Trapping Rain Water

Difficulty: **Hard** Accuracy: **33.14%** Submissions: **462K+** Points: **8**

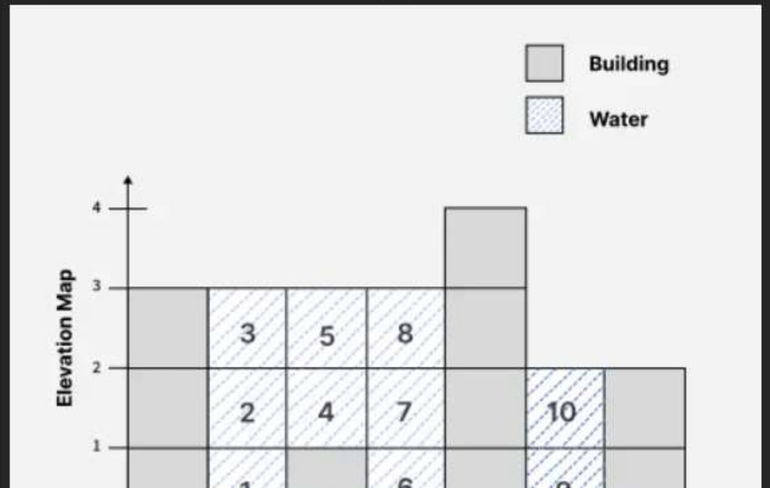
Given an array **arr[]** with non-negative integers representing the height of blocks. If the width of each block is 1, compute how much water can be trapped between the blocks during the rainy season.

Examples:

Input: arr[] = [3, 0, 1, 0, 4, 0 2]

Output: 10

Explanation: Total water trapped = 0 + 3 + 2 + 3 + 0 + 2 + 0 = 10 units.



Java (1.8) | Average Time: 20m | Start Timer

```java
// } Driver Code Ends
class Solution {
    public int maxWater(int arr[]) {
        int n = arr.length;
        if (n <= 1) return 0;
        int[] leftMax = new int[n];
        int[] rightMax = new int[n];
        leftMax[0] = arr[0];
        for (int i = 1; i < n; i++) {
            leftMax[i] = Math.max(leftMax[i - 1], arr[i]);
        }
        rightMax[n - 1] = arr[n - 1];
        for (int i = n - 2; i >= 0; i--) {
            rightMax[i] = Math.max(rightMax[i + 1], arr[i]);
        }
        int waterTrapped = 0;
        for (int i = 0; i < n; i++) {
            waterTrapped += Math.min(leftMax[i], rightMax[i]) - arr[i];
        }
        return waterTrapped;
    }
}
```

Custom Input | Compile & Run | Submit