Description | Solution | Discuss (636) | Submissions

266. Palindrome Permutation

Easy 748 64 Add to List Share

Given a string `s`, return `true` if a permutation of the string could form a palindrome.

Example 1:

```
Input: s = "code"
Output: false
```

Example 2:

```
Input: s = "aab"
Output: true
```

Example 3:

```
Input: s = "carerac"
Output: true
```

Constraints:

- `1 <= s.length <= 5000`
- `s` consists of only lowercase English letters.

Accepted 144,701 | Submissions 225,436

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Facebook | 12 Microsoft | 2

Related Topics

Hash Table String Bit Manipulation

Similar Questions

Longest Palindromic Substring	Medium
Valid Anagram	Easy
Palindrome Permutation II	Medium
Longest Palindrome	Easy

Hide Hint 1

Consider the palindromes of odd vs even length. What difference do you notice?

Hide Hint 2

Count the frequency of each character.

Hide Hint 3

If each character occurs even number of times, then it must be a palindrome. How about character which occurs odd number of times?

Java Autocomplete

```java
public class Solution {
 public boolean canPermutePalindrome(String s) {
     HashMap < Character, Integer > map = new HashMap < > ();
     for (int i = 0; i < s.length(); i++) {
         map.put(s.charAt(i), map.getOrDefault(s.charAt(i), 0) + 1);
     }
     int count = 0;
     for (char key: map.keySet()) {
         count += map.get(key) % 2;
     }
     return count <= 1;
 }
}


public class Solution {
    public boolean canPermutePalindrome(String s) {
        int count = 0;
        for (char i = 0; i < 128 && count <= 1; i++) {
            int ct = 0;
            for (int j = 0; j < s.length(); j++) {
                if (s.charAt(j) == i)
                    ct++;
            }
            count += ct % 2;
        }
        return count <= 1;
    }
}
```