

Trapping Rain Water Problem

The 'Trapping Rain Water' problem is a classic coding challenge that involves calculating the amount of water that can be trapped between bars of different heights. Given an array of non-negative integers representing the heights of bars, the goal is to compute how much water can be collected between the bars after rainfall.

Code Explanation:

The code provided uses a two-pointer approach to solve the problem efficiently. Here's a detailed breakdown:

1. **Initialization**: Two pointers (L and R) are initialized to the start and end of the array, respectively. Two variables, LMAX and RMAX, store the maximum heights encountered so far from the left and right.
2. **Two-Pointer Approach**:
 - Move the pointers inward until they meet.
 - Compare LMAX and RMAX to decide which pointer to move.
 - If LMAX is smaller, calculate the water trapped at L, update LMAX, and move the left pointer.
 - If RMAX is smaller, calculate the water trapped at R, update RMAX, and move the right pointer.
3. **Calculation**: At each step, the trapped water is added to the total water storage.
4. **Return**: Finally, the total water trapped is returned.

Time and Space Complexity:

- **Time Complexity**: $O(n)$, where n is the length of the array. Each bar is processed once.

- **Space Complexity**: $O(1)$, as the solution uses a constant amount of extra space.

Code Implementation:

```
1  class Solution:
2      def trap(self, height: List[int]) -> int:
3          L, R = 0, len(height) - 1
4          LMAX, RMAX = height[L], height[R]
5          totalWaterStorage = 0
6
7          while L < R:
8              if LMAX < RMAX:
9                  L += 1
10                 if LMAX > height[L]:
11                     totalWaterStorage += LMAX - height[L]
12                     LMAX = max(LMAX, height[L])
13             else:
14                 R -= 1
15                 if RMAX > height[R]:
16                     totalWaterStorage += RMAX - height[R]
17                     RMAX = max(RMAX, height[R])
18         return totalWaterStorage
19
```