第 5 章

字符串问题

判断两个字符串是否互为变形词

【题目】

给定两个字符串 str1 和 str2，如果 str1 和 str2 中出现的字符种类一样且每种字符出现的次数也一样，那么 str1 与 str2 互为变形词。请实现函数判断两个字符串是否互为变形词。

【举例】

str1="123"，str2="231"，返回 true。

str1="123"，str2="2331"，返回 false。

【难度】

士 ★☆☆☆

【解答】

如果字符串 str1 和 str2 长度不同，直接返回 false。如果长度相同，假设出现字符的编码值在 0~255 之间，那么先申请一个长度为 256 的整型数组 map，map[a]=b 代表字符编码为 a 的字符出现了 b 次，初始时 map[0..255]的值都是 0。然后遍历字符串 str1，统计每种字符出现的数量，比如遍历到字符'a'，其编码值为 97，则令 map[97]++。这样 map 就成了 str1 中每种字符的词频统计表。然后遍历字符串 str2，每遍历到一个字符都在 map 中把词频减

下来，比如遍历到字符'a'，其编码值为 97，则令 map[97]--，如果减少之后的值小于 0，直接返回 false。如果遍历完 str2，map 中的值也没出现负值，则返回 true。

具体请参看如下代码中的 isDeformation 方法。

```
public boolean isDeformation(String str1, String str2) {
        if (str1 == null || str2 == null || str1.length() != str2.length()) {
                return false;
        }
        char[] chas1 = str1.toCharArray();
        char[] chas2 = str2.toCharArray();
        int[] map = new int[256];
        for (int i = 0; i < chas1.length; i++) {
                map[chas1[i]]++;
        }
        for (int i = 0; i < chas2.length; i++) {
                if (map[chas2[i]]-- == 0) {
                        return false;
                }
        }
        return true;
}
```

如果字符的类型很多，可以用哈希表代替长度为 256 的整型数组，但整体过程不变。如果字符的种类为 M，str1 和 str2 的长度为 N，那么该方法的时间复杂度为 $O(N)$，额外空间复杂度为 $O(M)$。

字符串中数字子串的求和

【题目】

给定一个字符串 str，求其中全部数字串所代表的数字之和。

【要求】

1．忽略小数点字符，例如"A1.3"，其中包含两个数字 1 和 3。

2．如果紧贴数字子串的左侧出现字符"-"，当连续出现的数量为奇数时，则数字视为负，连续出现的数量为偶数时，则数字视为正。例如，"A-1BC--12"，其中包含数字为-1 和 12。

【举例】

str="A1CD2E33"，返回 36。