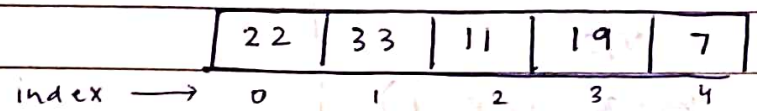
MAX. OF AN ARRAY



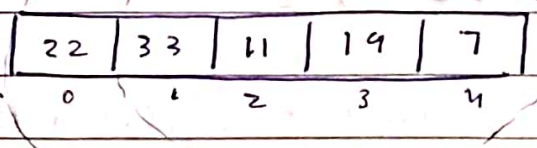
∴ max = 33

Expectation

max (arr, 0) se yeh umeedey hai, ki vo zero (0) se lekar end tak sabko compare krega! {Sabki ladaii krayega} Aur inn sab me se jo sabse bada hai usko find kyke hume return krega!

Faith

max (arr, 1) :→ 1 ke baad sabki ladai krayega! aur inme jo max hai vo hume dega!



Aur inn sabka champion pucha jaye!

→ Toh mai nikal sakta hoon!

→ Agar mujhe inn sabka champion pata chal jaye!

22 bacha hai

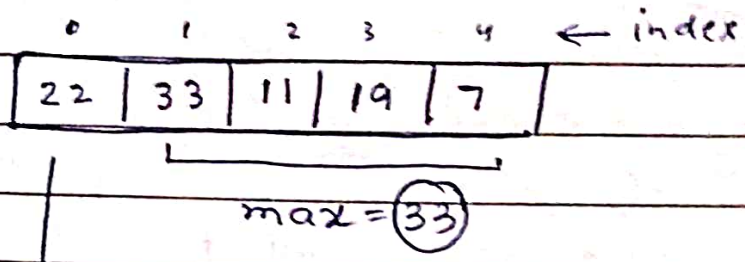
Kese

inn sabka champion hai (33).

aab (33) ko (22) ke sath ladai krni hai .... agar jeeta toh (33) sabka raja mana jayega!

Expectation meet Faith

max (arr, 0) = [arr [0] v/s max (arr, 1)]



→ compare (33) with (22) ∴

```
if (a[0] > 33)
    return a[0];
else
    return 33;
```

```
psvm (s[] a) {

Scanner s = new Scanner (System.in);
int n = s.nextInt();

int [] arr = new int [n];
for (int i = 0; i < arr.length ; i++) {
    arr [i] = s.nextInt();
}

int max = maxOfArray (arr, 0);
System.out.println(max);

}
```

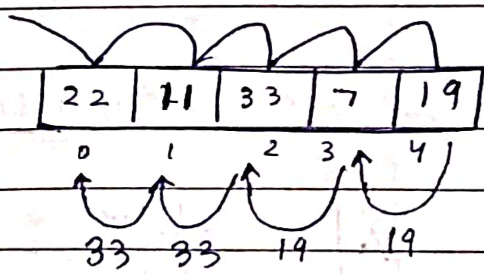
// Yeh (idx) se leke end tak sabki ladi krake
hume (max) dedega!


```
p s in maxOfArray (int[] arr, int idx){
        // maxinsmallerArray                  * BASE CASE
    int misa = maxOfArray (arr, idx+1);        (1)
```

// yeh (idx+1) se leke end tak sabki laddi krake
hume dedega! (max)
// Aab (idx) se leke (idx+1) tak max nikalna
hai!
∴

```
    if ( misa > arr[idx]){
            return misa;
    } else {
            return arr[idx];                   (2)
    }
}
```

DRY RUN



* BASE CASE

```
if (idx == arr.length-1){
    return arr[idx];
}
```

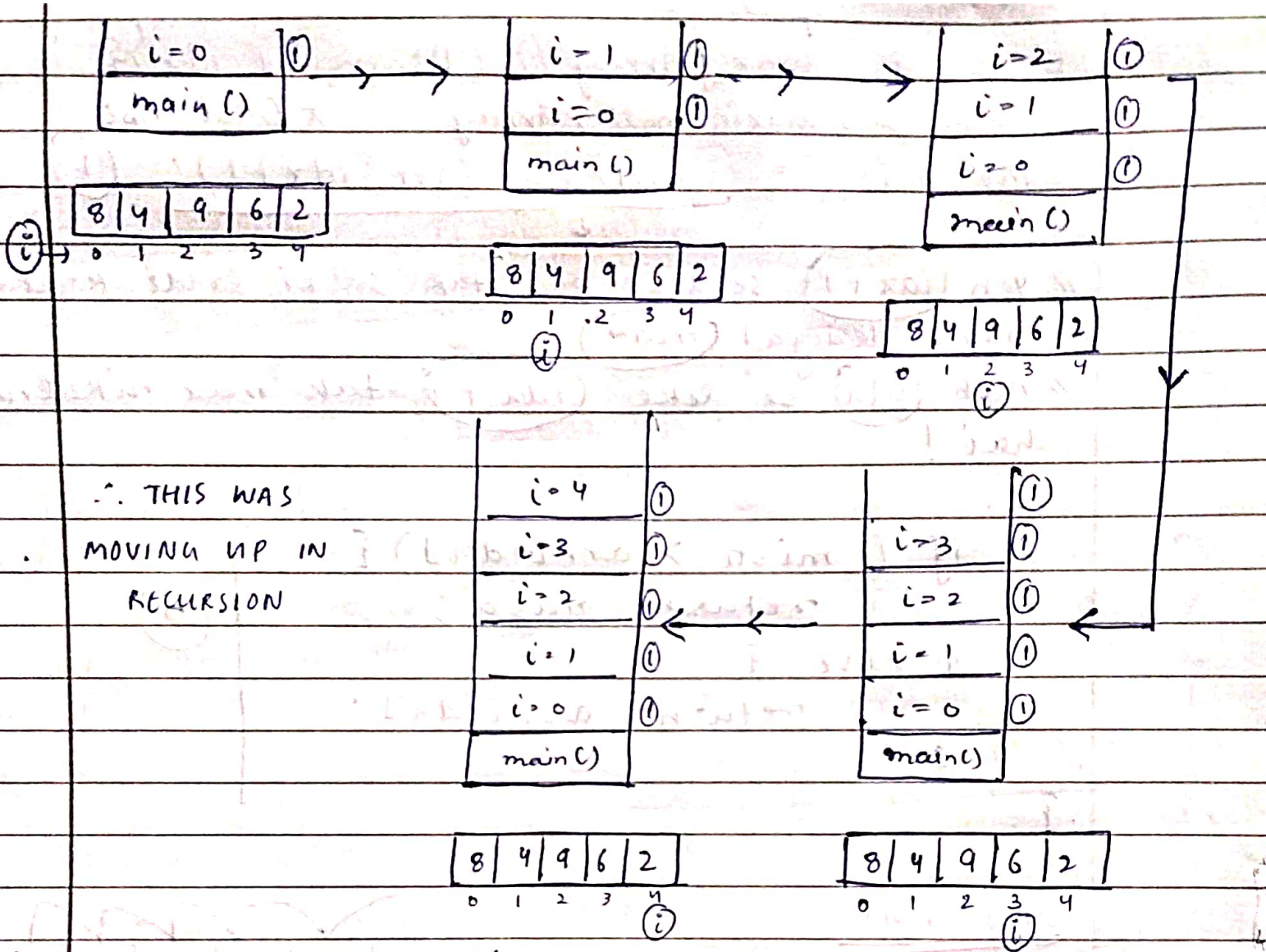
	arr	idx	
	4k	4	19
	4k	3	(1)(2) ∴ 17<19
	4k	2	(1)(2) 19 ∴ 33>19
	4k	1	(1)(2) 33 ~~11<33~~
mofarr	4k	0	(1)(2) 33

∴ 22<33

(33)

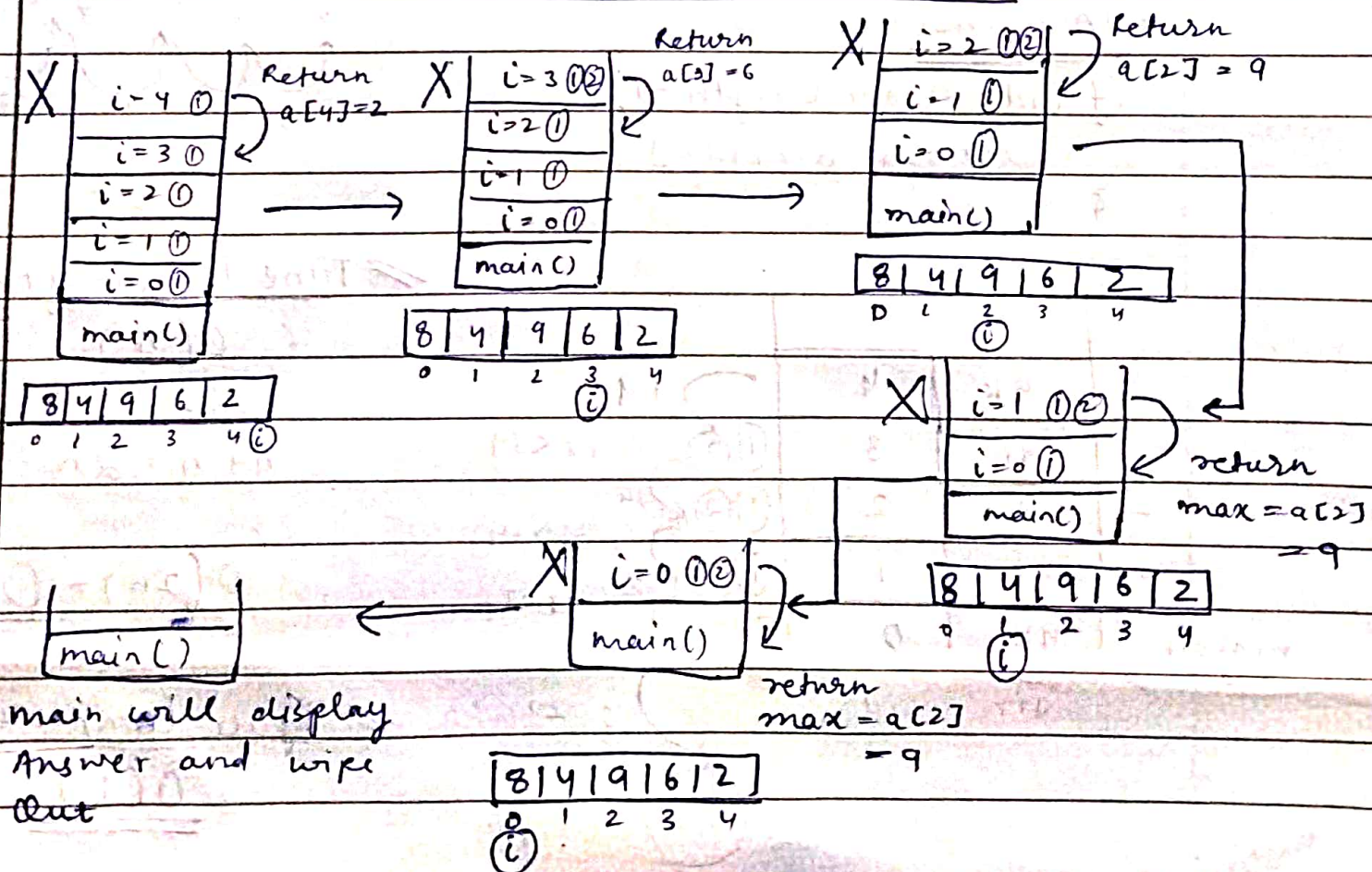
// Time Complexity
∴ O(n)
∴
n + n = 2n
∴
O(2n) = O(n)

// Space Complexity
∴ O(1)



∴ THIS WAS MOVING UP IN RECURSION

∴ NOW MOVING DOWN / COMING OUT OF RECURSION



main will display Answer and wipe out