

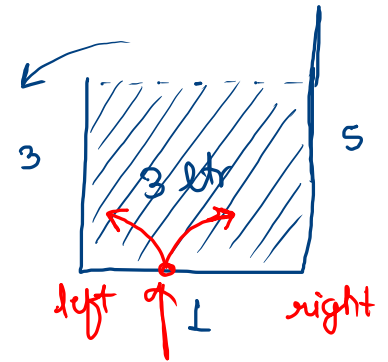
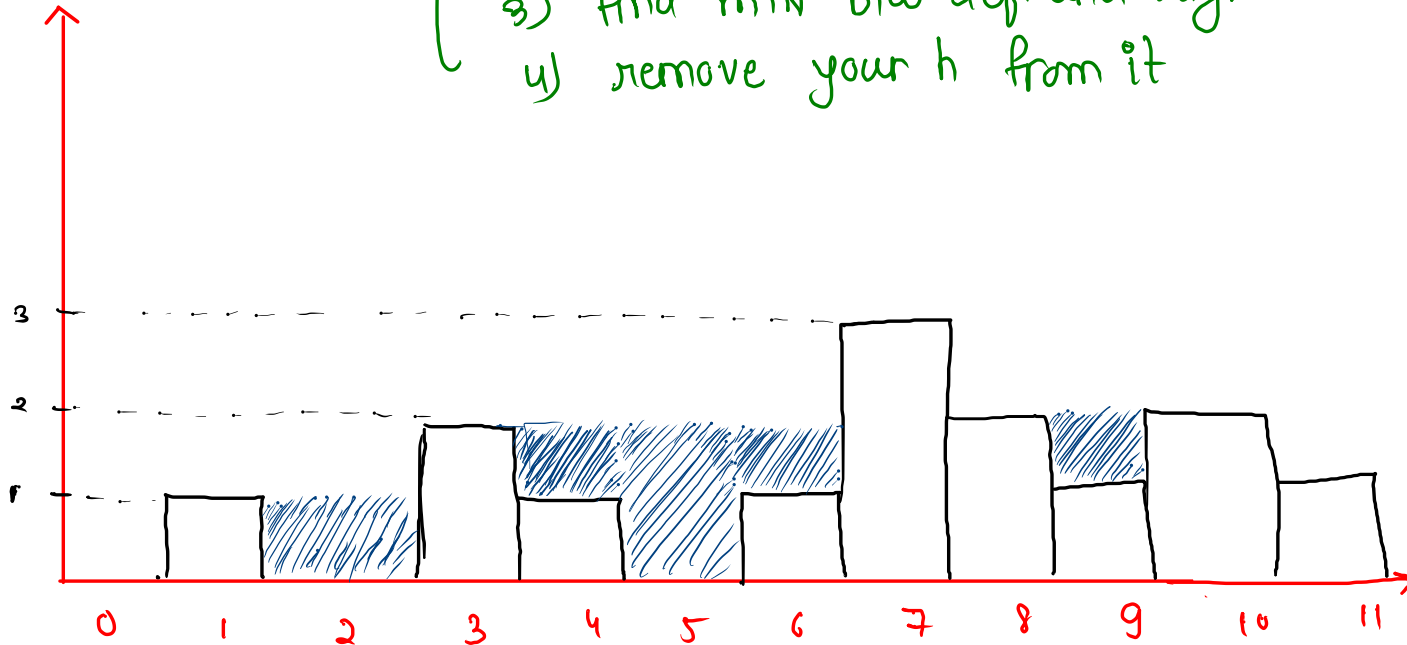
height

0 1 2 3 4 5 6 7 8 9 10 11
[0,1,0,2,1,0,1,3,2,1,2,1]

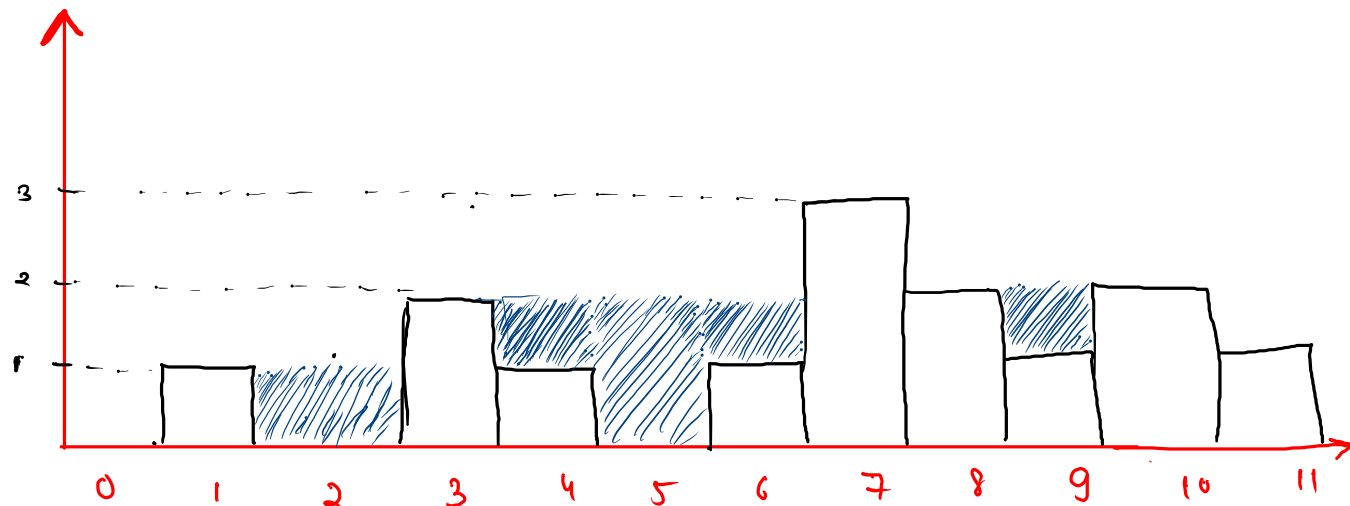
pseudo code:-

- 1) find max h on left
- 2) find max h on right
- 3) find min b/w left and right
- 4) remove your h from it

logic:- we will calc.
water stored on
top of each block



for each index, i will find maximum height on my left
and on my right as well



(h)

$$i=0, l=0, r=3, \min=0-0=0$$

$$i=1, l=1, r=3, \min=1-1=0$$

$$i=2, l=1, r=3, \min=1-0=1$$

$$i=3, l=2, r=3, \min=2-2=0$$

$$i=4, l=2, r=3, \min=2-1=1$$

$$i=5, l=2, r=3, \min=2-0=2$$

$$i=6, l=2, r=3, \min=2-1=1$$

ans = ~~0~~ ~~0~~ ~~0~~ ~~1~~ ~~1~~ ~~2~~ ~~4~~ ~~5~~ ~~6~~

$$i=7, l=3, r=3, \min=3-3=0$$

$$i=8, l=3, r=2, \min=2-2=0$$

$$i=9, l=3, r=2, \min=2-1=1$$

$$i=10, l=3, r=2, \min=2-2=0$$

$$i=11, l=3, r=1, \min=1-1=0$$

Same code

solve(a, b, c)

solve(int a, int b, int c)

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] height = new int[n];
    for (int i = 0; i < n; i++) {
        height[i] = scn.nextInt();
    }

    System.out.println(solve(height));
}
```

```
public static int solve(int[] height) {
    int result = 0;
    for (int i = 0; i < height.length; i++) {
        int left = height[i];
        for (int j = 0; j < i; j++) {
            if (height[j] > left) {
                left = height[j];
            }
        }

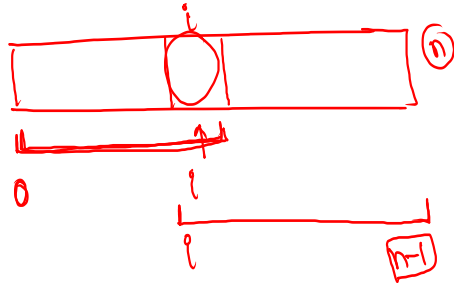
        int right = height[i];
        for (int j = i + 1; j < height.length; j++) {
            if (height[j] > right) {
                right = height[j];
            }
        }

        int mini = Math.min(left, right);
        int ans = mini - height[i];

        result += ans;
    }
    return result;
}
```

left maximum height

right maximum height



```
public static int solve(int[] height) {
    int result = 0;
    for (int i = 0; i < height.length; i++) {
        int left = Integer.MIN_VALUE;
        for (int j = 0; j <= i; j++) {
            if (height[j] > left) {
                left = height[j];
            }
        }

        int right = Integer.MIN_VALUE;
        for (int j = i; j < height.length; j++) {
            if (height[j] > right) {
                right = height[j];
            }
        }

        int mini = Math.min(left, right);
        int ans = mini - height[i];

        result += ans;
    }
    return result;
}
```

armstrong no.

↙ ↙ ↙
n = 153

digits = 3

$$\Rightarrow \underline{\underline{1^3 + 5^3 + 3^3}} = \textcircled{153}$$