

[leetcode 双周赛 6] 1153 字符串转化

阅读目录(Content)

- 1153 String Transforms Into Another String 字符串转化
 - 描述
 - 思路
 - 代码实现
 - 使用两个HashMap来构建映射表
 - 使用数组来构建映射表

回到顶部(go to top).

1153 String Transforms Into Another String 字符串转化

描述

给出两个长度相同的字符串，分别是 `str1` 和 `str2`。请你帮忙判断字符串 `str1` 能不能在 `零次` 或 `多次` 转化后变成字符串 `str2`。每一次转化时，将会一次性将 `str1` 中出现的 `所有` 相同字母变成其他 `任何` 小写英文字母（见示例）。

只有在字符串 `str1` 能够通过上述方式顺利转化为字符串 `str2` 时才能返回 `True`，否则返回 `False`。

- 示例 1：

输入：str1 = "aabcc", str2 = "ccdee"
输出：true
解释：将 'c' 变成 'e'，然后把 'b' 变成 'd'，接着再把 'a' 变成 'c'。注意，转化的顺序也很重要。

- 示例 2：

输入：str1 = "leetcode", str2 = "codeleet"
输出：false
解释：我们没有办法能够把 str1 转化为 str2。

- 提示：

1 <= str1.length == str2.length <= 10^4
str1 和 str2 中都只会出现 `小写英文字母`

思路

`转化` 的定义: 一次性将字符串中出现的 `所有` `相同字母` 变成其他 `任何` `小写英文字母`
例如:aabcc --> ccbcc(a转化成c) --> eebee(c转化成e)

公告

昵称：slowbird
园龄：1年
粉丝：2
关注：0
[+加关注](#)

<	2020年			
日	一	二	三	
29	30	31	1	
5	6	7	8	
12	13	14	15	
19	20	21	22	
26	27	28	29	
3	4	5	6	

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

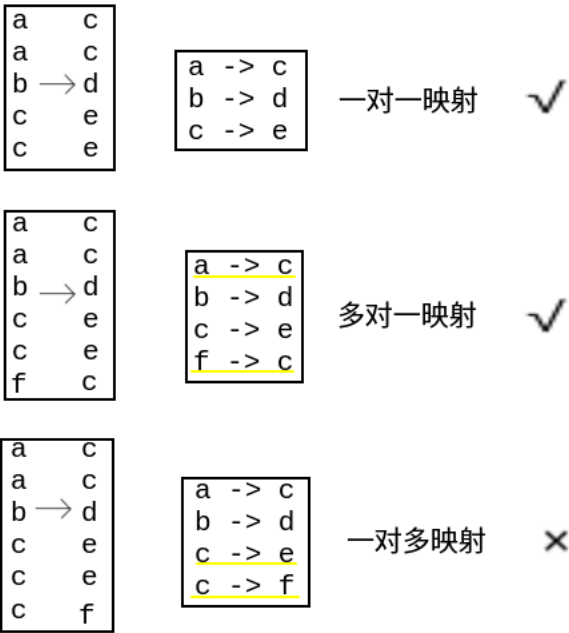
我的标签

算法(39)
题解(39)
leetcode(36)
leetcode 周赛(34)
数据库(10)
SQL(8)
练习(6)
剑指Offer(3)
集群(2)
Redis(2)
更多

1153 String Transforms Into ...

描述
思路
代码实现

3)

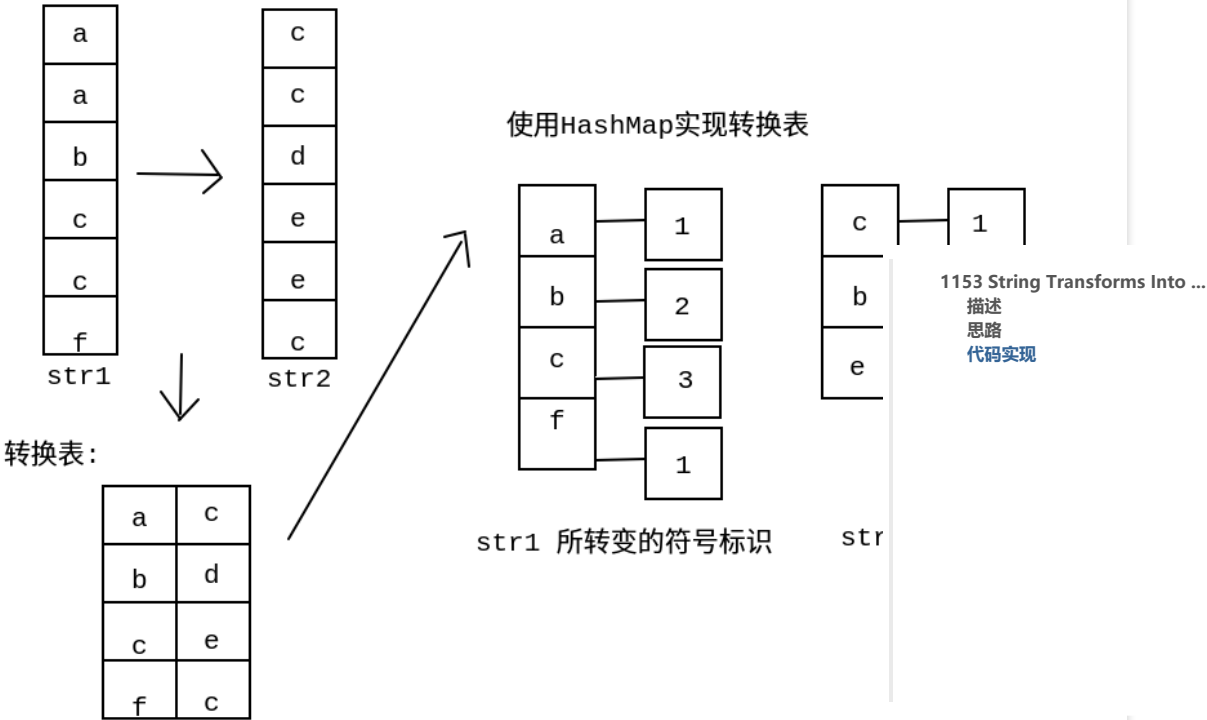


由上图可以看出:
转化的真正意义是 **多对一映射** 如此可以得出以下判断依据:

- 判断依据一: **字符种类**
转化后的字符种类必定小于或者刚好等于转化前的字符种类
注意:这种映射是**有限制**的, 它只能转化成 `小写字母`, 即它只能在26个小写字母之间转化,
所以目标字符串的字符种类**不能为26**(满值), 除非两字符串相等, 否则必定无法转化
- 判断依据二: **映射表**
可以建立映射表, 查看是否为多对一的映射

代码实现

- 使用两个HashMap来构建映射表



阅读排行榜1. [leetcode 双周赛]2. [Redis] 使用redi:3. [Redis] 手动搭建4. [leetcode 双周赛]5. [leetcode 周赛 1]

评论排行榜1. [牛客数据库SQL3]

推荐排行榜1. [Redis] 手动搭建

```
class Solution {
    public boolean canConvert(String str1, String str2) {
        if (null == str1 | null == str2) return false;
        // 两字符串长度不一必定无法转化
        if (str1.length() != str2.length()) return false;
        // 两字符串相等 无需转化
        if (str1.equals(str2)) return true;

        // 求元素种类
        HashSet<Character> set = new HashSet<>();
        for (char c : str1.toCharArray()) set.add(c);
        int size1 = set.size();
        set.clear();
        for (char c : str2.toCharArray()) set.add(c);
        int size2 = set.size();
        // 转换字符串str2元素种类满26 必定无法转化
        // 转化后字符串元素种类比转化前还多
        if (size2 == 26 | size1 < size2) return false;

        // 建立映射表
        // str1ToStr2 存放被转化字符串的字符及其转化后字符所对应的标识label 字符 -> label 多对一
        HashMap<Character, Integer> str1ToStr2 = new HashMap<>();
        // str2Table 存放转化后字符所对应的标识label 字符 -> label 一对一
        HashMap<Character, Integer> str2Table = new HashMap<>();
        // label 转化后字符所对应的标识label
        int label = 0;
        // 遍历字符
        char c1, c2;
        for (int i = 0; i < str1.length(); i++) {
            c1 = str1.charAt(i);
            c2 = str2.charAt(i);

            // c1 --> c2
            // c2没有被标识
            if (!str2Table.containsKey(c2)) {
                // c1已经有转化对应字符了 c1!->c2 多对多
                if (str1ToStr2.containsKey(c1)) return false;
                // c1 -> label
                str1ToStr2.put(c1, label);
                // c2 -> label
                str2Table.put(c2, label);
                label++;
            }
            // c2已经被标识 c1没有被标识 (还没有确定转化字符)
            if (!str1ToStr2.containsKey(c1)) {
                // c1 -> label -> c2
                str1ToStr2.put(c1, str2Table.get(c2));
            }
            // c1,c2都已经被标识 查看两者标识是否相同
            // c1 -> label ?-> c2
            if (str1ToStr2.get(c1) != str2Table.get(c2)) {
                return false;
            }
        }
        return true;
    }
}
```

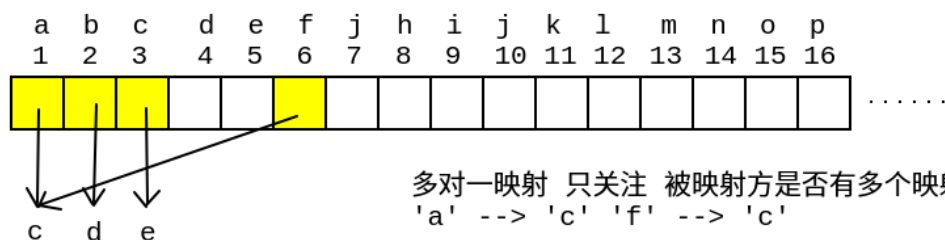
1153 String Transforms Into ...

描述

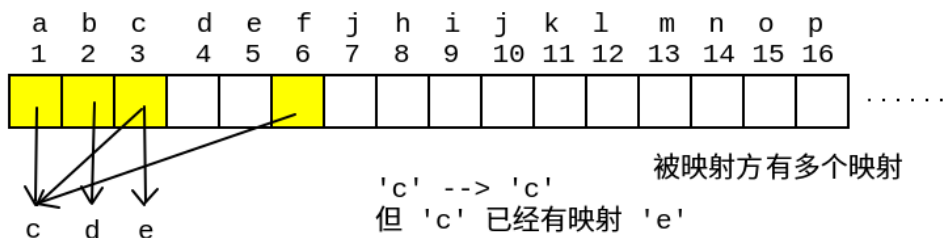
思路

代码实现

aabbccfc --> ccdeecc



"aabbccfc" --> "ccdeecc"



使用数组来构建映射表

```

class Solution {
    public boolean canConvert(String str1, String str2) {
        if (null == str1 || null == str2) return false;
        // 两字符串长度不一必定无法转化
        if (str1.length() != str2.length()) return false;
        // 两字符串相等 无需转化
        if (str1.equals(str2)) return true;

        // 求元素种类
        int n = str1.length(), size1 = 0, size2 = 0, p = 0, q = 0;
        boolean[] v1 = new boolean[30];
        boolean[] v2 = new boolean[30];
        for (int i = 0; i < n; i++) {
            p = str1.charAt(i) - 'a';
            q = str2.charAt(i) - 'a';
            if (!v1[p]) {
                size1++;
                v1[p] = true;
            }
            if (!v2[q]) {
                size2++;
                v2[q] = true;
            }
        }
        // 转换字符串str2元素种类满26 必定无法转化
        // 转化后字符串元素种类比转化前还多
        if (size2 == 26 || size1 < size2) return false;

        // 建立映射表
        // s1ToS2 映射表 下标表示原字符-'a' 值表示目标字符
        char[] s1ToS2 = new char[30];
        char c = '#';
        // 映射表初始化为全'#'
        for (int i = 0; i < 26; i++) s1ToS2[i] = '#';
        for (int i = 0; i < n; i++) {
            p = str1.charAt(i) - 'a';
            c = str2.charAt(i);

            // 该字符p还没有映射
            // 或者
            // 该字符p映射已经映射 且映射单一
            if (s1ToS2[p] == '#' || s1ToS2[p] == c) s1ToS2[p] = c;
            else return false;
        }

        return true;
    }
}

```

1153 String Transforms Into ...

描述
思路
代码实现

slowbird

博客园 首页 新随笔 联系 管理

随笔 - 49 文

好文要顶 关注我 收藏该文





slowbird

关注 - 0

粉丝 - 2

+加关注

« 上一篇 : [leetcode 双周赛 6] 1152 用户网站访问行为分析

» 下一篇 : [leetcode 周赛 149] 1154 一年中的第几天

posted @ 2019-08-15 11:02 slowbird 阅读(793) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)， [访问](#) 网站首页。

Copyright © 2020 slowbird

Powered by .NET Core on Kubernetes

1153 String Transforms Into ...

描述

思路

代码实现

https://www.cnblogs.com/slowbirdoflsh/p/11356674.html

5/5