Given a string s, we make queries on substrings of s.

For each query queries[i] = [left, right, k], we may **rearrange** the substring s[left], ..., s[right], and then choose **up to** k of them to replace with any lowercase English letter.

If the substring is possible to be a palindrome string after the operations above, the result of the query is true. Otherwise, the result is false.

Return an array answer[], where answer[i] is the result of the i-th query queries[i].

Note that: Each letter is counted **individually** for replacement so if for example s[left..right] = "aaa", and k = 2, we can only replace two of the letters.  (Also, note that the initial string s is never modified by any query.)

**Example :**

**Input:** s = "abcda", queries = [[3,3,0],[1,2,0],[0,3,1],[0,3,2],[0,4,1]]

**Output:** [true,false,false,true,true]

**Explanation:**

queries[0] : substring = "d", is palidrome.

queries[1] : substring = "bc", is not palidrome.

queries[2] : substring = "abcd", is not palidrome after replacing only 1 character.

queries[3] : substring = "abcd", could be changed to "abba" which is palidrome. Also this can be changed to "baab" first rearrange it "bacd" then replace "cd" with "ab".

queries[4] : substring = "abcda", could be changed to "abcba" which is palidrome.

**Constraints:**

* 1 <= s.length, queries.length <= 10^5
* 0 <= queries[i][0] <= queries[i][1] < s.length
* 0 <= queries[i][2] <= s.length
* s only contains lowercase English letters.