

# Interval Power Updates 1 - Solution Outline

## Key Ideas

Applying each update directly is too slow. A difference array stores how each range update changes the boundary, and a single prefix sum reconstructs the final array.

## Algorithm

1. Create an array `diff` of length `n` filled with 0.
2. For each update  $(l, r, x)$ :
  - Add `x` to `diff[l]`.
  - If  $r + 1 < n$ , subtract `x` from `diff[r + 1]`.
3. Compute prefix sums of `diff` to obtain the final values.
4. Print the reconstructed array.

## Correctness Sketch

Each update adds `x` to all positions from `l` to `r`. The difference array encodes exactly these changes at the boundaries. Taking prefix sums accumulates all contributions that cover each position, yielding the same values as applying updates individually.

## Complexity

- Time:  $O(n + q)$
- Space:  $O(n)$