

[作业]排序算法练习（一）

作业时间：04/21/2015

作业题目：

题号	题目	分类	难度
P1436	排序算法_众数	排序算法	1
P1438	排序算法_军事机密	排序算法	1
P1439	排序算法_输油管道	排序算法	2
P1442	排序算法_谁是组长	排序算法	2

P1436 排序算法_众数

【问题描述】 由文件给出 N 个 1 到 30000 间无序数正整数，其中 $1 \leq N \leq 10000$ ，同一个正整数可能会出现多次，出现次数最多的整数称为众数。求出它的众数及它出现的次数。							
【输入格式】 输入文件第一行是正整数的个数 N，第二行开始为 N 个正整数。							
【输出格式】 输出文件有若干行，每行两个数，第 1 个是众数，第 2 个是众数出现的次数。							
【输入输出样例】							
<table border="1"><tr><td>masses.in</td><td>masses.out</td></tr><tr><td>12</td><td>2 4</td></tr><tr><td>2 4 2 3 2 5 3 7 2 3 4 3</td><td>3 4</td></tr></table>	masses.in	masses.out	12	2 4	2 4 2 3 2 5 3 7 2 3 4 3	3 4	
masses.in	masses.out						
12	2 4						
2 4 2 3 2 5 3 7 2 3 4 3	3 4						
【时间限制】 各个测试点 1s							

分析：容易想到不用排序，直接用一个数组记录每个数字出现次数即可。

时间： $O(N + M)$ ，N 为数据个数，M 为数据中最大数；

空间： $O(M)$

代码：

<pre>program P1436; var a:array[1..30000] of longint; i,j,k,m,n:longint; begin readln(n); fillchar(a,sizeof(a),0); m:=0; for i:=1 to n do begin read(j); if j>m then m:=j; inc(a[j]); end;</pre>

```

readln;
k:=0;
for i:=1 to m do
  if a[i]>k then
    k:=a[i];
for i:=1 to m do
  if a[i]=k then
    writeln(i, ' ',k);
end.

```

评测结果:

评测记录号: [52957](#)

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
 24 Lines compiled, 0.1 sec

#01: **Accepted** (125ms, 756KB)
 #02: **Accepted** (140ms, 756KB)
 #03: **Accepted** (187ms, 756KB)
 #04: **Accepted** (31ms, 756KB)
 #05: **Accepted** (109ms, 756KB)
 #06: **Accepted** (171ms, 756KB)
 #07: **Accepted** (109ms, 756KB)
 #08: **Accepted** (109ms, 756KB)
 #09: **Accepted** (109ms, 756KB)
 #10: **Accepted** (109ms, 756KB)

Accepted / 100 / 1203ms / 756KB

P1438 排序算法_军事机密

【问题描述】

军方截获的信息由 n ($n \leq 30000$) 个数字组成, 因为是敌国的高端秘密, 所以一时不能破获。最原始的想法就是对这 n 个数进行小到大排序, 每个数都对应一个序号, 然后对第 i 个是什么数感兴趣, 现在要求编程完成。

【输入格式】

第一行 n , 接着是 n 个截获的数字, 接着一行是数字 k , 接着是 k 行要输出数的序号。

【输出格式】

k 行序号对应的数字。

【输入输出样例】

Secret.in	Secret.out
5	7
121 1 126 123 7	123
3	121
2	
4	
3	

【时间限制】

各个测试点 1s

分析：存数组里，快排。

时间：平均 $O(N \lg N + K)$

空间： $O(N)$

代码：

```
program P1438;
var
a:array[1..30000] of longint;
i,k,l,n:longint;
procedure qsort(l,r: longint);
var
  i,j,x,y: longint;
begin
  i:=l;
  j:=r;
  x:=a[(l+r) div 2];
  repeat
    while a[i]<x do
      inc(i);
    while x<a[j] do
      dec(j);
    if not(i>j) then
      begin
        y:=a[i]; a[i]:=a[j]; a[j]:=y;
        inc(i); dec(j);
      end;
  until i>j;
  if l<j then
    qsort(l,j);
  if i<r then
    qsort(i,r);
end;
begin
  readln(n);
  for i:=1 to n do
    read(a[i]);
  qsort(1,n);
  readln(k);
  for i:=1 to k do
  begin
    readln(l);
    writeln(a[l]);
  end;
end.
```

评测结果：

评测记录号：[52958](#)

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
40 Lines compiled, 0.1 sec

#01: Accepted (125ms, 756KB)
#02: Accepted (109ms, 756KB)
#03: Accepted (31ms, 756KB)
#04: Accepted (171ms, 756KB)
#05: Accepted (187ms, 756KB)
#06: Accepted (187ms, 756KB)
#07: Accepted (46ms, 756KB)
#08: Accepted (125ms, 756KB)
#09: Accepted (109ms, 756KB)
#10: Accepted (125ms, 756KB)
Accepted / 100 / 1218ms / 756KB

```

P1439 排序算法_输油管道

【问题描述】
 某石油公司计划建造一条由东向西的主输油管道。该管道要穿过一个有 n 口油井的油田。从每口油井都要有一条输油管道沿最短路径(或南或北)与主管道相连。如果给定 n 口油井的位置，即它们的 x 坐标（东西向）和 y 坐标（南北向），应如何确定主管道的最优位置，即使各油井到主管道之间的输油管道长度总和最小的位置？证明可在线性时间内确定主管道的最优位置。

【编程任务】
 给定 n 口油井的位置,编程计算各油井到主管道之间的输油管道最小长度总和。

【输入格式】
 由文件 pipe.in 提供输入数据。文件的第 1 行是油井数 n ， $1 \leq n \leq 10000$ 。接下来 n 行是油井的位置，每行 2 个整数 x 和 y ， $-10000 \leq x, y \leq 10000$ 。

【输出格式】
 程序运行结束时，将计算结果输出到文件 pipe.out 中。文件的第 1 行中的数是油井到主管道之间的输油管道最小长度总和。

【输入输出样例】

pipe.in	pipe.out
5	6
1 2	
2 2	
1 3	
3 -2	
3 3	

【时间限制】
 各个测试点 1s

分析：根据点到直线距离最短，在最短方案中，所有的连线必是垂直于管道的，即南北方向。因此给出的 x 坐标没有作用。

现在，问题转化为：

已知 n 个数 y_1, y_2, \dots, y_n ，求一个数 y_0 ，使得 $\sum_{i=1}^n |y_i - y_0|$ 的值最小。

首先对 y 数组排序，不妨认为之后 $y_1 \leq y_2 \leq y_3 \leq \dots \leq y_n$ 。

若 n 为偶数，可将这些数分为 $n/2$ 组： $\{y_1, y_n\}, \{y_2, y_{n-1}\}, \dots, \{y_{\frac{n}{2}}, y_{n+1-\frac{n}{2}}\}$;

对 $\{y_1, y_n\}$ ，易证当 $y_1 \leq y_0 \leq y_n$ 时，有 $|y_1 - y_0| + |y_n - y_0| = y_n - y_1$ ，若 y_0 在此区域之外，则 $|y_1 - y_0| + |y_n - y_0| > y_n - y_1$ 。

此式对任意一组均成立，最终得出当 $y_{\frac{n}{2}} \leq y_0 \leq y_{n+1-\frac{n}{2}}$ 时总和最小。为了程序简便，可以取二者中任意一个。

若 n 为奇数，则分出 $n/2$ 组后，还剩余一个数 $y_{\frac{n}{2}+1}$ ，显然令 $y_0 = y_{\frac{n}{2}+1}$ 即可，仍然是求中位数，即令 $y_0 = y_{(n+1) \text{ div } 2}$ 。

综上，先将 y 数组排序，然后利用快排求 $y_{(n+1) \text{ div } 2}$ ，根据要求输出距离之和。

时间复杂度： $O(N \lg N)$

空间复杂度： $O(N)$

代码：

```
program P1439;
var
  a,b:array[1..10000] of longint;
  i,k,l,n:longint;
procedure qsort(l,r: longint);
var
  i,j,x,y: longint;
begin
  i:=l;
  j:=r;
  x:=b[(l+r) div 2];
  repeat
    while b[i]<x do
      inc(i);
    while x<b[j] do
      dec(j);
    if not(i>j) then
      begin
        y:=a[i]; a[i]:=a[j]; a[j]:=y;
        y:=b[i]; b[i]:=b[j]; b[j]:=y;
        inc(i); dec(j);
      end;
  until i>j;
  if l<j then
    qsort(l,j);
  if i<r then
    qsort(i,r);
end;
begin
  readln(n);
  for i:=1 to n do
    readln(a[i],b[i]);
  qsort(1,n);
  k:=b[(n+1) div 2];
```

```

l:=0;
for i:=1 to n do
  inc(l,abs(b[i]-k));
writeln(l);
end.

```

评测结果:

评测记录号: 52959
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 39 Lines compiled, 0.2 sec
#01: Accepted (109ms, 716KB)
#02: Accepted (125ms, 716KB)
#03: Accepted (187ms, 716KB)
#04: Accepted (140ms, 716KB)
#05: Accepted (140ms, 716KB)
#06: Accepted (156ms, 716KB)
#07: Accepted (140ms, 716KB)
#08: Accepted (140ms, 716KB)
#09: Accepted (109ms, 716KB)
#10: Accepted (125ms, 716KB)
Accepted / 100 / 1375ms / 716KB

P1442 排序算法_谁是组长

<p>【问题描述】 安阳一中信息学奥赛小组需要选一个组长。信息学奥赛小组一共有 n 个人，分别用 1 到 n 编号，其中 m 个人参与了投票。得票数过半（票数大于 $m \div 2$）的人将被选为组长。 输入数据将告知这 m 个人分别将票投给了谁，请统计出谁将担任安阳一中信息学奥赛小组的组长。</p>							
<p>输入数据 第一行两个数 n 和 m。 第二行有 m 个数，这些数都是不超过 n 的正整数，表明这 m 个人的选择。</p>							
<p>输出数据 输出将被选为组长的人。如果没有人的票数过半，请输出-1。</p>							
<p>【输入输出样例】</p> <table border="1"> <thead> <tr> <th>leader.in</th> <th>leader.out</th> </tr> </thead> <tbody> <tr> <td>7 4</td> <td>7</td> </tr> <tr> <td>7 7 2 7</td> <td></td> </tr> </tbody> </table>		leader.in	leader.out	7 4	7	7 7 2 7	
leader.in	leader.out						
7 4	7						
7 7 2 7							
<p>【时间限制】 各个测试点 1s</p>							

分析: N 值很大, 直接存下来每个人的票数不可能。注意最多只有 10000 票, 所以真正得到票的也不会超过 10000 人, 这样就好办了。

一种方案是离散化（这个词有些高端，不过很简单），把存储的数组做成记录类型，两个域分别是编号和得票数。不过这样的话，每读入一个数，都要花 $O(M)$ 的时间（或把记录的数组按编号排序，然后二分，这样也不会快多少）找到所在的位置，然后再处理，时间复杂度为 $O(M^2)$ ，有可能超时。

另一种方案是把读入的数据排序，这样相同的编号必然在一起。然后可以在 $O(N)$ 时间内扫描一遍，得到给有编号及他们出现的次数。这里是不用存储的，找到过半的输出，并立刻结束程序（显然不会有两个过半的）。

时间复杂度： $O(M \lg M)$

空间复杂度： $O(M)$

代码：

```
program P1442;
var
  n,m,i,k,l:longint;
  a:array[1..10001] of longint;
procedure qsort(l,r: longint);
var
  i,j,x,y: longint;
begin
  i:=l;
  j:=r;
  x:=a[(l+r) div 2];
  repeat
    while a[i]<x do
      inc(i);
    while x<a[j] do
      dec(j);
    if not(i>j) then
      begin
        y:=a[i]; a[i]:=a[j]; a[j]:=y;
        inc(i); dec(j);
      end;
  until i>j;
  if l<j then
    qsort(l,j);
  if i<r then
    qsort(i,r);
end;
begin
  readln(n,m);
  for i:=1 to m do
    read(a[i]);
  qsort(1,m);
  k:=0;
  while k<m do
  begin
    inc(k);
    l:=k;
    while (a[l]=a[k])and(l<=m) do
      inc(l);
    if (l-k)>(m div 2) then
```

```
begin
  writeln(a[k]);
  halt;
end;
k:=l-1;
end;
writeln(-1);
end.
```

评测结果:

评测记录号: [52960](#)

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

foo.pas(3,3) Note: Local variable "n" is assigned but never used

Linking foo.exe

48 Lines compiled, 0.1 sec

#01: **Accepted** (109ms, 680KB)

#02: **Accepted** (125ms, 680KB)

#03: **Accepted** (187ms, 680KB)

#04: **Accepted** (171ms, 680KB)

#05: **Accepted** (156ms, 680KB)

#06: **Accepted** (171ms, 680KB)

#07: **Accepted** (125ms, 680KB)

#08: **Accepted** (125ms, 680KB)

#09: **Accepted** (125ms, 680KB)

#10: **Accepted** (109ms, 680KB)

Accepted / 100 / 1406ms / 680KB

总结: 作为排序算法的入门练习, 这四道题难度并不大, 主要考查对快排的熟练应用。