# 数组中的逆序对

## 题目

[牛客网](https://www.nowcoder.com/practice/96bd6684e04a44eb80e6a68efc0ec6c5?tpId=13&tqId=11188&tPage=2&rp=2&ru=%2Fta%2Fcoding-interviews&qru=%2Fta%2Fcoding-interviews%2Fquestion-ranking)

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P。并将P对1000000007取模的结果输出。 即输出P%1000000007

输入描述: 题目保证输入的数组中没有的相同的数字

数据范围：

 对于%50的数据,size<=10^4

 对于%75的数据,size<=10^5

 对于%100的数据,size<=2\*10^5

## 解题思路

1. 使用归并排序的方式，划分子数组
2. 两个子数组进行对比，有两个分别指向两个数组末尾的指针 `f,s`，数组分割下标为 `mid`，如果 `array[f] > array[s]`那么，就有`s - mid`个 `array[f]` 的逆序
3. 依此类推，最终将数组排序，并且获得结果

public int InversePairs(int[] array) {
long[] sum = {0};
if (array == null || array.length == 0) {
 return (int) sum[0];
}

int[] temp = new int[array.length];
mergeSort(array, 0, array.length - 1, temp, sum);
return (int) (sum[0] % 1000000007);
}

private void mergeSort(int[] array, int start, int end, int[] temp, long[] sum) {
if (start == end) {
 return;
}

int mid = (start + end) / 2;

mergeSort(array, start, mid, temp, sum);
mergeSort(array, mid + 1, end, temp, sum);

int f = mid, s = end;
int t = end;
while (f >= start && s >= mid + 1) {
 if (array[f] > array[s]) {
 temp[t--] = array[f--];
 sum[0] += s - mid;
 } else {
 temp[t--] = array[s--];
 }
}

while (f >= start) {
 temp[t--] = array[f--];
}

while (s >= mid + 1) {
 temp[t--] = array[s--];
}

for (int i = end, j = end; i >= start; ) {
 array[j--] = temp[i--];
}
}