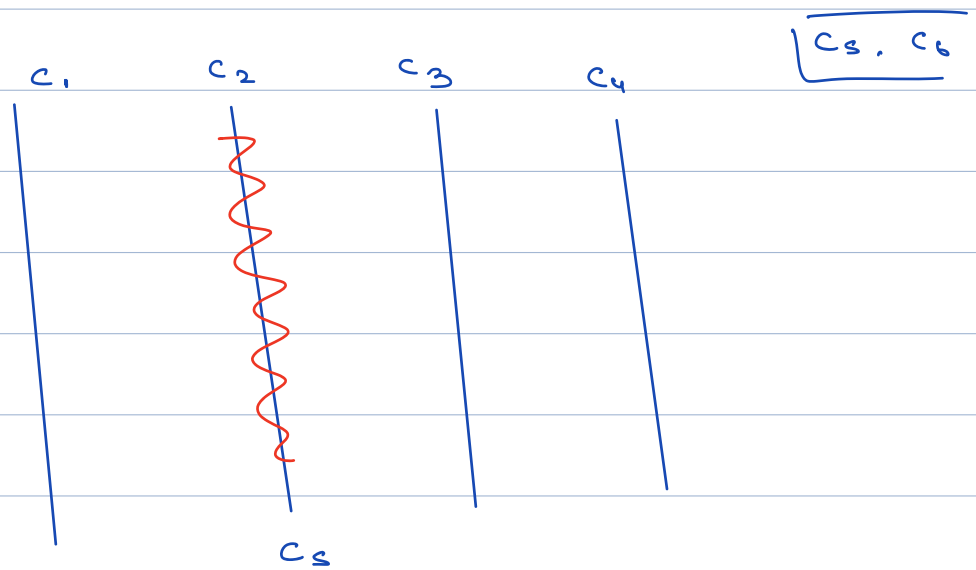
Agenda :-

Executors

Callable

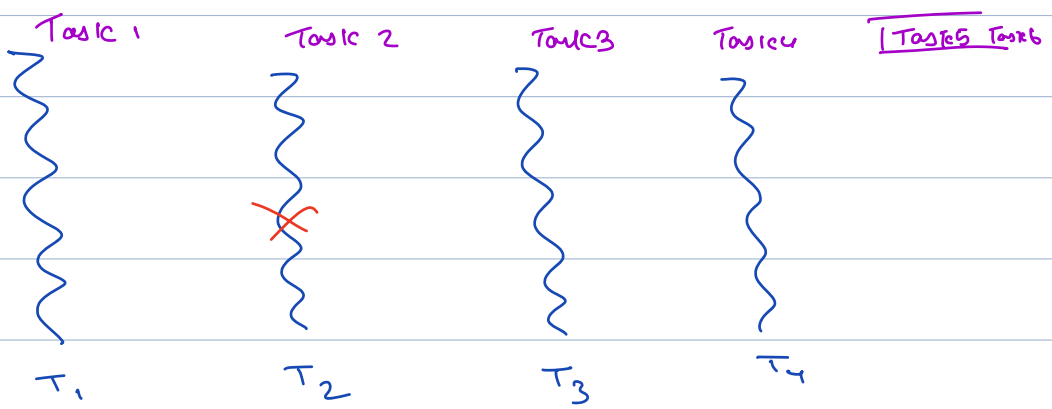
Merge Sort

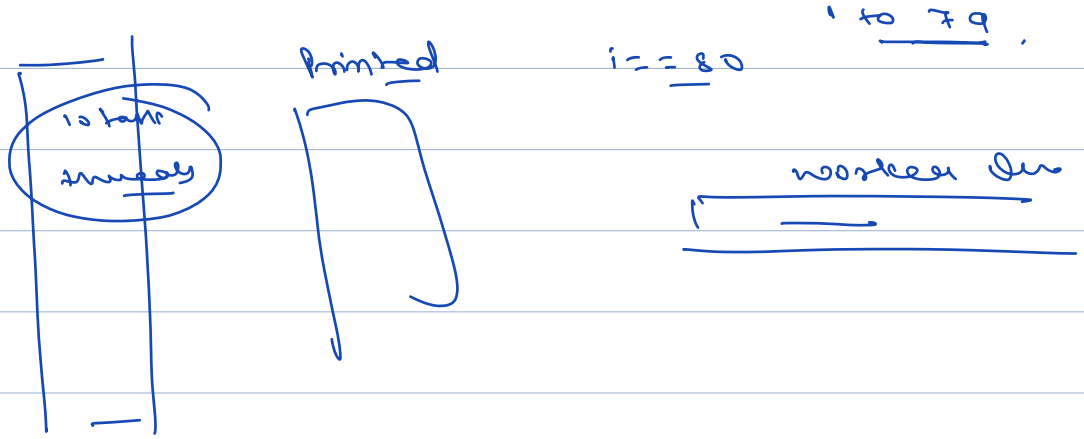
```
    }
                    1000
for(int i=1;i<=1000;i++){
    NumberPrinter numberPrinter = new NumberPrinter(i);
    Thread thread = new Thread(numberPrinter);
    thread.start();
}
```



1 to 10

Thread Pool





```
ExecutorService es = Executors.newFixedThreadPool( nThreads: 10);
for(int i=1;i<=100;i++) {
    if(i==80){
        continue;
    }
    NumberPrinter x1 = new NumberPrinter(i);
    es.execute(x1);
}
```

fixed

↓

no. of Threads fixed.

Todo → 1 Million fixed Threads.

Cache

☺

It creates a new thread if all the existing threads are busy.

Runnable

run method

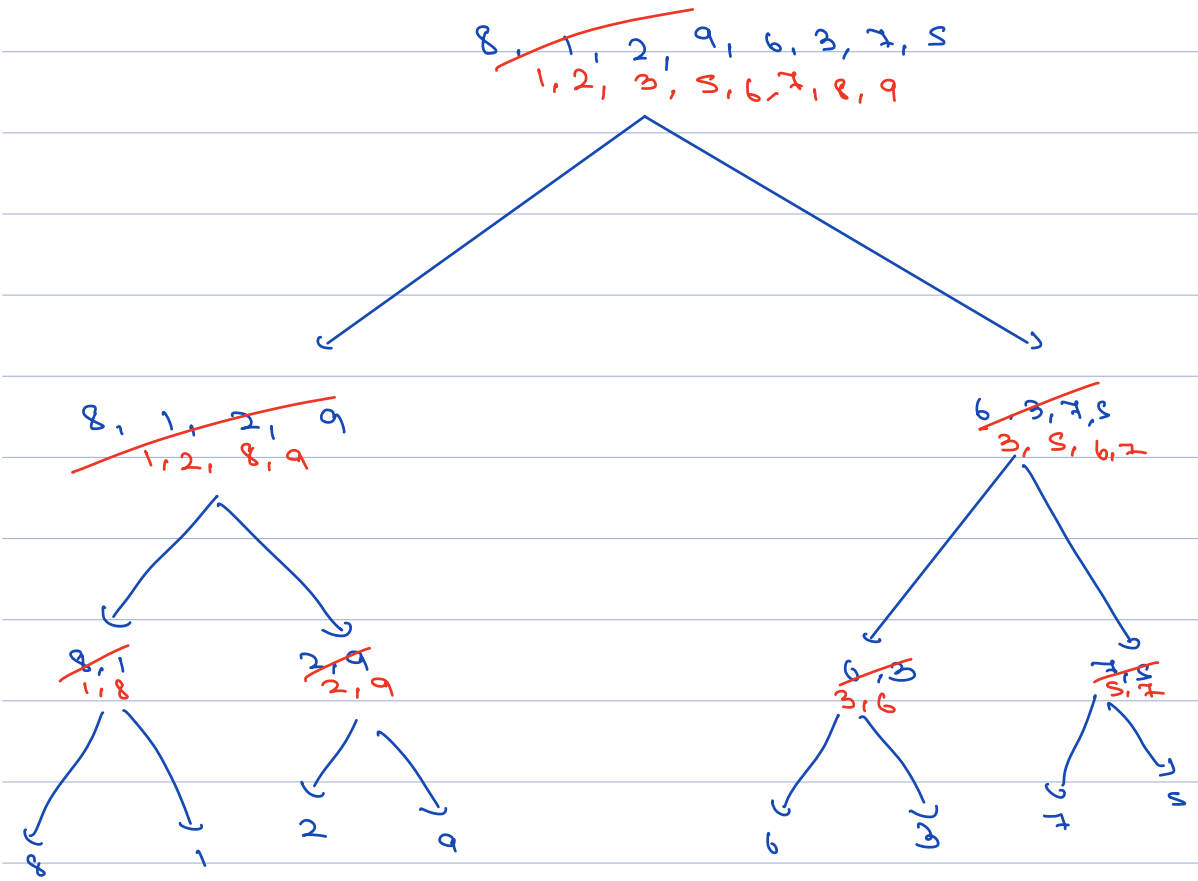
void run();

Can't return anything.

Callable :→ Type T (Generic Type)

T call() → method.

Merge Sort



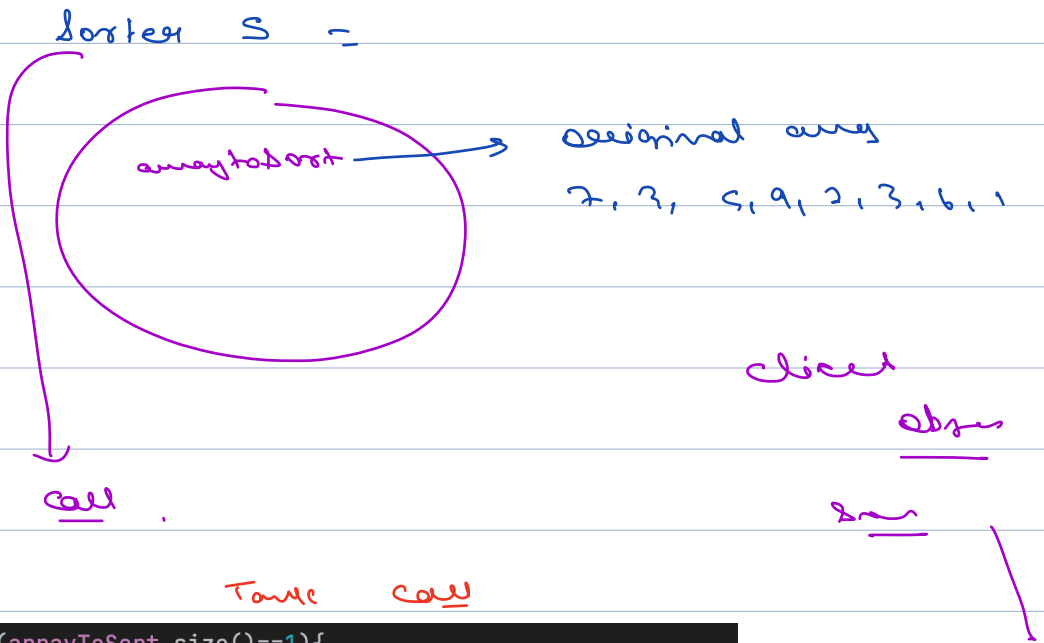
Sorter s = new Sorter(a); → call

Sorter s2 = new Sorter(arr2); → call.

class Sorter {

array to sort

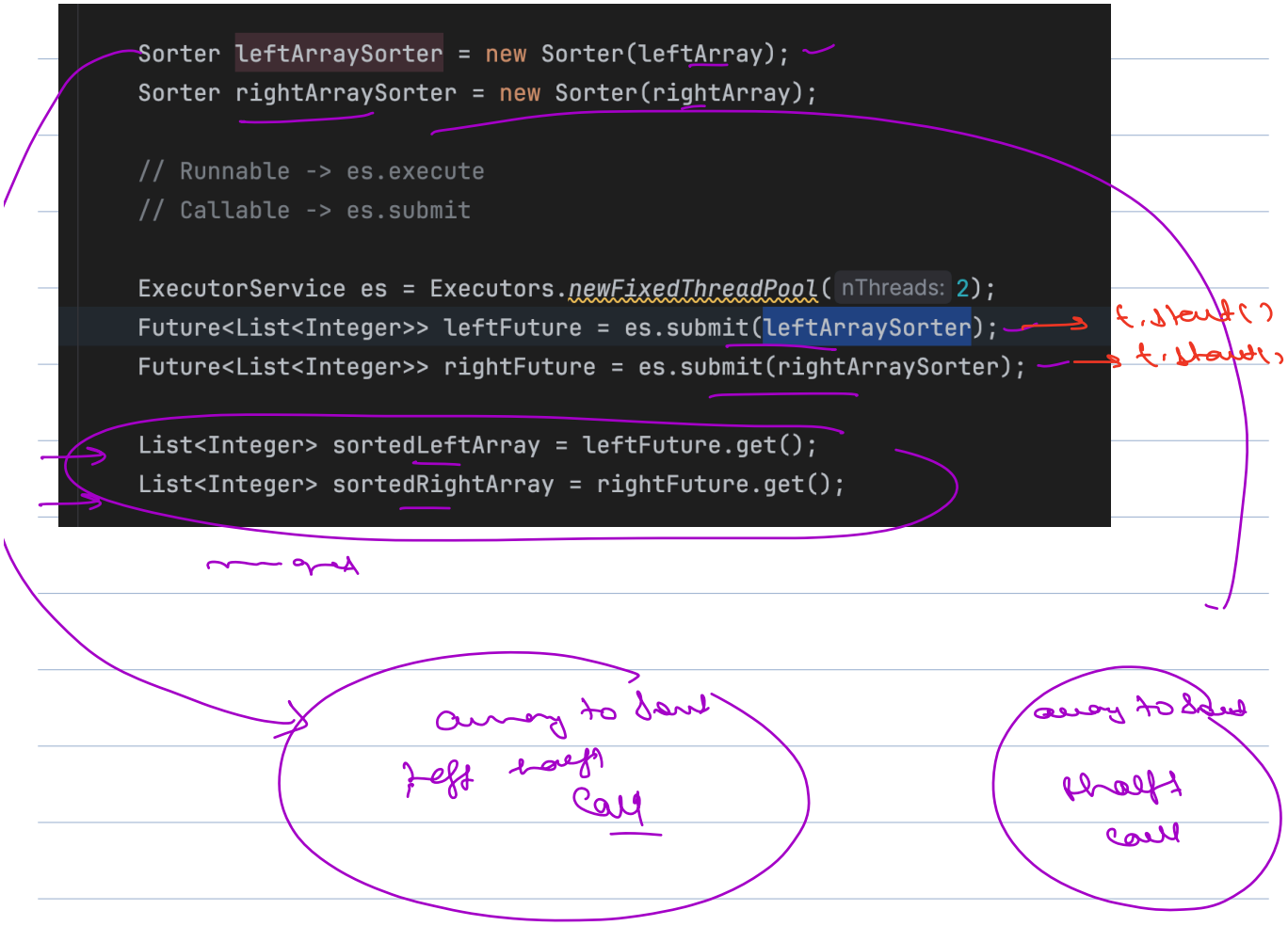
}



Same call

```java
if(arrayToSort.size()==1){
    return arrayToSort;
}

int mid = arrayToSort.size()/2;

List<Integer> leftArray = new ArrayList<>();
List<Integer> rightArray = new ArrayList<>();

for(int i=0;i<mid;i++){
    leftArray.add(arrayToSort.get(i));
}

for(int i=mid+1;i<arrayToSort.size();i++){
    rightArray.add(arrayToSort.get(i));
}
```

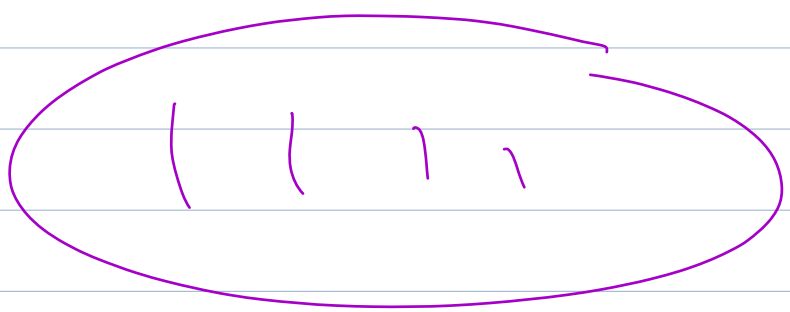
```java
Sorter leftArraySorter = new Sorter(leftArray);
Sorter rightArraySorter = new Sorter(rightArray);

// Runnable -> es.execute
// Callable -> es.submit

ExecutorService es = Executors.newFixedThreadPool( nThreads: 2);
Future<List<Integer>> leftFuture = es.submit(leftArraySorter);
Future<List<Integer>> rightFuture = es.submit(rightArraySorter);

List<Integer> sortedLeftArray = leftFuture.get();
List<Integer> sortedRightArray = rightFuture.get();
```

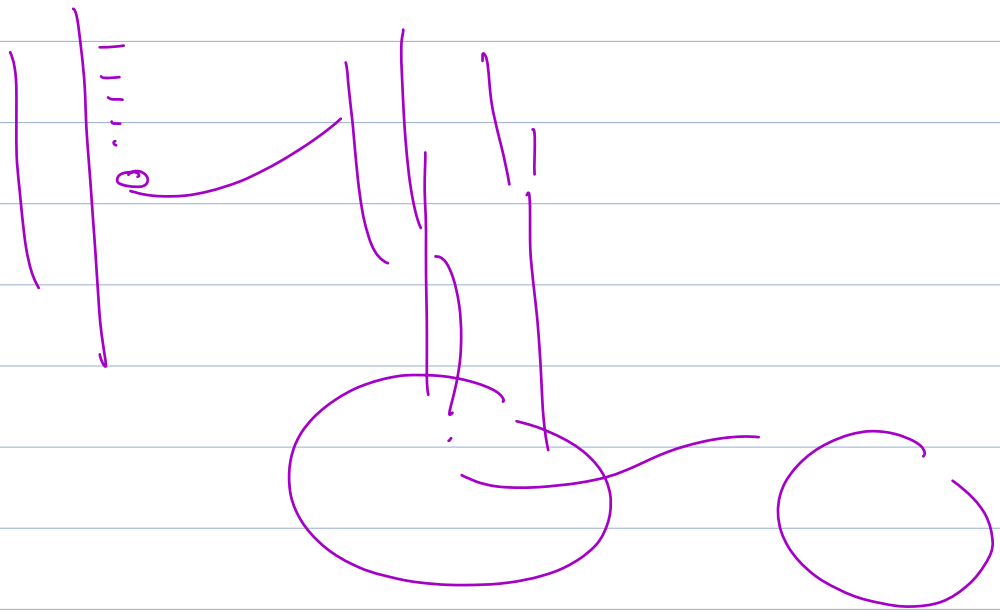


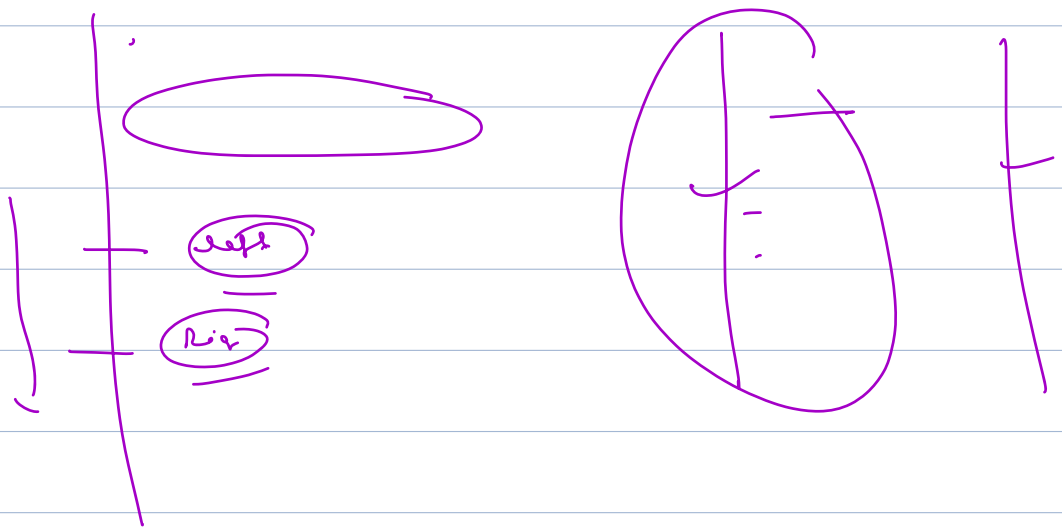

main
left arr
right arr.
left arr
right
left =
right =

Breaks 10:38pm - 10:48pm

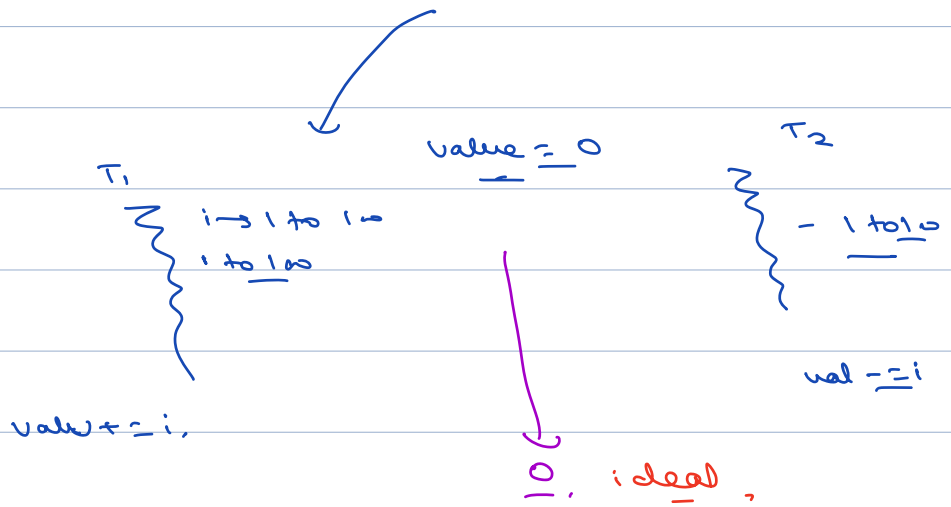
Lecture class

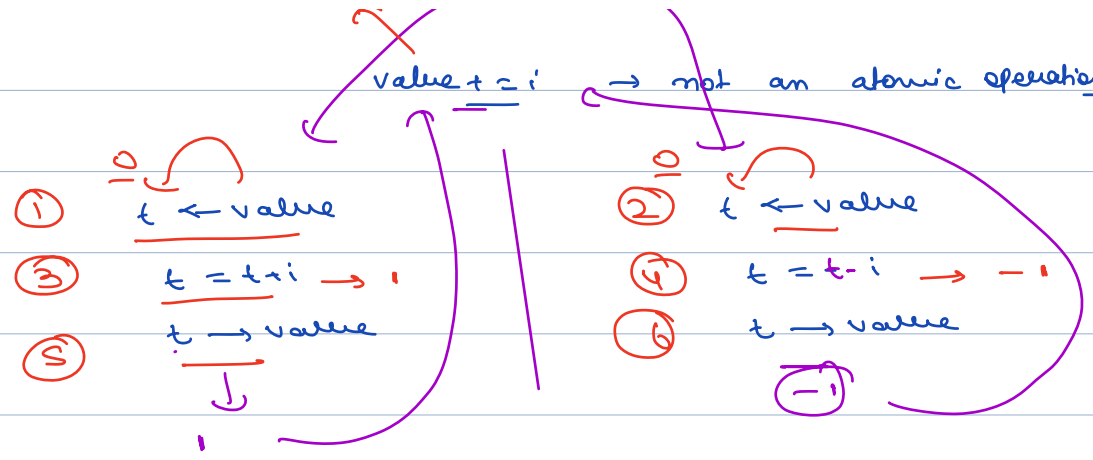




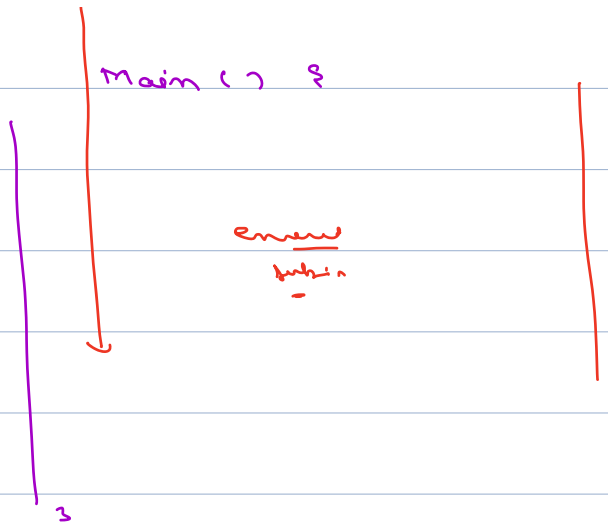


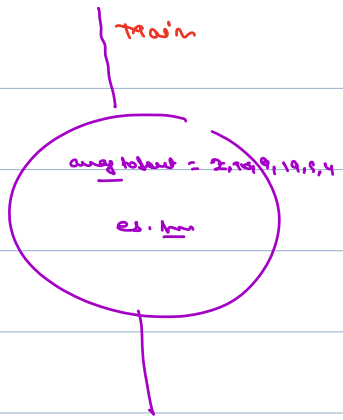
Adder / Subtractor





1. critical section
2. race cond
3. preemptiveness


Main ( ) {
3



```java
public class Client {  aman
    public static void main(String[] args) throws ExecutionException, InterruptedException
        List<Integer> list = List.of(8   , 1 ,2 , 9 , 6 , 3 , 7, 5);
        ExecutorService es= Executors.newCachedThreadPool();

        Sorter sorter = new Sorter(list,es);

        Future<List<Integer>> listFuture = es.submit(sorter);

        List<Integer> ans = listFuture.get();
        System.out.println(ans);
```

```java
    for(int i=mid;i<arrayToSort.size();i++){
        rightArray.add(arrayToSort.get(i));
    }

    Sorter leftArraySorter = new Sorter(leftArray,executor);
    Sorter rightArraySorter = new Sorter(rightArray,executor);

    // Runnable -> es.execute
    // Callable -> es.submit

      ExecutorService es = Executors.newFixedThreadPool(2);
    Future<List<Integer>> leftFuture = executor.submit(leftArraySorter);
    Future<List<Integer>> rightFuture = executor.submit(rightArraySorter);
```