**难度: 简单**

给定一个字符串，编写一个函数判定其是否为某个回文串的排列之一。

回文串是指正反两个方向都一样的单词或短语。排列是指字母的重新排列。

回文串不一定是字典当中的单词。

示例1：

输入："tactcoa"

输出：true（排列有"tacocat"、"atcocta"，等等）

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/palindrome-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:**

1). 创建一个HashSet集合

2). 循环把字符串的数组添加到集合里

3). 如果添加到HashSet集合之后,长度没有变化,则表示相同,移除

4). 最后看一下HashSet集合的长度是否为1或者为0,如果是,则是回文串

|  |
| --- |
| public boolean canPermutePalindrome(String s) {  //创建一个HashSet集合  Set<Character> set = new HashSet<Character>();  for (int i = 0; i < s.length(); i++) {  //开始的长度  int startSize = set.size();  set.add(s.charAt(i));  //比较两则的长度是否相同  if (set.size() == startSize) {  set.remove(s.charAt(i));  }  }  //比较长度是否为1或者为0  return set.size() == 1 || set.size() == 0;  } |

**思路2:**

1). 创建一个数组,存储相同字符的个数

2). 循环存储字符串

3). 循环比较存储的字符串是否有剩余

4). 当字符串剩余的个数累加大于1说明有两个不同的直接返回false;

5). 当循环结束则表示字符串剩余的个数累加不大于1,直接返回true

|  |
| --- |
| public boolean canPermutePalindrome2(String s) {  //创建一个数组,存储相同字符的个数  int[] counts = new int[128];  //循环存储  for (int i = 0; i < s.length(); i++) {  counts[s.charAt(i)]++;  }  //标记个数  int flag = 0;  //循环比较  for (int i = 0; i < counts.length; i++) {  //如果是奇数,说明还有剩余  if (counts[i] % 2 == 1) {  flag++;  }  //只要剩余的个数大于1,就说明不是回文数了  if (flag > 1) {  return false;  }  }  return true;  } |