NOIP2004 普及组解题报告

天津市益中 阎骥洲

第一题 不高兴的津津

【问题描述】

津津上初中了。妈妈认为津津应该更加用功学习，所以津津除了上学之外，还要参加妈妈为她报名的各科复习班。另外每周妈妈还会送她去学习朗诵、舞蹈和钢琴。但是津津如果一天上课超过八个小时就会不高兴，而且上得越久就会越不高兴。假设津津不会因为其它事不高兴，并且她的不高兴不会持续到第二天。请你帮忙检查一下津津下周的日程安排，看看下周她会不会不高兴；如果会的话，哪天最不高兴。

【输入文件】

输入文件 unhappy.in 包括七行数据，分别表示周一到周日的日程安排。每行包括两个小于 10 的非负整数，用空格隔开，分别表示津津在学校上课的时间和妈妈安排她上课的时间。

【输出文件】

输出文件 unhappy.out 包括一行，这一行只包含一个数字。如果不会不高兴则输出 0，如果会则输出最不高兴的是周几(用 1，2，3，4，5，6，7 分别表示周一，周二，周三，周四，周五，周六，周日)。如果有两天或两天以上不高兴的程度相当，则输出时间最靠前的一天。

【样例输入】

```
5 3
6 2
7 2
5 3
5 4
0 4
0 6
```

【样例输出】

```
3
```

[问题分析]

由于这道题数据规模是已知的，所以应把每天的两个时间加起来，在比较出最大的一个数据 m,如果 m 大于 8 则 m 所在的这一天 n 为结果；如果 m 小于等于 8 则没有不高兴的天，输出零。

[程序清单]

```
program unhappy;
var
  a:array[1..7,1..2] of integer;
  i,j,m,n:integer;
  f1,f2:text;
begin
   assign(f1,'unhappy.in');
   assign(f2,'unhappy.out');
   reset(f1);
   for i:=1 to 7 do
      begin
         read(f1,a[i,1]);
         readln(f1,a[i,2]);
      end;
   close(f1);
   m:=0;
   n:=0;
   for i:=1 to 7 do
      if a[i,1]+a[i,2]>m then
        begin
           m:=a[i,1]+a[i,2];
           n:=i;
        end;
   rewrite(f2);
   if m>8 then
     writeln(f2,n)
   else
     writeln(f2,'0');
   close(f2);
end.
```

小结：这道题作为第一题是很简单的，但是有些细节地方要多加注意，如：如果有多天不高兴的程度相当，则输出最靠前的一天；小于等于 8 时输出零等。

第二题 花生采摘

【问题描述】

鲁宾逊先生有一只宠物猴，名叫多多。这天，他们两个正沿着乡间小路散步，突然发现路边的告示牌上贴着一张小小的纸条：“欢迎免费品尝我种的花生！——熊字”。

鲁宾逊先生和多多都很开心，因为花生正是他们的最爱。在告示牌背后，路边真的有一块花生田，花生植株整齐地排列成矩形网格(如图 1)。有经验的多多一眼就能看出，每棵花生植株下的花生有多少。为了训练多多的算术，鲁宾逊先生说：“你先找出花生最多的植株，去采摘它的花生；然后再找出剩下的植株里花生最多的，去采摘它的花生；依此类推，不过你一定要在我限定的时间内回到路边。”

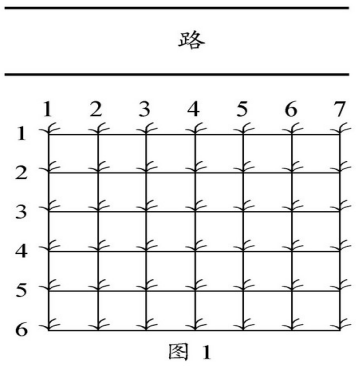


图 1

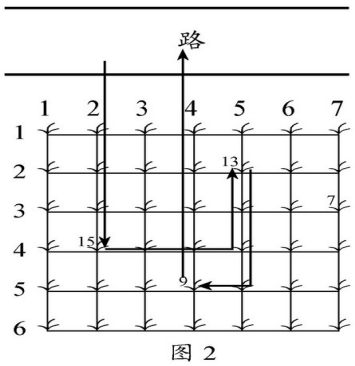


图 2

我们假定多多在每个单位时间内，可以做下列四件事情中的一件：

1)、从路边跳到最靠近路边 (即第一行)的某棵花生植株；

2)、从一棵植株跳到前后左右与之相邻的另一棵植株；

3)、采摘一棵植袜下的花生；

4)、从最靠近路边 (即第一行)的某棵花生植株跳回路边。

现在给定一块花生田的大小和花生的分布，请问在限定时间内，多多最多可以采到多少个花生？注意可能只有部分植株下面长有花生，假设这些植株下的花生个数各不相同。

例如在图 2 所示的花生田里，只有位于(2，5)，(3，7)，(4，2)，(5，4)的植株下长有花生，个数分别为 13，7，15，9。沿着图示的路线，多多在 21 个单位时间内，最多可以采到 37 个花生。

【输入文件】

输入文件 peanuts.in 的第一行包括三个整数，M，N 和 K，用空格隔开；表示花生田的大小为 M×N(1≤M，N≤20)，多多采花生的限定时间为 K(0≤K≤1000)个单位时间。接下来的 M 行，每行包括 N 个非负整数，也用空格隔开；第 i+1 行的第 j 个整数 P_{ij} (0≤P_{ij}≤500)表示花生田里植株(i,j)下花生的数目，0 表示该植株下没有花生。

【样例输入 1】

```
6 7 21
0 0 0 0 0 0 0
0 0 0 0 13 0 0
0 0 0 0 0 0 7
0 15 0 0 0 0 0
0 0 0 9 0 0 0
0 0 0 0 0 0 0
```

【样例输出 1】

```
37
```

【样例输入 2】

```
6 7 20
0 0 0 0 0 0 0
```

0 0 0 0 13 0 0
0 0 0 0 0 0 7
0 15 0 0 0 0 0
0 0 0 9 0 0 0
0 0 0 0 0 0 0

【样例输出 2】
28

[问题分析]

由于采摘的顺序已经给出（先采最多的，在采次多的……），所以本题应采用模拟的方法：

1）确定第一个目标——最大的节点（X_1,Y_1）和初始位置（0，Y_1）；

2）计算从当前位置(P,Q)到目标位置(X,Y)、摘下花生、回到路边的时间是否超过规定时间；

3）若不超过则时间 t 累加上从当前位置到目标位置的时间，数目 I 累加上目标位置的权值 data[X,Y]，当前位置赋为目标位置，当前目标位置权值赋为 0，生成当前目标位置，重复步骤 2；

4）若超过则输出，结束。

程序中，主程序用来模拟采摘过程，函数 notime 计算是否超时使用,过程 make 生成新的目标位置。A 树组存放花生田的状态。

[程序清单]

```
program peanuts;
var
  m,n,tt:integer;
  a:array[1..20,1..20] of integer;
procedure make(var x:integer;var y:integer);
var
  i,j,max:integer;
begin
    max:=0;
    for i:=1 to m do
       for j:=1 to n do
          if a[i,j]>max then
            begin
               max:=a[i,j];
               x:=i;
               y:=j;
            end;
end;

function notime(x,y,p,q,t:integer):boolean;
var
  stamp:integer;
```

```
begin
    stamp:=abs(x-p)+abs(y-q)+1+x;
    if stamp+t>tt then
      notime:=true
    else
      notime:=false;
end;

var
  i,j,k,t,l,x,y,p,q:integer;
  f1,f2:text;
begin
    assign(f1,'peanuts.in');
    assign(f2,'peanuts.out');
    reset(f1);
    read(f1,m);
    read(f1,n);
    readln(f1,tt);
    for i:=1 to m do
       for j:=1 to n do
          read(f1,a[i,j]);
    close(f1);

    make(x,y);
    p:=0;
    q:=y;
    rewrite(f2);
    repeat
        if notime(x,y,p,q,t) then
          begin
             writeln(f2,l);
             break;
          end;
        t:=t+abs(x-p)+abs(y-q)+1;
        l:=l+a[x,y];
        a[x,y]:=0;
        p:=x;
        q:=y;
        make(x,y)
    until t>tt;
    close(f2);
end.
```

小结：这道题的采摘顺序已经给出，用模拟法就可以了，若不给出只说让得到最多的花生，那就是一道较难的题了，有能力的读者可以自己试一试。

第三题 FBI 树

【问题描述】

我们可以把由“0”和“1”组成的字符串分为三类：全“0”串称为 B 串，全“1”串称为 I 串，既含“0”又含“1”的串则称为 F 串。

FBI 树是一种二叉树[1]，它的结点类型也包括 F 结点，B 结点和 I 结点三种。由一个长度为 2^N 的“01”串 S 可以构造出一棵 FBI 树 T，递归的构造方法如下：

1)、T 的根结点为 R，其类型与串 S 的类型相同；

2)、若串 S 的长度大于 1，将串 S 从中间分开，分为等长的左右子串 S1 和 S2；由左子串 S1 构造 R 的左子树 T1，由右子串 S2 构造 R 的右子树 T2。

现在给定一个长度为 2^N 的“01”串，请用上述构造方法构造出一棵 FBI 树，并输出它的后序遍历[2]序列。

【输入文件】

输入文件 fbi.in 的第一行是一个整数 N(0≤N≤10)，第二行是一个长度为 2^N 的“01”串。

【输出文件】

输出文件 fbi.out 包括一行，这一行只包含一个字符串，即 FBI 树的后序遍历序列。

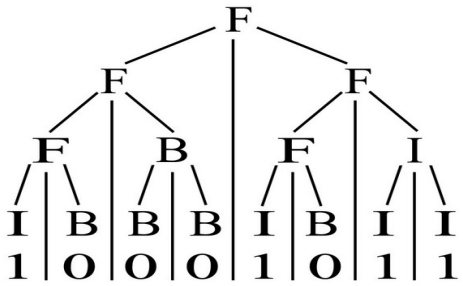
【样例输入】

3

10001011

【样例输出 】

IBFBBBFIBFIIIFF



[问题分析]

这道题如果使用深度优先搜索的话，则在空间方面：当 n 为最大值 10 时，为一个深度为 11 的满二叉树，底层数据有 1024 个，所以最多使用 2^{11}-1 个数据，在空间上是可行的；在时间方面：当 n 为最大值 10 时，搜索次数为 2^{10}，递规层数为 10 层，它的时间复杂度为 $O(2^n)$（0<n<=10）,在时间上也是可行的。所以本题采用递归搜索算法。而且由于存放数据的数组已有较确定的范围，所以无需使用链表的数据结构。

程序中，主程序用来判断和生成二叉树，tree 过程用来递规生成后序遍历。Fbi 数组存放二叉树，stap 数组存放后序遍历。

[程序清单]

```
program fbitree;
var
  fbi:array[1..11,1..1024] of 0..2;
```

```
  tb:char;
  a:array[1..1024] of 0..1;
  stap:array[1..2048] of 0..2;
  sttr:array[0..2] of string[1];
  i,j,k,l,m,n,p,q:integer;
  f1,f2:text;

procedure tree(x,y:integer);
begin
    if x=n then
      begin
          p:=p+1;
          stap[p]:=fbi[x+1,y*2-1];
          p:=p+1;
          stap[p]:=fbi[x+1,y*2];
          p:=p+1;
          stap[p]:=fbi[x,y];
          exit;
      end;
    tree(x+1,y*2-1);
    tree(x+1,y*2);
    p:=p+1;
    stap[p]:=fbi[x,y];
end;

begin
    assign(f1,'fbi.in');
    assign(f2,'fbi.out');
    reset(f1);
    readln(f1,n);
    m:=1;
    for i:=1 to n do
       m:=m*2;
    for i:=1 to m do
       begin
          read(f1,tb);
          fbi[n+1,i]:=ord(tb)-48;
       end;
    close(f1);

    p:=m;
    for i:=n downto 1 do
       begin
          p:=p div 2;
```

```
        for j:=1 to p do
          begin
            if (fbi[i+1,j*2-1]<>fbi[i+1,j*2]) then
              fbi[i,j]:=2;
            if (fbi[i+1,j*2-1]=fbi[i+1,j*2]) then
              fbi[i,j]:=fbi[i+1,j*2-1];
          end;
      end;

  p:=0;
  tree(1,1);

  sttr[0]:='B';
  sttr[1]:='I';
  sttr[2]:='F';
  rewrite(f2);
  for i:=1 to p do
    write(f2,sttr[stap[i]]);
  close(f2);
end.
```

小结：本题的算法，只适用于 n 值较小的情况。当 n 值较大时，就要使用链表来存储，或设计其他算法，在此就不做讨论了。

第四题 火星人

【问题描述】

人类终于登上了火星的土地并且见到了神秘的火星人。人类和火星人都无法理解对方的语言，但是我们的科学家发明了一种用数字交流的方法。这种交流方法是这样的，首先，火星人把一个非常大的数字告诉人类科学家，科学家破解这个数字的含义后，再把一个很小的数字加到这个大数上面，把结果告诉火星人，作为人类的回答。

火星人用一种非常简单的方式来表示数字——掰手指。火星人只有一只手，但这只手上有成千上万的手指，这些手指排成一列，分别编号为 1，2，3……。火星人的任意两根手指都能随意交换位置，他们就是通过这方法计数的。

一个火星人用一个人类的手演示了如何用手指计数。如果把五根手指——拇指、食指、中指、无名指和小指分别编号为 1，2，3，4 和 5，当它们按正常顺序排列时，形成了 5 位数 12345，当你交换无名指和小指的位置时，会形成 5 位数 12354，当你把五个手指的顺序完全颠倒时，会形成 54321，在所有能够形成的 120 个 5 位数中，12345 最小，它表示 1；12354 第二小，它表示 2；54321 最大，它表示 120。下表展示了只有 3 根手指时能够形成的 6 个 3 位数和它们代表的数字：

三进制数	123	132	213	231	312	321

代表的数字	1	2	3	4	5	6

现在你有幸成为了第一个和火星人交流的地球人。一个火星人会让你看他的手指，科学家会告诉你要加上去的很小的数。你的任务是，把火星人用手指表示的数与科学家告诉你的数相加，并根据相加的结果改变火星人手指的排列顺序。输入数据保证这个结果不会超出火星人手指能表示的范围。

【输入文件】

输入文件 martian.in 包括三行，第一行有一个正整数 N，表示火星人手指的数目 (1≤N≤10000)。第二行是一个正整数 M，表示要加上去的小整数 (1≤M≤100)。下一行是 1 到 N 这 N 个整数的一个排列，用空格隔开，表示火星人手指的排列顺序。

【输出文件】

输出文件 martian.out 只有一行，这一行含有 N 个整数，表示改变后的火星人手指的排列顺序。每两个相邻的数中间用一个空格分开，不能有多余的空格。

【样例输入】

5
3
1 2 3 4 5

【样例输出 】

1 2 4 5 3

[问题分析]

先看这道题的数据规模，最多为 10000 个数据，那么生成排列再进行排序是完全不可能的（时间复杂度为 O(2*n!)，空间复杂度为 O(n*n!)），故这道题每一步生成的排列必须是有序的。因此这道题采用递归的方法比较好：

1．对于一个排列，进行从 y 到 x（该排列的上下界）的循环；

2．当满足 a[i-1]<a[i]时，对 a[i-1]到 a[y]进行从小到大的排列，找出原 a[i-1]在现数组中的位置 la；

3．进行从 la+1 到 y 的循环，使 a[i-1]=a[y]，剩余步数 l=l-1，再进行 a[i]到 a[y]的排列；

4．当 l=0 时，结束递归，输出，结束。

为了不生成多余的排列（在 l=0 之前的部分排列），我作出了一个优化：对于第三步中，若 a[i]到 a[y]的排列总数加以排列总数小于 M，则 l 减去 a[i]到 a[y]的排列总数,a[i]到 a[y]的数据进行倒序排列。

程序中，主程序主要进行输入，过程 finger 进行递归调用，过程 swap 进行数据交换（pascal 中没有数据交换函数），过程 print 控制输出，a 数组存储当前排列，l 为剩余步数，as 数组存储 a[i]到 a[y]排列的原形（为了复原变形后的数据）。

[程序清单]

```
program martain;
var
  a:array[1..10000] of integer;
  l,m,n:integer;
  f1,f2:text;

procedure swap(var a:integer;var b:integer);
var
  t:integer;
begin
    t:=a;
    a:=b;
    b:=t;
end;

procedure print;
var
  i,j:integer;
begin

    rewrite(f2);
    for i:=1 to n do
       write(f2,a[i],' ');
    close(f2);
    halt;
end;

procedure finger(x,y:integer);
var
  i,j,k,p,q,st1,la,z:integer;
  as:array[1..1000] of integer;
begin

    if l=0 then
      print;
    if x=y then
      exit;
    for i:=y downto x+1 do
       if a[i]>a[i-1] then
          begin
             st1:=a[i-1];
             for j:=i-1 to y-1 do
                for k:=j+1 to y do
                   if a[j]>a[k] then
```

```
                swap(a[j],a[k]);
            for j:=i-1 to y do
              if a[j]=st1 then
                begin
                  la:=j;
                  break;
                end;

            for z:=la+1 to y do
              begin
                swap(a[i-1],a[z]);
                for p:=i-1 to y do
                  as[p-i+2]:=a[p];
                p:=1;
                q:=1;
                for j:=1 to y-i+1 do
                  begin
                    p:=p*j;
                    if l-p<=0 then
                      begin
                        q:=0;
                        break;
                      end;
                  end;
                if q=1 then
                  begin
                    for j:=i to y-1 do
                      for k:=j+1 to y do
                        if a[j]<a[k] then
                          swap(a[j],a[k]);
                    l:=l-p;
                  end
                else
                  begin
                    l:=l-1;
                    finger(i,y);
                  end;
                for p:=i-1 to y do
                  a[p]:=as[p-i+2];
              end;
          end;

end;
```

```
var
  i:integer;
begin
    assign(f1,'martian.in');
    assign(f2,'martian.out');
    reset(f1);
    readln(f1,n);
    readln(f1,m);
    for i:=1 to n do
       read(f1,a[i]);
    close(f1);

    l:=m;
    finger(1,n);
end.
```

小结：这道题关键在于生成有序的排列，递归不一定是最好的方法，读者可以自己尝试更好的方法，可以和本人进行讨论（本人导师的邮箱 mailto:jiang_zhx@126.com）。