Plan:

- end of term timing
- tail call optimization
- memory allocation?

Comps gala tomorrow!
3:30 pm

wk 9 | wk 10 | beyond

M

if self-scheduled must answer yes in Moodle survey

T

W today

last class!
part 9

Th

10:15am index card due

firm deadline

if self scheduling

F primitives

reading days

Sa

3:30p-6p
scheduled exam slots
CoK #4

If you do Cok #4 at scheduled slot
- bring index card with you

If you do it self-scheduled
- Can take it Sa, Su, or Mon
- you have to turn in your index card to my mailbox by Th wk 10 at 10:15 am

---

College rule:
- if you think you might take it self-scheduled, must tell me in advance

- even if you said you might take it self-scheduled, you can still show up at the scheduled exam time if you wish
- but please respect the trees

Tail call optimization

Memory downsides with recursion
- every recursive call creates a new frame

"Tail call" - a function call where it happens as the very last step of another function

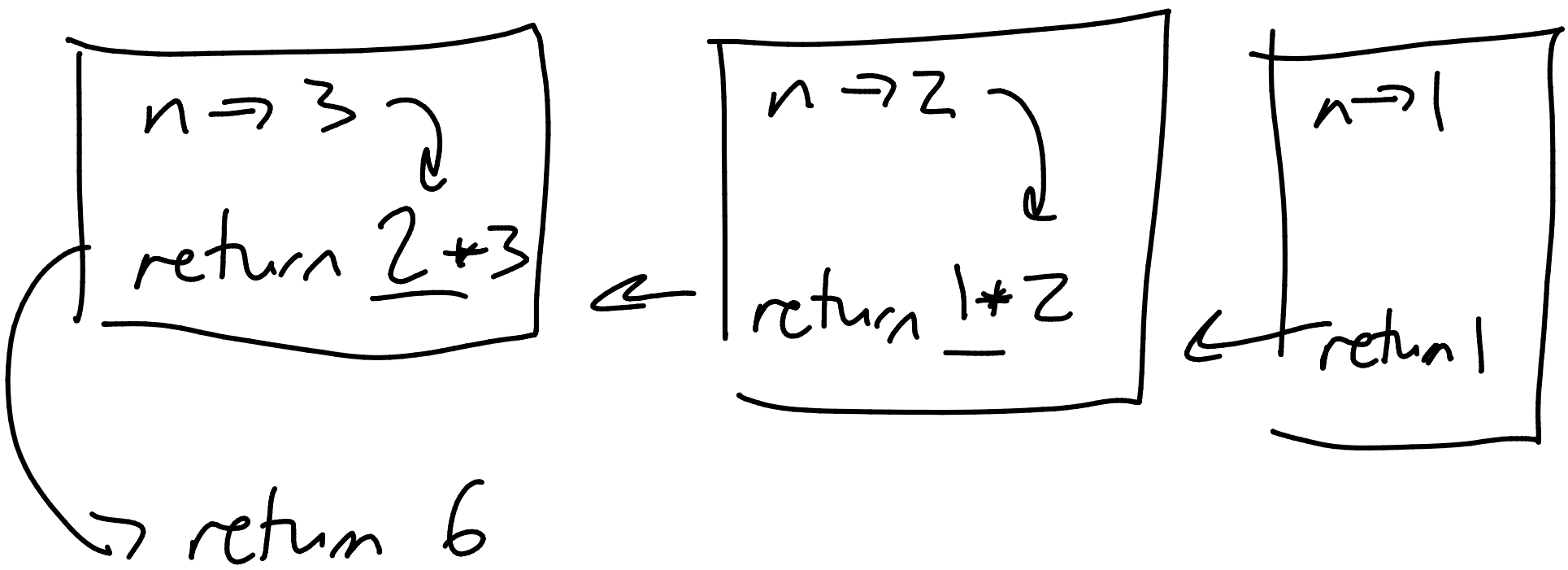
```
(define inc
  (lambda (x)
    (+ x 1)))

(define f
  (lambda (y)
    (inc y)))
```


tail call
we don't need f's frame anymore - we could delek it

In the case of fact (as we wrote it)

(fact 3)



all frames needed until end

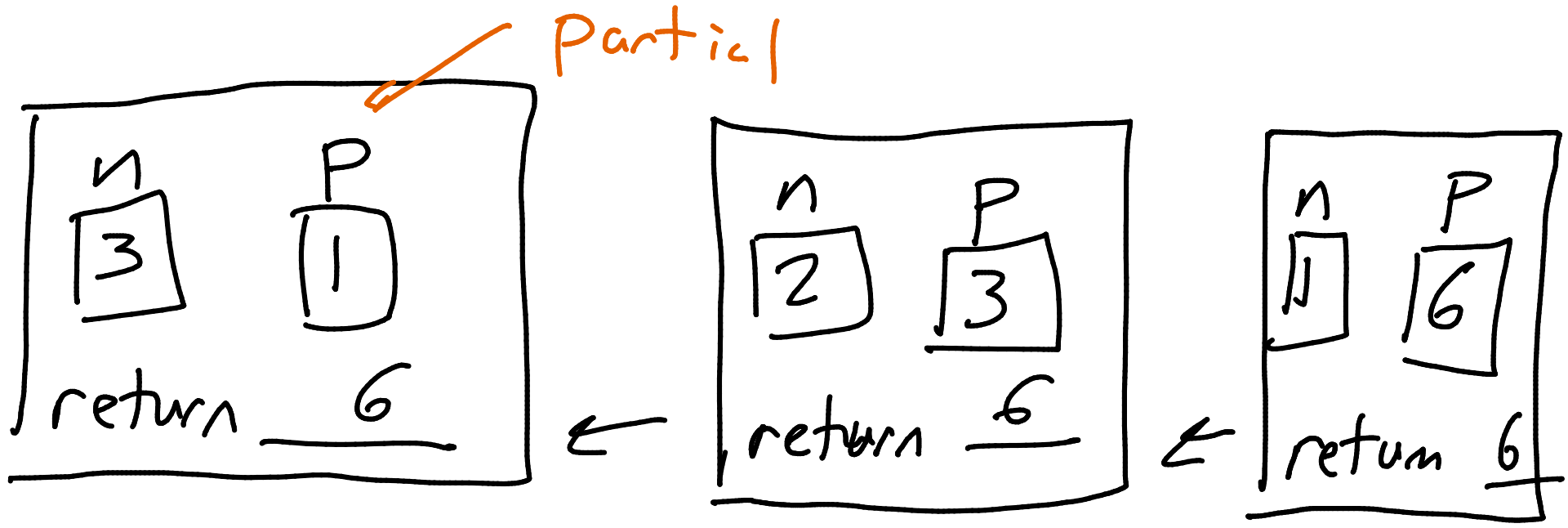
If you write your recursive functions so that the recursive call is a tail call, a smart compiler/interpreter can get away w/o keeping all the frames

Two questions:

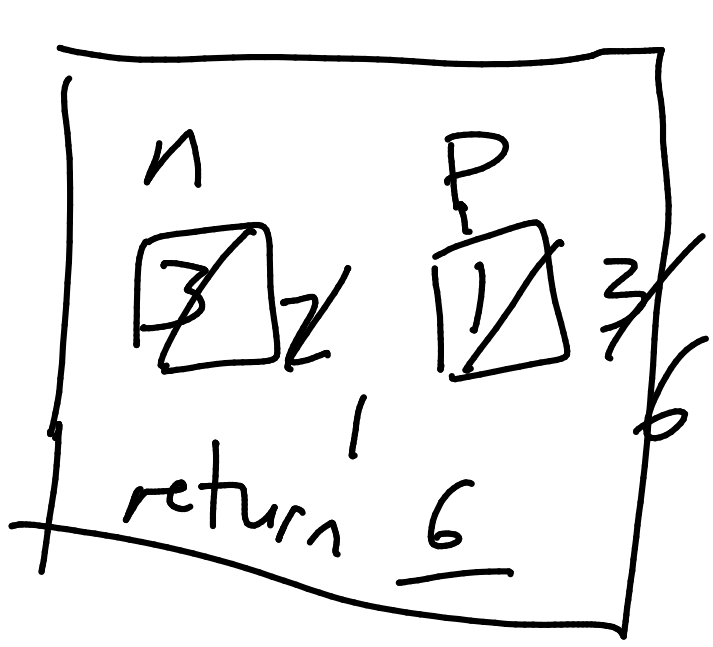
- can we rewrite factorial to use a tail call instead?
- does it matter? Does Guile optimize for tail recursion?
- does python.org Python? Oracle Java? other?

↳ controversial

(fact-tail 3 1)



A good ~~optimizing~~ tail call optimizing compile interpeta wouldn't waste memory on 3 frames, instead:



it would just reuse the same frame

Most functional programming environments do this.

How does malloc work?

- actual malloc, but also memory allocation in general on the heap?

Java

String x = new String("—")

Python

x = "—" ⇒ object, goes in heap

How do we allocate memory in the heap?