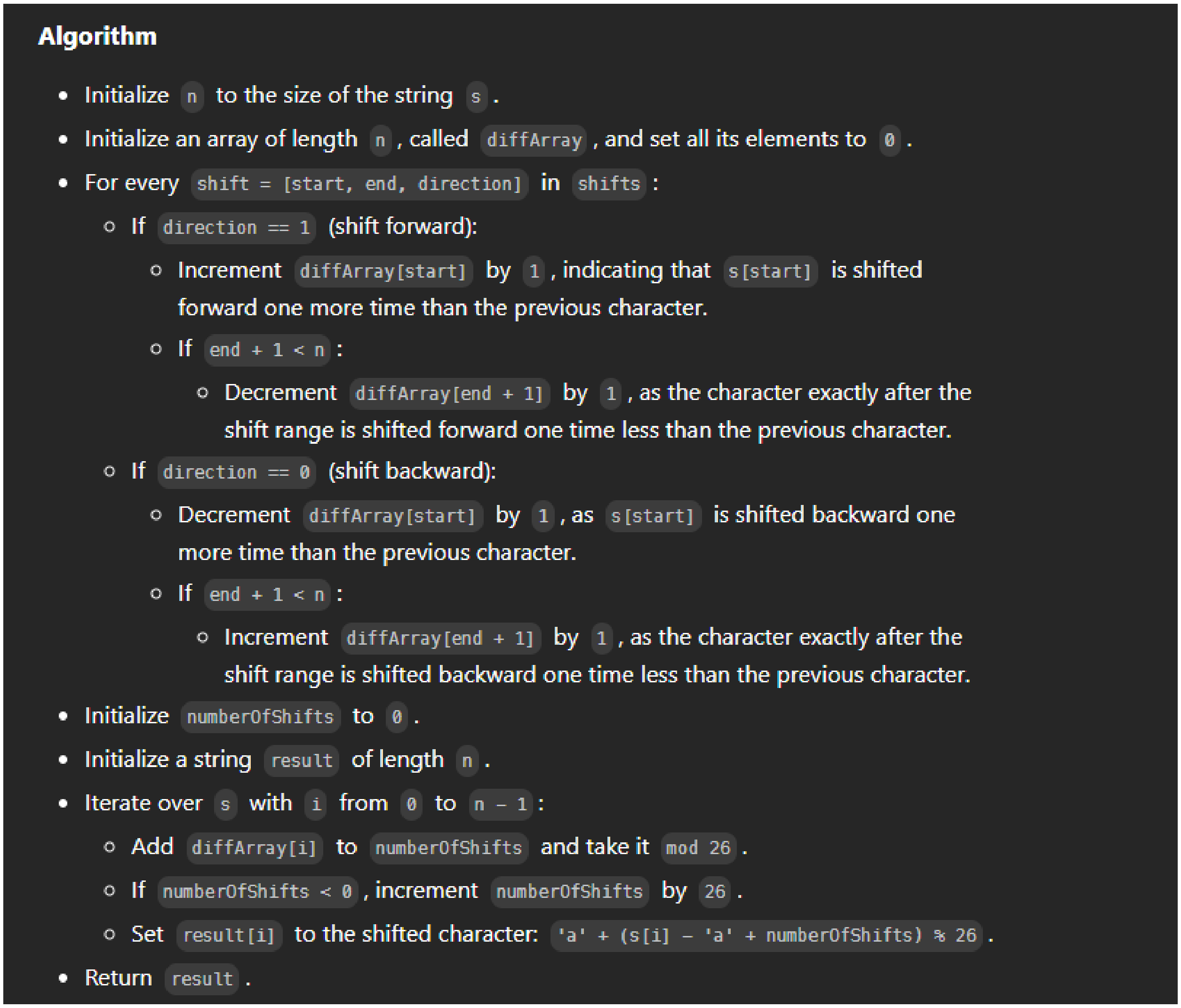
Intution

Intuition

Building on the idea of cumulative sums, we can use a difference array to handle range updates more efficiently. A difference array helps us record changes in values between consecutive elements rather than updating every element in a range directly.

Instead of keeping track of how many shifts should be applied to each character in the alphabet, we'll use the difference array to store how many more shifts should be applied to the current character compared to the previous one. This allows us to record changes only at the starting and ending points of shifts, rather than updating each character in the range.

For convenience, a positive shift means that the character must move forward in the alphabet, and a negative shift means that it must move backward.



Algorithm

- Initialize `n` to the size of the string `s`.
- Initialize an array of length `n`, called `diffArray`, and set all its elements to `0`.
- For every `shift = [start, end, direction]` in `shifts`:
    - If `direction == 1` (shift forward):
        - Increment `diffArray[start]` by `1`, indicating that `s[start]` is shifted forward one more time than the previous character.
        - If `end + 1 < n`:
            - Decrement `diffArray[end + 1]` by `1`, as the character exactly after the shift range is shifted forward one time less than the previous character.
    - If `direction == 0` (shift backward):
        - Decrement `diffArray[start]` by `1`, as `s[start]` is shifted backward one more time than the previous character.
        - If `end + 1 < n`:
            - Increment `diffArray[end + 1]` by `1`, as the character exactly after the shift range is shifted backward one time less than the previous character.
- Initialize `numberOfShifts` to `0`.
- Initialize a string `result` of length `n`.
- Iterate over `s` with `i` from `0` to `n - 1`:
    - Add `diffArray[i]` to `numberOfShifts` and take it `mod 26`.
    - If `numberOfShifts < 0`, increment `numberOfShifts` by `26`.
    - Set `result[i]` to the shifted character: `'a' + (s[i] - 'a' + numberOfShifts) % 26`.
- Return `result`.