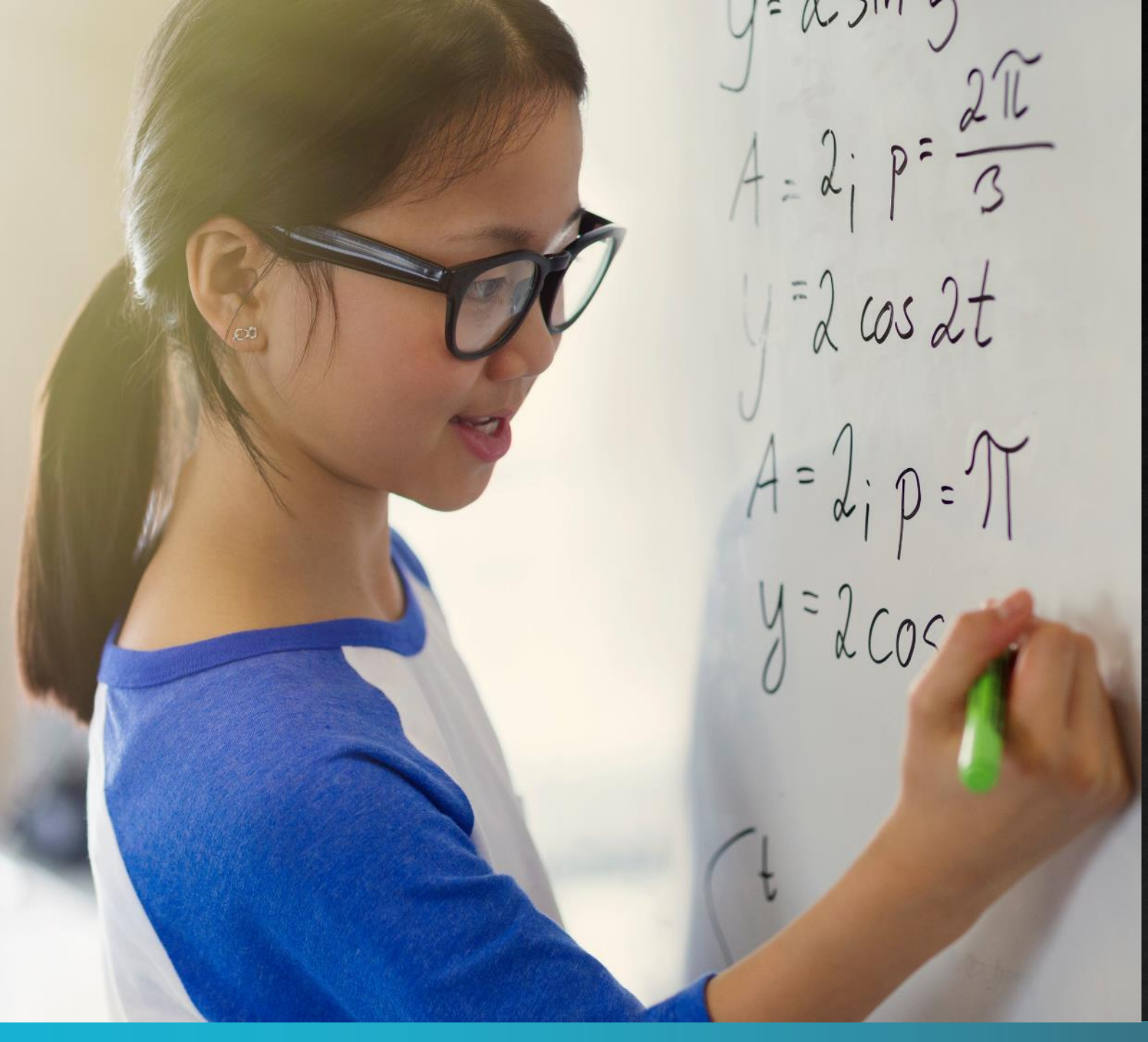


Trapping Rain Water | Algorithm Problem

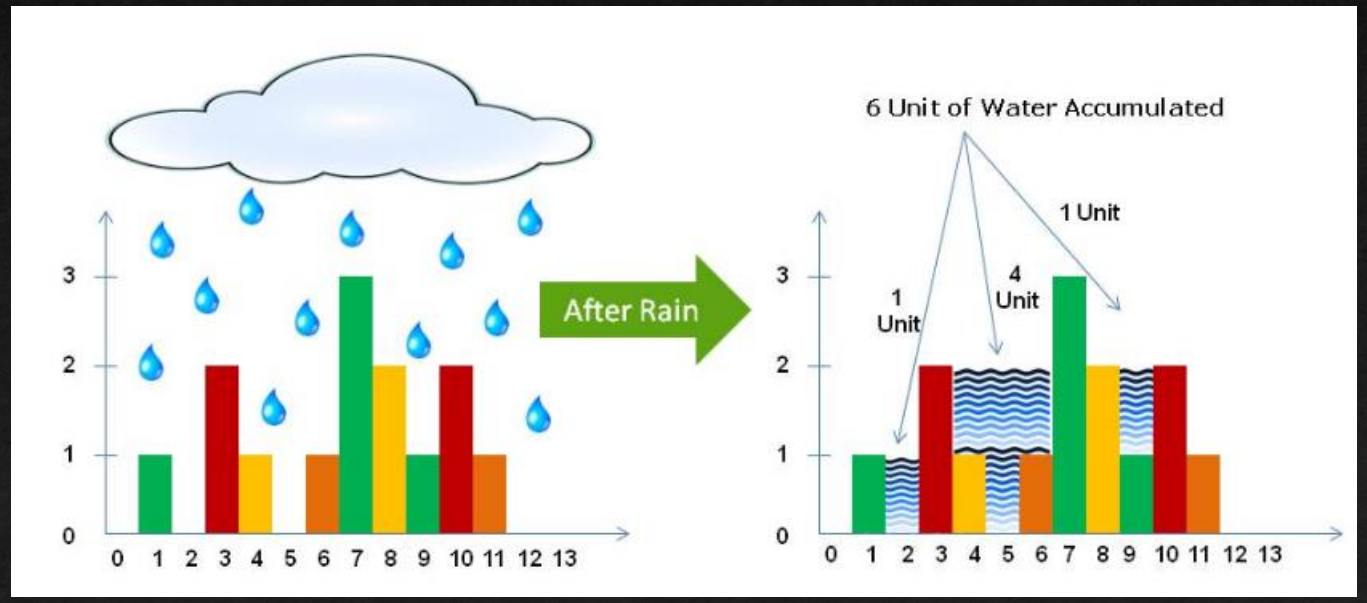
SUBMITTED BY:
ANKIT KUMAR JHA
20BPS1050

Problem Statement

- To find the maximum volume of water that can be stored in between buildings or bars as shown in the below image. Assume that width of each bar is 1.
- In other words, Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

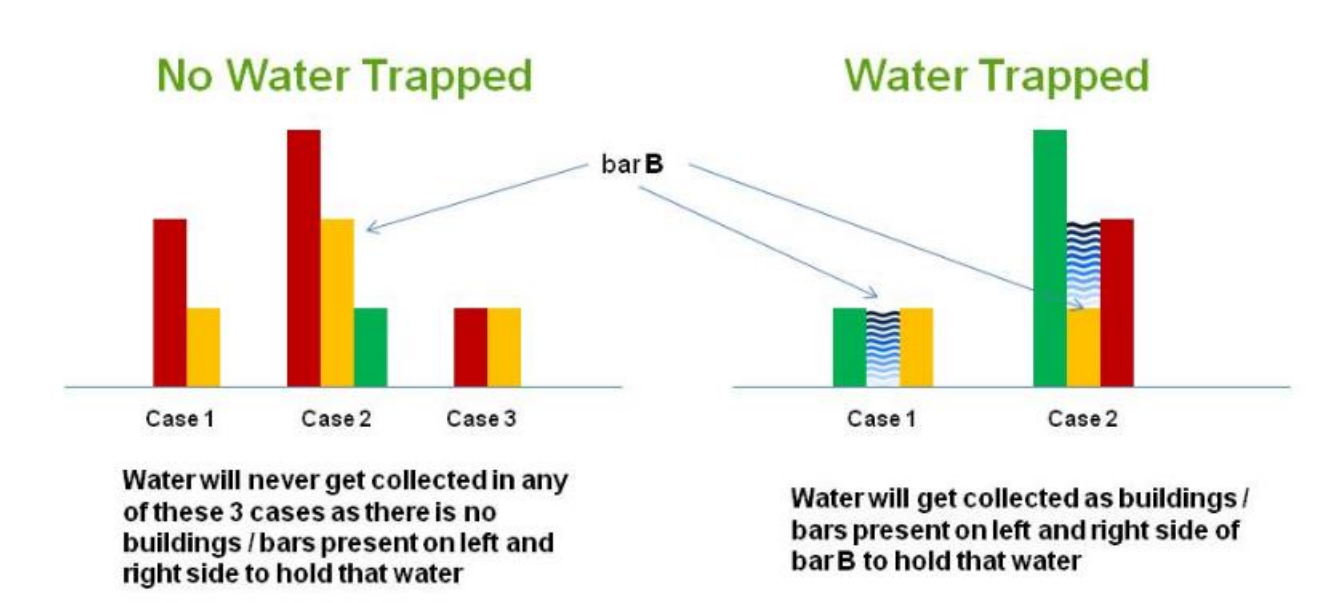
  int arr[] = {0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1};

  Answer : 6 Unit

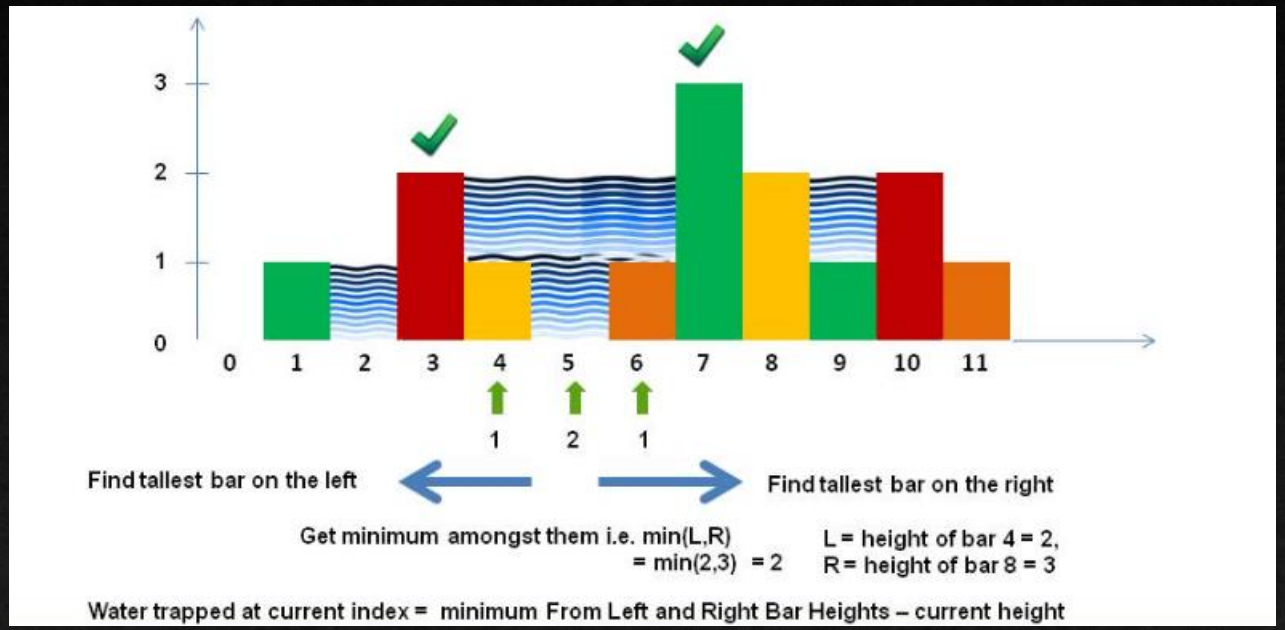


When Will Rain Water Be Trapped On Any Bar?

- For water to get accumulated on bar B, there should be higher sized bars present on left and right side of that bar B. (i.e. bars with height greater than the hight of bar B.



How Much Rain Water Will Be Trapped On Each Bar?



So for example,

water trapped at index 5 = min(2,3) – 1 = 1

index 6 = min(2,3) – 0 = 2

index 7 = min(2,3) – 1 = 1

Solution Approaches

- Using above facts, we can think of two approaches to solve this problem:
- **Approach 1 (Naive Approach)**

1. Traverse every array element
2. For each element,
   – Find the highest bars on left and right sides.
   – Take the smaller of two heights.
   – The difference between smaller height and height of current element is the amount of water that can be stored in this array element.
3. Time complexity of this solution is $O(n^2)$ as
   – we are traversing each element i.e. n times +
   – for each element we are searching left and right bar with max hight i.e. again n times.

- This is not very efficient solution for the given problem.

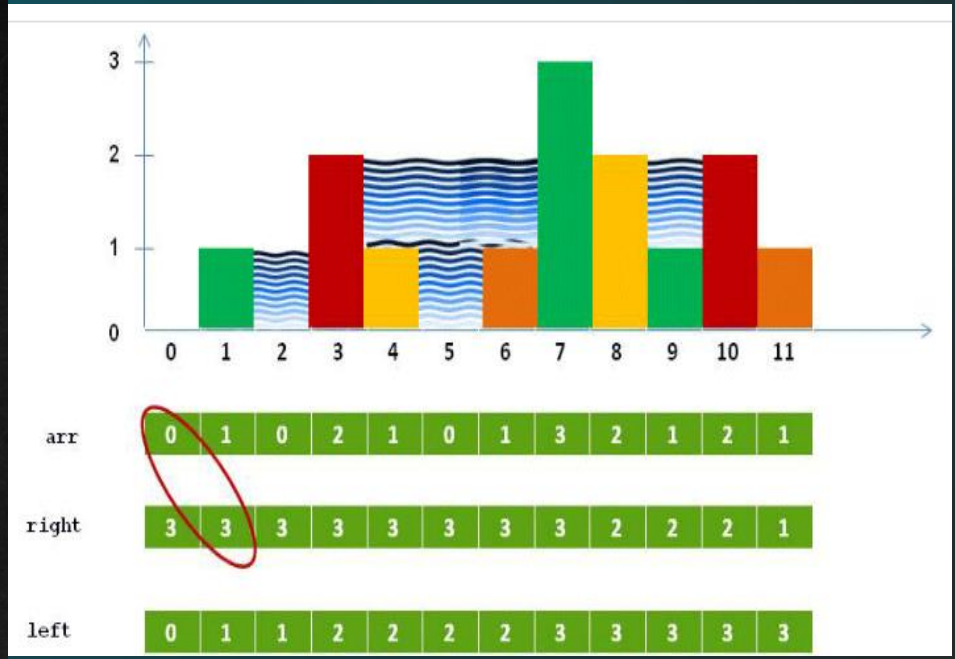
Approach 2

- We can to prre-compute highest bar on left and right of every bar in O(n) time. Then use these pre-computed values to find the amount of water in every array element. This solution is O(n) and more efficient than the previous one.

- We will implement the second solution:

1. First compute highest bar on the left of every bar in O(n) – by getting maximum between height of current bar and height of the previous tallest bar. Store this maximum value at the current index of left array.
2. Similarly compute highest bar on the right of every bar in O(n) and store it in right array.
3. Now at any point of the bar the maximum water that can be trapped will be minimum of left height and right height minus the height of the bar.
4. Keep adding the value for each bar.

Amount of water stored on the top of the bar i = (maximum of left(i) and right(i)) – arr(i)



Algorithm

- Create two arrays *left* and *right* of size n. create a variable *max_ = INT_MIN*.
- Run one loop from start to end. In each iteration update max_ as *max_ = max(max_, arr[i])* and also assign *left[i] = max_*
- Update max_ = INT_MIN.
- Run another loop from end to start. In each iteration update max_ as *max_ = max(max_, arr[i])* and also assign *right[i] = max_*
- Traverse the array from start to end.
- The amount of water that will be stored in this column is *min(a,b) – array[i]*,(where a = left[i] and b = right[i]) add this value to total amount of water stored
- Print the total amount of water stored.

REFERENCE:

1. https://www.geeksforgeeks.org/trapping-rain-water/
2. https://www.techiedelight.com/trapping-rain-water-within-given-set-bars/
3. https://www.codechef.com/problems/cres102

THANK YOU!!!!!