Agenda

Concurrency 2

→ Callables and Futures

↳ Merge Sort using Callables and Futures

→ Adder Subtractor Problem
(Synchronization Issues | Race Cond^n)

↳ Synchronized

→ Mutex and Semaphores

→ Reentrant Locks

→ Read Write locks ⇐

Concurrency 3

Con cu

→ Fairness of locks

→ Atomic Data Types / Concurrent HashMap

→ Volatile

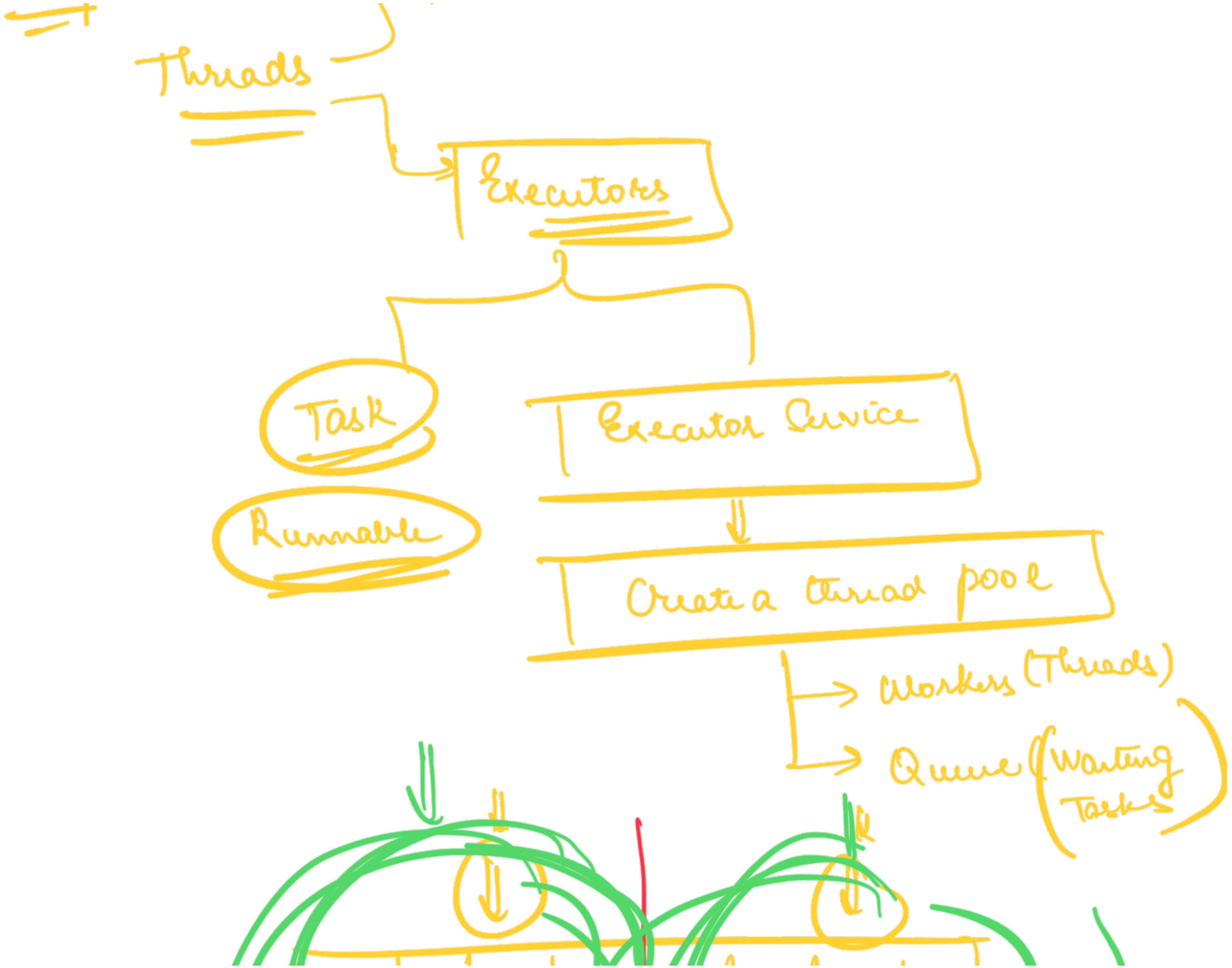
3 classes

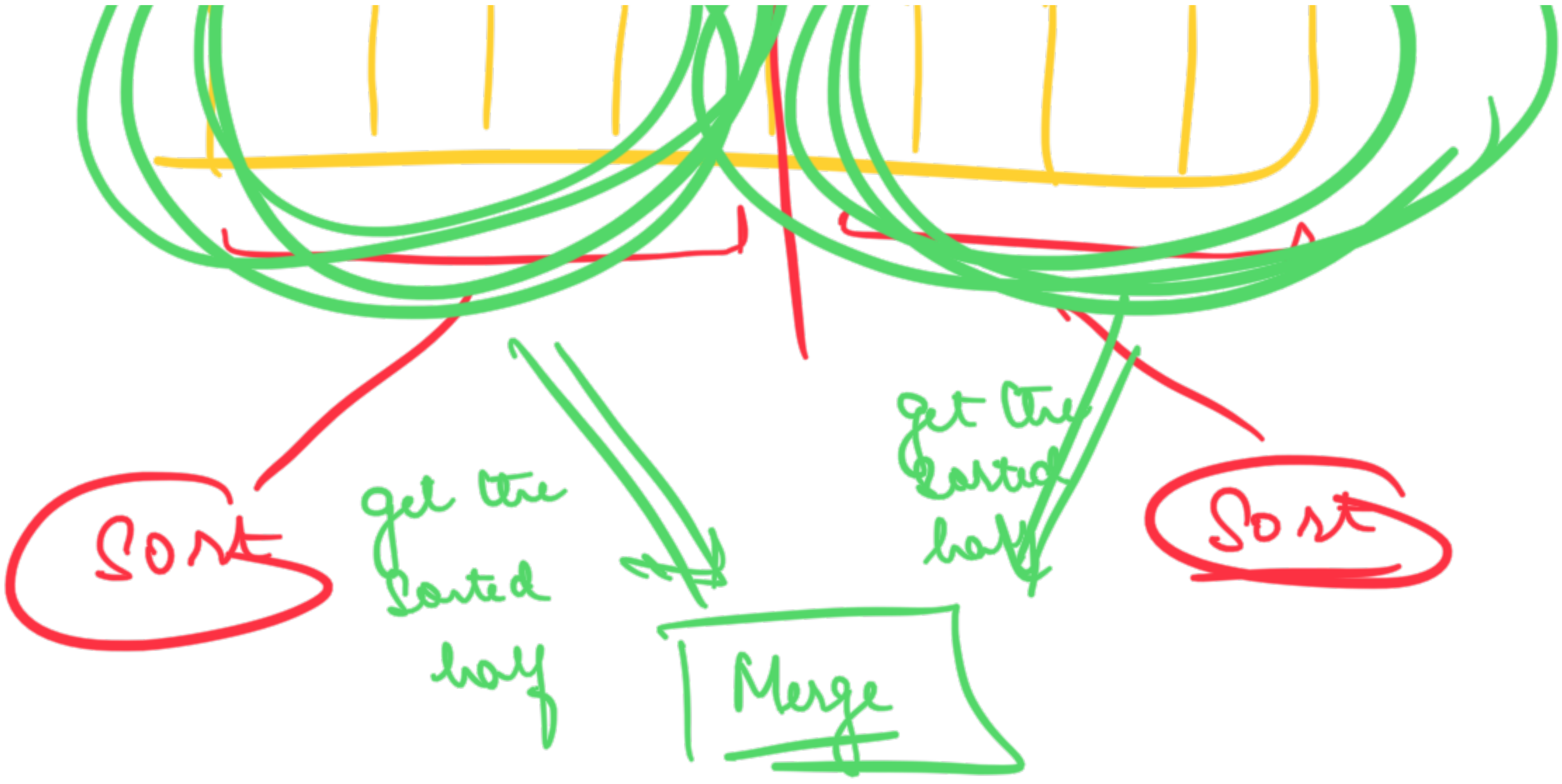
→ Language Module

Java

Recap

→ Runnable


Threads
Executors
Task
Runnable
Executor Service
Create a thread pool
Workers (Threads)
Queue (Waiting Tasks)



Callables and Futures

We created a Task using runnable

run( )

Create a Task that returns something

$\boxed{\text{Callable}}$

Call()

Class Sorter implement $\boxed{\text{Callable<List<Integer>>}}$

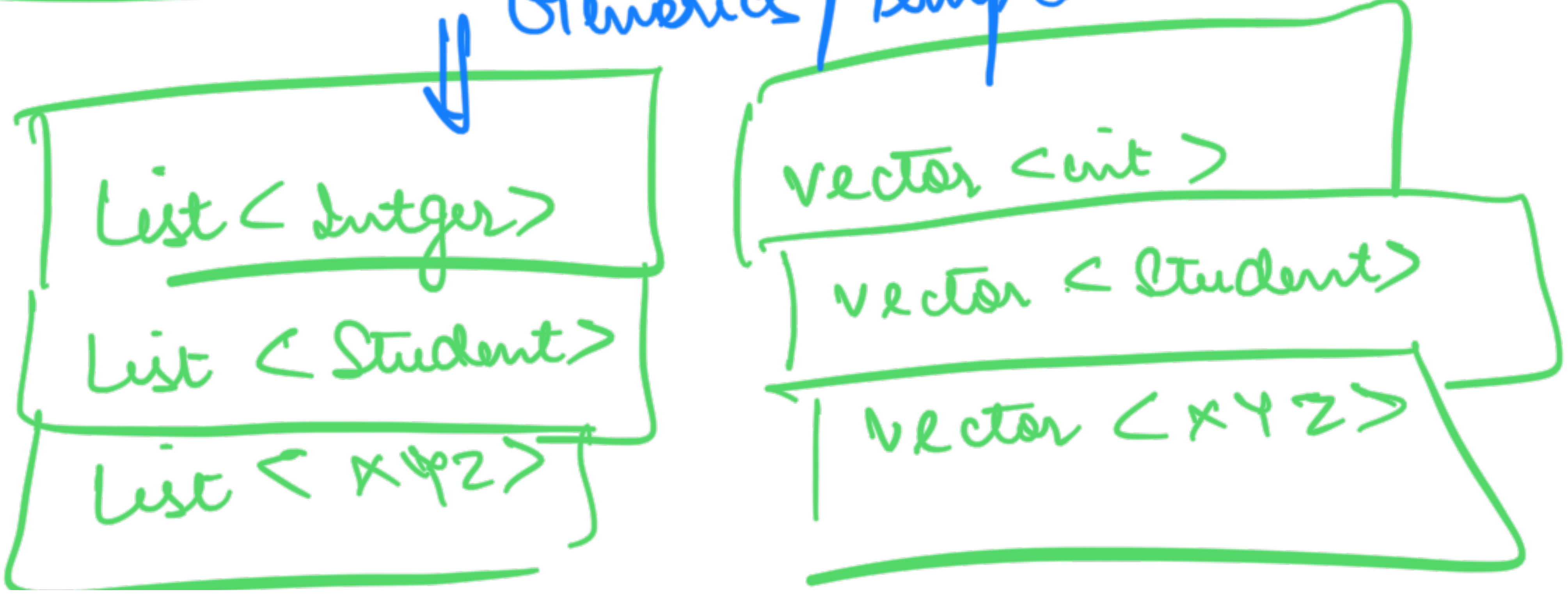
List<Integer> call() {

}

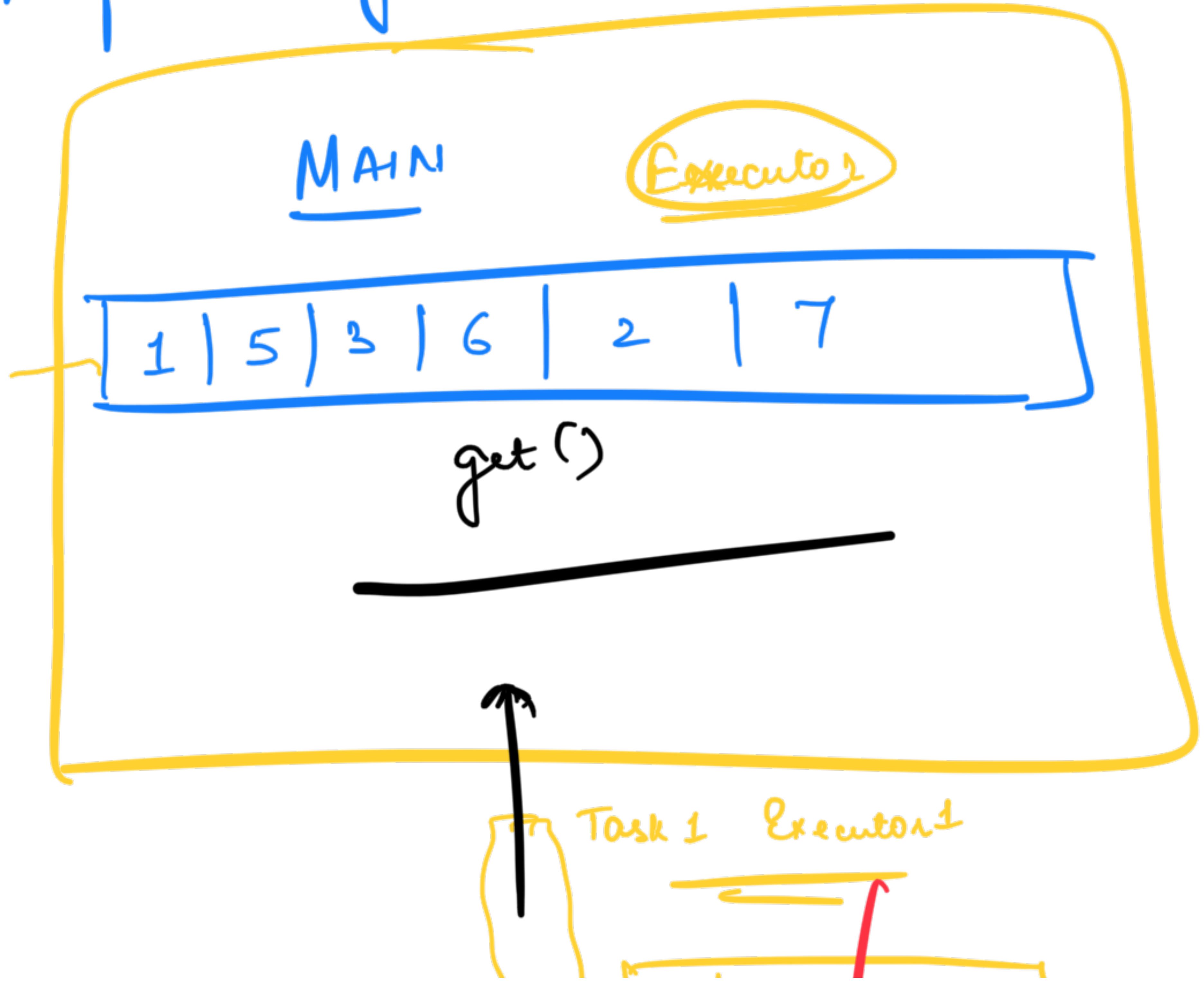
generics

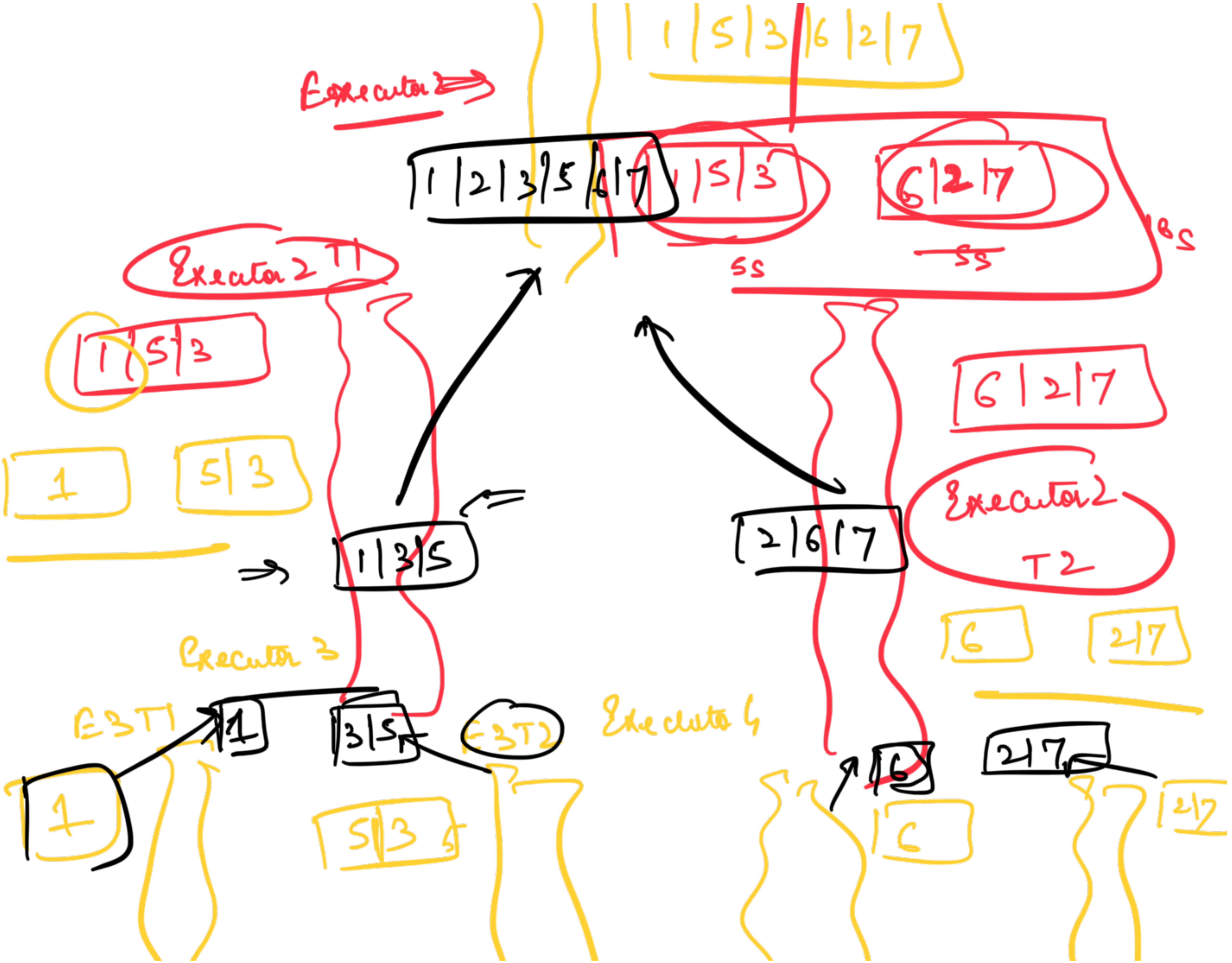
templates

Generics | Templates



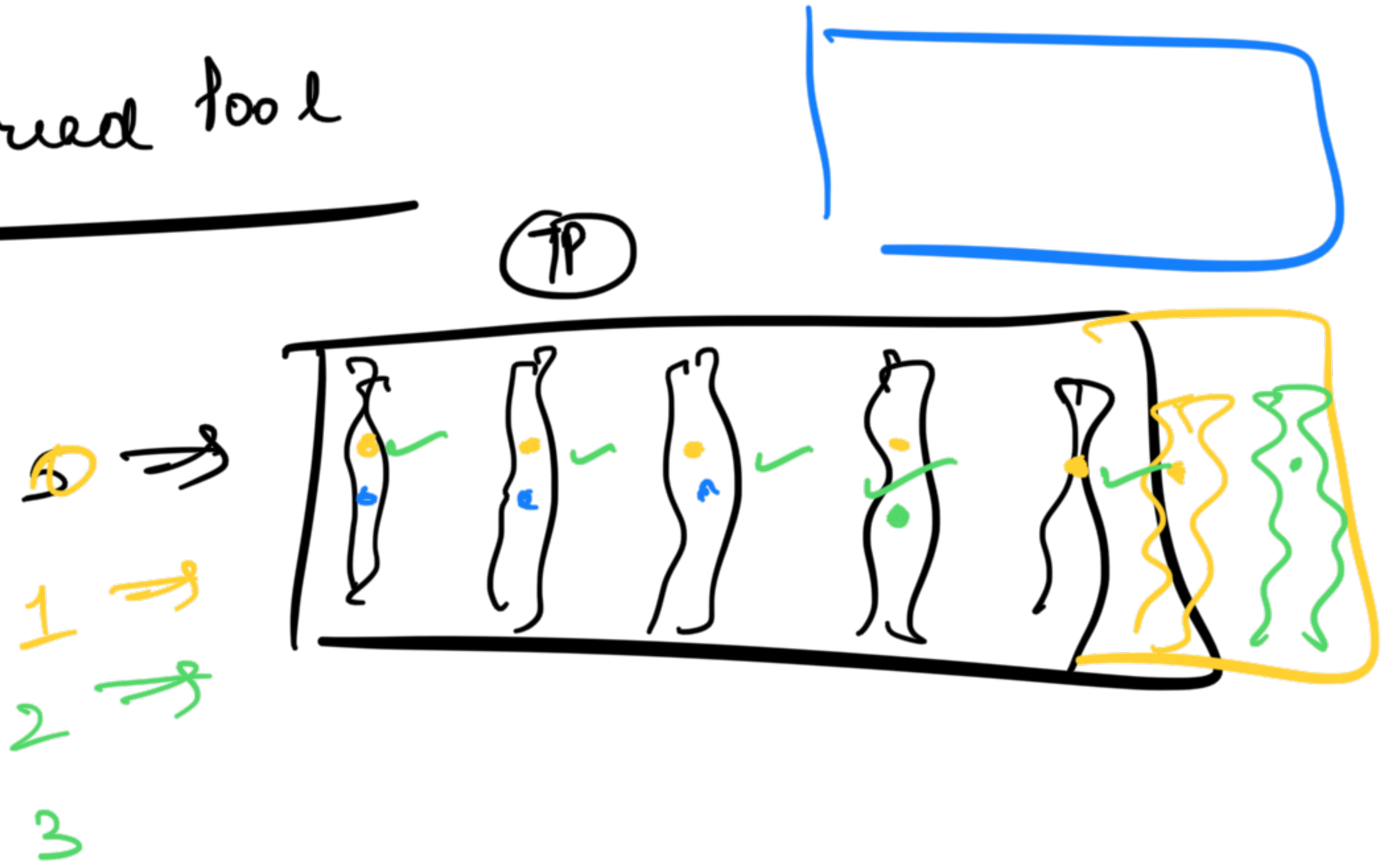
Map < String, int >

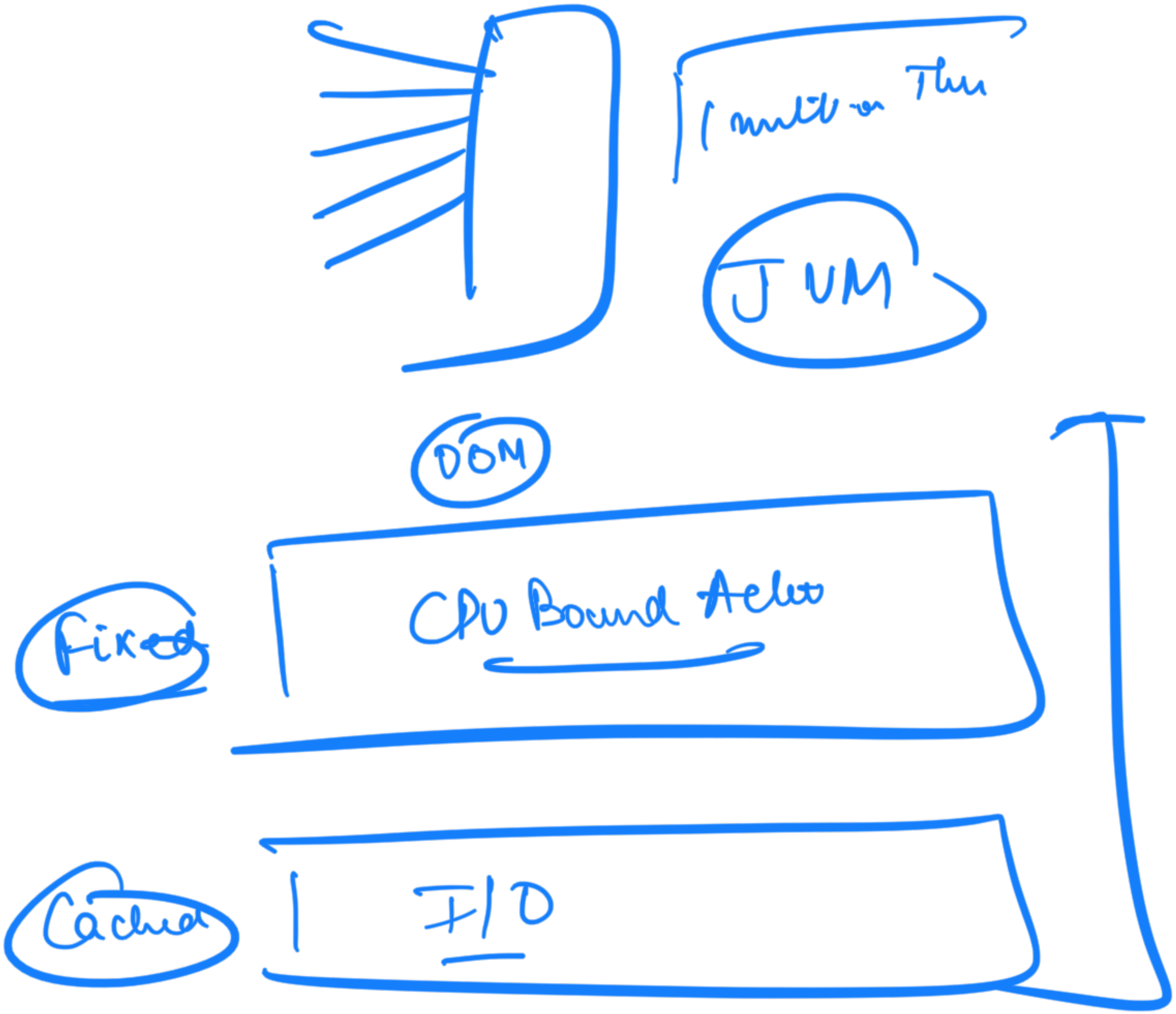




1 | 5 | 3 | 6 | 2 | 7
Executor 2
1 | 2 | 3 | 5 | 6 | 7
1 | 5 | 3
6 | 2 | 7
SS
SS
BS
Executor 2 T1
1 | 5 | 3
1
5 | 3
1 | 3 | 5
2 | 6 | 7
6 | 2 | 7
Executor 2 T2
6
2 | 7
Executor 3
E3T1
1
3 | 5
E3T2
Executor 4
6
2 | 7
1
5 | 3
6
2 | 7

Cached Thread Pool




multi-on Thu
JVM
DOM
CPU Bound Acho
Fixed
Cached
I/O

I/O
I/O
I/O
FTP
Size=3
CPU

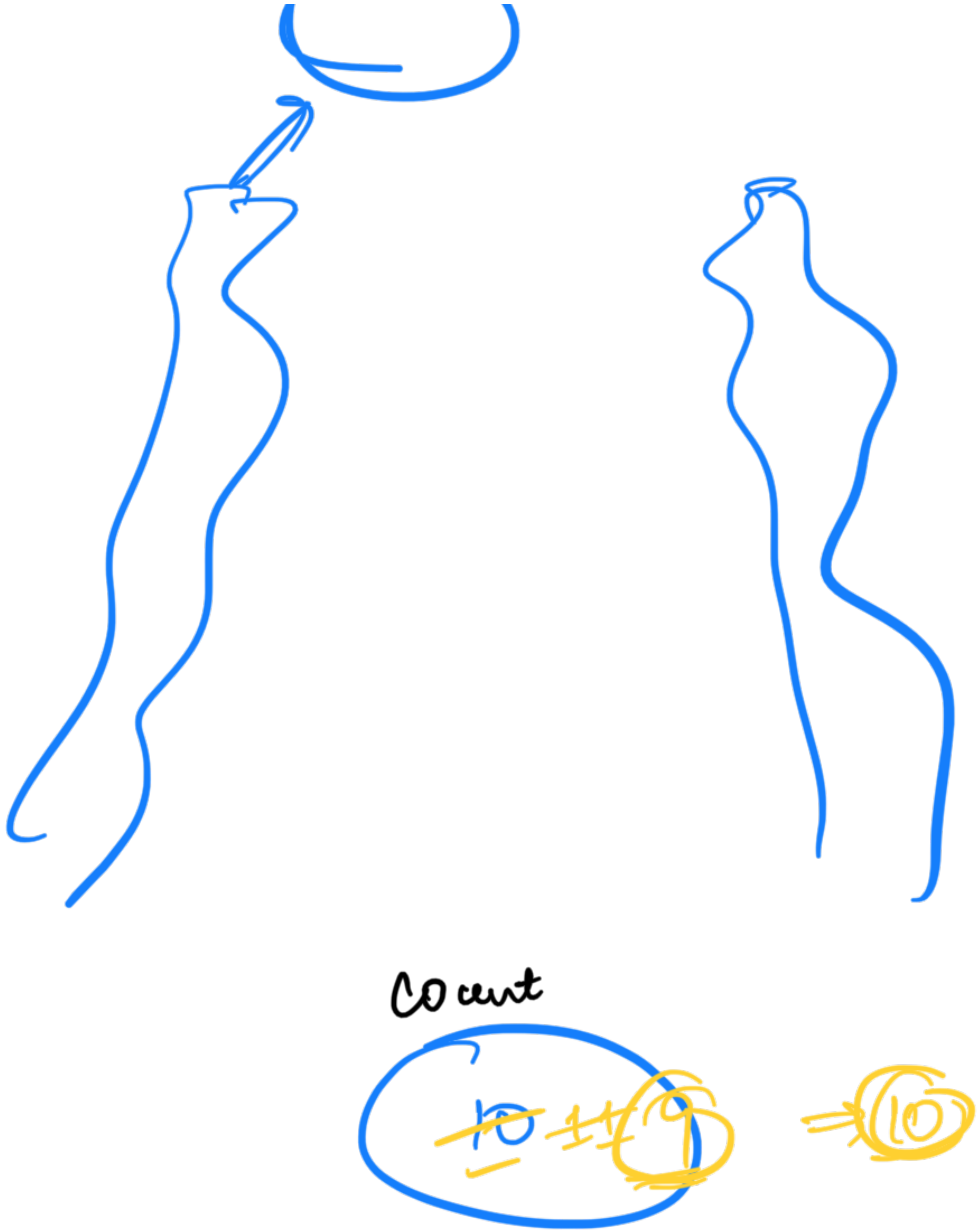
Assgn

Multithreading

① Implement Quick Sort via

→ Adder Subtractor Problem

Data Sharing with threads is a headache


Count
~~10~~ ~~11~~ 9
→ 10

Race Condition

Adder 1

count += 1

Subtractor 1

count -= 1

Critical section

1 Read count ← 10

2 INC count ← 11

4 Store count ←

3 → Read count → 10

5 → DEC count ← 9

6 → Store count

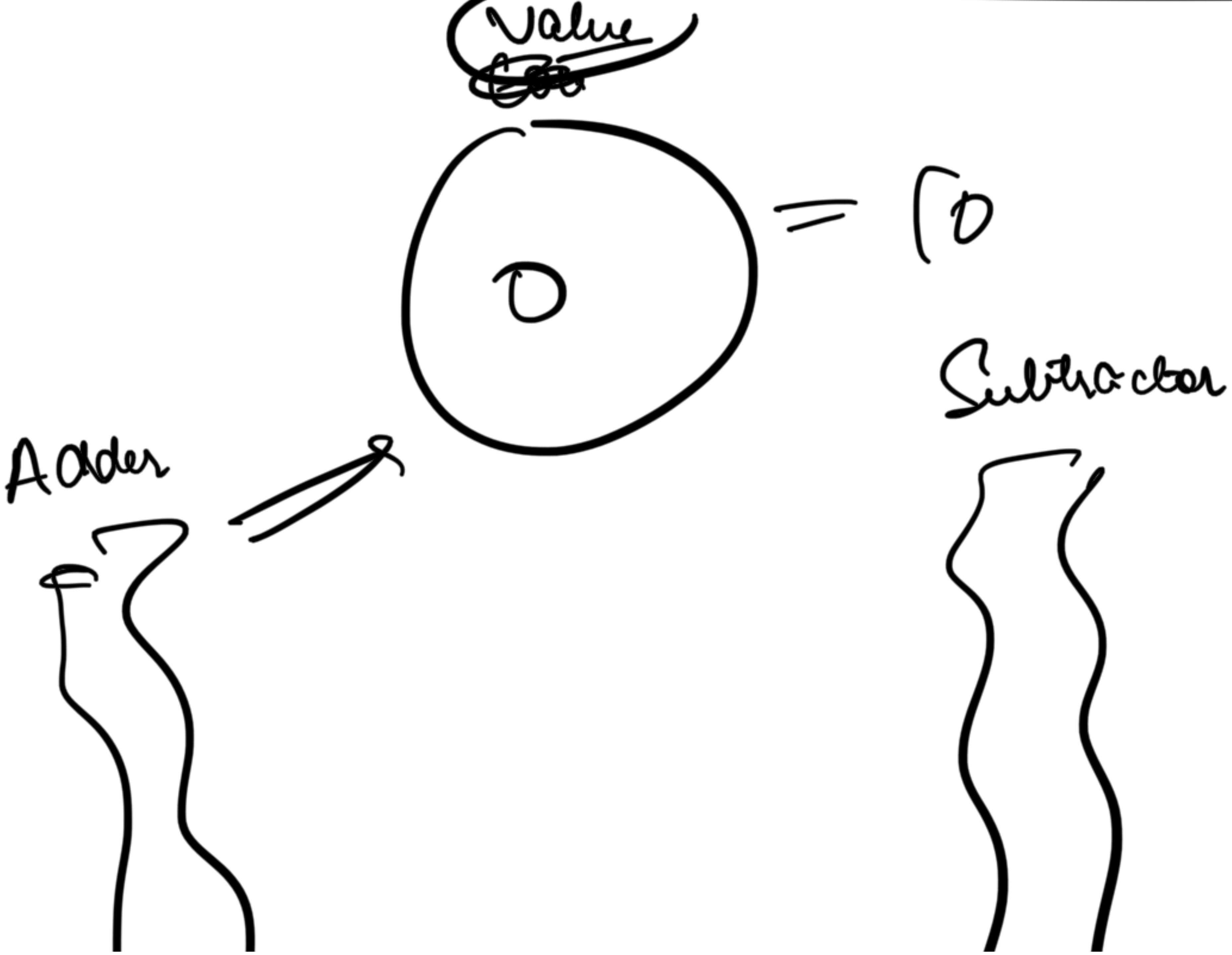
→ Yes

Yes → Check Singer

① ⇒ Check Single Yes

A ② ⇒ Send Rose

Propose



Class {

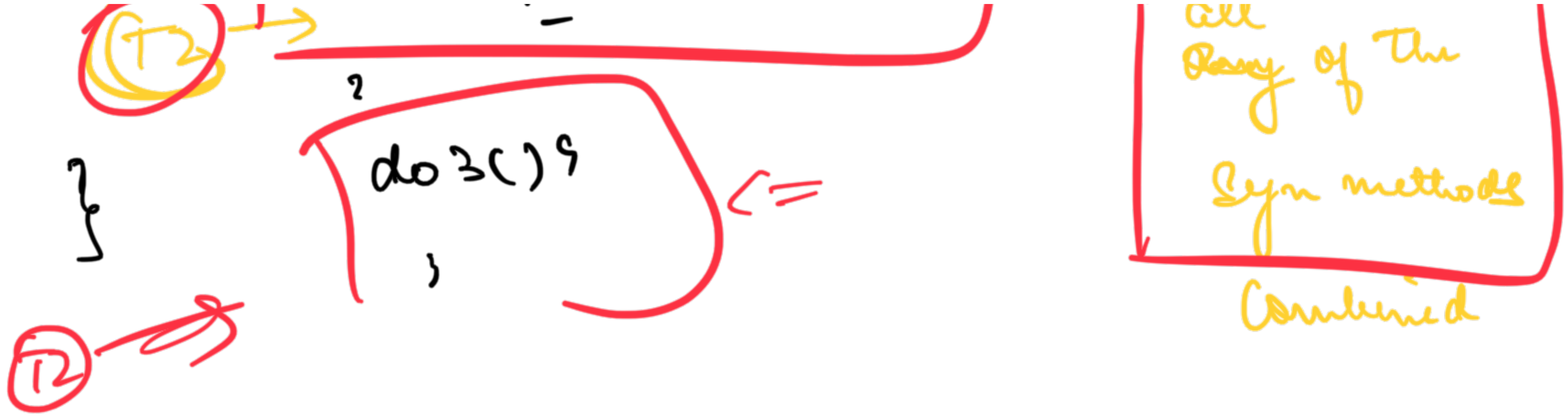
Only one
T1
T2
T

Only on
I
T1

```
Synchronized   do1(){
    ==
}

Synchronized   do2(){
```

For 1 object at 1 time only 1 thread can be there in



Semaphores

If you have sync methods they have to be in the Subject

```
Value {
    get()
    set(i)
}

Adder {
    int cur = value.get
    int new = cur+1
```

value. set (New)

}