

[作业]排序算法练习（二）

作业时间：05/10/2015

作业题目：

题号	题目	分类	难度
P1434	分治算法_求逆序对	排序算法	1
P1438	排序算法_军事机密	排序算法	1

P1434 分治算法_求逆序对

【问题描述】 给定一个序列 a_1, a_2, \dots, a_n ，如果存在 $i < j$ 并且 $a_i > a_j$ ，那么我们称之为逆序对，求逆序对的数目。	
【输入格式】 第一行为 n ，表示序列长度，接下来的 n 行，第 $i+1$ 行表示序列中的第 i 个数。	
【输出格式】 所有逆序对总数。	
【输入输出样例】	
<pre>*.in 4 3 2 3 2</pre>	<pre>*.out 3</pre>
【时间限制】 各个测试点 1s	
【来源】 思路：与归并排序联系起来。	

分析：直接枚举要 $O(N^2)$ ，显然会超时。根据题中所述，考虑下面这种情况：

1	3	5	7	9	2	4	6	8	10
---	---	---	---	---	---	---	---	---	----

其中前半部分有序，后半部分有序。有两个变量 i, j ，初始时分别指向第一个和第六个元素，和归并排序一样。

第一步： $1 < 2$ ， i 右移。

第二步： $3 > 2$ ， j 右移。注意既然有 $3 > 2$ ，那么3后面的5,7,9均大于2，一共有4个逆序对。

第三步： $3 < 4$ ， i 右移。

第四步： $5 > 4$ ， j 右移。注意既然有 $5 > 4$ ，那么5后面的7,9均大于4，一共有3个逆序对。

.....

这样算法的框架就有了。定义：

```
function exmerge(l,r:longint):int64;
```

表示区间l到r之间的逆序对个数，并且确保返回值前，这段数据已经有序。

根据归并排序的思想，首先有

```
exmerge(l,r):=exmerge(l,mid)+exmerge(mid+1,r);
```

然后做上述步骤即可，容易看出，只需要在归并排序基础上加一句话，即在j右移时累加。还有，由于规模较大，要用int64存储。

时间: $O(N \lg N)$

空间: $O(N)$

代码:

```
program P1434;
var
  n,i:longint;
  ans:int64;
  a,ls:array[1..100005] of longint;
function exmerge(l,r:longint):int64;
var
  i,j,mid,cnt:longint;
  sum:int64;
begin
  if l=r then exit(0);
  mid:=(l+r) div 2;
  sum:=exmerge(l,mid)+exmerge(mid+1,r);
  i:=l; j:=mid+1;
  cnt:=0;
  while cnt<r-l+1 do
  begin
    if a[i]<=a[j] then
    begin
      inc(cnt);
      ls[cnt]:=a[i];
      inc(i);
    end
    else
    begin
      inc(cnt);
      ls[cnt]:=a[j];
      inc(j);
      {The Next Line is IMPORTANT}
      sum:=sum+(mid-i+1);
    end;
    if i>mid then
    begin
      for j:=j to r do
      begin
        inc(cnt);
        ls[cnt]:=a[j];
      end;
    end;
    if j>r then
```

```
begin
  for i:=i to mid do
  begin
    inc(cnt);
    ls[cnt]:=a[i];
  end;
end;
end;
for i:=1 to r do
  a[i]:=ls[i-1+1];
exit(sum);
end;
begin
  readln(n);
  for i:=1 to n do
  read(a[i]);
  ans:=exmerge(1,n);
  writeln(ans);
end.
```

评测结果:

<p>评测记录号: 53041</p> <p>VijosNT Mini 2.0.5.6</p> <p>Free Pascal Compiler version 2.0.4 [2006/08/21] for i386 Copyright (c) 1993-2006 by Florian Klaempfl Target OS: Win32 for i386 Compiling foo.pas Linking foo.exe 60 Lines compiled, 0.3 sec</p> <p>#01: Accepted (93ms, 1420KB) #02: Accepted (109ms, 1420KB) #03: Accepted (187ms, 1420KB) #04: Accepted (187ms, 1420KB) #05: Accepted (156ms, 1420KB) #06: Accepted (156ms, 1420KB) #07: Accepted (109ms, 1420KB) #08: Accepted (125ms, 1420KB) #09: Accepted (125ms, 1420KB) #10: Accepted (125ms, 1420KB)</p> <p>Accepted / 100 / 1375ms / 1420KB</p>
--

P1437 排序算法_车厢重组

<p>【问题描述】</p> <p>在一个旧式的火车站旁边有一座桥，其桥面可以绕河中心的桥墩水平旋转。一个车站的职工发现桥的长度最多能容纳两节车厢，如果将桥旋转 180 度，则可以把相邻两节车厢的位置交换，用这种方法可以重新排列车厢的顺序。于是他就负责用这座桥将进站的车厢按车厢号从小到大排列。他退休后，火车站决定将这一工作自动化，其中一项重要的工作是编一个程序，输入初始的车厢顺序，计算最少用多少步就能将车厢排序。</p>
<p>【输入格式】</p> <p>输入文件有两行数据，第一行是车厢总数 N（不大于 10000），第二行是 N 个不同的数表示初始的车厢顺序。</p>

【输出格式】 一个数据，是最少的旋转次数。	
【输入输出样例】	
carry.in	carry.out
4	6
4 3 2 1	
【时间限制】 各个测试点 1s	

分析：每次交换两个相邻元素，显然是在模拟冒泡排序，我们也做一遍就行了。

时间： $O(N^2)$

空间： $O(N)$

代码：

```

program P1437;
var
  a:array[1..10005] of longint;
  i,j,n,t:longint;
  sum:int64;
begin
  readln(n);
  sum:=0;
  for i:=1 to n do
    read(a[i]);
  for i:=1 to n-1 do
    for j:=1 to n-1 do
      if a[j]>a[j+1] then
        begin
          sum:=sum+1;
          t:=a[j]; a[j]:=a[j+1]; a[j+1]:=t;
        end;
    writeln(sum);
  end.

```

评测结果：

```

评测记录号: 53042
VijosNT Mini 2.0.5.6
Free Pascal Compiler version 2.0.4 [2006/08/21] for i386
Copyright (c) 1993-2006 by Florian Klaempfl
Target OS: Win32 for i386
Compiling foo.pas
Linking foo.exe
19 Lines compiled, 0.1 sec

#01: Accepted (109ms, 680KB)
#02: Accepted (109ms, 680KB)
#03: Accepted (171ms, 680KB)
#04: Accepted (125ms, 680KB)
#05: Accepted (156ms, 680KB)
#06: Accepted (140ms, 680KB)
#07: Accepted (125ms, 680KB)
#08: Accepted (109ms, 680KB)

```

#09: Accepted (125ms, 680KB)
#10: Accepted (109ms, 680KB)
Accepted / 100 / 1281ms / 680KB

总结：这两道题都不是直接的套模板了，需要动一点脑子。其实车厢重组那道题求的也是逆序对个数，具体怎么证明不再叙述。这样时间复杂度又可以优化一个量级。逆序对问题很经典，应该引起重视。