

# Peak Value Control 1 - Solution Outline

## Key Ideas

Values can only move to the left. For any prefix, its total sum must stay inside that prefix, so the maximum value in the final array must be at least the ceiling of the average of each prefix sum.

## Algorithm

1. Initialize `prefixSum = 0` and `answer = 0`.
2. For each index `i` from 0 to `n - 1`:
  - Add `a[i]` to `prefixSum`.
  - Compute `ceil(prefixSum / (i + 1))`.
  - Update `answer` with the maximum of these values.
3. Output `answer`.

## Correctness Sketch

Because movement is only leftward, the sum of any prefix cannot be reduced or moved to the right. Thus, those `i + 1` positions must hold their total sum, forcing a maximum of at least the ceiling of the average. A greedy leftward redistribution can achieve this bound, so the maximum over all prefixes is both necessary and sufficient.

## Complexity

- Time:  $O(n)$
- Space:  $O(1)$