**难度: 简单**

给定一个赎金信 (ransom) 字符串和一个杂志(magazine)字符串，判断第一个字符串 ransom 能不能由第二个字符串 magazines 里面的字符构成。如果可以构成，返回 true ；否则返回 false。

(题目说明：为了不暴露赎金信字迹，要从杂志上搜索各个需要的字母，组成单词来表达意思。杂志字符串中的每个字符只能在赎金信字符串中使用一次。)

注意：

你可以假设两个字符串均只含有小写字母。

canConstruct("a", "b") -> false

canConstruct("aa", "ab") -> false

canConstruct("aa", "aab") -> true

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/ransom-note

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(代码有详细的注释):**

|  |
| --- |
| public boolean canConstruct(String ransomNote, String magazine) {  //比较一下长度,当我们的信的字符大于杂志的,就可以肯定是false了  if(ransomNote.length() > magazine.length()) {  return false;  }  //分别创建两个存放字符的的字符数组  int[] ransomCharCount = new int[128];  int[] magazineCharCount = new int[128];  //把ransomNote的字符全部放到ransomCharCount中  for (int i = 0; i < ransomNote.length(); i++) {  ransomCharCount[ransomNote.charAt(i)]++;  }  //把magazine的字符全部放到magazineCharCount中  for (int i = 0; i < magazine.length(); i++) {  magazineCharCount[magazine.charAt(i)]++;  }  //a是97,z是122  for (int i = 97; i < 128; i++) {  //比较两个字符数组的大小,当赎金信的字符大于杂志的字符就可以直接返回false  if (ransomCharCount[i] > magazineCharCount[i]) {  return false;  }  }  return true;  } |

**思路2(进行了小部分的优化,把三个for循环变成两个,两个数组变成一个,但是好像也快不了多少):**

**代码有注释**

|  |
| --- |
| public boolean canConstruct2(String ransomNote, String magazine) {  //比较一下长度,当我们的信的字符大于杂志的,就可以肯定是false了  if(ransomNote.length() > magazine.length()) {  return false;  }  //分别创建两个存放字符的的字符数组  int[] resultCharCount = new int[128];  //把magazine的字符全部放到resultCharCountt中  for (int i = 0; i < magazine.length(); i++) {  resultCharCount[magazine.charAt(i)]++;  }  for (int i = 0; i < ransomNote.length(); i++) {  //用ransomNote的字符来减resultCharCount的字符  resultCharCount[ransomNote.charAt(i)]--;  //当resultCharCount小于0说明已经不存在了,所以直接返回false  if (resultCharCount[ransomNote.charAt(i)] < 0) {  return false;  }  }  return true;  } |