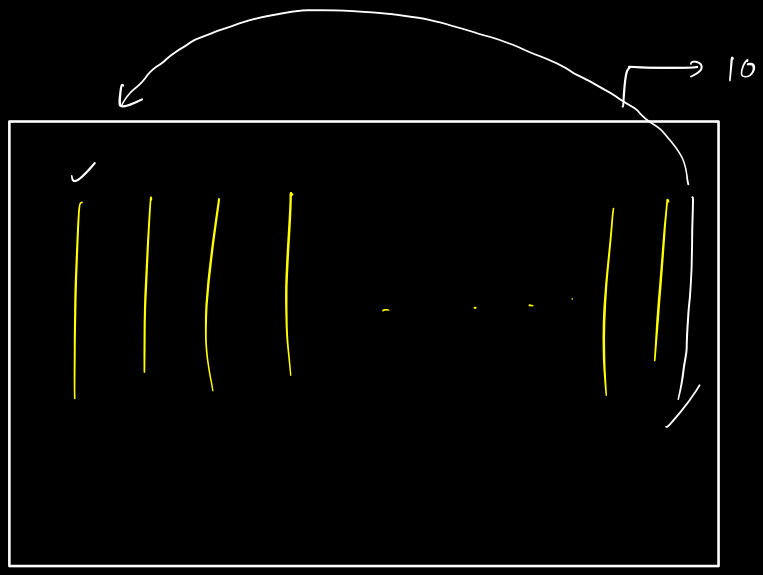
Today's Agenda :-

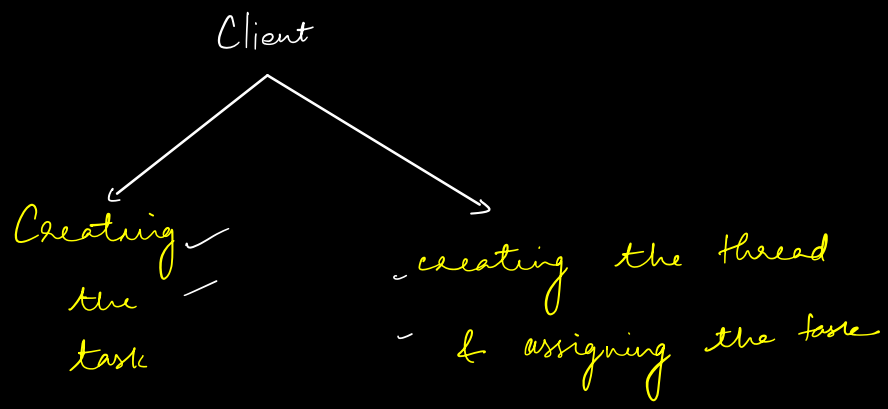
1) Executors

2) Callables

3) Multithreaded Merge Sort

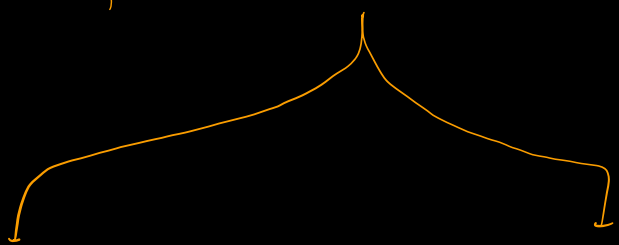
4) Intro to Adder Subtractor Problem.





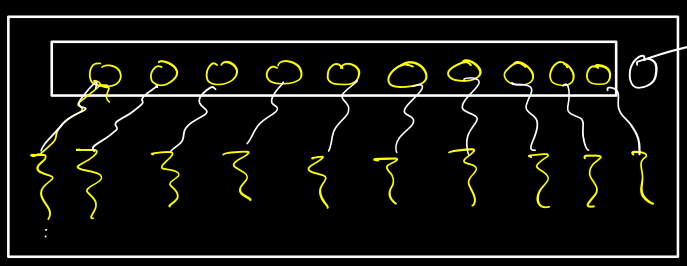
In a multi threaded environment,

Divide the responsibilities into 2 parts :-



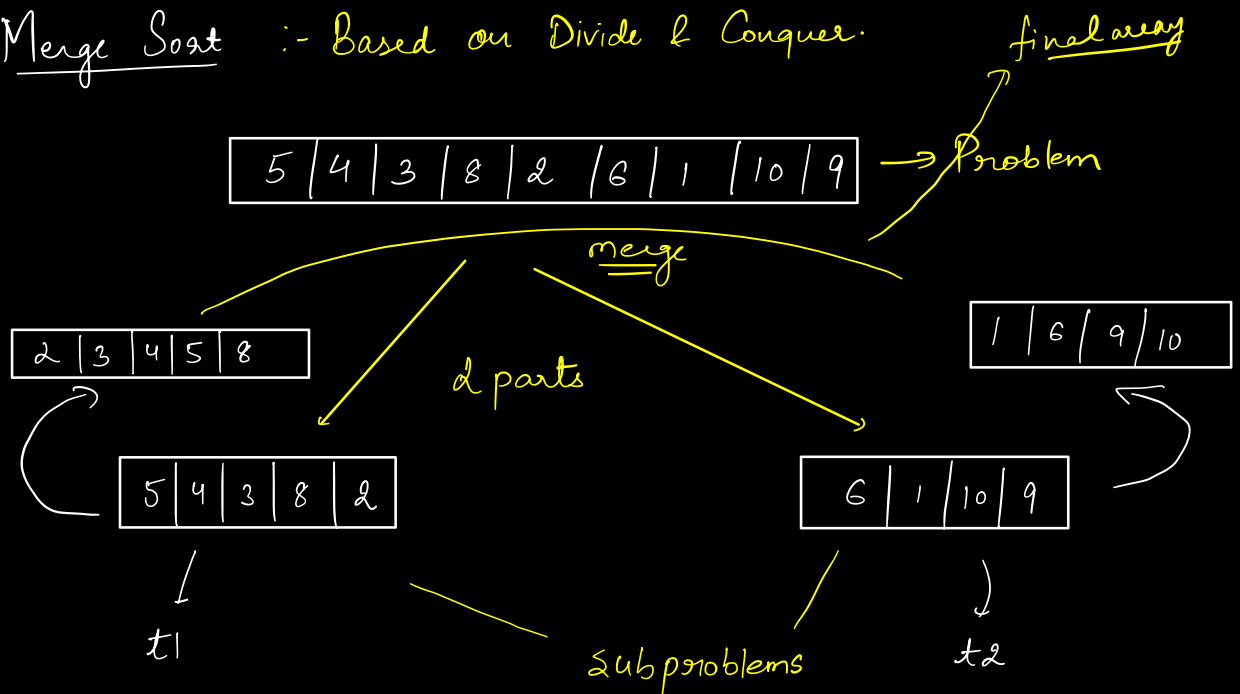
Thread pool (10)

Executor framework



↳ factory

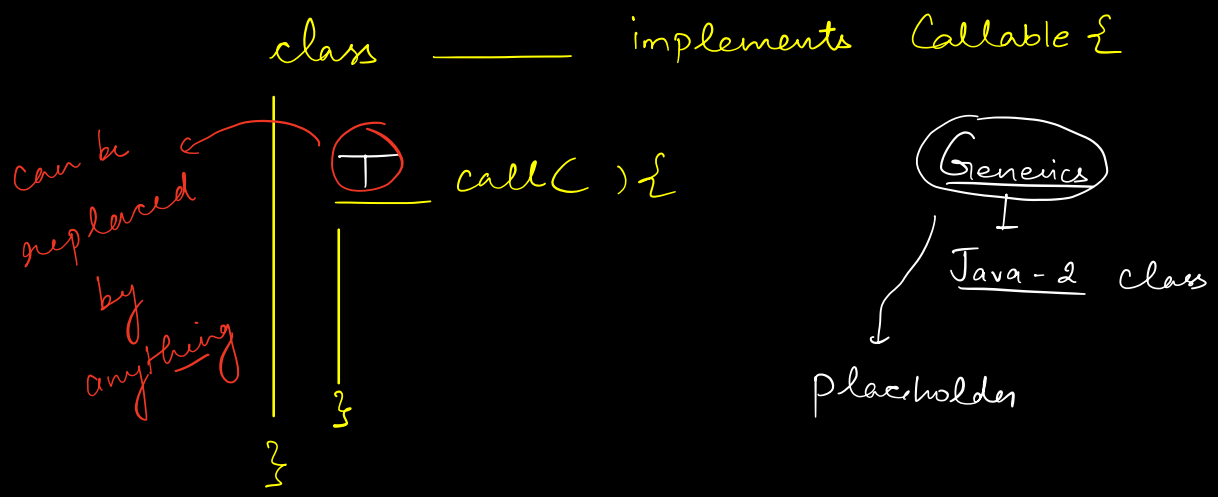
Merge Sort :- Based on Divide & Conquer.

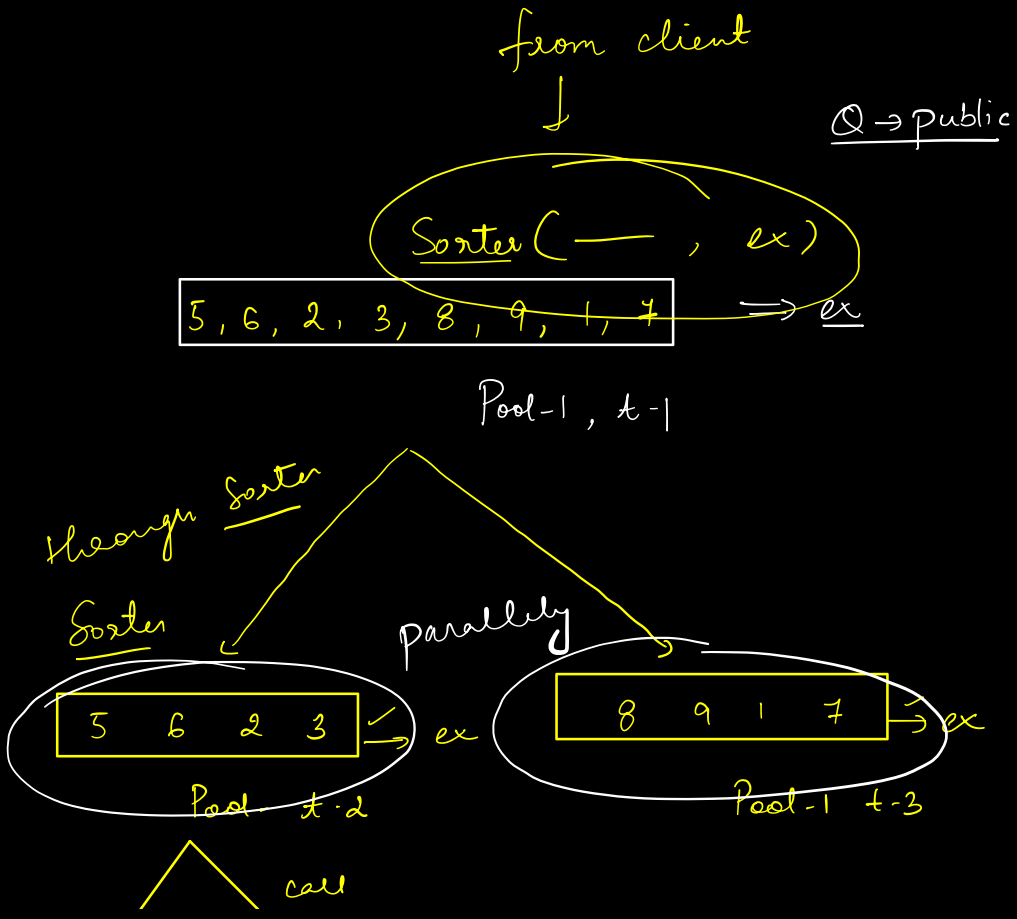


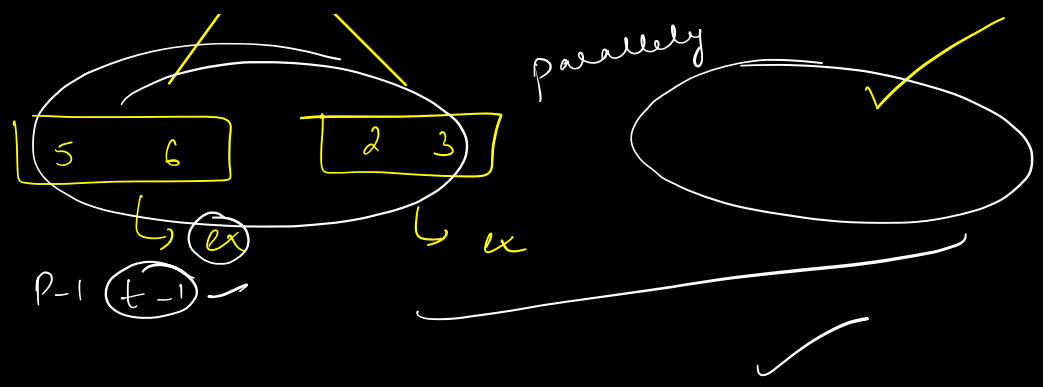
Sorter (which accepts an array, Sorts the array & return the final sorted array)

```
class task implements Runnable {
    void run() {
        // } code you want to run as a part of the task.
    }
}
```

Callable :- Runnable + <Returns some data>



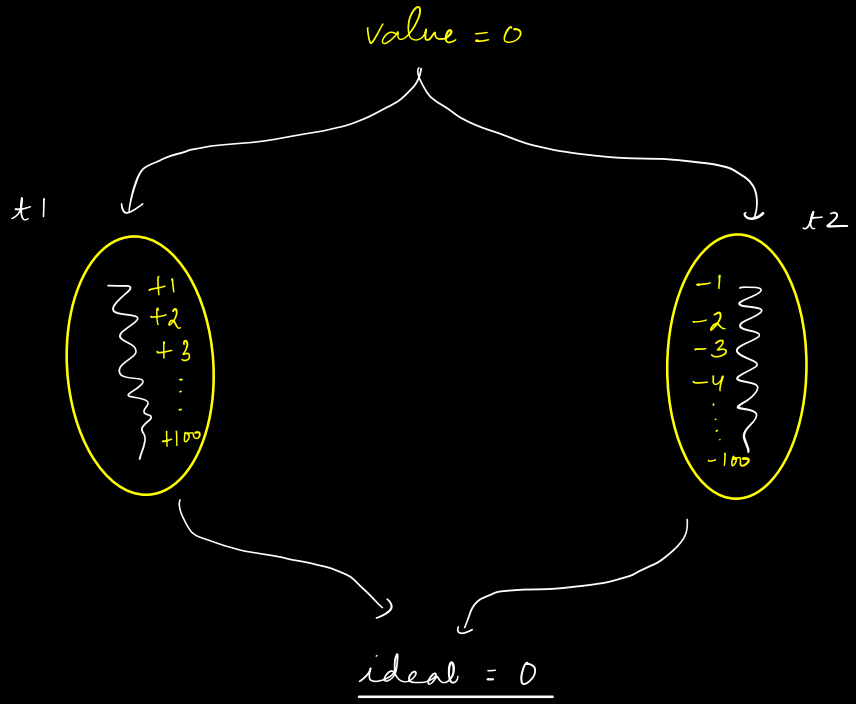




advantage

Synchronisation Problem

Adder Subtractor Problem

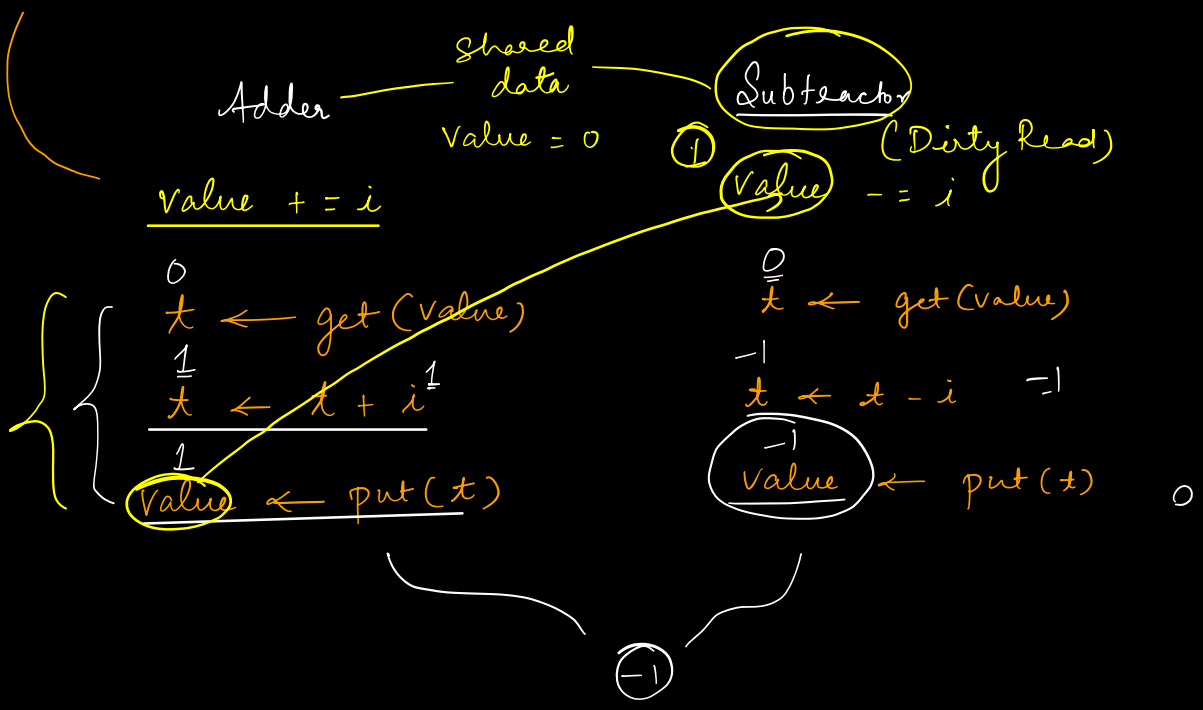


Let's see this in action.

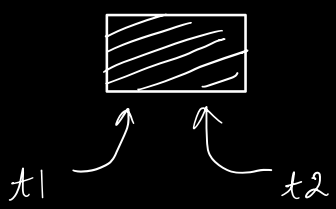
Read Uncommitted

↳ It's not a single operation for computers

↳ in multithreaded env.

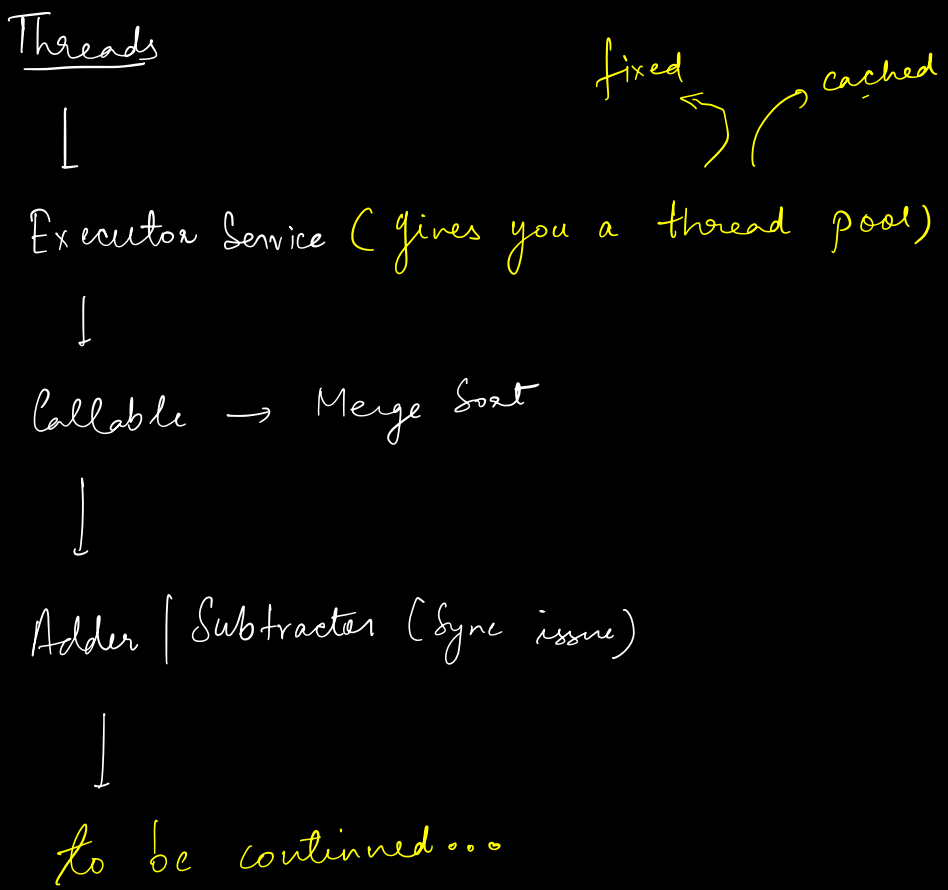


Data Sync issue



To Solve this problem →
- Semaphores
- Mutex / Lock
- Sync Methods
- Sync in Java

Next 2 classes


Threads
fixed
cached
Executor Service (gives you a thread pool)
Callable → Merge Sort
Adder | Subtractor (Sync issue)
to be continued...