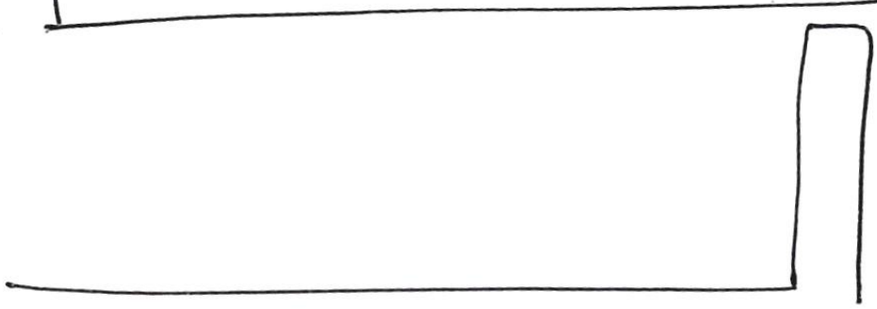
3 cases of

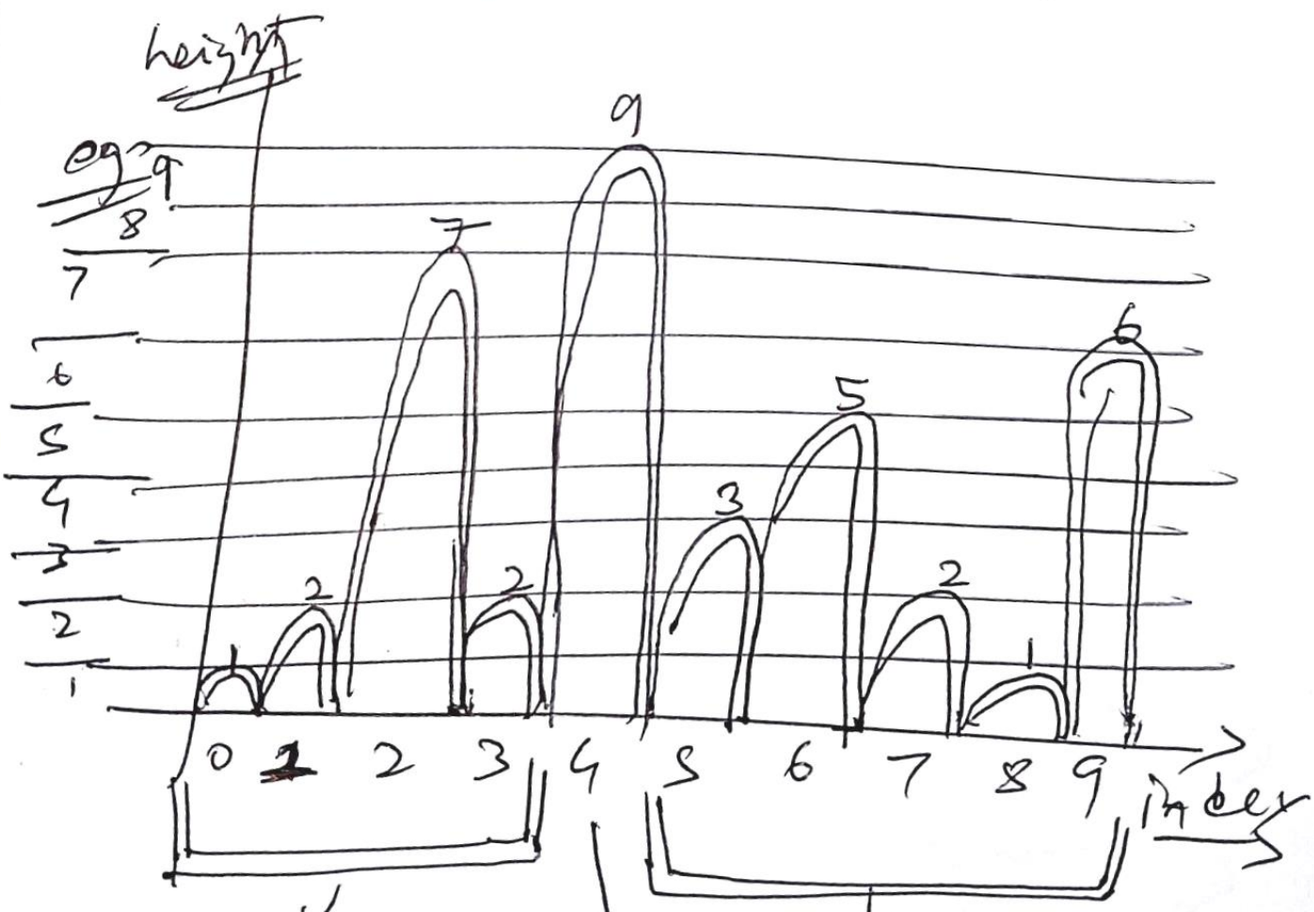
Max arr

(i)

(ii)

(iii)





arr2 (left arr)

=> malloc 4 units

j = max of arr = 4

arr3 (right arr)

malloc n-j-1 = 10-4-1 = 5 units

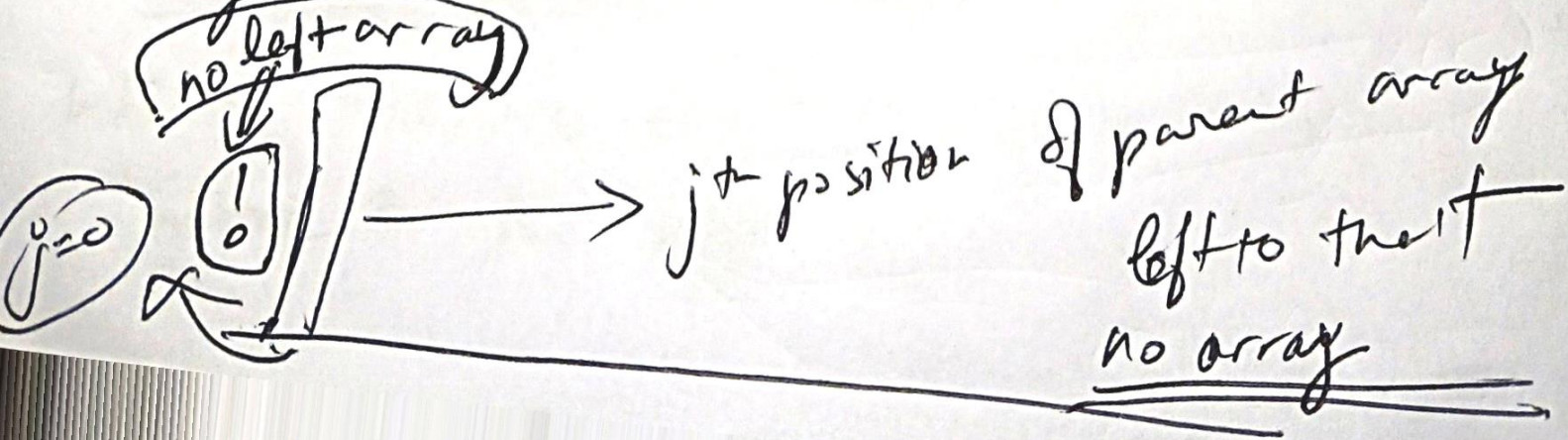
side side max left function is called

↳ to get left array max.

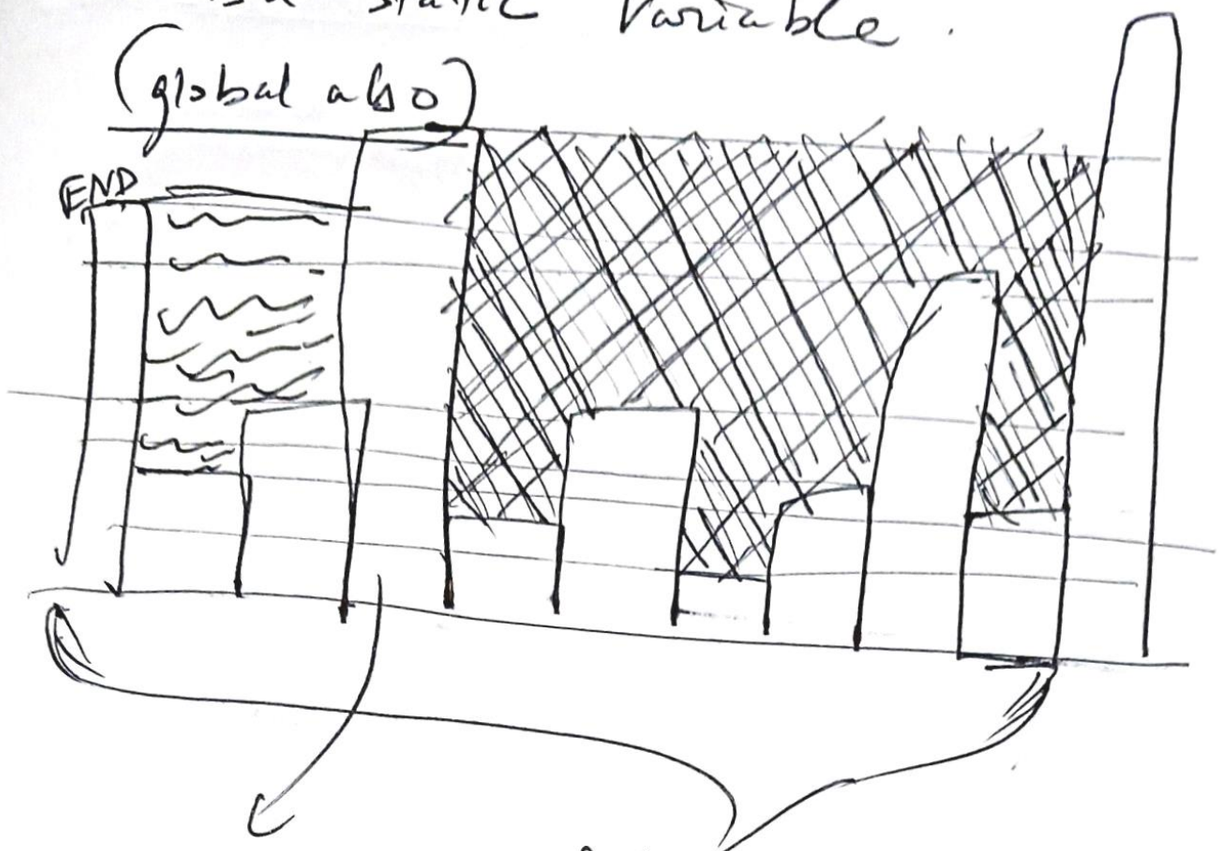
in which j is passed to max of arr function

line (12) to (15) (refer)

↳ here if j==0 ; the left array doesn't exist as =>



Sum is a static variable.
(global also)



line 33
index = k

leftarray

trapped rain water

b/w ← max leftarr (&) max of parent array

line 38 sum = adding the  regions

as shown in drawing

next checking if k>0, i.e,

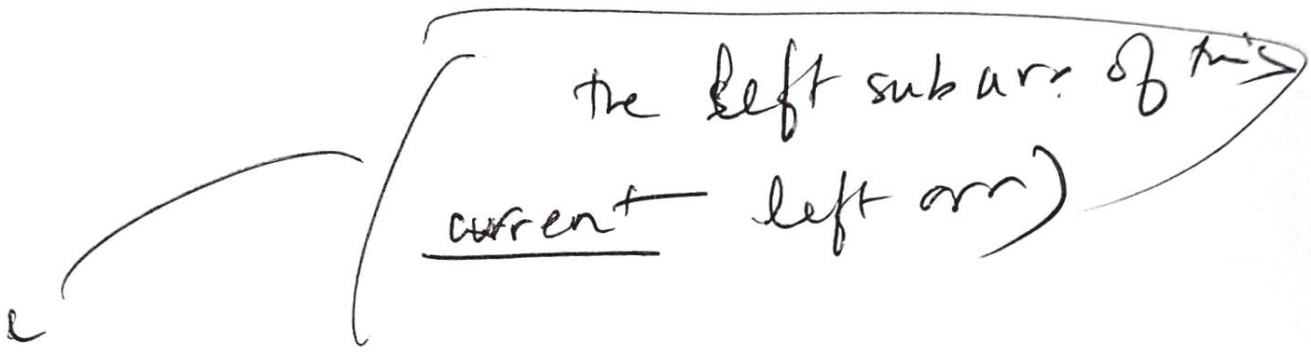
max leftarr not in the END

which means water can be trapped more in the  left too

so recursively called

side side left max

arguments → ~~arguments~~ (this current leftarr , the curent left arr max ,



also malloced

---

for right array : same thing applies but in right direction

=> and the final rainwater is added in main() function

(left + right) ~~water~~ water