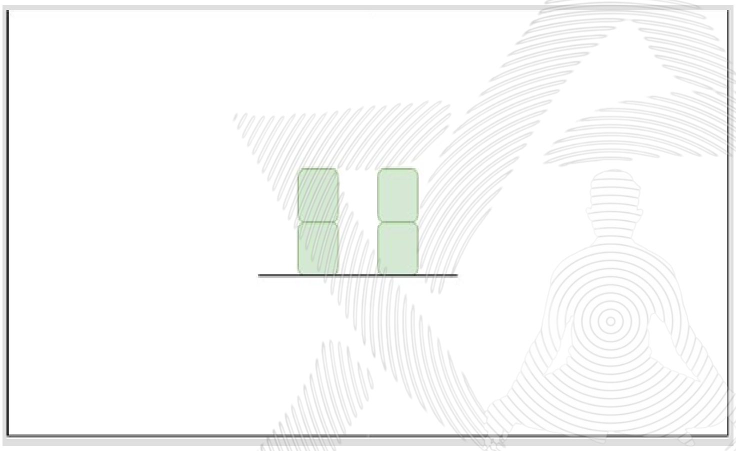
TRAPPING RAIN WATER

Given an array **arr[]** of **N** non-negative integers representing height of blocks at index **i** as **A**$_i$ where the width of each block is 1. Compute how much water can be trapped in between blocks after raining.

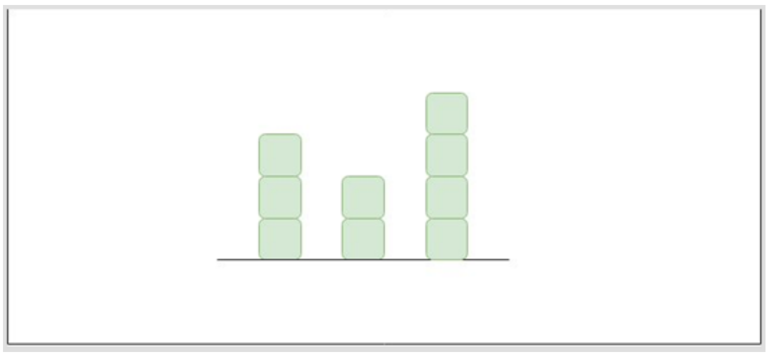
Input: arr[] = {2, 0, 2}
Output: 2
Explanation:

The structure is like below



We can trap 2 units of water in the middle gap.
Input: arr[] = {3, 0, 2, 0, 4}

Output: 7
Explanation:

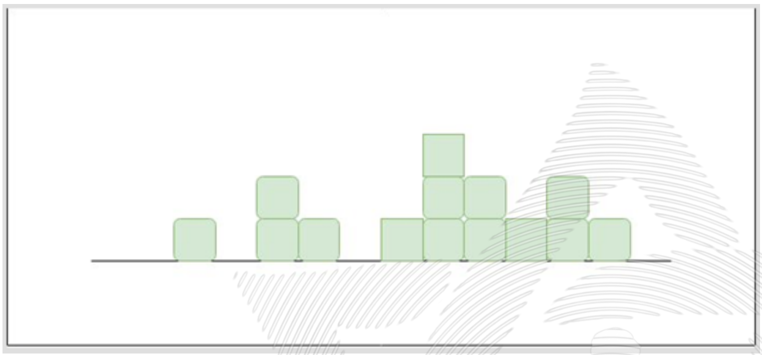


We can trap "3 units" of water between 3 and 2, "1 unit" on top of bar 2 and "3 units" between 2 and 4. See below diagram also.

Input: arr[] = [0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1]
Output: 6

Explanation:

The structure is like below



Trap "1 unit" between first 1 and 2, "4 units" between first 2 and 3 and "1 unit" between second last 1 and last 2

Algorithm:

1. Traverse the array from start to end.
2. For every element (bar) , find the max value (bar with maximum height)on the left side as well as right side .
3. Take the minimum of these two values and subtract that particular element(height of the bar) for which the minimum of these maximums is calculated.This gives water stored by that bar.
4. Take the sum of water stored by each bar.

CODE :

```cpp
#include <bits/stdc++.h>
#include <iostream>
using namespace std;

vector<int> left(int a[], int n)
{
    int i = 0;
    vector<int> v;
    int m = a[0];

    v.push_back(m);
    for (i = 1; i < n; i++)
    {
        if (m < a[i])
        {
            m = a[i];
            v.push_back(m);
            continue;
        }
        v.push_back(m);
    }
    return v;
}
vector<int> right(int a[], int n)
{
    int i = 0;
    vector<int> v;
    int m = a[n - 1];
    v.push_back(m);
    for (i = n - 2; i >= 0; i--)
    {
        if (m < a[i])
        {
            m = a[i];
            v.push_back(m);
            continue;
        }
        v.push_back(m);
    }
    vector<int> v1;
    for (i = 0; i < n; i++)
    v1.push_back(v[n - i - 1]);
    return v1;
}
```

```cpp
int main()

{

    int t;

    cin >> t;

    while (t--)

    {

        int n;

        cin >> n;

        int a[n];

        for (int i = 0; i < n; i++) cin >> a[i];

        vector<int> vl = left(a, n);

        vector<int> vr = right(a, n);

        int m = 0;

        for (int i = 0; i < n; i++)

        {

            m = m + min(vl[i], vr[i]) - a[i];

        }

        cout << m << endl;

    }

    return 0;

}
```