

763-划分字母区间

题述

763. 划分字母区间

难度 中等 681 ☆ 100% 100% 100%

字符串 `S` 由小写字母组成。我们要把这个字符串划分为尽可能多的片段，同一字母最多出现在一个片段中。返回一个表示每个字符串片段的长度的列表。

示例：

输入：S = "ababcbacadefegdehijhklij"

输出：[9,7,8]

解释：

划分结果为 "ababcbaca", "defegde", "hijhklij"。

每个字母最多出现在一个片段中。

像 "ababcbacadefegde", "hijhklij" 的划分是错误的，因为划分的片段数较少。

提示：

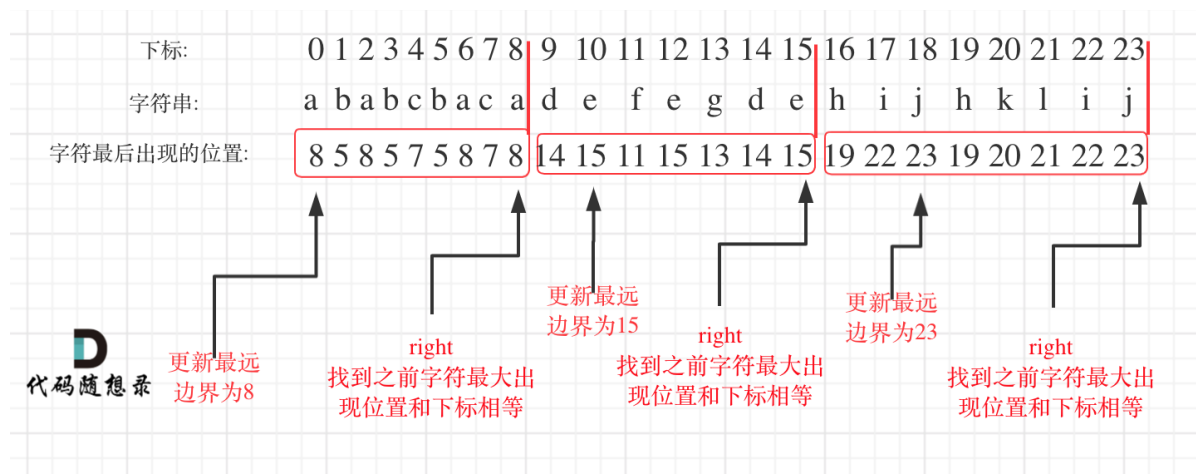
- `S` 的长度在 `[1, 500]` 之间。
- `S` 只包含小写字母 `'a'` 到 `'z'`。

思路

题目要求同一字母最多出现在一个片段中，那么如何把同一个字母的都圈在同一个区间里呢？

在遍历的过程中相当于是要找每一个字母的边界，**如果找到之前遍历过的所有字母的最远边界，说明这个边界就是分割点了**。此时前面出现过所有字母，最远也就到这个边界了。

- 统计每一个字符最后出现的位置
- 从头遍历字符，并更新字符的最远出现下标，如果找到字符最远出现位置下标和当前下标相等了，则找到了分割点



题解

C++

```
class Solution {
public:
    vector<int> partitionLabels(string s)
    {
        //要求同一字母最多出现在一个片段中
        //遍历的过程中相当于是要找每一个字母的边界
        int hash[27]={0}; //i为字符 hash[i]即为字符出现的最后位置
        for(int i = 0; i < s.size();i++)
        {
            //统计每一个字符最后出现的位置
            hash[s[i] - 'a'] = i;
        }
        vector<int> res;
        int left = 0;
        int right = 0;
        for(int i = 0; i < s.size();i++)
        {
            right = max(right,hash[s[i] - 'a']); //找到字符出现的最远边界
            if(i == right)
            {
                res.push_back(right - left + 1);
                left = i + 1;
            }
        }
        return res;
    }
};
```

执行结果: **通过** [显示详情 >](#)

[添加备注](#)

执行用时: **4 ms** , 在所有 C++ 提交中击败了 **64.20%** 的用户

内存消耗: **6.5 MB** , 在所有 C++ 提交中击败了 **44.82%** 的用户

通过测试用例: **117 / 117**

炫耀一下:



[写题解, 分享我的解题思路](#)

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	4 ms	6.5 MB	C++	2022/03/20 19:53	添加备注

Python

```
class Solution:
    def partitionLabels(self, s: str) -> List[int]:
        hash = [0] * 26
        for i in range(len(s)):
            hash[ord(s[i]) - ord('a')] = i
        result = []
        left = 0
        right = 0
        for i in range(len(s)):
            right = max(right, hash[ord(s[i]) - ord('a')])
            if i == right:
                result.append(right - left + 1)
                left = i + 1
        return result
```

执行结果: **通过** [显示详情 >](#)

[添加备注](#)

执行用时: **72 ms** , 在所有 Python3 提交中击败了 **5.95%** 的用户

内存消耗: **14.7 MB** , 在所有 Python3 提交中击败了 **99.30%** 的用户

通过测试用例: **117 / 117**

炫耀一下:



[写题解, 分享我的解题思路](#)

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	72 ms	14.7 MB	Python3	2022/03/20 20:01	▶
通过	52 ms	14.9 MB	Python3	2022/03/20 19:59	▶
通过	4 ms	6.5 MB	C++	2022/03/20 19:53	▶

思考

说实话, 感觉和贪心没啥关系, 没有感受到局部最优推出全局最优的过程