

OS - 3 → Callables and Futures (Practical → Java)



→ Synchronization

→ Producer / Consumer Problem

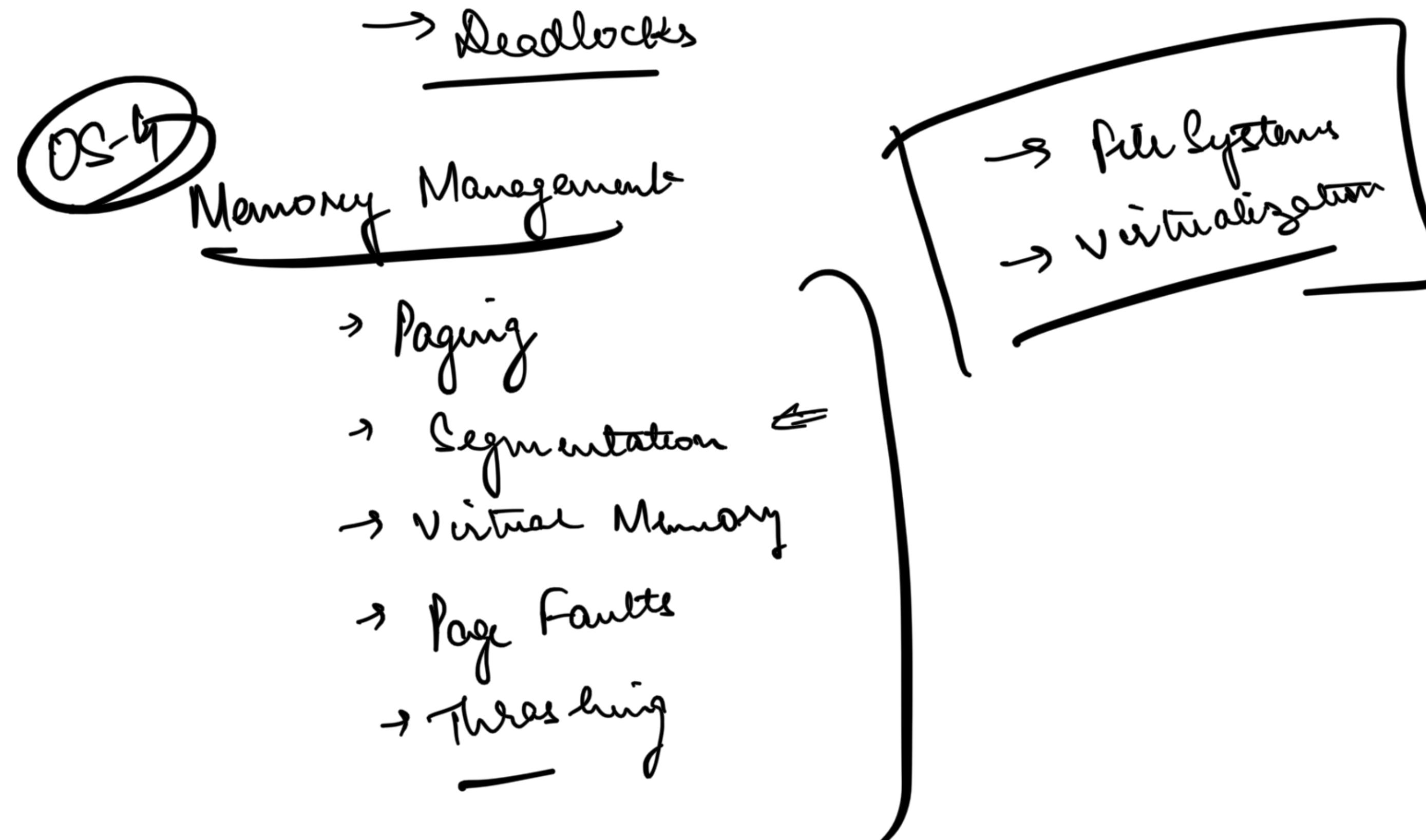
Code → T (Adder / Subtractor Problem)

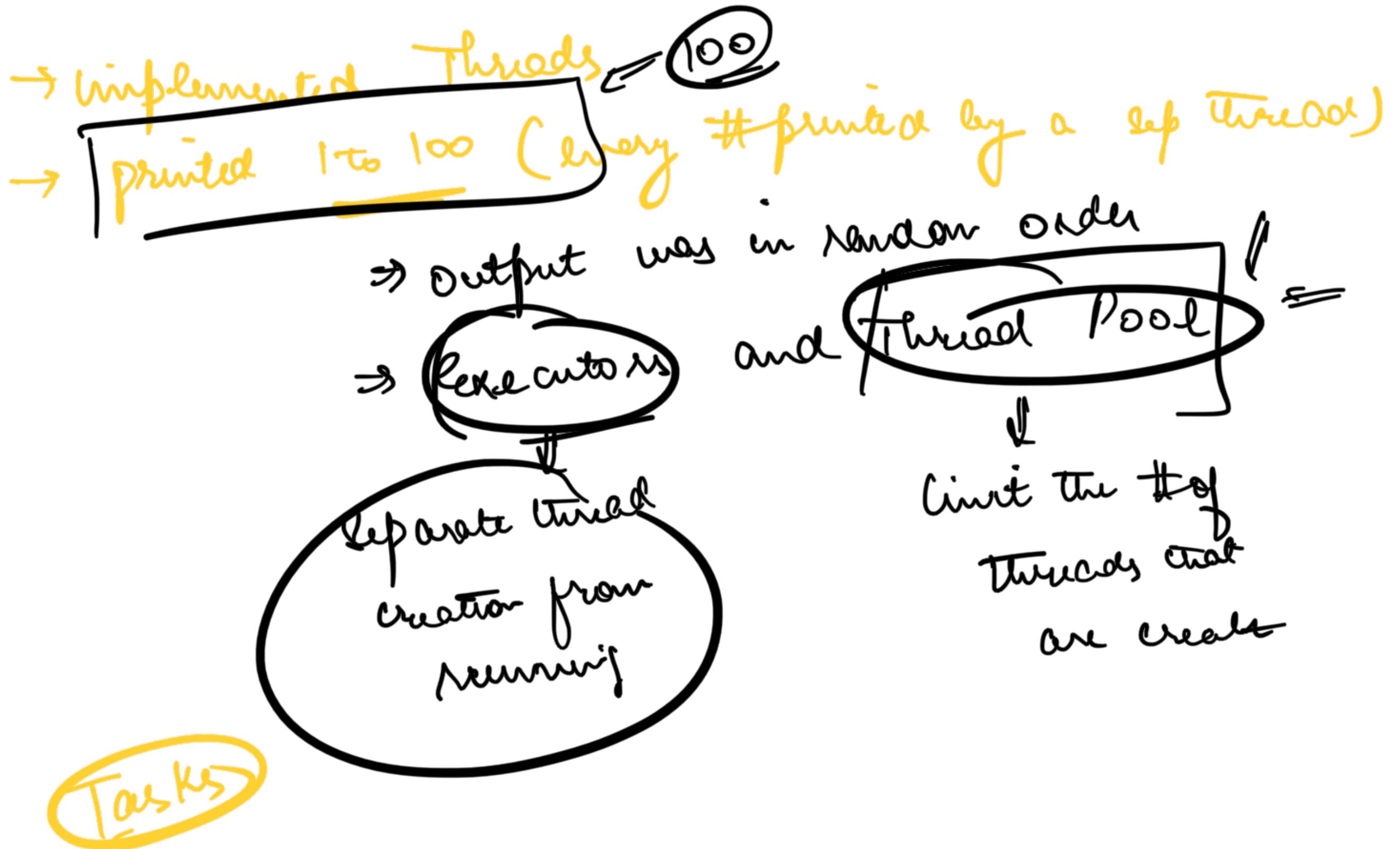
→ How to Solve

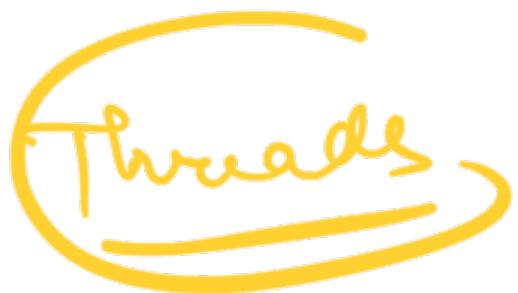
→ Hardware Sol<sup>m</sup>

→ Peterson Sol<sup>m</sup>

→ Practical :  
Mutex & Semaphore







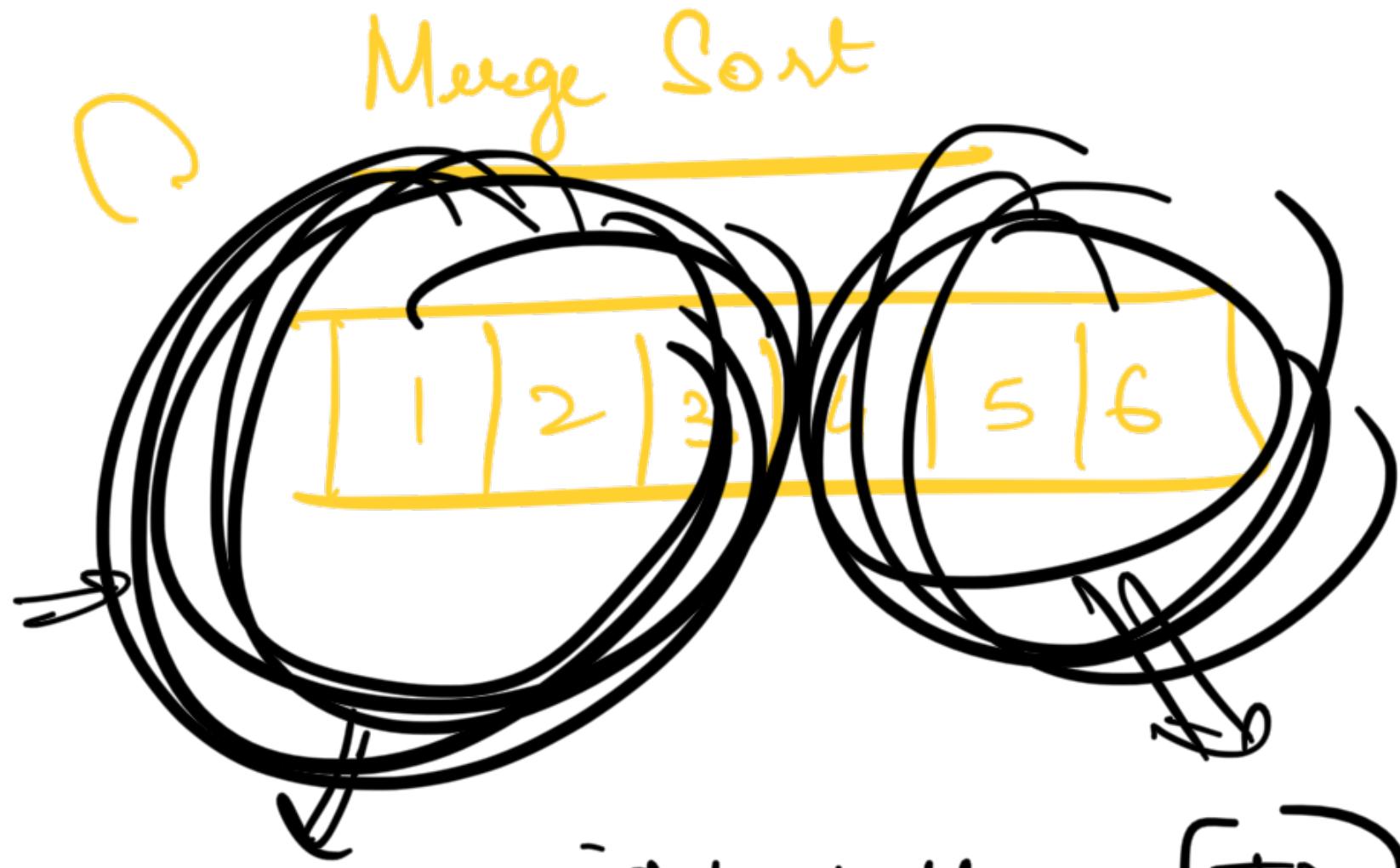
run() {

    → in normal f<sup>nc</sup>  
    I expect return

Value

?



~~Merge~~

Callable

Runnable

Integer call() {  
 ↓  
 list<Animal>  
}

class

Sorter

implements



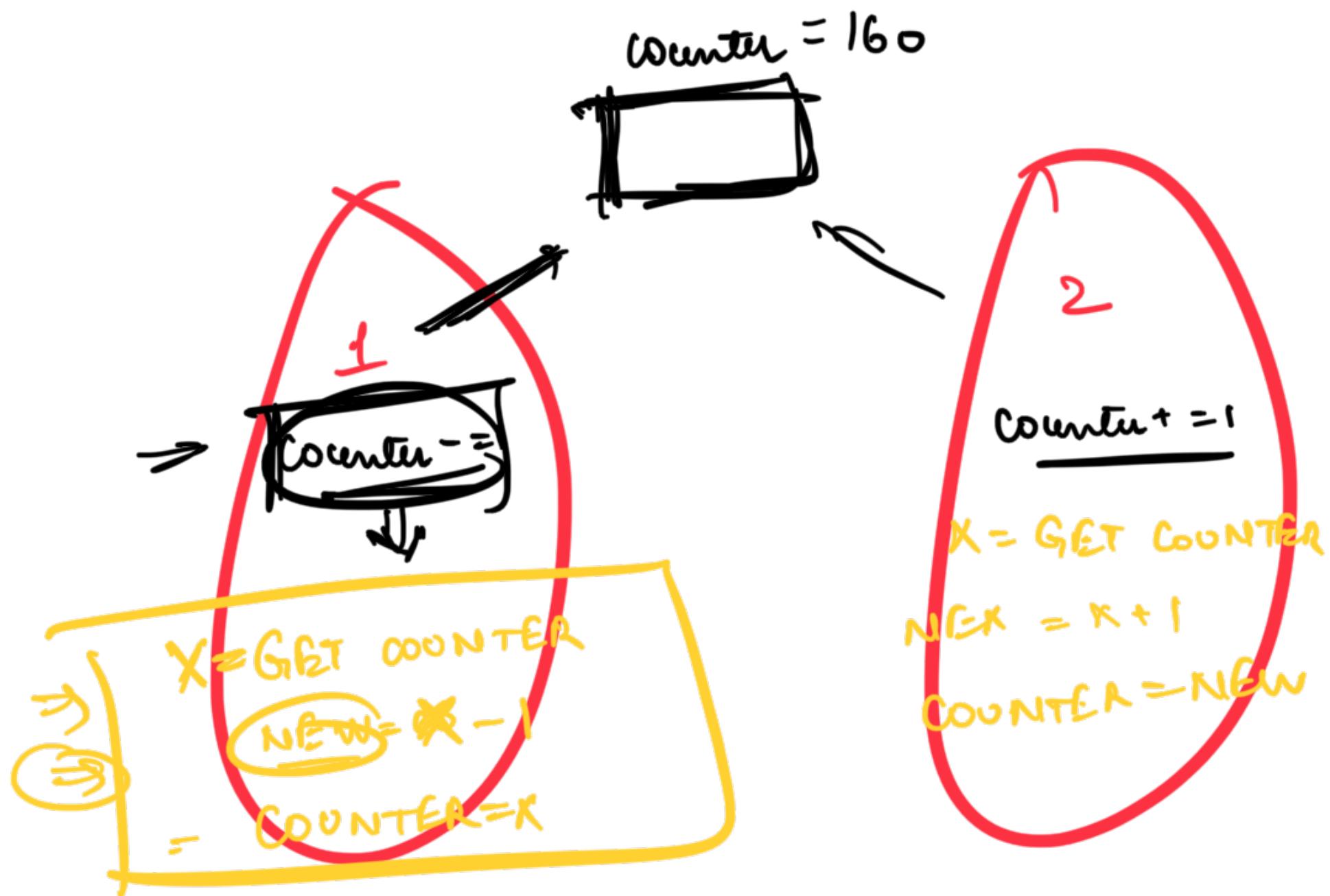
@Override

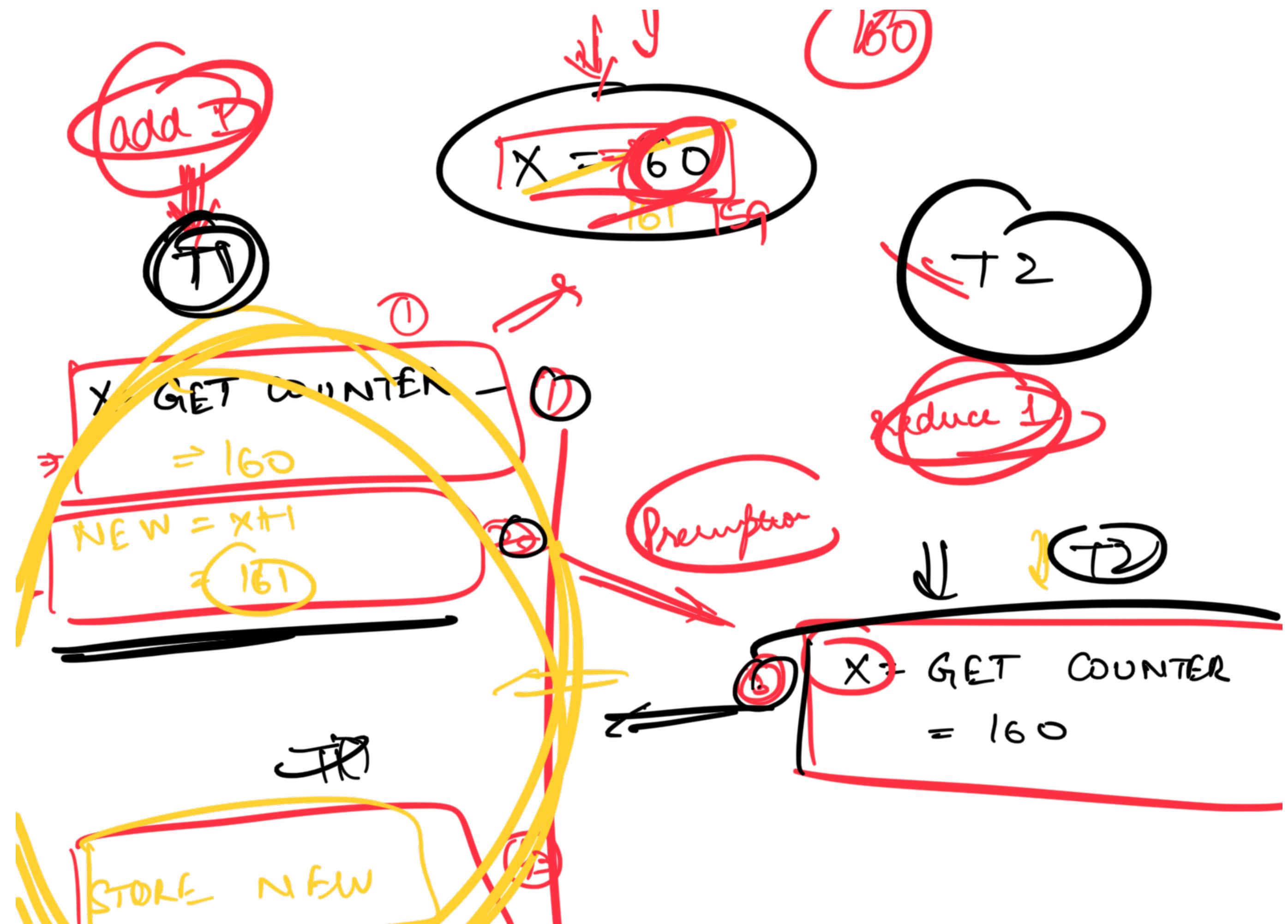
list < Integer

coll ()

Synchronizator Problem

## Data Sharing





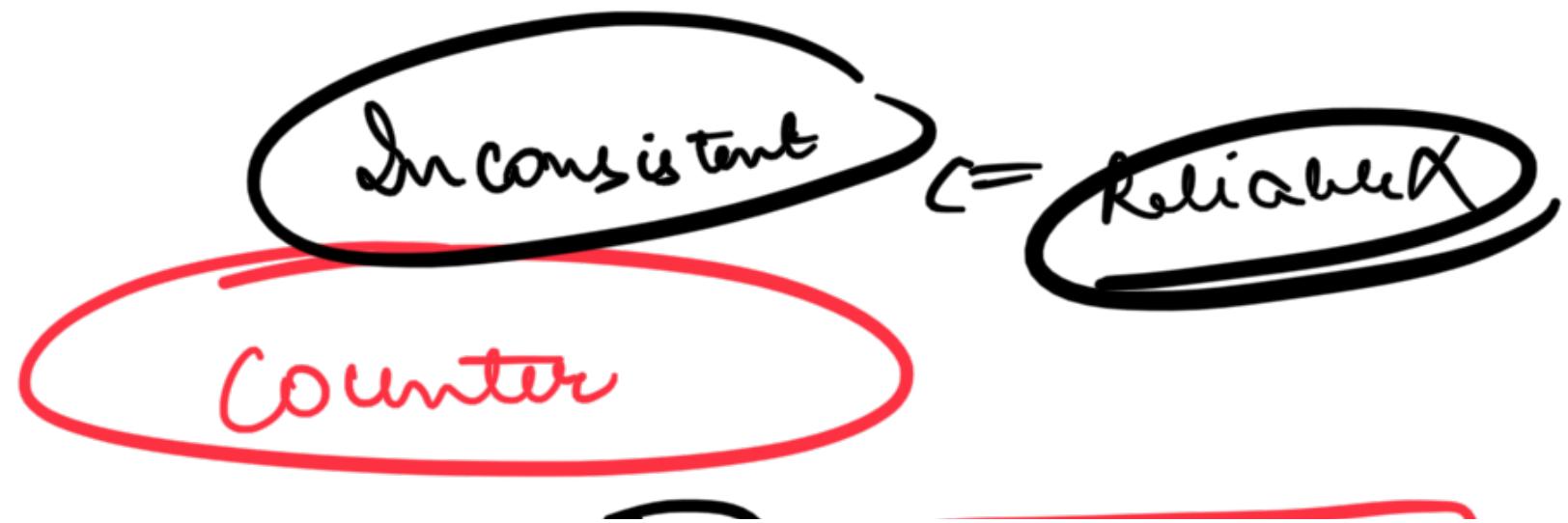
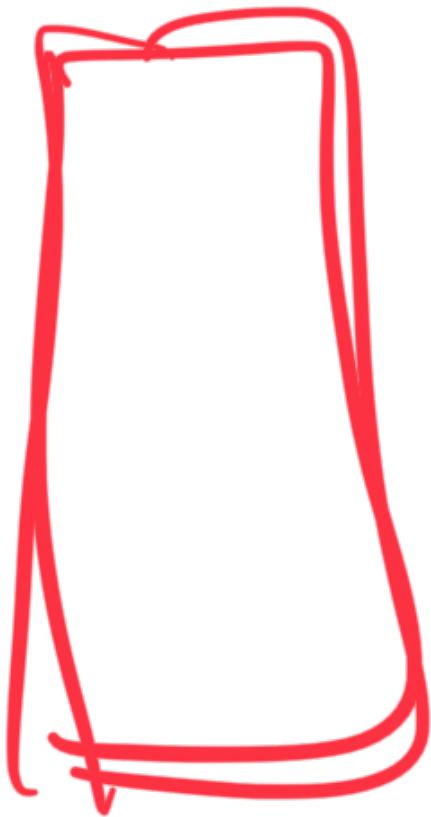
⑤  $\text{NEW} = \text{X} - 1$   
= 159

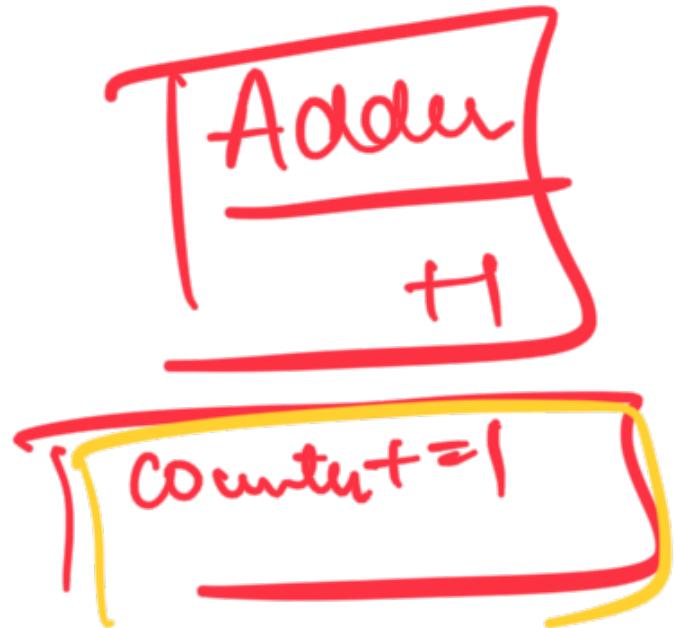
⑥ STORE NEW

Race Condition :

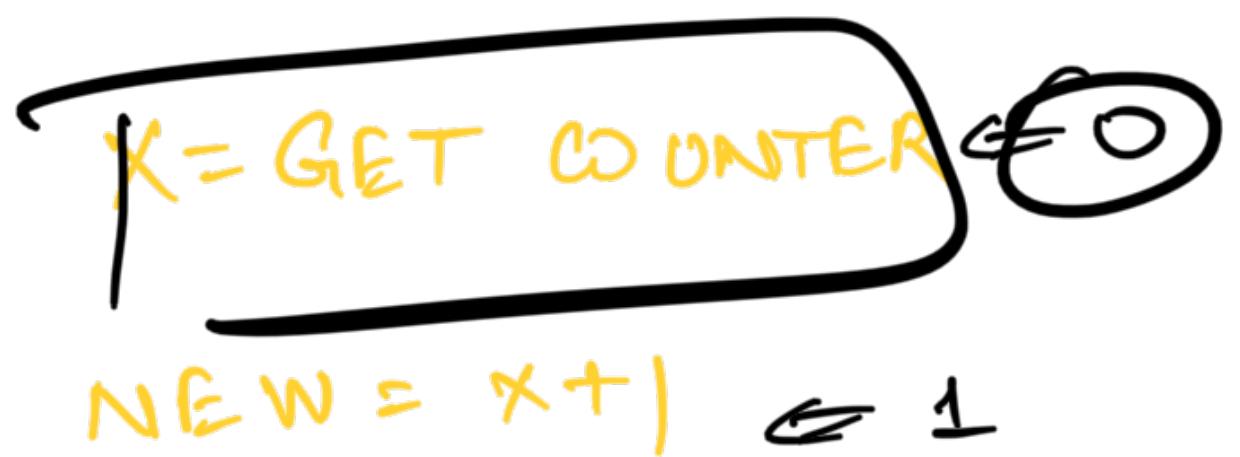
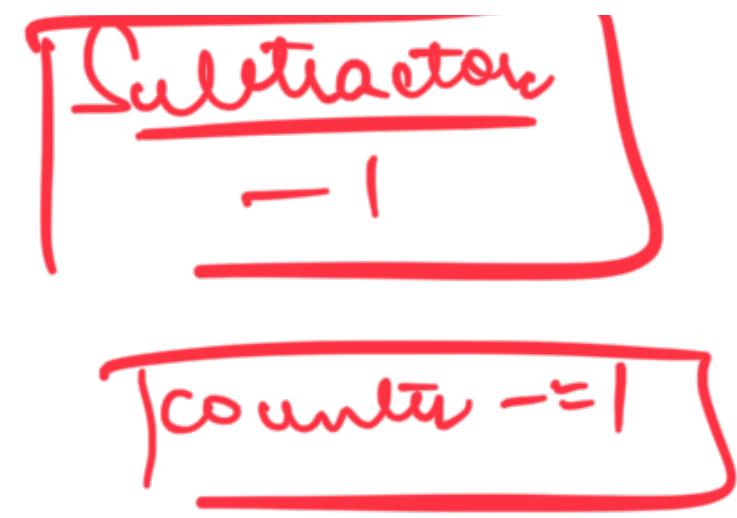
When more than 1 threads are manipulating  
the same value and the final answers  
changes based on who executed in what

Order:



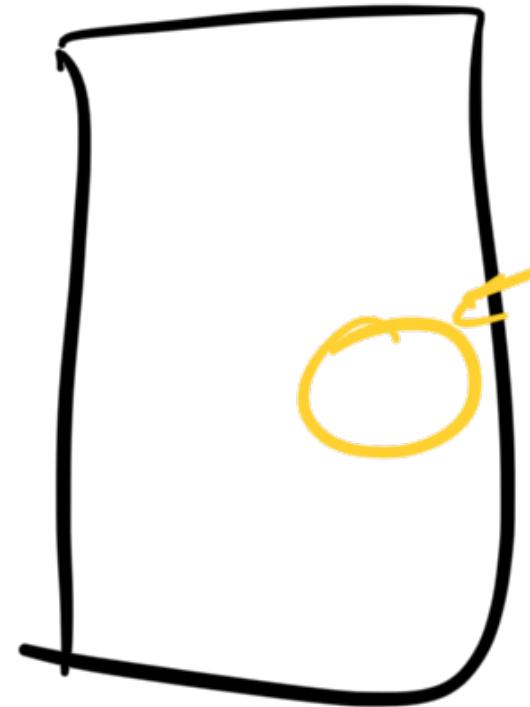


~~0  $\oplus$  1~~



COUNTER = NEW  $\leftarrow 1$

$\Rightarrow$  COUNTER = NEW



Le mushroom available

X = GET MUSHROOM

CAN MUSHROOM



→ 3 sec  
AWA →

X = GET WASHROOM  
WENT CONSIR.

WENT IN SIR

## Critical Section

---

Part of the code where a shared variable is being accessed.

---

### WHY SUCH ISSUES HAPPEN

---

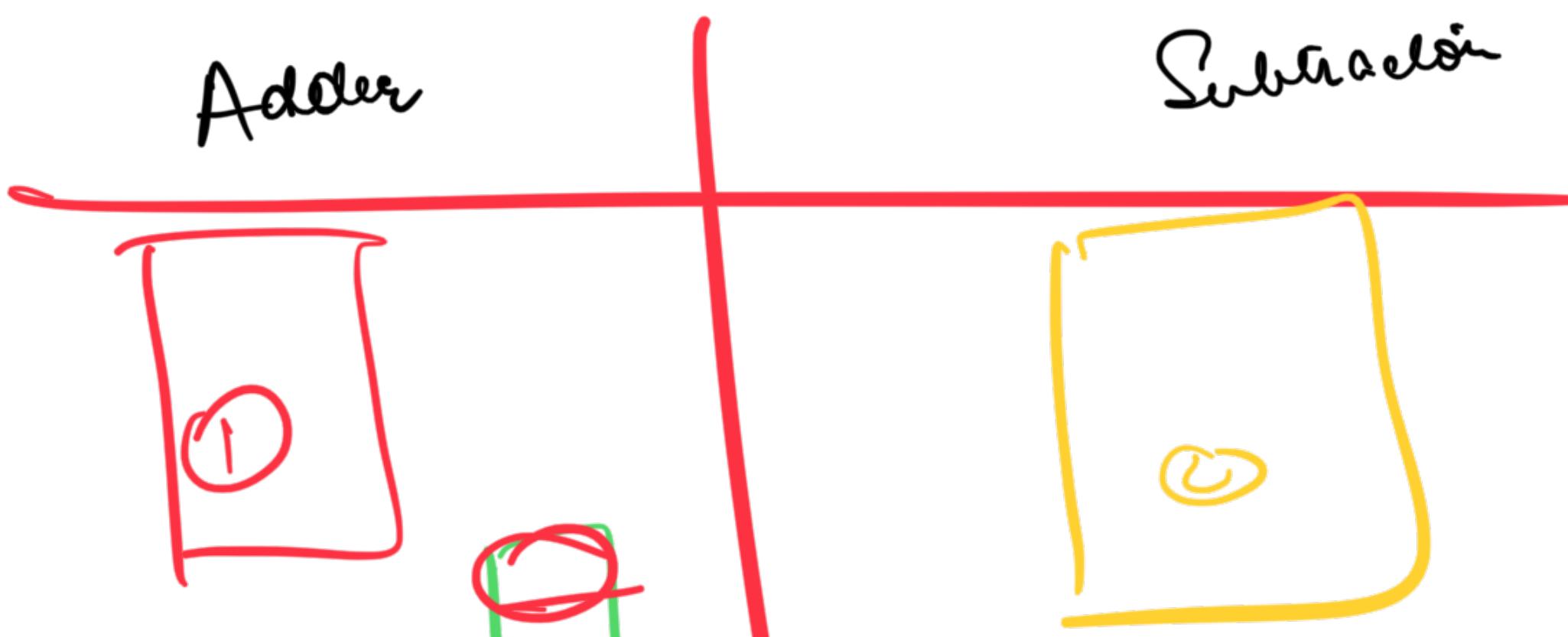
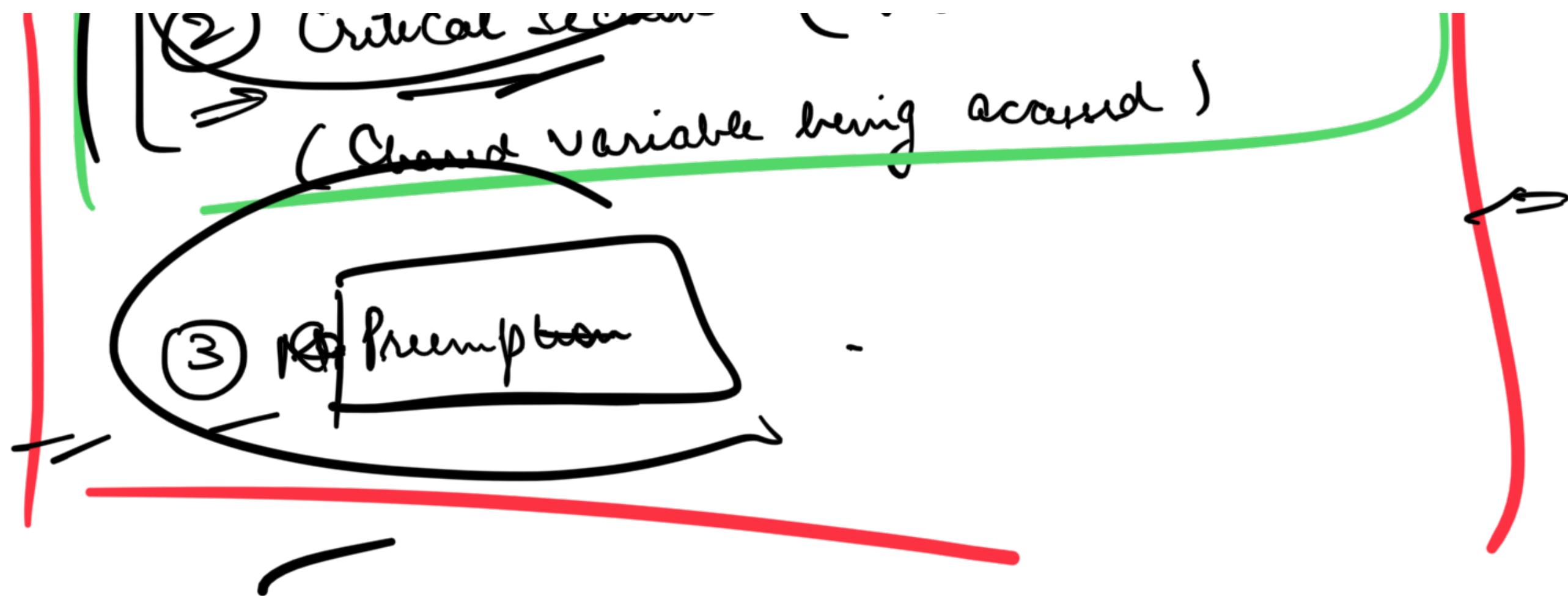
① Race Condition

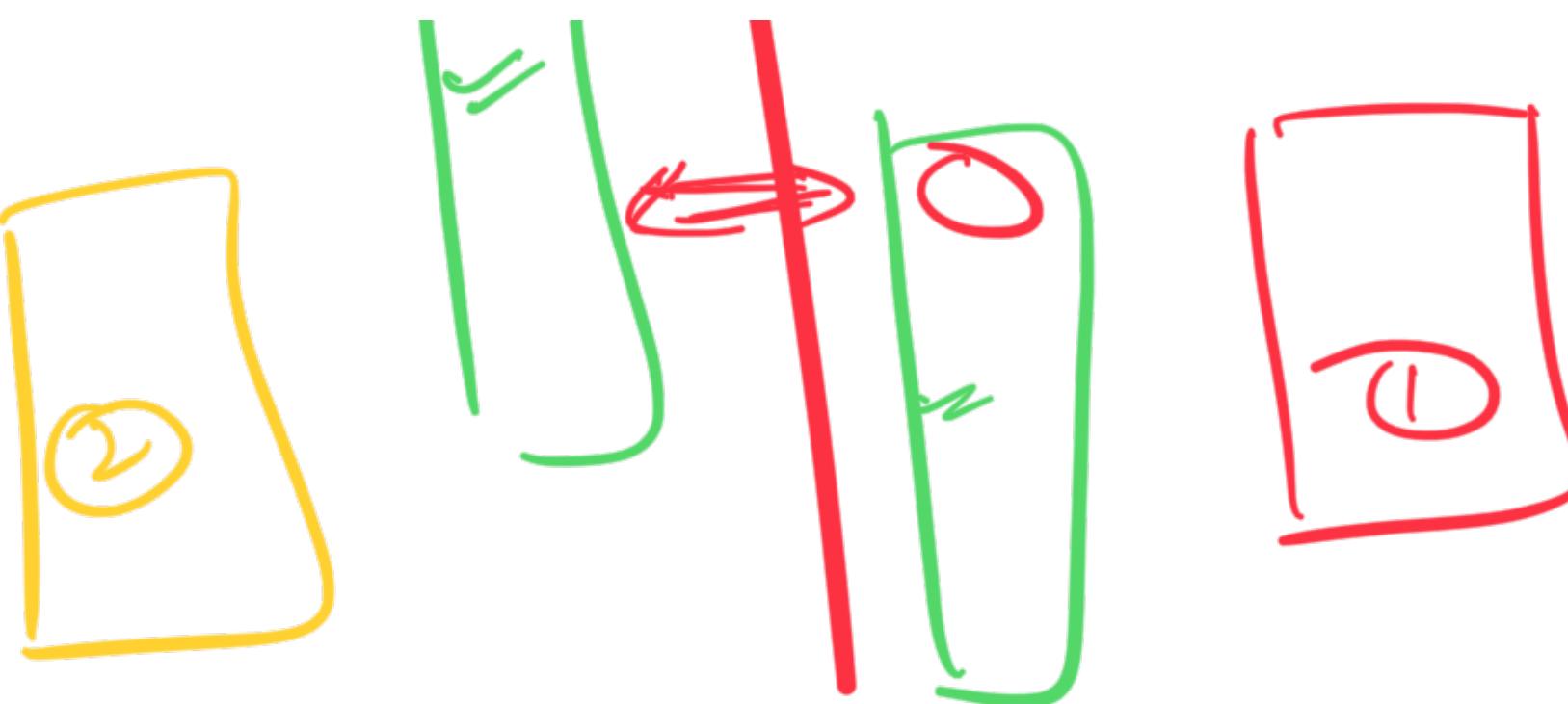
(2 people going to)

= (More than 1 thing is happening in parallel)

Concurrent access

(washroom)





How to Solve

What is a good sol<sup>n</sup>

# ① Mutual Exclusion

Only one ~~person~~ thread should be working

# ② Progress

Everything stops X

Overall system should keep moving

# ③ Bounded Waiting

No one should wait infinite

# ④ No busy waiting

||

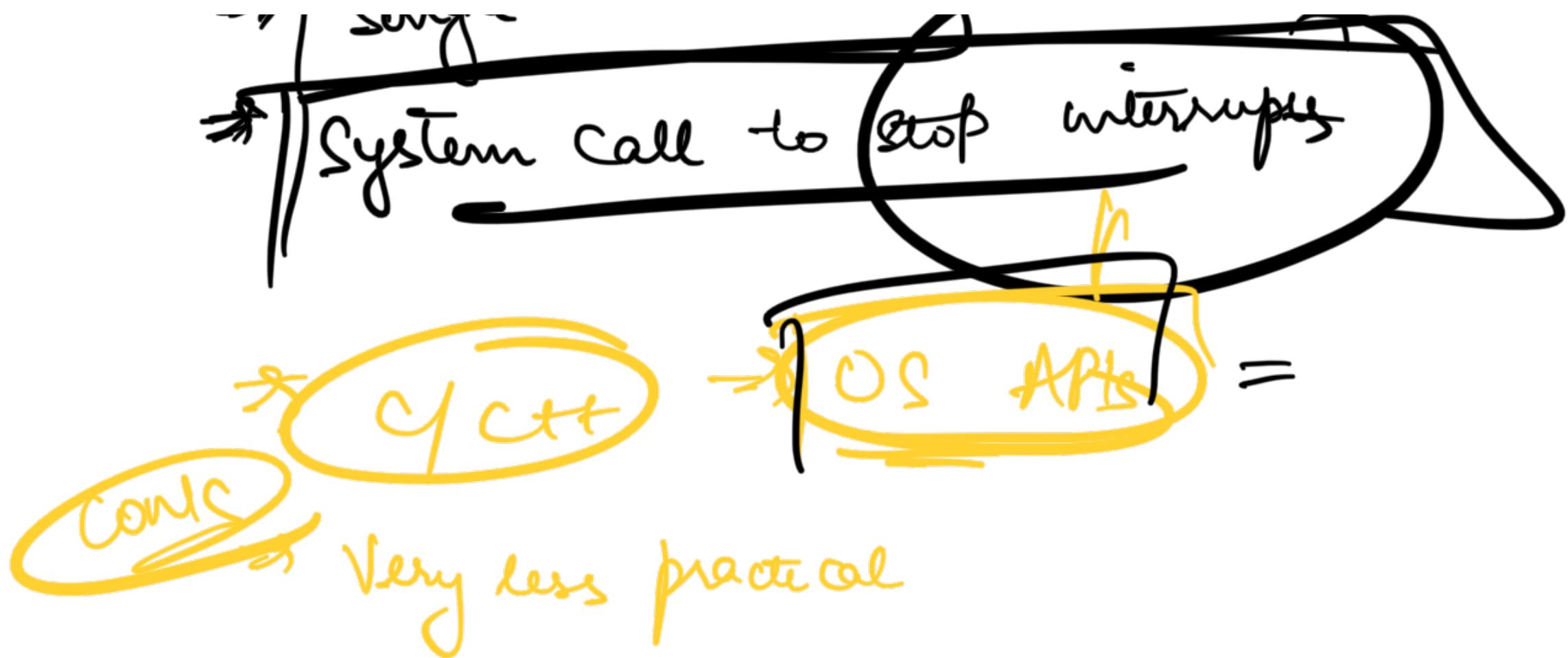
white (~~Be!~~ washaar Anstreichen)



Solution

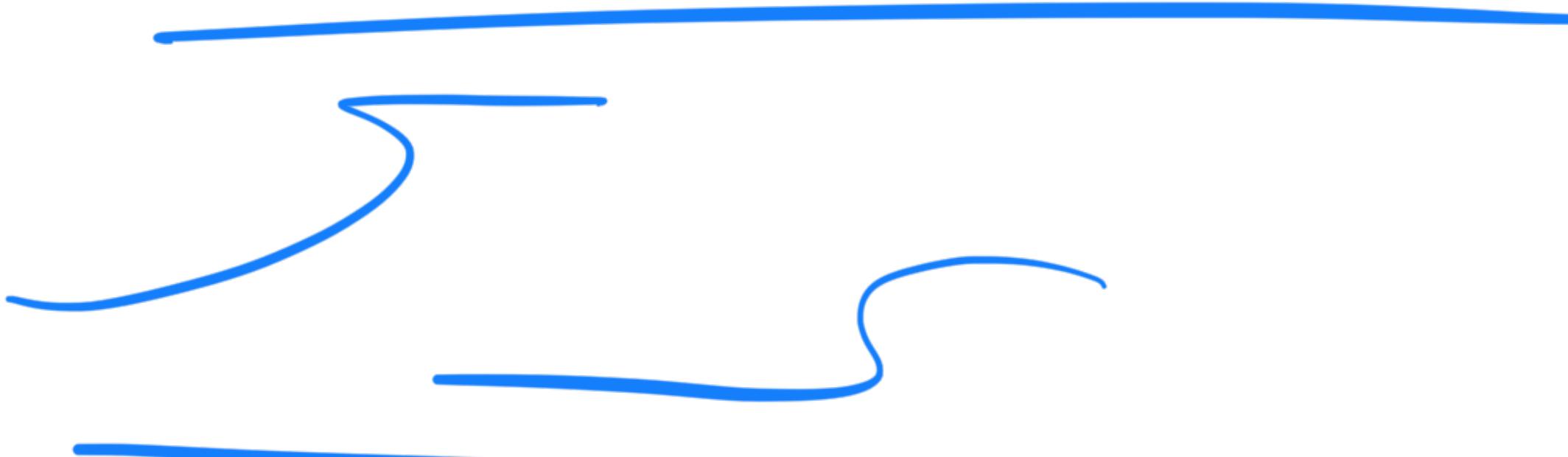
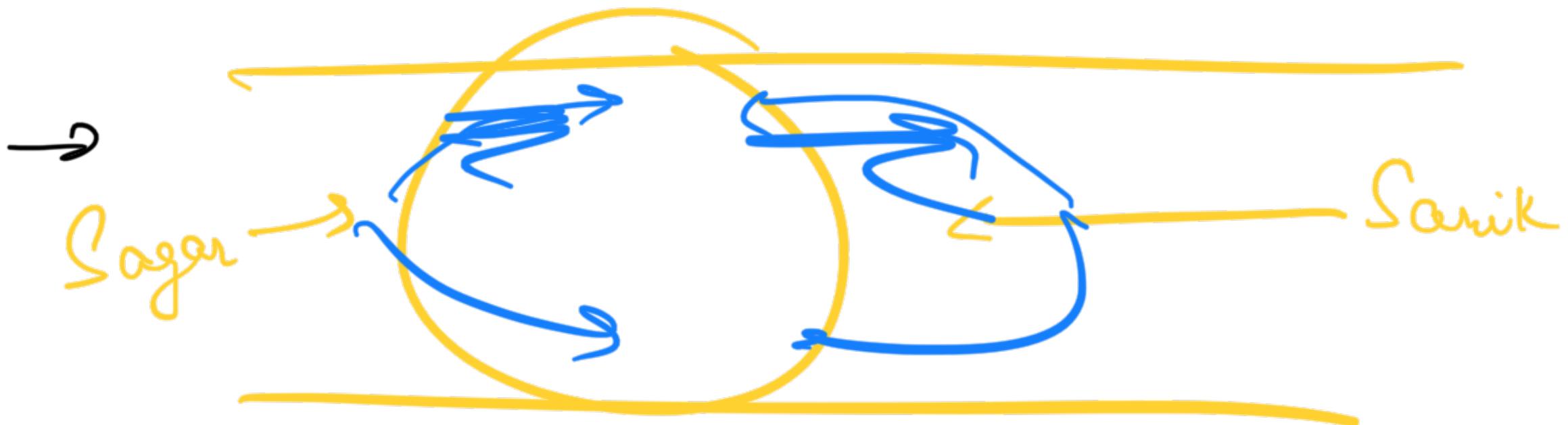
→ Hardware Sol<sup>n</sup>

→ viele von CPU's



## Software Solution

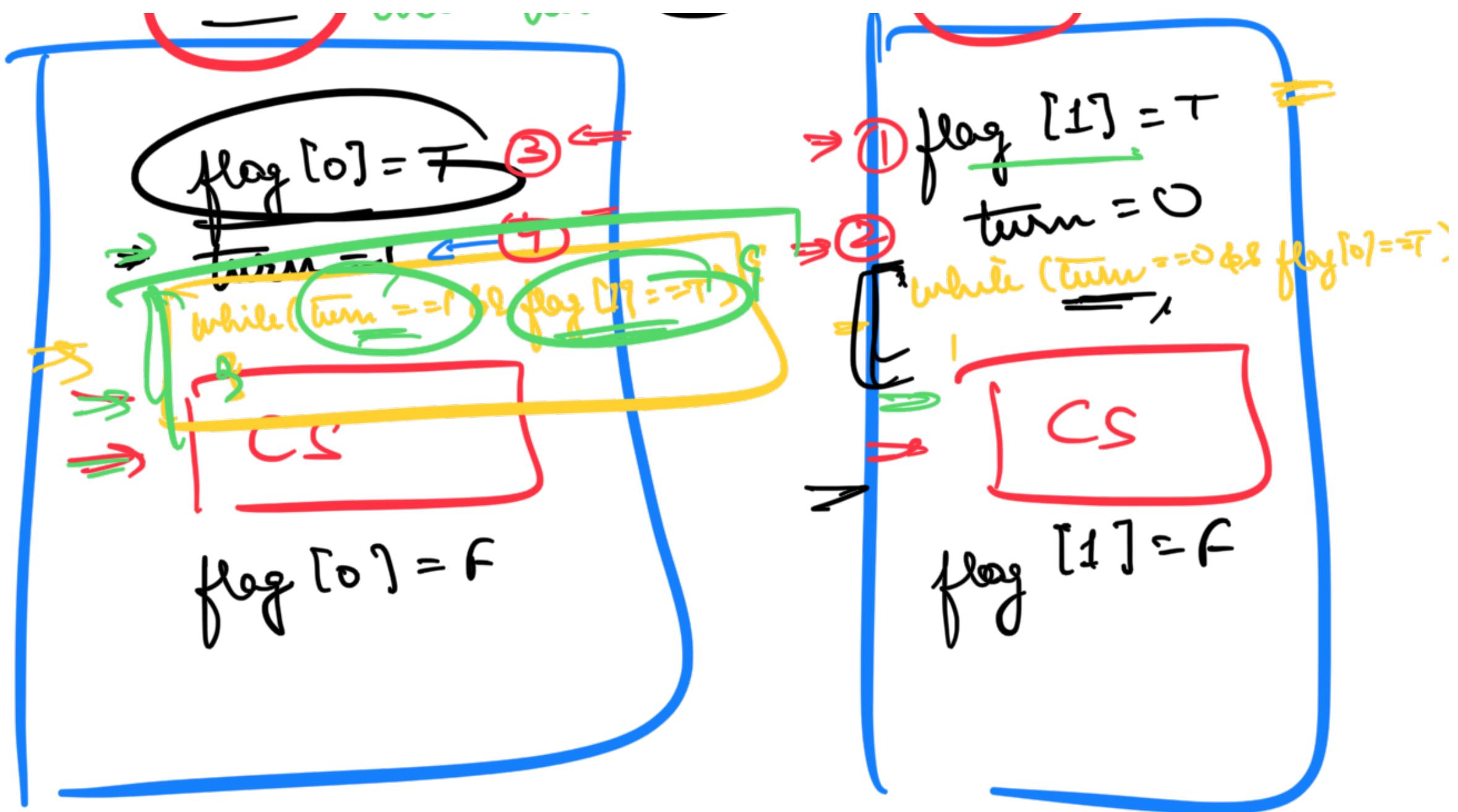
Peter's own "Sol"



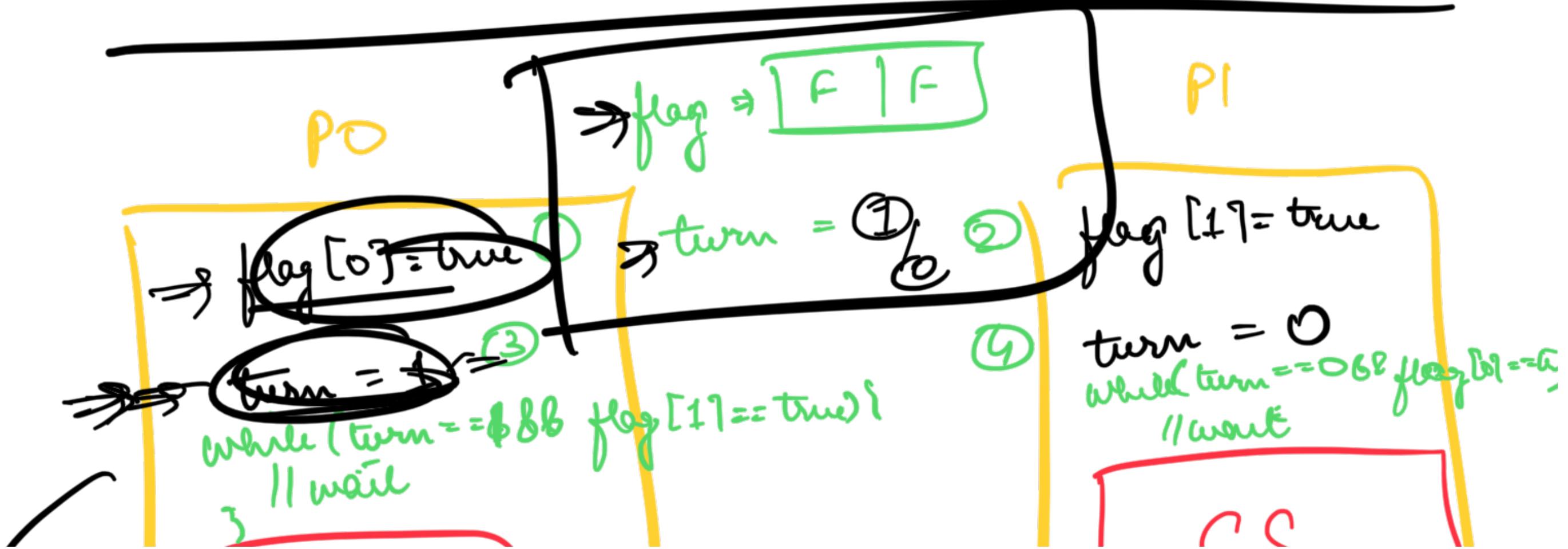
1) boolean  
ext[ ] flag = 

F		F
---	--	---

  
int turn = 1  
(P0) V (P1) V



C Q



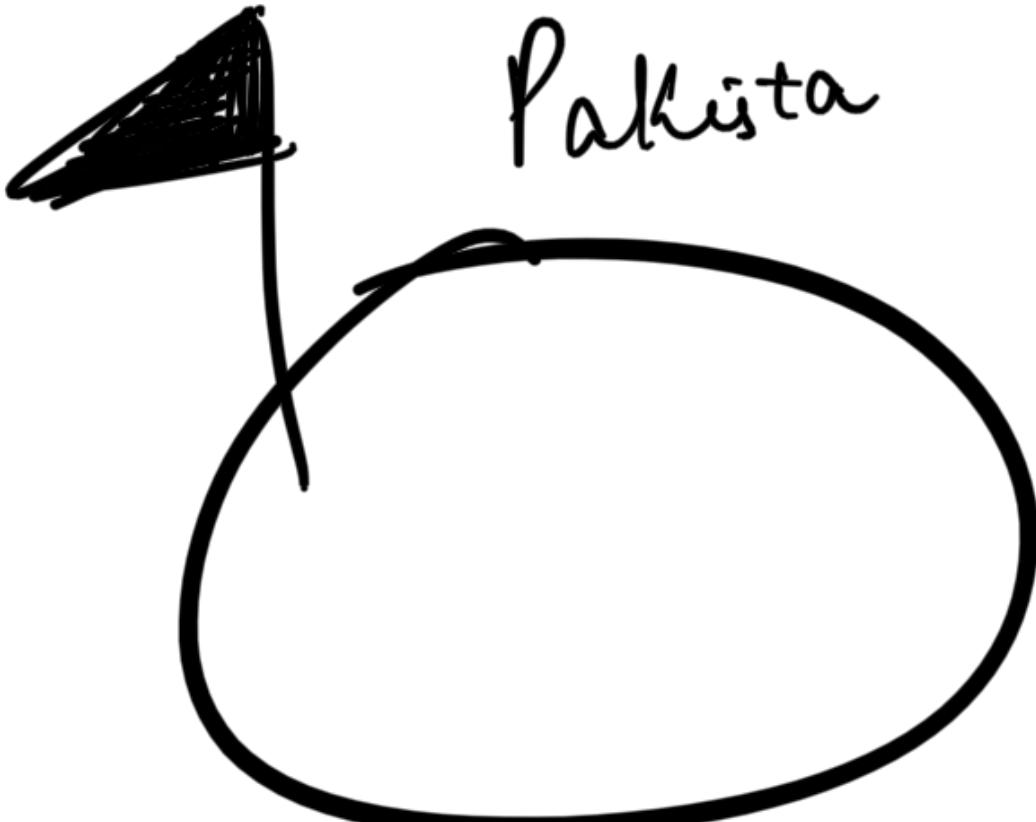
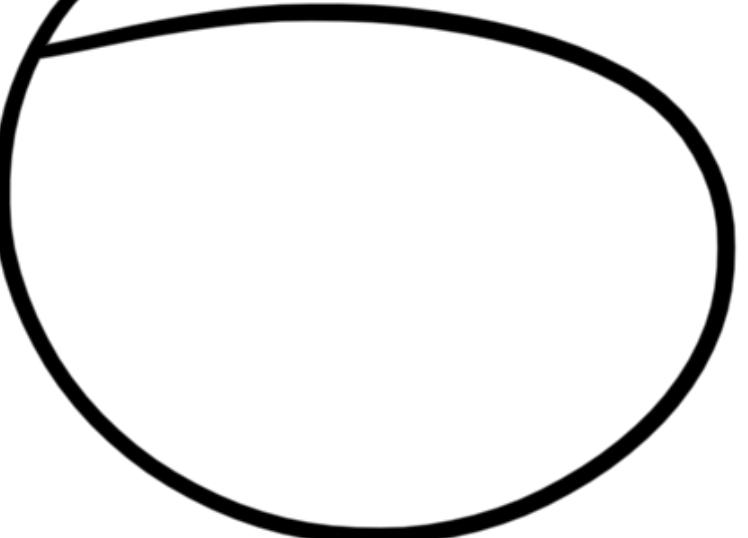
ICS

flag[0] = false

flag[6] = true

If PO goes to CS

- ① flag[1] = false
- ② turn = 2 India



N

~~If I~~ went to CS

- ① either you didn't want to go [ $\text{flag} = \text{false}$ ]
- ② or you told me to go [ $\text{turn} = 0$ ]

OS - 1

HW

C

Quick sort via Threads

^ . . . n

②

Rewatch Peterson's Sol

