

438 找到字符串中所有字母异位词

Label: 滑动窗口、Hash

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。

异位词 指字母相同，但排列不同的字符串。

输入: $s = \text{"cbaebabacd"}, p = \text{"abc"}$

输出: $[0,6]$

输入: $s = \text{"abab"}, p = \text{"ab"}$

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。

$1 \leq s.length, p.length \leq 3 * 10^4$

s 和 p 仅包含小写字母

- 暴力遍历

```
class Solution {
    public List<Integer> findAnagrams(String s, String p) {
        List<Integer> re = new ArrayList<>();
        Map<Character, Integer> map = new HashMap<>();

        for (int i = 0; i < p.length(); i++) { // 统计
            map.put(p.charAt(i), map.getOrDefault(p.charAt(i), 0) + 1);
        }

        for (int i = 0; i < s.length() - p.length() + 1; i++) {
            Map<Character, Integer> newMap = new HashMap<>(map);
            String subStr = s.substring(i, i + p.length());

            for (int j = 0; j < subStr.length(); j++) {
                Character c = subStr.charAt(j);
                if (newMap.containsKey(c)) {
                    if (newMap.get(c) == 1) {
                        newMap.remove(c);
                    } else {
                        newMap.put(c, newMap.get(c) - 1);
                    }
                } else break;
            }

            if (newMap.size() == 0) re.add(i);
        }
        return re;
    }
}
```

- 滑动窗口 + 数组计数

```
class Solution {
    public List<Integer> findAnagrams(String s, String p) {
        int n = s.length(), m = p.length();
        List<Integer> res = new ArrayList<>();
        if(n < m) return res;
        int[] pCnt = new int[26]; // 记录 p 的频次
        int[] sCnt = new int[26]; // 记录 s 前m的频次

        for (int i = 0; i < m; i++) { // 统计
            pCnt[p.charAt(i) - 'a']++;
            sCnt[s.charAt(i) - 'a']++;
        }

        if(Arrays.equals(sCnt, pCnt)){
            res.add(0);
        }

        for (int i = m; i < n; i++) {
            sCnt[s.charAt(i - m) - 'a']--; // 除去左侧旧字母
            sCnt[s.charAt(i) - 'a']++;    // 添加右侧新字母

            if (Arrays.equals(sCnt, pCnt)){
                res.add(i - m + 1);
            }
        }
        return res;
    }
}
```