

剑指Offer（三十五）：数组中的逆序对

🕒 2018年1月14日 18:26:12 🗨 14 🌡 8,890 °C 📄 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、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

输入：

1	1,2,3,4,5,6,7,0
---	-----------------

输出：

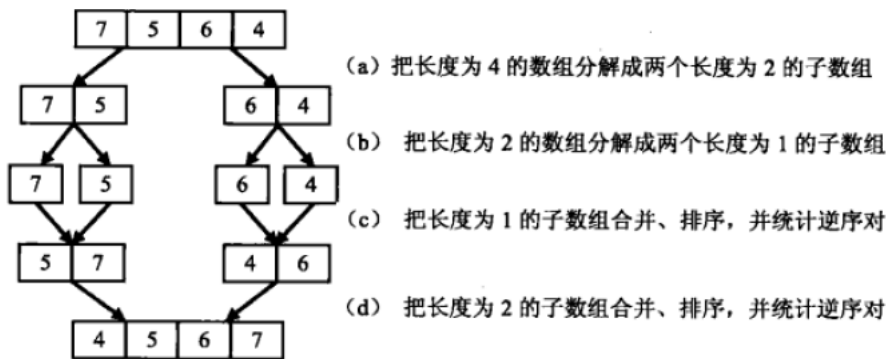
1	7
---	---

1、思路

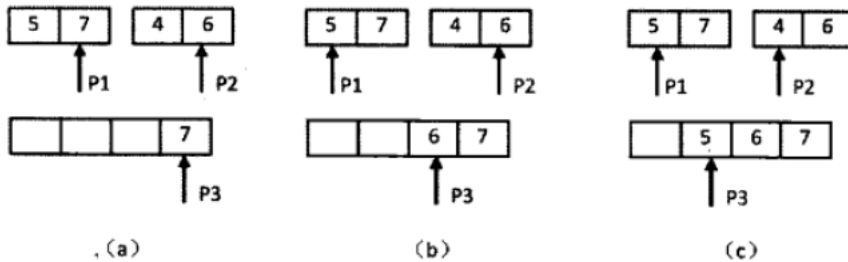
如数组{7,5,6,4}，逆序对总共有5对，{7,5}，{7,6}，{7,4}，{5,4}，{6,4}；

思路1：暴力解法，顺序扫描整个数组，每扫描到一个数字的时候，逐个比较该数字和它后面的数字的大小。如果后面的数字比它小，则这两个数字就成一个逆序对。假设数组中含有n个数字，由于每个数字都要和O(n)个数字作比较，因此这个算法的时间复杂度是O(n^2)。

思路2：分治思想，采用归并排序的思路来处理，如下图，先分后治：



先把数组分解成两个长度为2的子数组，再把这两个子数组分解成两个长度为1的子数组。接下来一边合并相邻的子数组，一边统计逆序对的数目。在第一对长度为1的子数组{7}、{5}中7>5，因此 (7,5) 组成一个逆序对。同样在第二对长度为1的子数组{6}、{4}中也有逆序对 (6,4)，由于已经统计了这两对数组内部的逆序对，因此需要把这两对子数组进行排序，避免在之后的统计过程中重复统计。



逆序对的总数 = 左边数组中的逆序对的数量 + 右边数组中逆序对的数量 + 左右结合成新的顺序数组时中出现的逆序对的数量

总结一下：

这是一个归并排序的合并过程，主要是考虑合并两个有序序列时，计算逆序对数。

对于两个升序序列，设置两个下标：两个有序序列的末尾。每次比较两个末尾值，如果前末尾大于后末尾值，则有“后序列当前长度”个逆序对；否则不构成逆序对。然后把较大值拷贝到辅助数组的末尾，即最终要将两个有序序列合并到辅助数组并有序。

这样，每次在合并前，先递归地处理左半段、右半段，则左、右半段有序，且左右半段的逆序对数可得到，再计算左右半段合并时逆序对的个数。

## 2、代码

C++：

注意：InversePairsCore形参的顺序是(data,copy)，而递归调用时实参是(copy,data)。

要明白递归函数InversePairsCore的作用就行了，它是对data的左右半段进行合并，复制到辅助数组copy中有序。

```

1 class Solution {
2 public:
3     int InversePairs(vector<int> data) {
4         if(data.size() == 0){
5             return 0;
6         }
7         // 排序的辅助数组
8         vector<int> copy;
9         for(int i = 0; i < data.size(); ++i){
10             copy.push_back(data[i]);
11         }
12         return InversePairsCore(data, copy, 0, data.size() - 1) % 1000000007;
13     }
14     long InversePairsCore(vector<int> &data, vector<int> &copy, int begin, int end){
15         // 如果指向相同位置，则没有逆序对。
16         if(begin == end){
17             copy[begin] = data[end];
18             return 0;
19         }
20         // 求中点
21         int mid = (end + begin) >> 1;
22         // 使data左半段有序，并返回左半段逆序对的数量
23         long leftCount = InversePairsCore(copy, data, begin, mid);
24         // 使data右半段有序，并返回右半段逆序对的数量
25         long rightCount = InversePairsCore(copy, data, mid + 1, end);
26
27         int i = mid; // i初始化为前半段最后一个数字的下标
28         int j = end; // j初始化为后半段最后一个数字的下标
29         int indexcopy = end; // 辅助数组复制的数组的最后一个数字的下标
30         long count = 0; // 计数，逆序对的个数，注意类型
31
32         while(i >= begin && j >= mid + 1){
33             if(data[i] > data[j]){
34                 copy[indexcopy--] = data[i--];
35                 count += j - mid;
36             }
37             else{

```

C+

```
38         copy[indexcopy--] = data[j--];
39     }
40 }
41 for(; i >= begin; --i){
42     copy[indexcopy--] = data[i];
43 }
44 for(; j >= mid + 1; --j){
45     copy[indexcopy--] = data[j];
46 }
47 return leftCount + rightCount + count;
48 }
49 };
```

### Python:

```
1  #-*- coding:utf-8 -*-
2  class Solution:
3      def InversePairs(self, data):
4          # write code here
5          if not data:
6              return 0
7          temp = [i for i in data]
8          return self.mergeSort(temp, data, 0, len(data)-1) % 1000000007
9
10     def mergeSort(self, temp, data, low, high):
11         if low >= high:
12             temp[low] = data[low]
13             return 0
14         mid = (low + high) / 2
15         left = self.mergeSort(data, temp, low, mid)
16         right = self.mergeSort(data, temp, mid+1, high)
17
18         count = 0
19         i = low
20         j = mid+1
21         index = low
22         while i <= mid and j <= high:
23             if data[i] <= data[j]:
24                 temp[index] = data[i]
25                 i += 1
26             else:
27                 temp[index] = data[j]
28                 count += mid-i+1
29                 j += 1
30             index += 1
31         while i <= mid:
32             temp[index] = data[i]
33             i += 1
34             index += 1
35         while j <= high:
36             temp[index] = data[j]
37             j += 1
38             index += 1
39         return count + left + right
```

Pytho



微信公众号

分享技术，乐享生活：微信公众号搜索

「JackCui-AI」关注一个在互联网摸爬滚  
打的潜行者。

活着就是为了改变世界，难道还有其他原因吗？ --- 乔布斯