# 题目

输入一个整数数组，实现一个函数来调整该数组中数字的顺序，使得所有奇数位于数组的前半部分，所有偶数位于数组的后半部分。

**示例：**

输入：nums = [1,2,3,4]

输出：[1,3,2,4]

注：[3,1,2,4] 也是正确的答案之一。

**提示：**

1 <= nums.length <= 50000

1 <= nums[i] <= 10000

类似题目：Leetcode 905（奇偶顺序不一样）

# 分析

## 方法一：双指针

**思路：**

**代码：**

class Solution {

public:

vector<int> exchange(vector<int>& nums) {

int i = 0, j = nums.size()-1;

while(i<j){

while(nums[i]%2==1 && i<j) i++;

while(nums[j]%2==0 && i<j) j--;

swap(nums[i],nums[j]);

}

return nums;

}

};

## 方法二：两个数组

**思路：**

**代码：**

class Solution {

public:

vector<int> exchange(vector<int>& nums) {

vector<int> ji,ou;

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

{

if(nums.at(i)%2==0)

{

ou.push\_back(nums.at(i));

}

else

{

ji.push\_back(nums.at(i));

}

}

ji.insert(ji.end(),ou.begin(),ou.end());

return ji;

}

};