

Trapping Rain Water

This problem is solvable in $O(n)$ time and $O(1)$ space using the two-pointers technique.

[Key Idea:]

The water trapped at position i is determined by:

$$\text{water}[i] = \min(\text{max_left}[i], \text{max_right}[i]) - \text{height}[i]$$

But instead of computing two arrays, we can do this in one pass using two pointers.

[Steps:]

1. Initialize two pointers:
 - $\text{left} = 0, \text{right} = n-1$
2. Maintain two variables:
 - $\text{left_max} = 0, \text{right_max} = 0$
3. While $\text{left} < \text{right}$:
 - If $\text{height}[\text{left}] < \text{height}[\text{right}]$:
 - If $\text{height}[\text{left}] \geq \text{left_max}$, update left_max
 - Else add $\text{left_max} - \text{height}[\text{left}]$ to result
 - Move $\text{left}++$
 - Else:
 - If $\text{height}[\text{right}] \geq \text{right_max}$, update right_max
 - Else add $\text{right_max} - \text{height}[\text{right}]$ to result
 - Move $\text{right}--$

[Complexity:]

- Time: $O(n)$ — single pass
- Space: $O(1)$ — constant extra space