

5月8日报

本日学习内容

- 1. 学习面向对象下的协议与委托，使用类别实现非正式协议，遵守实现协议，看学长学姐的分享博客，手敲代码。
- 2. 完成每日算法题

今日算法题

题目1: [438. 找到字符串中所有字母异位词](#)

438. 找到字符串中所有字母异位词

已解答

中等

相关标签

相关企业

Ax

给定两个字符串 `s` 和 `p`，找到 `s` 中所有 `p` 的 **异位词** 的子串，返回这些子串的起始索引。不考虑答案输出的顺序。

示例 1:

输入: `s = "cbaebabacd"`, `p = "abc"`

输出: `[0,6]`

解释:

起始索引等于 0 的子串是 "cba", 它是 "abc" 的异位词。

起始索引等于 6 的子串是 "bac", 它是 "abc" 的异位词。

示例 2:

输入: `s = "abab"`, `p = "ab"`

输出: `[0,1,2]`

解释:

起始索引等于 0 的子串是 "ab", 它是 "ab" 的异位词。

起始索引等于 1 的子串是 "ba", 它是 "ab" 的异位词。

起始索引等于 2 的子串是 "ab", 它是 "ab" 的异位词。

提示:

- `1 <= s.length, p.length <= 3 * 104`
- `s` 和 `p` 仅包含小写字母

面试中遇到过这道题? 1/5

是

否

通过次数 681.6K | 提交次数 1.3M | 通过率 54.2%

```
class Solution {
```

```

public:
    vector<int> findAnagrams(string s, string p) {
        if (s.size() < p.size()) return {};
        vector<int> windowcount(26);
        vector<int> pcount(26);
        vector<int> ans;
        for(int i = 0; i < p.size(); i++){
            pcount[p[i] - 'a']++;
        }
        for(int i = 0; i < p.size(); i++){
            windowcount[s[i] - 'a']++;
        }
        for(int i = p.size(); i <= s.size(); i++){
            if(windowcount == pcount){
                ans.push_back(i - p.size());
            }
            if(i < s.size()) {
                windowcount[s[i-p.size()] - 'a']--;
                windowcount[s[i] - 'a']++;
            }
        }
        return ans;
    }
};

```

本日遇到的问题

1. 最初不太理解协议和接口的概念，不理解什么是非正式协议

明日学习计划

1. 开始学习面向对象下的自动应用计数，@autoreleasepool块
2. 将每日新学的内容整理为博客