# 题目

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）

# 分析

## 方法一：哈希表

**思路：**

哈希表统计每个字符出现次数，如果是回文串的排列，则字符出现次数为奇数的个数不能大于1。

**代码：**

class Solution {

public:

bool canPermutePalindrome(string s) {

vector<int> hash(256, 0);

for (auto c : s) {

++hash[c];

}

int cnt = 0;

for (int i = 0; i < hash.size(); ++i) {

if (hash[i] % 2 == 1) ++cnt;

}

return cnt <= 1;

}

};

## 方法二：位运算

**思路：**

偶数次出现次数对应的bit为0，奇数次则为1。

最后只需要统计bitset中1的个数是否是0或1即可。

**代码：**

class Solution {

public:

bool canPermutePalindrome(string s) {

bitset<128> bits;

for (char c : s) {

bits.flip(c);

}

return bits.none() || bits.count() == 1;

}

};