# Problem 2955: Number of Same-End Substrings

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 61.51%
**Paid Only:** Yes
**Tags:** Array, Hash Table, String, Counting, Prefix Sum

## Problem Description

You are given a **0-indexed** string `s`, and a 2D array of integers `queries`, where `queries[i] = [li, ri]` indicates a substring of `s` starting from the index `li` and ending at the index `ri` (both **inclusive**), i.e. `s[li..ri]`.

Return _an array_`ans` _where_ `ans[i]` _is the number of**same-end** substrings of_ `queries[i]`.

A **0-indexed** string `t` of length `n` is called **same-end** if it has the same character at both of its ends, i.e., `t[0] == t[n - 1]`.

A **substring** is a contiguous non-empty sequence of characters within a string.

**Example 1:**

**Input:** s = "abcaab", queries = [[0,0],[1,4],[2,5],[0,5]] **Output:** [1,5,5,10] **Explanation:** Here is the same-end substrings of each query: 1st query: s[0..0] is "a" which has 1 same-end substring: "**_a_** ". 2nd query: s[1..4] is "bcaa" which has 5 same-end substrings: "**_b_** caa", "b** _c_** aa", "bc** _a_** a", "bca** _a_** ", "bc** _aa_** ". 3rd query: s[2..5] is "caab" which has 5 same-end substrings: "**_c_** aab", "c** _a_** ab", "ca** _a_** b", "caa** _b_** ", "c** _aa_** b". 4th query: s[0..5] is "abcaab" which has 10 same-end substrings: "**_a_** bcaab", "a** _b_** caab", "ab** _c_** aab", "abc** _a_** ab", "abca** _a_** b", "abcaa** _b_** ", "abc** _aa_** b", "**_abca_** ab", "**_abcaa_** b", "a** _bcaab_** ".

**Example 2:**

**Input:** s = "abcd", queries = [[0,3]] **Output:** [4] **Explanation:** The only query is s[0..3] which is "abcd". It has 4 same-end substrings: "**_a_** bcd", "a** _b_** cd", "ab** _c_** d", "abc** _d_** ".

**Constraints:**

* `2 <= s.length <= 3 * 104` * `s` consists only of lowercase English letters. * `1 <= queries.length <= 3 * 104` * `queries[i] = [li, ri]` * `0 <= li <= ri < s.length`

## Code Snippets

**C++:**

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

}
};
```

**Java:**

```
class Solution {
public int[] sameEndSubstringCount(String s, int[][] queries) {

}
}
```

**Python3:**

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