

# 面试题21：调整数组顺序使奇数位于偶数前面

## 题目描述

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，使得所有的奇数位于数组的前半部分，所有的偶数位于数组的后半部分，并保证奇数和奇数，偶数和偶数之间的相对位置不变。

## 解题思路

再创建一个数组

## Java

```
1  /*
2  相对位置不变--->保持稳定性；奇数位于前面，偶数位于后面 --->存在判断，挪动元素位置；
3  这些都和内部排序算法相似，考虑到具有稳定性的排序算法不多，例如插入排序，归并排序等；这里采用插入排序的思想实现。
4  */
5  public class Solution {
6      public void reOrderArray(int [] array) {
7          //相对位置不变，稳定性
```

```

8      //插入排序的思想
9      int m = array.length;
10     int k = 0; //记录已经摆好位置的奇数的个数
11     for(int i = 0;i < m;i++){
12         if(array[i] % 2 == 1){
13             int j = i;
14             while(j > k){ //j >= k + 1
15                 int tmp = array[j];
16                 array[j] = array[j - 1];
17                 array[j-1] = tmp;
18                 j--;
19             }
20             k++;
21         }
22     }
23 }
24 }

```

## C++

```

1  //再创建一个数组
2  class Solution {
3  public:
4      void reOrderArray(vector<int> &array) {
5          vector<int> array_temp;
6          vector<int>::iterator ib1,ie1;

```

```

7         ib1 =array.begin();
8
9         for(;ib1 != array.end();){ //遇见偶数，就
保存到新数组，同时从原数组中删除
10             if(*ib1 % 2 == 0){
11                 array_temp.push_back(*ib1);
12                 ib1 = array.erase(ib1);
13             }else{
14                 ib1++;
15             }
16         }
17         vector<int>::iterator ib2,ie2;
18         ib2 = array_temp.begin();
19         ie2 = array_temp.end();
20         for(;ib2 != ie2;ib2++){ //将新数组的数添加
到老数组
21             array.push_back(*ib2);
22         }
23     }
24 };

```

## Python2.7

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def reOrderArray(self, array):
4         # write code here
5         #return sorted(array,key=lambda
c:c%2,reverse=True) 一行解法, 使用lambda表达式
6         odd,even=[],[]
7         for i in array:
8             odd.append(i) if i%2==1 else
even.append(i)
9         return odd+even
```