NOIP2005 提高组解题报告
天津南开中学 薛原
谁拿了最多奖学金(scholar)

题目概述：

已知每个学生的个人信息，求出获得奖学金最多的学生姓名、金额，以及全部奖学金金额。

算法分析：

模拟

其中涉及简单的字符处理，特别要注意数据类型的应用。如：学生姓名可采用 char 和 string 相结合的方法处理，奖学金金额用 longint 较为适宜。

程序：

```
program scholar;
var name:array[1..100] of string;
    a1,a2,a5:array[1..100] of longint;
    a3,a4:array[1..100] of char;
    n,i,max,total,p:longint;
    maxname:string;
    ch:char;
    f:text;
begin
     assign(f,'scholar.in');reset(f);
     readln(f,n);
     for i:=1 to n do
     begin
          read(f,ch);
          while ch<>' ' do
          begin
               name[i]:=name[i]+ch;
               read(f,ch);
          end;
          readln(f,a1[i],a2[i],ch,a3[i],ch,a4[i],ch,a5[i]);
     end;
     close(f);
     for i:=1 to n do
     begin
          p:=0;
          if (a1[i]>80) and (a5[i]>=1) then inc(p,8000);
          if (a1[i]>85) and (a2[i]>80) then inc(p,4000);
          if (a1[i]>90) then inc(p,2000);
          if (a1[i]>85) and (a4[i]='Y') then inc(p,1000);
          if (a2[i]>80) and (a3[i]='Y') then inc(p,850);
          if p>max then
          begin
```

```
                max:=p;
                maxname:=name[i];
           end;
           inc(total,p);
      end;
      assign(f,'scholar.out');rewrite(f);
      writeln(f,maxname);
      writeln(f,max);
      writeln(f,total);
      close(f);
end.
```

[align=center]过河(River)[/align]

题目概述：

在一条长为L数轴上有若干障碍点，每次前进距离为S到T之间的任意正整数（包括S,T），求走过 L或大于L的距离，遇到最少的障碍点。

算法分析：

看到题目首先想到的是时间复杂度为O(L)的递推算法。但是L的上限为10^9，这种算法显然是不行的。

仔细思考，可以得到下面的结论：

存在N0，当n> N0时，n可以由若干S到T之间的正整数（包括S,T）组成。

因此，将障碍点按升序排列，当两相邻障碍点之间距离较大时，可适当缩小两障碍点之间距离，但不影响最终结果。

根据上述结论，改进递推算法。由于障碍点之间距离大大缩减，算法的复杂度是可以承受的。

特别地，当S=T时需要单独处理。

程序：

```
program river;
const max=105;
var a,a1:array[0..101] of longint;
    b:array[0..100] of boolean;
    c,d:array[0..10000] of longint;
    l,s,t,m,ans,low,i,j,k,temp:longint;
    flag:boolean;
    f:text;
procedure init;
begin
     assign(f,'river9.in');reset(f);
     readln(f,l);
     readln(f,s,t,m);
     for i:=1 to m do read(f,a[i]);
     a[0]:=0;a[m+1]:=l;
     for i:=1 to m-1 do
     for j:=i+1 to m do
     if a[i]>a[j] then
```

```
     begin
          temp:=a[i];a[i]:=a[j];a[j]:=temp;
     end;
     close(f);
end;
procedure work1;
begin
     for i:=1 to m do
     if a[i] mod s=0 then inc(ans);
end;
procedure work2;
begin
     fillchar(b,sizeof(b),false);
     b[0]:=true;
     for i:=s to t do
     begin
          for j:=0 to 100 do
          if b[j] then
          begin
               k:=1;
               while k*i+j<=100 do
               begin
                    b[k*i+j]:=true;
                    inc(k);
               end;
          end;
     end;
     for i:=1 to 100 do
     begin
          flag:=true;
          for j:=0 to t-1 do
          if not b[i+j] then begin flag:=false;break;end;
          if flag then
          begin
               low:=i;
               break;
          end;
     end;
     if low<t then low:=t;
     for i:=1 to m+1 do
     begin
          a1[i]:=(a[i]-a[i-1]-low) mod low+a1[i-1]+low;
     end;
     a:=a1;
```

```
     for i:=1 to m do d[a[i]]:=1;
     l:=a[m+1];
     for i:=1 to l+t-1 do c[i]:=max;
     for i:=1 to l+t-1 do
     for j:=s to t do
     if (i-j>=0) and (c[i]>c[i-j]+d[i]) then
        c[i]:=c[i-j]+d[i];
     ans:=max;
     for i:=l to l+t-1 do
     if ans>c[i] then ans:=c[i];
end;
begin
     init;
     if s=t then work1
     else work2;
     assign(f,'river.out');rewrite(f);
     writeln(f,ans);
     close(f);
end.
```

[align=center]篝火晚会（fire）[/align]

题目概述：

根据一定的移动规则，将初始圆环转化为满足一定条件的目标圆环。

算法分析：

从第一个人处断开，将圆环的问题转化为序列的问题。如果可以，求出目标序列。求出目标序列复杂度O(n).

求出目标序列右移0至n-1位置时，不需要移动的人数。将目标序列反转，再求出目标序列右移0至n-1位置时，不需要移动的人数。不需要移动的人数最大等价于需要移动的人数最小。复杂度O(n)。

程序：

```
program fire;
var a:array[1..50000] of longint;
    b:array[1..50000,1..2] of longint;
    d:array[1..50000] of longint;
    w:array[0..50000] of longint;
    n,ans,i,j,t,max:longint;
    flag:boolean;
    f:text;
procedure init;
begin
     assign(f,'fire.in');reset(f);
     readln(f,n);
     for i:=1 to n do
     begin
```

```
          readln(f,b[i,1],b[i,2]);
          inc(d[b[i,1]]);
          inc(d[b[i,2]]);
     end;
     close(f);
     for i:=1 to n do
     if d[i]<>2 then begin flag:=false;exit;end;
end;
procedure circle;
begin
     a[1]:=1;a[2]:=b[1,1];
     for i:=3 to n do
     if b[a[i-1],1]<>a[i-2] then a[i]:=b[a[i-1],1]
     else a[i]:=b[a[i-1],2];
     if a[n]<>b[1,2] then flag:=false;
end;
procedure min;
begin
     fillchar(w,sizeof(w),0);
     for i:=1 to n do
         inc(w[(a[i]-i+n) mod n]);
     for i:=0 to n-1 do
     if max<w[i] then max:=w[i];
     for i:=1 to (n+1) div 2 do
     begin
          t:=a[i];a[i]:=a[n+1-i];a[n+1-i]:=t;
     end;
     fillchar(w,sizeof(w),0);
     for i:=1 to n do
         inc(w[(a[i]-i+n) mod n]);
     for i:=0 to n-1 do
     if max<w[i] then max:=w[i];
     ans:=n-max;
end;
begin
     flag:=true;
     init;
     if flag then circle;
     if flag then min;
     assign(f,'fire.out');rewrite(f);
     if flag then writeln(f,ans) else writeln(f,-1);
     close(f);
end.
```

[align=center]等价表达式[/align]

题目概述：
判断两表达式是否等价。
算法分析：
用栈的方法求表达式的值是经典的算法。考虑到多项式的处理比较麻烦，不妨对变量 a 进行多次赋值以判断表达式是否等价。

值得注意，由于进行数值运算，采用哪种数据类型成为程序是否正确的关键。下面的程序，采取 mod m 的方法，其中 m 为任意正整数。当对 a 多次赋值，且 m 取不同的较大的正整数时，可以保证算法的正确性。
程序：

```
program equal;
const max=maxlongint;
const com:array[1..7,1..7] of char=(('>','>','<','<','<','>','>'),
                                    ('>','>','<','<','<','>','>'),
                                    ('>','>','>','<','<','>','>'),
                                    ('>','>','>','>','<','>','>'),
                                    ('<','<','<','<','<','=','X'),
                                    ('>','>','>','>','X','>','>'),
                                    ('<','<','<','<','<','X','X'));

var there:char;
    oped:array[1..1000] of longint;
    optr:array[1..1000] of char;
    ned,ntr:int64;
    a,b:int64;
    flag:boolean;
    s:array[0..26] of string;
    value:array[0..26,-4..4] of int64;
    ans:array[0..26] of boolean;
    n,i,j,p,q:longint;
    f:text;
function compare(w1,w2:char):char;
var x1,x2:integer;
begin
     case w1 of
          '+':x1:=1;
          '-':x1:=2;
          '*':x1:=3;
          '^':x1:=4;
          '(':x1:=5;
          ')':x1:=6;
          '#':x1:=7;
```

```
     end;
     case w2 of
          '+':x2:=1;
          '-':x2:=2;
          '*':x2:=3;
          '^':x2:=4;
          '(':x2:=5;
          ')':x2:=6;
          '#':x2:=7;
     end;
     compare:=com[x1,x2];
end;
function operation(a:int64;there:char;b:int64):int64;
var i:longint;
begin
     case there of
          '+':operation:=(a+b) mod max;
          '-':operation:=(a-b) mod max;
          '*':operation:=(a*b) mod max;
          '^':begin operation:=1;for i:=1 to b do
operation:=operation*a mod max;end;
     end;
end;
function exp(s:string;aa:int64):int64;
var i:int64;
begin
     s:=s+'#';
     i:=1;
     ned:=0;ntr:=1;
     fillchar(oped,sizeof(oped),0);
     optr:='';
     optr[1]:='#';flag:=false;
     while not ((s[i]='#')and (optr[ntr]='#')) do
     begin
          if s[i] in ['0'..'9'] then
          begin
               if not flag then
               begin
                    ned:=ned+1;
                    oped[ned]:=ord(s[i])-ord('0');
                    flag:=true;
                    inc(i);
               end
               else
```

```
                begin
                    oped[ned]:=oped[ned]*10+ord(s[i])-ord('0');
                    inc(i);
                end;
            end
            else
            if s[i]='a' then
            begin
                inc(ned);
                oped[ned]:=aa;
                inc(i);
            end
            else if s[i]=' ' then inc(i)
            else
            begin
                flag:=false;
                case compare(optr[ntr],s[i]) of
                    '<':begin ntr:=ntr+1;optr[ntr]:=s[i];inc(i);end;
                    '>':begin
                            there:=optr[ntr];ntr:=ntr-1;
                            b:=oped[ned];ned:=ned-1;
                            a:=oped[ned];
                            oped[ned]:=operation(a,there,b);
                        end;
                    '=':begin ntr:=ntr-1;inc(i);end;
                end;
            end;
        end;
        exp:=oped[1];
end;
begin
        assign(f,'equal.in');reset(f);
        readln(f,s[0]);
        readln(f,n);
        for i:=1 to n do readln(f,s[i]);
        fillchar(ans,sizeof(ans),true);
        close(f);
        for i:=0 to n do
        if ans[i] then
        for j:=-4 to 4 do
        begin
            value[i,j]:=exp(s[i],j);
            if value[i,j]<>value[0,j] then begin ans[i]:=false;break;end;
        end;
```

```
    assign(f,'equal.out');rewrite(f);
    for i:=1 to n do
        if ans[i] then write(f,chr(ord('A')+i-1));
    writeln(f);
    close(f);
end.
```