Problem 1177: Can Make Palindrome from Substring

Problem Information

Difficulty: Medium
Acceptance Rate: 40.97%
Paid Only: No
Tags: Array, Hash Table, String, Bit Manipulation, Prefix Sum

Problem Description

You are given a string `s` and array `queries` where `queries[i] = [lefti, righti, ki]`. We may rearrange the substring `s[lefti...righti]` for each query and then choose up to `ki` 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 a boolean array `answer` where `answer[i]` is the result of the `ith` query `queries[i]`.

Note that each letter is counted individually for replacement, so if, for example `s[lefti...righti] = "aaa"`, and `ki = 2`, we can only replace two of the letters. Also, note that no query modifies the initial string `s`.

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.

Example 2:

Input: s = "lyb", queries = [[0,1,0],[2,2,1]] **Output:** [false,true]

Constraints:

* `1 <= s.length, queries.length <= 105` * `0 <= lefti <= righti < s.length` * `0 <= ki <= s.length`
* `s` consists of lowercase English letters.

Code Snippets

C++:

```cpp
class Solution {
public:
vector<bool> canMakePaliQueries(string s, vector<vector<int>>& queries) {


}
};
```

Java:

```java
class Solution {
public List<Boolean> canMakePaliQueries(String s, int[][] queries) {


}
}
```

Python3:

```python
class Solution:
def canMakePaliQueries(self, s: str, queries: List[List[int]]) -> List[bool]:
```