# 51. 数组中的逆序对

[NowCoder](https://www.nowcoder.com/practice/96bd6684e04a44eb80e6a68efc0ec6c5?tpId=13&tqId=11188&tPage=1&rp=1&ru=/ta/coding-interviews&qru=/ta/coding-interviews/question-ranking&from=cyc_github)

## 题目描述

在数组中的两个数字，如果前面一个数字大于后面的数字，则这两个数字组成一个逆序对。输入一个数组，求出这个数组中的逆序对的总数。

## 解题思路

private long cnt = 0;
private int[] tmp; // 在这里声明辅助数组，而不是在 merge() 递归函数中声明

public int InversePairs(int[] nums) {
 tmp = new int[nums.length];
 mergeSort(nums, 0, nums.length - 1);
 return (int) (cnt % 1000000007);
}

private void mergeSort(int[] nums, int l, int h) {
 if (h - l < 1)
 return;
 int m = l + (h - l) / 2;
 mergeSort(nums, l, m);
 mergeSort(nums, m + 1, h);
 merge(nums, l, m, h);
}

private void merge(int[] nums, int l, int m, int h) {
 int i = l, j = m + 1, k = l;
 while (i <= m || j <= h) {
 if (i > m)
 tmp[k] = nums[j++];
 else if (j > h)
 tmp[k] = nums[i++];
 else if (nums[i] <= nums[j])
 tmp[k] = nums[i++];
 else {
 tmp[k] = nums[j++];
 this.cnt += m - i + 1; // nums[i] > nums[j]，说明 nums[i...mid] 都大于 nums[j]
 }
 k++;
 }
 for (k = l; k <= h; k++)
 nums[k] = tmp[k];
}