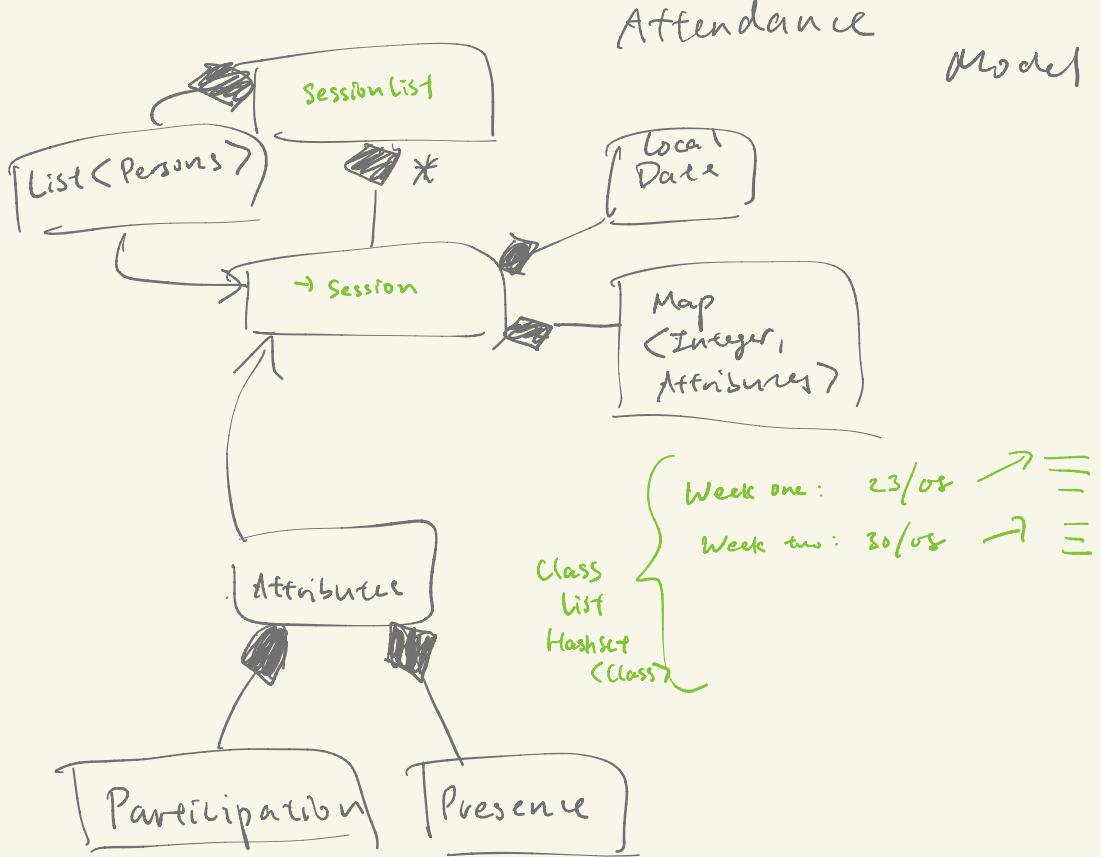


- ② Wishful thinking:
permutations (remove(x , list))

- ③ Map list in wishful thinking solution
map ($p \Rightarrow$ pair (x , p),
permutations (remove (x , list)))

- ④ Map element in list
map ($x \Rightarrow$ map ($p \Rightarrow$ pair (x , p),
permutations (remove (x , list))),
list)

- ⑤ Final step: combine solutions using accumulate (append, null, ...)

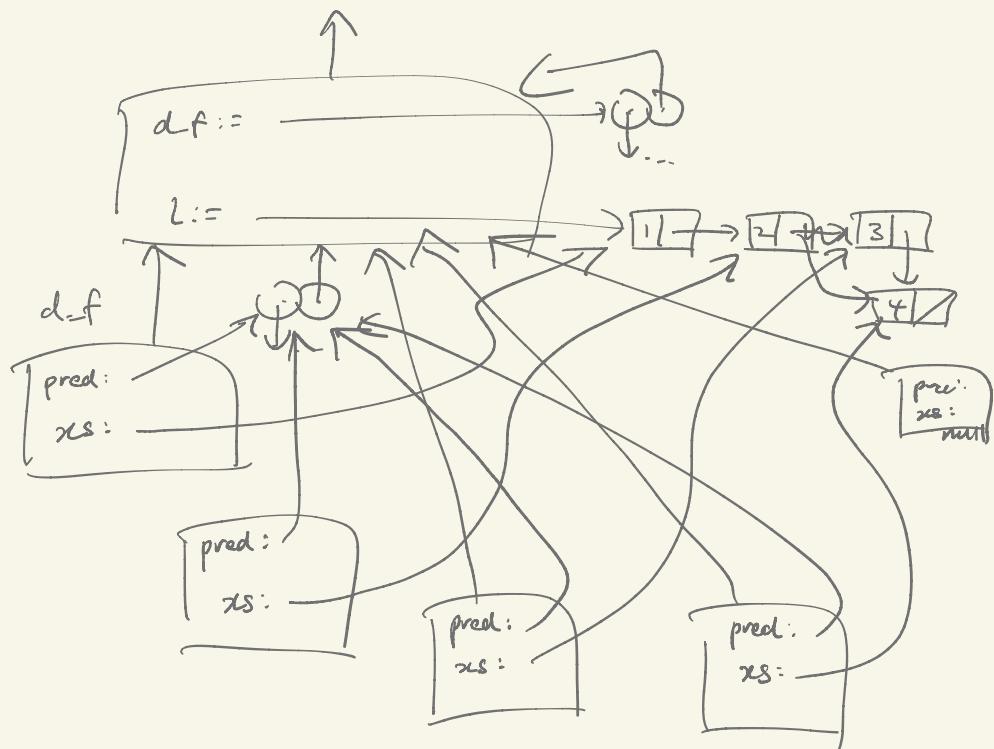
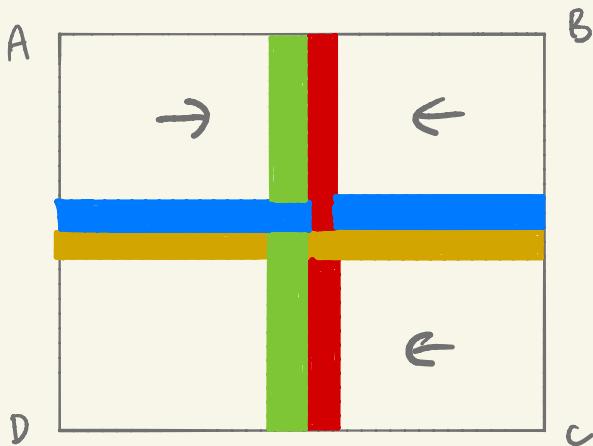


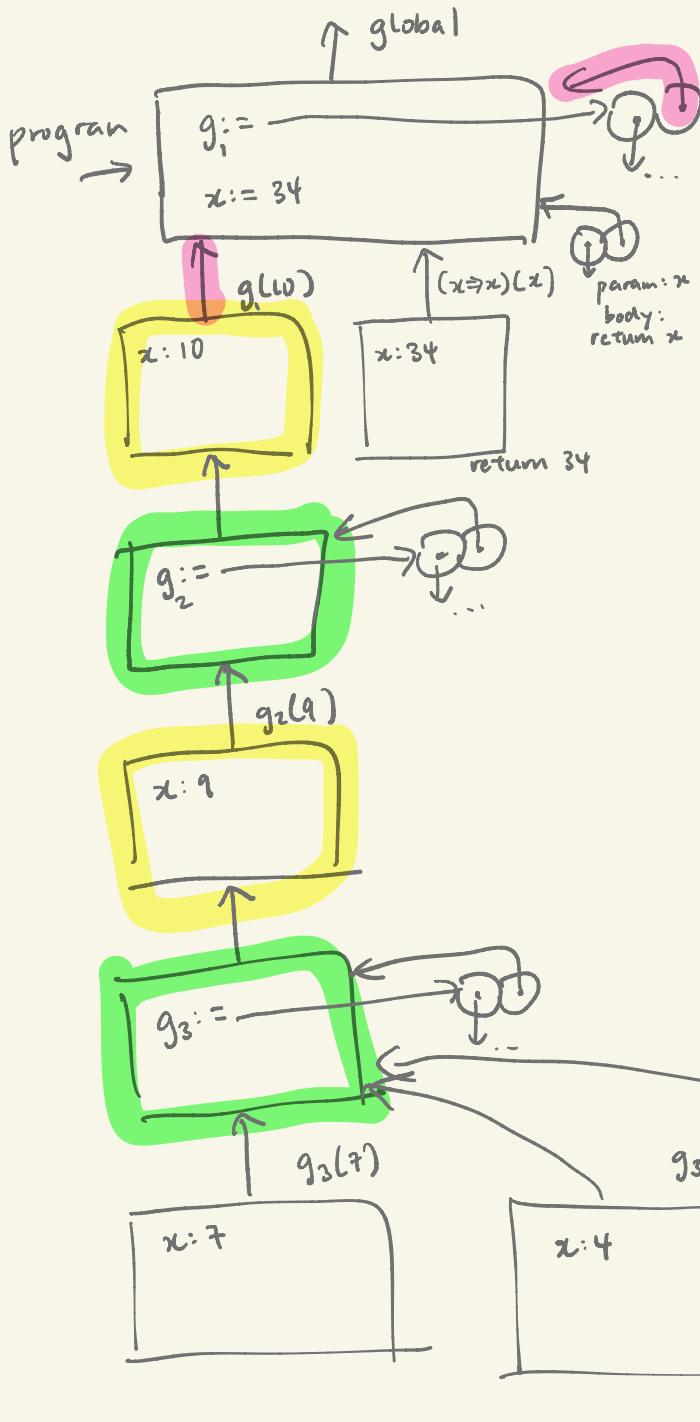
chain(twice, 3)
 $\rightarrow \text{twice(twice)(twice)}$
 $\rightarrow \text{four-times(twice)}$
 $\rightarrow \text{four-times(fff)}$
 $\rightarrow \text{three-times(ffff)}$
 $\rightarrow \text{two-times(ffff...f)}$
 $\rightarrow \text{ff...f}_{16} = 2^{2^2} = 32$

$t(b, n-1) : \Theta(n^{-1}b)$
 $t(b, n-2) : \Theta(n^{-2}b) < \Theta(n^{-1}b)$
 $t(b, n-3) : \Theta(n^{-3}b) < \Theta(n^{-1}b)$
 \vdots
 $t(b, 1) : \Theta(1) < \Theta(n^{-1}b)$

n

- 1111 1111
11 1111 $2^7 - 1$



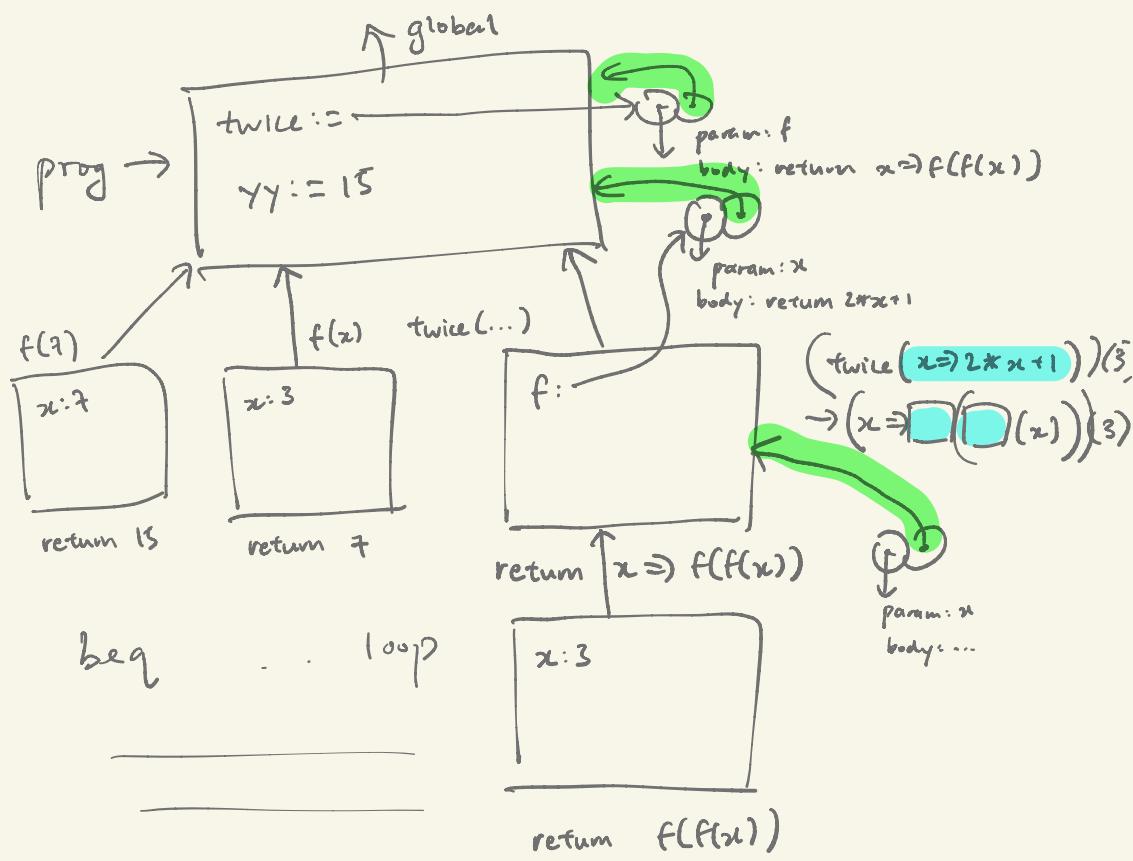


Func. app.

- ↳ extend frame
- ↳ extend from frame in which fn obj. is created
- ↳ declare names
 - parameters
 - body (names)

Func. obj.





beg . . . loop?

loop:

