

Shifting Letters II

You are given a string s of lowercase English letters and a 2D integer array $shifts$ where $shifts[i] = [start_i, end_i, direction_i]$. For every i , **shift** the characters in s from the index $start_i$ to the index end_i (**inclusive**) forward if $direction_i = 1$, or shift the characters backward if $direction_i = 0$.

Shifting a character **forward** means replacing it with the **next** letter in the alphabet (wrapping around so that 'z' becomes 'a'). Similarly, shifting a character **backward** means replacing it with the **previous** letter in the alphabet (wrapping around so that 'a' becomes 'z').

Return *the final string after all such shifts to s are applied*.

Example 1:

Input: $s = \text{"abc"}$, $shifts = [[0,1,0],[1,2,1],[0,2,1]]$

Output: "ace"

Explanation: Firstly, shift the characters from index 0 to index 1 backward. Now $s = \text{"zac"}$.

Secondly, shift the characters from index 1 to index 2 forward. Now $s = \text{"zbd"}$.

Finally, shift the characters from index 0 to index 2 forward. Now $s = \text{"ace"}$.

Example 2:

Input: $s = \text{"dztz"}$, $shifts = [[0,0,0],[1,1,1]]$

Output: "catz"

Explanation: Firstly, shift the characters from index 0 to index 0 backward. Now $s = \text{"cztz"}$.

Finally, shift the characters from index 1 to index 1 forward. Now $s = \text{"catz"}$.

Constraints:

- $1 \leq s.length, shifts.length \leq 5 * 10^4$
- $shifts[i].length == 3$
- $0 \leq start_i \leq end_i < s.length$
- $0 \leq direction_i \leq 1$
- s consists of lowercase English letters.