NOIP2008 解题报告

张恩权

一、ISBN 号码

基础字符串处理题，心细一点的基本都能得满分。

参考程序：

```
program isbn;
const
  inp='isbn.in';
  oup='isbn.out';

var
  i,j,k,ans:longint;
  s:string;
  ch:char;
procedure flink;
  begin
    assign(input,inp);
    reset(input);
    assign(output,oup);
    rewrite(output);
  end;
procedure fclose;
  begin
    close(input);
    close(output);
  end;

begin
  flink;
  readln(s);// 输入字符串
  j:=0;
```

```
 i:=1;
  ans:=0;
 while j<9 do
  begin
   if s[i] in ['0'..'9']  then//如果是数字,那么累加到 ans 中,共 9 个数字
     begin
      inc(j);
      inc(ans,(ord(s[i])-ord('0'))*j);
     end;
    inc(i);
  end;
 ans:=ans mod 11;计算识别码
 if ans=10 then ch:='X' else ch:=chr(ord('0')+ans);//把识别码转换成字符,方便输出
 if s[length(s)]=ch
  then write('Right')
  else write(copy(s,1,12)+ch);//输出正确的识别码
 fclose;
end.
```

二、排座椅

用的是赛前集训时提到的贪心，当时说某些题目用贪心可以得部分分，但是本题贪心可以得满分的。

当然本题的贪心需要预处理下，开 2 个一维数组，row[i]录如果在第 i 行加通道,可以分割多少对调皮学生,col[i]记录如果在第 j 列加通道,可以分割多少对调皮学生,最后贪心法输出分割学生最多的前 K 行和前 L 列。

参考程序：

```
program seat;
const
 inp='seat.in';
 oup='seat.out';

var
```

```
  flag,m,n,k,l,d,i,j,x,y,x1,y1:longint;
  tmp,col,row:array[1..1000] of longint;
  s,s1:ansistring;
procedure flink;
  begin
    assign(input,inp);
    reset(input);
    assign(output,oup);
    rewrite(output);
  end;
procedure fclose;
  begin
    close(input);
    close(output);
  end;
function min(a,b:longint):longint;
  begin
    if a<b then exit(a); exit(b);
  end;
procedure qsort(m,n:Longint);//快排
  var
    i,j,k,t:longint;
  begin
    i:=m;  j:=n;  k:=tmp[(i+j) shr 1];
    repeat
      while tmp[i]>k do inc(i);
      while tmp[j]<k do dec(j);
      if i<=j  then
        begin
          t:=tmp[i]; tmp[i]:=tmp[j]; tmp[j]:=t;
          inc(I);  dec(J);
        end;
```

```
    until i>j;
    if m<j then qsort(m,j);
    if I<n then qsort(i,n);
  end;
begin
  flink;
  readln(m,n,k,L,d);
  fillchar(row,sizeof(row),0);
  fillchar(col,sizeof(col),0);
  for i:= 1 to d do  //统计在每行、每列添加通道可以分割的学生数
    begin
      readln(x,y,x1,y1);
      if (x=x1)
        then inc(col[min(y,y1)])
        else  inc(row[min(x,x1)]);
    end;
  j:=0;
  for i:= 1 to m do //把能没个行通道分割的学生数加入 tmp 数组,准备排序
    begin
      if row[i]>0 then
        begin
          inc(j);
          tmp[j]:=row[i];
        end;
    end;
  qsort(1,j);//对 tmp 数组排序
  flag:=tmp[k];//flag 为前 K 项的最小值
  i:=1; j:=0;
  while  (i<=n) and (j<k) do
    begin
      if row[i]>=flag then //如果该行能分割的人数不少于 flag,说明此处可以添加通道
        begin
```

```
          write(i);
          inc(j);
          if j<>k then write(' ');
        end;
      inc(i);
    end;
  writeln;
//下面是求列通道,思想同上
  j:=0;
  for i:= 1 to n do
    begin
      if col[i]>0 then
        begin
          inc(j);
          tmp[j]:=col[i];
        end;
    end;
  qsort(1,j);
  flag:=tmp[L];
  i:=1;  j:=0;
   while  (i<=m) and (j<L) do
    begin
      if col[i]>=flag then
        begin
          write(i);
          inc(j);
          if j<>L then write(' ');
        end;
      inc(i);
    end;
  fclose;
end.
```

三、传球游戏

直接 dp，似乎说递推更确切点。

f(i,k)表示经过 k 次传到编号为 i 的人手中的方案数。那么可以推出下面的方程：

f(i,k)=f(i-1,k-1)+f(i+1,k-1) （i=1 或 n 时，需单独处理）

边界条件：f(1,0)=1;

结果在 f(1,m)中

参考程序：

```
program ball;
const
  inp='ball.in';
  oup='ball.out';

var
  i,j,k,n,m:longint;
  f:array[0..30,0..30] of longint;
procedure flink;
  begin
    assign(input,inp);
    reset(input);
    assign(output,oup);
    rewrite(output);
  end;
procedure fclose;
  begin
    close(input);
    close(output);
  end;

begin
  flink;
  readln(n,m);
  fillchar(f,sizeof(f),0);
```

```
 f[1,0]:=1;
 for k:=1 to m do//注意此处 2 个循环的次序
  begin
   f[1,k]:=f[2,k-1]+f[n,k-1];
   for i:= 2 to n-1 do
    f[i,k]:=f[i-1,k-1]+f[i+1,k-1];
   f[n,k]:=f[n-1,k-1]+f[1,k-1];
  end;
 write(f[1,m]);

 fclose;
end.
```

四、立体图

Pku 原题，编号 2330

算不上难题，但是比较麻烦，细心点就 ok 了。

先计算好画布的大小，再写一个根据左下角坐标绘制一个单位立方体的子程序。

然后遵循下面法则，不停绘制若干个立方体。（此处能体现出分割程序的伟大）

因为要不停的覆盖，所以要遵循“视觉法则”：

1. 先绘里层再绘外层
2. 先绘底层再绘上层
3. 先回左边再绘右边

参考程序：

```
program drawing;
const
 inp='drawing.in';
 oup='drawing.out';

var
 m,n,i,j,k,x,y,h,tmp,maxx,maxy:longint;
 map:array[1..1000,1..1000] of char;//画布
 a:array[1..50,1..50] of integer;//记录输入的矩阵
```

```
procedure flink;
  begin
    assign(input,inp);
    reset(input);
    assign(output,oup);
    rewrite(output);
  end;
procedure fclose;
  begin
    close(input);
    close(output);
  end;
procedure print;//输出画布
  var
    i,j:longint;
  begin
    for i:= 1 to maxx do
    begin
      for j:= 1 to maxy do
        write(map[i,j]);
      if i<>maxx then writeln;
    end;


  end;
procedure  draw(x,y:longint);//在画布(map 数组)上绘制左下角坐标为(x,y)的一个单位立
方体
  begin
    map[x,y]:='+';map[x,y+1]:='-'; map[x,y+2]:='-';map[x,y+3]:='-';map[x,y+4]:='+';
    dec(x);
    map[x,y]:='|';map[x,y+1]:=' '; map[x,y+2]:=' ';map[x,y+3]:=' ';map[x,y+4]:='|';
    map[x,y+5]:='/';
    dec(x);
```

```
    map[x,y]:='|';map[x,y+1]:=' '; map[x,y+2]:=' ';map[x,y+3]:=' ';map[x,y+4]:='|';
    map[x,y+5]:=' ';map[x,y+6]:='+';
    dec(x);
    map[x,y]:='+';map[x,y+1]:='-'; map[x,y+2]:='-';map[x,y+3]:='-';map[x,y+4]:='+';
    map[x,y+5]:=' ';map[x,y+6]:='|';
    dec(x);  inc(y);
    map[x,y]:='/';map[x,y+1]:=' '; map[x,y+2]:=' ';map[x,y+3]:=' ';map[x,y+4]:='/';
    map[x,y+5]:='|';
    dec(x);inc(y);
    map[x,y]:='+';map[x,y+1]:='-'; map[x,y+2]:='-';map[x,y+3]:='-';map[x,y+4]:='+';

  end;

begin
  flink;
  for i:= 1 to 1000 do
    for j:= 1 to 1000 do
      map[i,j]:='.'; //初始化画布
  readln(m,n);
  //计算画布大小 maxx * maxy
  maxy:=n*4+1+m*2;
  maxx:=0;
  for i:= 1 to m do
    begin
      tmp:=0;
      for j:= 1 to n do
        begin
          read(a[i,j]);
          if a[i,j]>tmp then tmp:=a[i,j];
        end;
      tmp:=tmp*3+3+(m-i)*2;
      if tmp >maxx then maxx:=tmp;
```

```
    readln;
     end;
//开始往画布上绘图
for i:= 1 to m do
  for j:= 1 to n do
    begin
      x:=maxx-(m-i)*2;//第 i 层第 j 列最下方立方体左下角点的位置(x,y)
      y:=(m-i)*2+(j-1)*4+1;
      for k:= 1 to a[i,j] do
        draw(x-(k-1)*3,y);//绘制每层的若干个一个单位立方体
    end;
print;//输出画布

fclose;
end.
```