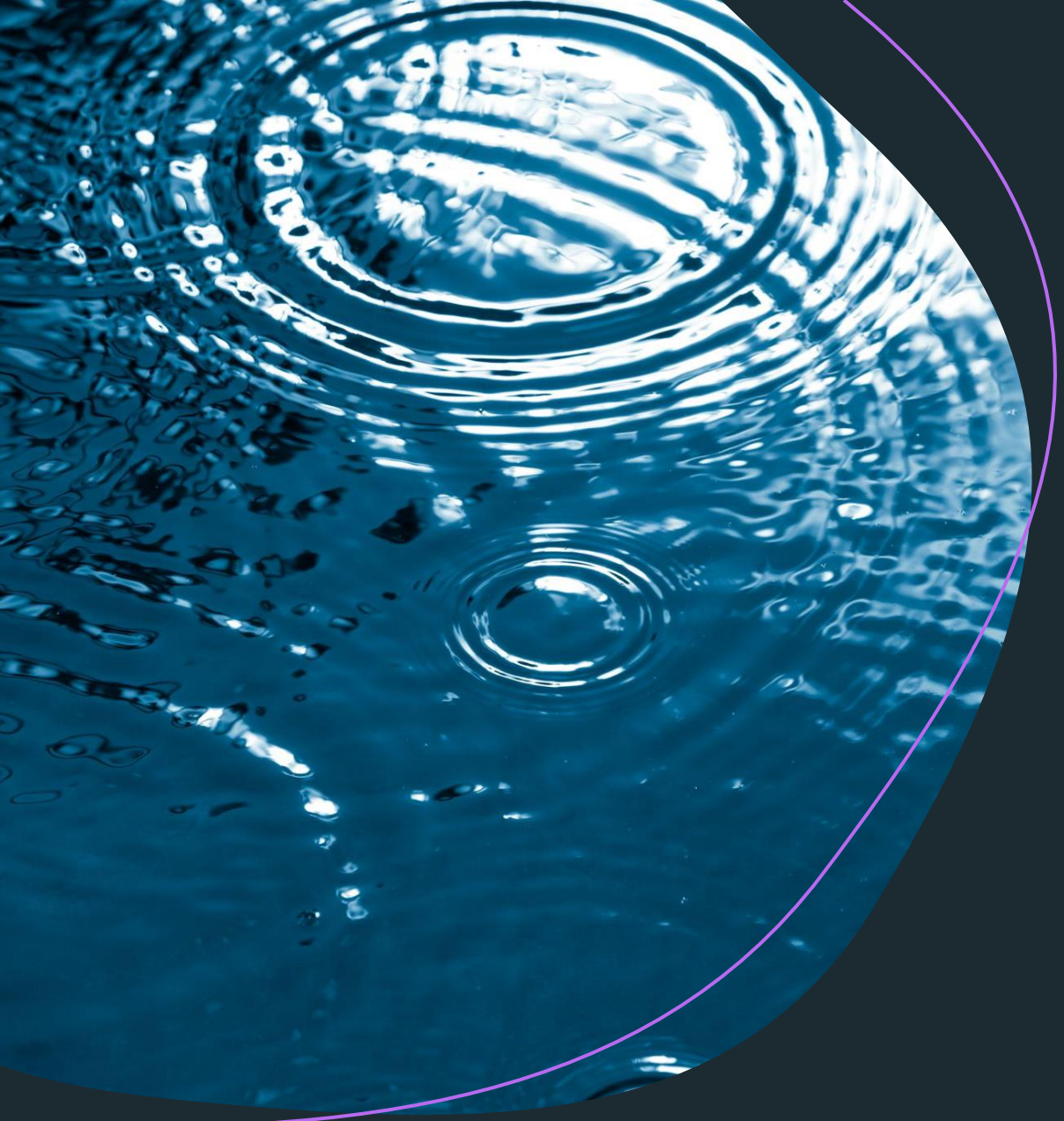
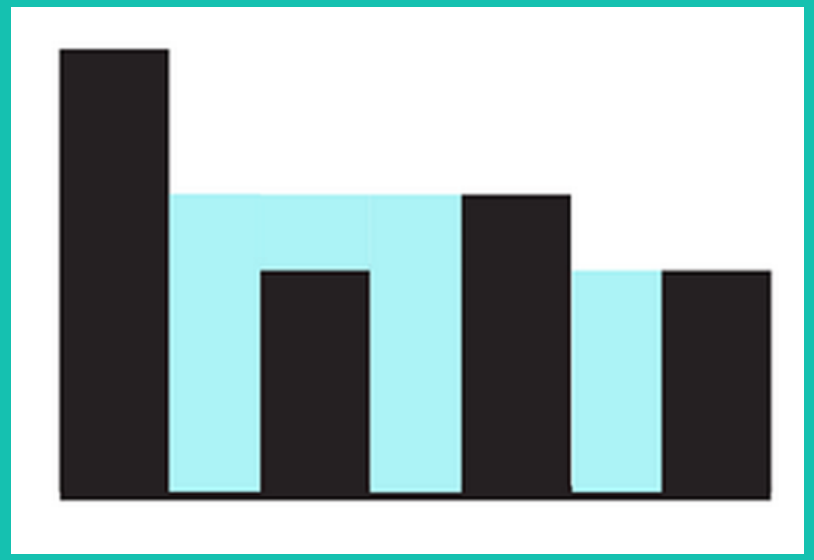


Trapping Rain Water



Problem – Trapping rain water

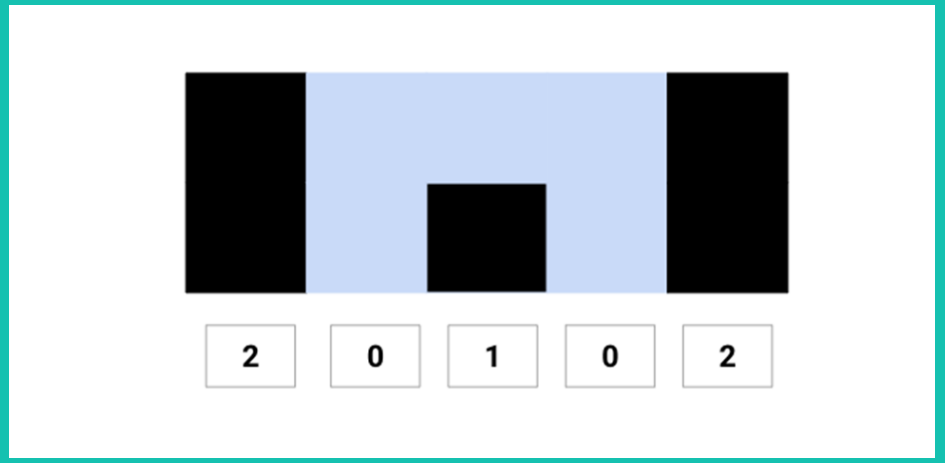
- Given n non-negative integers representing an elevation map where the width of each tower is 1, write a program to compute how much water it can trap after raining.

Example 1

Input: height[] = [2, 0, 1, 0, 2],
Output: 5

Explanation:

Trapped water = 2 x 1 + 1 x 1 + 2 x 1 = 5 (Area of the blue region in the following diagram)



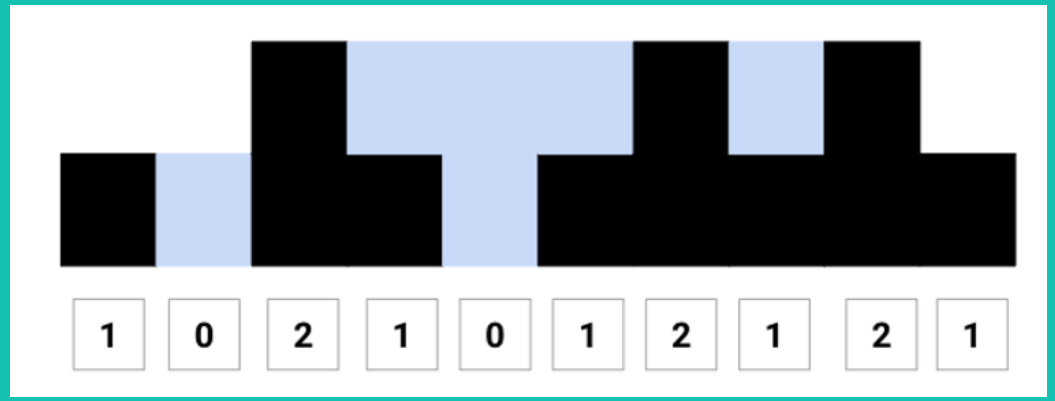
Example 2

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

Explanation:

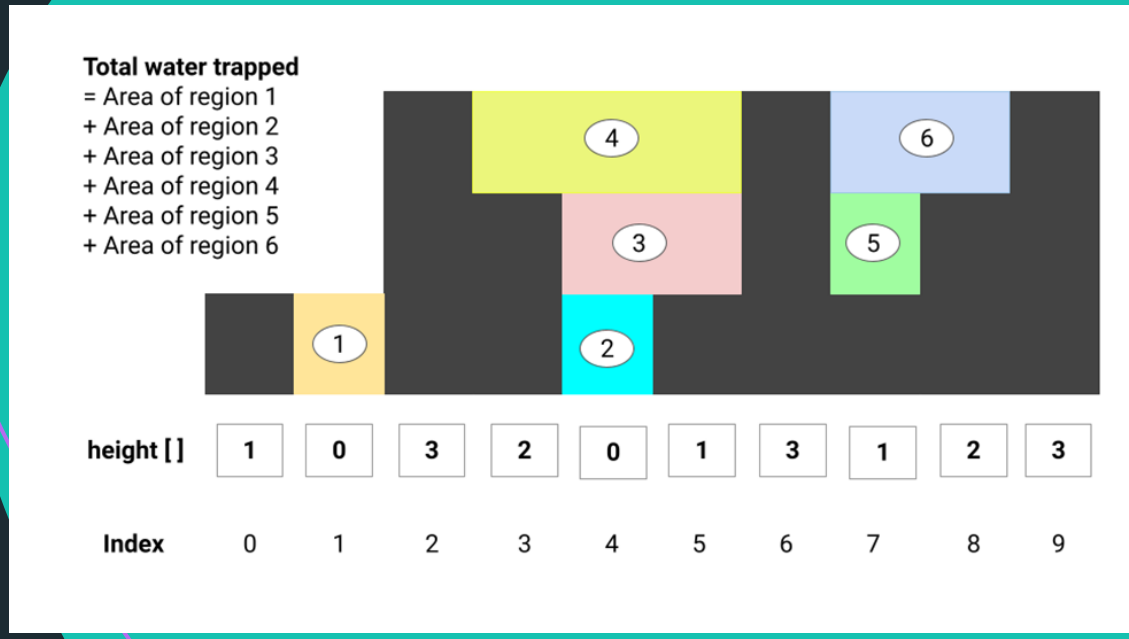
Trapped water = 1 x 1 + 1 x 1 + 2 x 1 + 1 x 1 + 1 x 1 = 6

(Area of the blue region in the following diagram)



Solution using data structure: Stack

- Area of region 1 = area confined between index 2 and 0
- Area of region 2 = area confined between index 5 and 3
- Area of region 3 = area confined between index 6 and 3
- Area of region 4 = area confined between index 6 and 2
- Area of region 5 = area confined between index 8 and 6
- Area of region 5 = area confined between index 9 and 6



Did we observe some pattern?

Algorithm Steps

- We declare a stack S to store the indices of the towers.
- Now we scan the height[n] using a loop variable curr < n
- If we found a current tower larger than that tower at the top of the stack, we can say that the tower at the top of the stack is confined between the current tower and a previous tower in the stack. Hence, we pop the stack and add water trapped between towers to the total water trapped. We can continue this step in a loop until we find the current tower smaller than the tower at the top of the stack.
- Water trapped between towers = Area of the rectangular region formed by the current tower, popped tower, and tower at the top of the stack

  Region length = (current index – index of the top stack element – 1)

  Region height = min (height of the current tower, height of tower at top of the stack) – height of the popped tower

  Water trapped = Region length x Region height
- If the current tower is smaller than or equal to the tower at the top of the stack, we push the index of the current tower to the stack and move to the next tower. It means current tower is confined with tower at the top of the stack.