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

**Solution:**

class Solution {

public int trap(int[] height) {

int n = height.length;

if(n<=2)

return 0;

int maxLeft = height[0];

int maxRight = height[n-1];

int trappedWater = 0;

int left = 1;

int right = n-2;

while(left<=right){

if(maxLeft<maxRight){

if(height[left] >= maxLeft)

maxLeft = height[left];

else

trappedWater += maxLeft - height[left];

left +=1;

}

else{

if(height[right] > maxRight)

maxRight = height[right];

else

trappedWater += maxRight - height[right];

right -=1;

}

}

return trappedWater;

}

}

Video: https://www.youtube.com/watch?v=C8UjlJZsHBw