# 3. 数组中重复的数字

## 题目链接

[牛客网](https://www.nowcoder.com/practice/6fe361ede7e54db1b84adc81d09d8524?tpId=13&tqId=11203&tab=answerKey&from=cyc_github)

## 题目描述

在一个长度为 n 的数组里的所有数字都在 0 到 n-1 的范围内。数组中某些数字是重复的，但不知道有几个数字是重复的，也不知道每个数字重复几次。请找出数组中任意一个重复的数字。

Input:
{2, 3, 1, 0, 2, 5}

Output:
2

## 解题思路

要求时间复杂度 O(N)，空间复杂度 O(1)。因此不能使用排序的方法，也不能使用额外的标记数组。

对于这种数组元素在 [0, n-1] 范围内的问题，可以将值为 i 的元素调整到第 i 个位置上进行求解。在调整过程中，如果第 i 位置上已经有一个值为 i 的元素，就可以知道 i 值重复。

以 (2, 3, 1, 0, 2, 5) 为例，遍历到位置 4 时，该位置上的数为 2，但是第 2 个位置上已经有一个 2 的值了，因此可以知道 2 重复：

public int duplicate(int[] nums) {
 for (int i = 0; i < nums.length; i++) {
 while (nums[i] != i) {
 if (nums[i] == nums[nums[i]]) {
 return nums[i];
 }
 swap(nums, i, nums[i]);
 }
 swap(nums, i, nums[i]);
 }
 return -1;
}

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