

面试题 51：数组中的逆序对

题目：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。例如，在数组{7, 5, 6, 4}中，一共存在 5 个逆序对，分别是(7, 6)、(7, 5)、(7, 4)、(6, 4)和(5, 4)。

解 归并排序 $O(n\log n)$

我们先用两个指针分别指向两个子数组的末尾，并每次比较两个指针指向的数字。如果第一个子数组中的数字大于第二个子数组中的数字，则构成逆序对，并且逆序对的数目等于第二个子数组中剩余数字的个数，如图 5.3 (a) 和图 5.3 (c) 所示。如果第一个数组中的数字小于或等于第二个

数组中的数字，则不构成逆序对，如图 5.3 (b) 所示。每次比较的时候，我们都把较大的数字从后往前复制到一个辅助数组，确保辅助数组中的数字是递增排序的。在把较大的数字复制到辅助数组之后，把对应的指针向前移动一位，接下来进行下一轮比较。

逆序对总数=左边逆序对数+右边逆序对数+合并时出现的逆序对数

```
class Solution {
public:
    int InversePairs(vector<int> data) {
        if(data.size() <= 0)
            return 0;
        vector<int> copy;
        for(int i=0; i<data.size(); i++)
        {
            copy.push_back(data[i]);
        }
        //int res=0;
        return help(data, copy, 0, data.size()-1)%1000000007;
    }
    long long help(vector<int> & data, vector<int> & copy, int start, int end)
    {
```

```

if(start==end)
{
    copy[start]=data[start];
    return 0;
}
int length=(end-start)/2;
long long left=help(copy,data,start,start+length);
long long right=help(copy,data,start+length+1,end);
int i=start+length,j=end;
int index=end;
long long count=0;
while(i>=start && j>=start+length+1)
{
    if(data[i]>data[j])
    {
        copy[index--]=data[i--];
        count+=j-length-start;
    }
    else
        copy[index--]=data[j--];
}
while(i>=start)
{
    copy[index--]=data[i--];
}
while(j>=start+length+1)
    copy[index--]=data[j--];

return left+right+count;
}
};

```