

# Problem 2983: Palindrome Rearrangement Queries

## Problem Information

**Difficulty:** Hard

**Acceptance Rate:** 24.23%

**Paid Only:** No

**Tags:** Hash Table, String, Prefix Sum

## Problem Description

You are given a \*\*0-indexed\*\* string `s` having an \*\*even\*\* length `n`.

You are also given a \*\*0-indexed\*\* 2D integer array, `queries`, where `queries[i] = [ai, bi, ci, di]`.

For each query `i`, you are allowed to perform the following operations:

- \* Rearrange the characters within the \*\*substring\*\* `s[ai:bi]`, where `0 <= ai <= bi < n / 2`.\*
- Rearrange the characters within the \*\*substring\*\* `s[ci:di]`, where `n / 2 <= ci <= di < n`.

For each query, your task is to determine whether it is possible to make `s` a \*\*palindrome\*\* by performing the operations.

Each query is answered \*\*independently\*\* of the others.

Return \_a\*\*0-indexed\*\* array \_`answer` \_, where\_`answer[i] == true` \_if it is possible to make\_`s` \_a palindrome by performing operations specified by the\_`ith` \_query, and\_`false` \_otherwise.\_

\* A \*\*substring\*\* is a contiguous sequence of characters within a string. \* `s[x:y]` represents the substring consisting of characters from the index `x` to index `y` in `s`, \*\*both inclusive\*\*.

\*\*Example 1:\*\*

**\*\*Input:\*\*** s = "abcabc", queries = [[1,1,3,5],[0,2,5,5]] **\*\*Output:\*\*** [true,true] **\*\*Explanation:\*\*** In this example, there are two queries: In the first query: - a0 = 1, b0 = 1, c0 = 3, d0 = 5. - So, you are allowed to rearrange s[1:1] => a \_b\_ cab and s[3:5] => abc \_abc\_. - To make s a palindrome, s[3:5] can be rearranged to become => abc \_cba\_. - Now, s is a palindrome. So, answer[0] = true. In the second query: - a1 = 0, b1 = 2, c1 = 5, d1 = 5. - So, you are allowed to rearrange s[0:2] => \_abc\_ abc and s[5:5] => abcab \_c\_. - To make s a palindrome, s[0:2] can be rearranged to become => \_cba\_ abc. - Now, s is a palindrome. So, answer[1] = true.

**\*\*Example 2:\*\***

**\*\*Input:\*\*** s = "abbcdecbba", queries = [[0,2,7,9]] **\*\*Output:\*\*** [false] **\*\*Explanation:\*\*** In this example, there is only one query. a0 = 0, b0 = 2, c0 = 7, d0 = 9. So, you are allowed to rearrange s[0:2] => \_abb\_ cdecbb and s[7:9] => abbcd \_bba\_. It is not possible to make s a palindrome by rearranging these substrings because s[3:6] is not a palindrome. So, answer[0] = false.

**\*\*Example 3:\*\***

**\*\*Input:\*\*** s = "acbcab", queries = [[1,2,4,5]] **\*\*Output:\*\*** [true] **\*\*Explanation:\*\*** In this example, there is only one query. a0 = 1, b0 = 2, c0 = 4, d0 = 5. So, you are allowed to rearrange s[1:2] => a \_cb\_ cab and s[4:5] => acbc \_ab\_. To make s a palindrome s[1:2] can be rearranged to become a \_bc\_ cab. Then, s[4:5] can be rearranged to become abcc \_ba\_. Now, s is a palindrome. So, answer[0] = true.

**\*\*Constraints:\*\***

\* `2 <= n == s.length <= 105` \* `1 <= queries.length <= 105` \* `queries[i].length == 4` \* `ai == queries[i][0], bi == queries[i][1]` \* `ci == queries[i][2], di == queries[i][3]` \* `0 <= ai <= bi < n / 2` \* `n / 2 <= ci <= di < n` \* `n` is even. \* `s` consists of only lowercase English letters.

## Code Snippets

**C++:**

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

```
};
```

**Java:**

```
class Solution {  
    public boolean[] canMakePalindromeQueries(String s, int[][][] queries) {  
  
    }  
}
```

**Python3:**

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