916 · Palindrome Permutation

Description

Given a string, determine if a permutation of the string could form a palindrome.

Example

Example1

```
Input: s = "code"
Output: False
Explanation:
No solution
```

Example2

```
Input: s = "aab"
Output: True
Explanation:
"aab" --> "aba"
```

Example3

```
Input: s = "carerac"
Output: True
Explanation:
"carerac" --> "carerac"
```

```
class Solution:
    """
    @param s: the given string
    @return: if a permutation of the string could form a palindrome
    """
    def canPermutePalindrome(self, s):
        # write your code here
        freqmap = {}
        for ele in s:
            if ele in freqmap:
                freqmap[ele] = freqmap[ele]+1
            else:
                freqmap[ele]=1
        oddChar = ''
```

```
        odd = 0
        count=0
        for key in freqmap.keys():
            if freqmap[key]%2!=0:
                oddChar = key
                freqmap[key] = freqmap[key]//2
                if freqmap[key]!=0:
                    count = freqmap[key]+count
                odd = odd+1
            else:
                freqmap[key] = freqmap[key]//2
                count = freqmap[key]+count
        if odd>1:
            return False
        return True
```