Question: https://leetcode.com/problems/trapping-rain-water/

The problem might seem to be difficult but break it into a sub-problem for a certain position, we need to know the right boundary and the left boundary then we can know the volume of water which can get stored at that position.

So a boundary for a certain position must have value greater than or equal to the height of that position.

So we maintain 2 arrays left and right for storing left boundary and right boundary of every position then we can simply find column of water trapped at each position, I.e., the lowest boundary between left and right - height of that position.

Code:  
class Solution {

public int trap(int[] height) {

int storage=0,i;

int[] left= new int[height.length], right= new int[height.length];

left[0]=height[0];

right[height.length-1]=height[height.length-1];

for(i=1;i<height.length;i++){

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

}

for(i=height.length-2;i>=0;i--){

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

}

for(i=1;i<height.length;i++){

storage+=(Math.min(left[i], right[i])-height[i])\*1;

}

return storage;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>