**[Trapping Rain Water](https://leetcode.com/problems/trapping-rain-water/)**

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.

**Example 1:**



**Input:** height = [0,1,0,2,1,0,1,3,2,1,2,1]

**Output:** 6

**Explanation:** The above elevation map (black section) is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

**Example 2:**

**Input:** height = [4,2,0,3,2,5]

**Output:** 9

**Constraints:**

* n == height.length
* 1 <= n <= 2 \* 104
* 0 <= height[i] <= 105

class Solution {

public:

int trap(vector<int>& height) {

int n = height.size();

if (n == 0) return 0;

vector<int> left(n);

vector<int> right(n);

// Fill left array

left[0] = height[0];

for (int i = 1; i < n; i++) {

left[i] = max(left[i - 1], height[i]);

}

// Fill right array

right[n - 1] = height[n - 1];

for (int i = n - 2; i >= 0; i--) {

right[i] = max(right[i + 1], height[i]);

}

// Calculate trapped water

int trappedWater = 0;

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

trappedWater += min(left[i], right[i]) - height[i];

}

return trappedWater;

}

};

Link : <https://leetcode.com/problems/trapping-rain-water/>