



链滴

调整数组顺序使奇数位于偶数前面

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518784217073>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

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

- 在原数组进行调整，时间复杂度较大 $O(n\log(n))$ ；
 - `index`保存第一个偶数的索引，一旦找到奇数，就把`index`和奇数中的所有数字向后移，`index`位设为此奇数，然后`index++`。
- 创建辅助数组，空间复杂度较大 $O(n)$ ，时间复杂度较小 $O(n)$ 。
 - 两头一起找，如果前面是奇数，存入辅助数组的前面；
 - 如果后面是偶数，存入辅助数组的后面；
 - 然后把辅助数组的数赋给原数组。

原数组：

```
public class Solution {
    public void reOrderArray(int [] array) {
        if (array == null || array.length == 0)
            return;
        int index = 0;
        for (int i = 0; i < array.length; i++) {
            if (array[i] % 2 == 1) {
                swap(array, index, i);
                index++;
            }
        }
    }

    private void swap(int[] array, int index, int i) {
        int tmp = array[i];
        for (int j = i; j > index; j--)
            array[j] = array[j-1];
        array[index] = tmp;
    }
}
```

辅助数组：

```
public class Solution {
    public void reOrderArray(int [] array) {
        int[] ret = new int[array.length];
        int index = 0;
        int odd = 0;
        int even = array.length-1;
        while (index < array.length) {
            if (array[index] % 2 == 1) {
                ret[odd] = array[index];
            }
        }
    }
}
```

```
        odd++;
    }
    if (array[array.length-index-1] % 2 == 0) {
        ret[even] = array[array.length-index-1];
        even--;
    }
    index++;
}
for (int i = 0; i < array.length; i++)
    array[i] = ret[i];
}
```