**难度: 简单**

实现一个算法，确定一个字符串 s 的所有字符是否全都不同。

示例 1：

输入: s = "leetcode"

输出: false

示例 2：

输入: s = "abc"

输出: true

限制：

0 <= len(s) <= 100

如果你不使用额外的数据结构，会很加分

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/check-permutation-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1:**

**从第一个开始循环逐一比较,如果其中有一个相同则直接结束循环,返回false**

**否则当循环结束,则表示全部都不相同,则返回true.**

|  |
| --- |
| **public boolean isUnique(String astr) {**  **//第一个循环用于整体的字符串的循环**  **for (int i = 0; i < astr.length(); i++) {**  **//第二个循环 i+ 1 是因为每次都是前一个与后一个进行比较,不需要和本身进行比较**  **for (int j = i + 1; j < astr.length(); j++) {**  **//如果相同直接返回false**  **if (astr.charAt(i) == astr.charAt(j)) {**  **return false;**  **}**  **}**  **}**  **//循环结束表示没有相同的**  **return true;**  **}** |

**思路2:**

**利用Hash中的子类HashSet进行添加, 然后判断里面的的个数是否增加…如果增加则表示不一样,然后里面的个数没有增加则表示是重复的**

|  |
| --- |
| **//创建一个HashSet**  **HashSet<Character> set = new HashSet<Character>();**  **int count = 0; //计算正常添加的个数**  **for (int i = 0; i < astr.length(); i++) {**  **set.add(astr.charAt(i));**  **count++;**  **//判断是否真的添加了**  **if (set.size() != count) {**  **return false;**  **}**  **}**  **return true;**  **}** |