# 21. 调整数组顺序使奇数位于偶数前面

## 题目链接

[牛客网](https://www.nowcoder.com/practice/ef1f53ef31ca408cada5093c8780f44b?tpId=13&tqId=11166&tab=answerKey&from=cyc_github)

## 题目描述

需要保证奇数和奇数，偶数和偶数之间的相对位置不变，这和书本不太一样。例如对于 [1,2,3,4,5]，调整后得到 [1,3,5,2,4]，而不能是 {5,1,3,4,2} 这种相对位置改变的结果。

## 解题思路

方法一：创建一个新数组，时间复杂度 O(N)，空间复杂度 O(N)。

public int[] reOrderArray (int[] nums) {
 // 奇数个数
 int oddCnt = 0;
 for (int x : nums)
 if (!isEven(x))
 oddCnt++;
 int[] copy = nums.clone();
 int i = 0, j = oddCnt;
 for (int num : copy) {
 if (num % 2 == 1)
 nums[i++] = num;
 else
 nums[j++] = num;
 }
 return nums;
}

private boolean isEven(int x) {
 return x % 2 == 0;
}

方法二：使用冒泡思想，每次都将当前偶数上浮到当前最右边。时间复杂度 O(N2)，空间复杂度 O(1)，时间换空间。

public int[] reOrderArray(int[] nums) {
 int N = nums.length;
 for (int i = N - 1; i > 0; i--) {
 for (int j = 0; j < i; j++) {
 if (isEven(nums[j]) && !isEven(nums[j + 1])) {
 swap(nums, j, j + 1);
 }
 }
 }
 return nums;
}

private boolean isEven(int x) {
 return x % 2 == 0;
}

private void swap(int[] nums, int i, int j) {
 int t = nums[i];
 nums[i] = nums[j];
 nums[j] = t;
}