

较大分组的位置

在一个由小写字母构成的字符串 s 中，包含由一些连续的相同字符所构成的分组。

例如，在字符串 $s = \text{"abbxxxxzzy"}$ 中，就含有 "a" , "bb" , "xxxx" , "z" 和 "yy" 这样的一些分组。

分组可以用区间 $[\text{start}, \text{end}]$ 表示，其中 start 和 end 分别表示该分组的起始和终止位置的下标。上例中的 "xxxx" 分组用区间表示为 $[3, 6]$ 。

我们称所有包含大于或等于三个连续字符的分组为 **较大分组**。

找到每一个 **较大分组** 的区间，按起始位置下标递增顺序排序后，返回结果。

示例 1:

输入: $s = \text{"abbxxxxzzy"}$

输出: $[[3, 6]]$

解释: "xxxx" 是一个起始于 3 且终止于 6 的较大分组。

示例 2:

输入: $s = \text{"abc"}$

输出: $[]$

解释: "a" , "b" 和 "c" 均不是符合要求的较大分组。

示例 3:

输入: $s = \text{"abcdddeeeeaabbbcd"}$

输出: $[[3, 5], [6, 9], [12, 14]]$

解释: 较大分组为 "ddd" , "eeee" 和 "bbb"

示例 4:

输入: $s = \text{"aba"}$

输出: $[]$

```
class Solution {
public:
    vector<vector<int>> largeGroupPositions(string s) {
        vector<vector<int>> res;
        if(s.size()==0)
        {
            return res;
        }
        int start=0;
        int end=0;
        while(end<s.size())
        {
            while(s[start]==s[end])
            {
                end++;//如果前后相同 连续存在 end 右移
            }
            if(end-start>=3)
            {
                res.push_back({start,end-1});
            }
            //找到一组符合要求的 继续找下一组 更新 start 起始位置
            start=end;
            end++;
        }
        return res;
    }
};
```