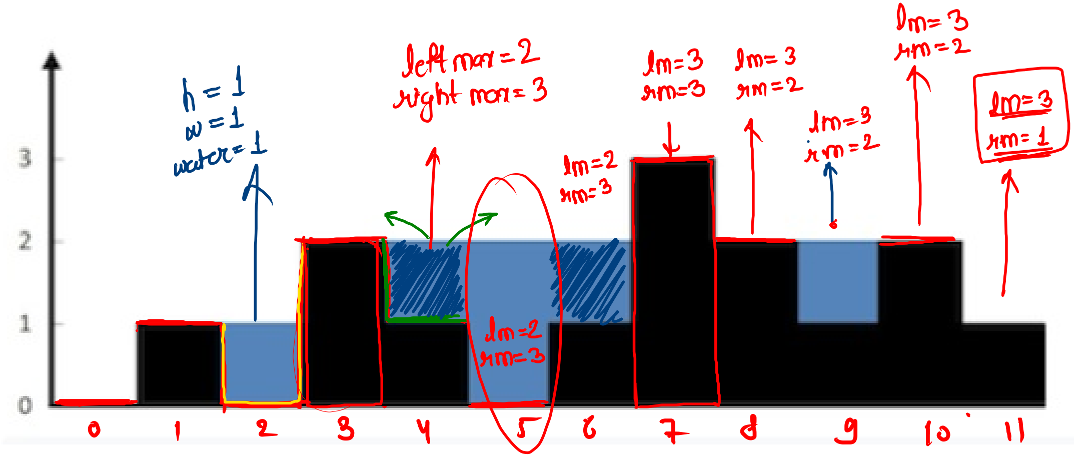
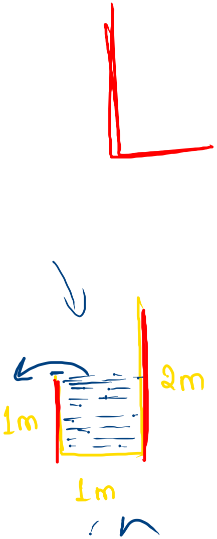
Store Maximum

(Trapping rain water)





arr =

0	1	0	2	1	0	1	3	2	1	2	1

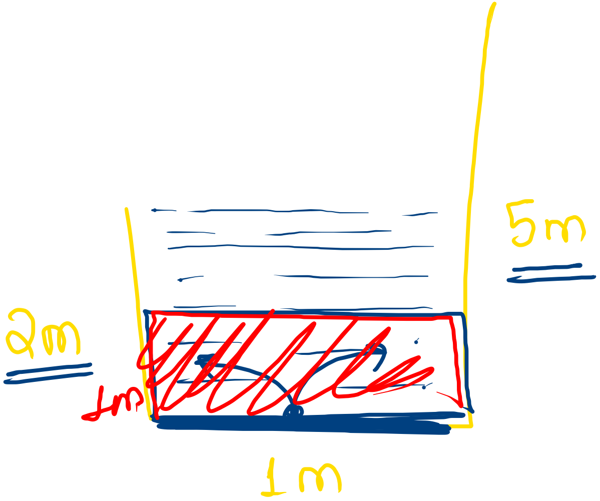
Note :- for each index, we need to find max. height wall on left side and max. height wall on right side.

formula :-

left max :- max. h on left side

right max :- max. h on right side

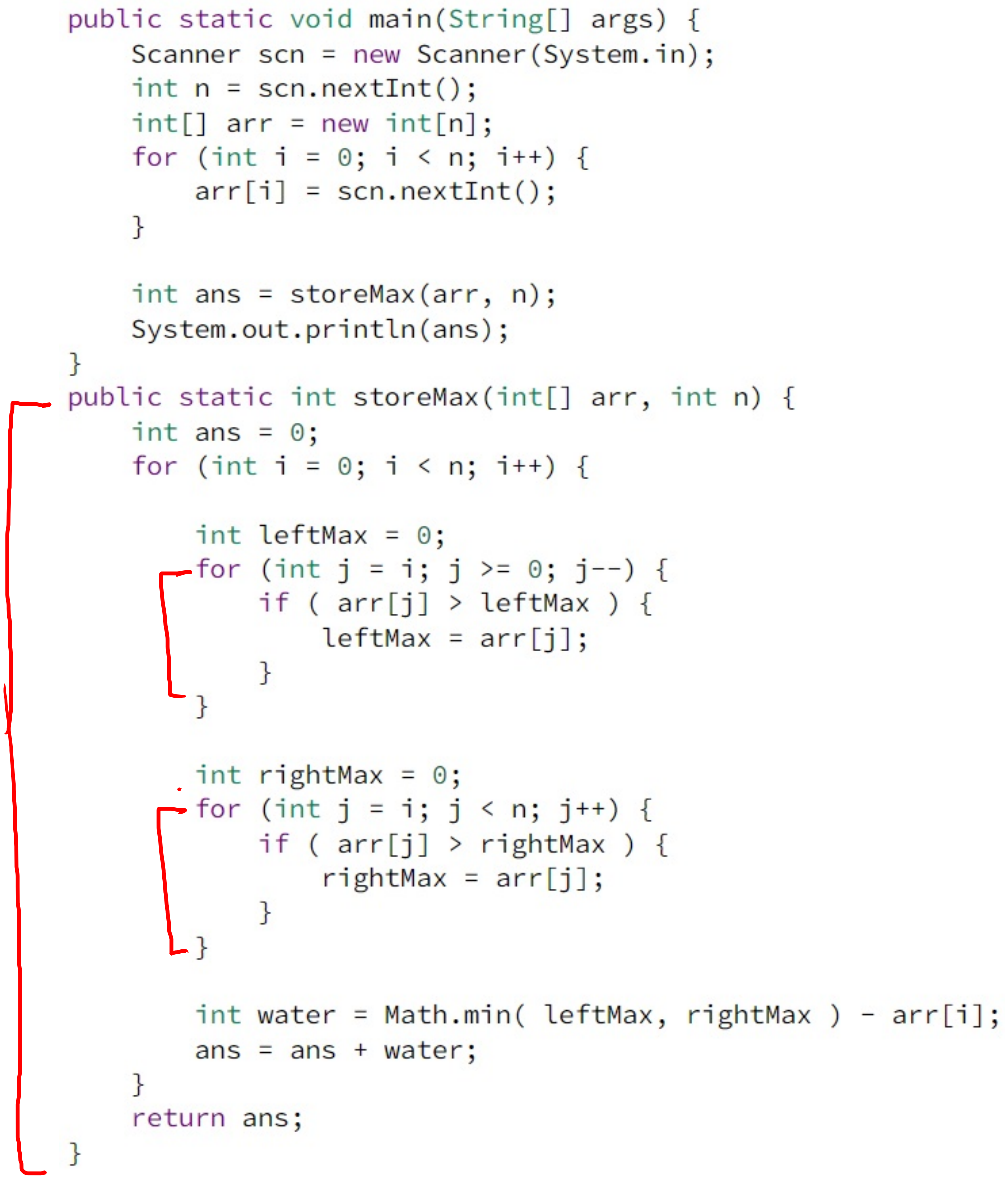
ans = min (leftmax, rightmax) - arr[i]



$2 - 1 = \boxed{1}$

psudo code

```
1) for each index i
    1.1.) traverse from i to 0
          and find left maximum
    1.2.) traverse from i to n-1
          and find right maximum
    1.3) water = min (leftmax, rightmax) - arr[i]
         ans += water
```



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

    int ans = storeMax(arr, n);
    System.out.println(ans);
}
public static int storeMax(int[] arr, int n) {
    int ans = 0;
    for (int i = 0; i < n; i++) {

        int leftMax = 0;
        for (int j = i; j >= 0; j--) {
            if ( arr[j] > leftMax ) {
                leftMax = arr[j];
            }
        }

        int rightMax = 0;
        for (int j = i; j < n; j++) {
            if ( arr[j] > rightMax ) {
                rightMax = arr[j];
            }
        }

        int water = Math.min( leftMax, rightMax ) - arr[i];
        ans = ans + water;
    }
    return ans;
}
```

⇒ Time Complexity (Time consumed by program to execute)

M.Imp :- TC can only be calculated using **no. of operation performed.**

ex :-

```
main() {
        Syso("Hello");   // 1 operation
        Syso("Hi");
}
```

Capital O

constant

$T.C = O(2)$

$\simeq \underline{O(1)}$

$O(200)$

$\simeq O(1)$

1, 5, 5000, 5M

```
main() {
    int n = scn.nextInt();
    for(int i=0; i<n; i++) {
        // statement with 1 operation
        syso("Hello");
    }
}
```

// N operations

T.C = O(N) → linear

where N is given input

ex :-

n = 1 ⟶ 1

n = 10 ⟶ 10

n = 1000 ⟶ 1000

n = 5000 ⟶ 5000

$$\boxed{T.C \propto n}$$

T.C = O(n)

Types of operations

- linear
- quadratic
- cubic
- logerethemic
- constant
- ⋮

Input	output operation
n	n
n	n^2
n	n^3
n	$\log(n)$
n	1

Ex:-

n = 5, m = 6

```
n = scn.nextInt();
m = scn.nextInt();
for(int i=0; i<n; i++){
    for(int j=0; j<m; j++){
        Syso("Hi");  ← 30
    }
}
```

T.C = O(m*n)

T.C = O(N^2)

where N is input