

1、 一些例题的原题

1、 奶牛浴场

奶牛浴场

【问题描述】

由于 John 建造了牛场围栏，激起了奶牛的愤怒，奶牛的产奶量急剧减少。为了讨好奶牛，John 决定在牛场中建造一个大型浴场。但是 John 的奶牛有一个奇怪的习惯，每头奶牛都必须在牛场中的一个固定的位置产奶，而奶牛显然不能在浴场中产奶，于是，John 希望所建造的浴场不覆盖这些产奶点。这回，他又要求助于 Clevow 了。你还能帮助 Clevow 吗？

John 的牛场和规划的浴场都是矩形。浴场要完全位于牛场之内，并且浴场的轮廓要与牛场的轮廓平行或者重合。浴场不能覆盖任何产奶点，但是产奶点可以位于浴场的轮廓上。

Clevow 当然希望浴场的面积尽可能大了，所以你的任务就是帮她计算浴场的最大面积。

【输入文件】

输入文件的第一行包含两个整数 L 和 W ，分别表示牛场的长和宽。文件的第二行包含一个整数 n ，表示产奶点的数量。以下 n 行每行包含两个整数 x 和 y ，表示一个产奶点的坐标。所有产奶点都位于牛场内，即： $0 < x < L$ ， $0 < y < W$ 。

【输出文件】

输出文件仅一行，包含一个整数 S ，表示浴场的最大面积。

【输入输出样例】

happy.in	happy.out
10 10	80
4	
1 1	
9 1	
1 9	
9 9	

【参数约定】

$0 < n < 5000$

$1 < L, W < 30000$

2、Candy

糖果盒 (Candy Box)

问题描述:

一个被分为 $n*m$ 个格子的糖果盒，第 i 行第 j 列位置的格子里面有 $a[i][j]$ 颗糖。本来 **tenshi** 打算送这盒糖果给某 **PPMM** 的，但是就在要送出糖果盒的前一天晚上，一只极其可恶的老鼠夜袭糖果盒，有部分格子被洗劫并且穿了洞。**tenshi** 必须尽快从这个糖果盒里面切割出一个矩形糖果盒，新的糖果盒不能有洞，并且 **tenshi** 希望保留在新糖果盒内的糖的总数尽量多。

任 务：

请帮 **tenshi** 设计一个程序 计算一下新糖果盒最多能够保留多少糖果。

输入格式:

从文件 **CANDY.INP** 读入数据。第一行有两个整数 $n、m$ 。第 $i + 1$ 行的第 j 个数表示 $a[i][j]$ ，如果这个数为 0 ，则表示这个位置的格子被洗劫过。其中：

$1 \leq n, m \leq 1000$
 $0 \leq a[i][j] \leq 255$

注意：本题提供 **16 MB** 内存，时间限制为 **2** 秒。

输出格式:

输出最大糖果数到 **CANDY.OUT**。

样例

CANDY.INP	CANDY.OUT
3 4	17
1 2 3 4	
5 0 6 3	
10 3 4 0	

注:

10 3 4
这个矩形的糖果数最大

3 、 Big Barn

Big Barn

A Special Treat

Farmer John wants to place a big square barn on his square farm. He hates to cut down trees on his farm and wants to find a location for his barn that enables him to build it only on land that is already clear of trees. For our purposes, his land is divided into $N \times N$ parcels. The input contains a list of parcels that contain trees. Your job is to determine and report the largest possible square barn that can be placed on his land without having to clear away trees. The barn sides must be parallel to the horizontal or vertical axis.

EXAMPLE

Consider the following grid of Farmer John's land where '.' represents a parcel with no trees and '#' represents a parcel with trees:

```
  1 2 3 4 5 6 7 8
1 . . . . . . .
2 . # . . # . .
3 . . . . . . .
4 . . . . . . .
5 . . . . . . .
6 . . # . . . .
7 . . . . . . .
8 . . . . . . .
```

The largest barn is 5×5 and can be placed in either of two locations in the lower right part of the grid.

PROGRAM NAME: bigbrn

INPUT FORMAT

Line 1: Two integers: N ($1 \leq N \leq 1000$), the number of

parcels on a side, and T ($1 \leq T \leq 10,000$) the number of parcels with trees

Lines Two integers ($1 \leq \text{each integer} \leq N$), the row and
2..T+1: column of a tree parcel

SAMPLE INPUT (file bigbrn.in)

```
8 3
2 2
2 6
6 3
```

OUTPUT FORMAT

The output file should consist of exactly one line, the maximum side length of John's barn.

SAMPLE OUTPUT (file bigbrn.out)

```
5
```

2、 一些例题的程序

1、 奶牛浴场用第一种算法解的程序

```
program happy;
var
  f:text;
  x,y:array[1..5002] of longint;
  maxl,n,best,a,b,c,w,l,i,j,high,low:longint;

procedure sort(l,r:longint);
var
  i,j:longint;
begin
  i:=l+random(r-l+1);
  a:=x[i]; b:=y[i]; i:=l; j:=r;
```

```

repeat
  while (x[i]<a) or ((x[i]=a) and (y[i]<b)) do i:=i+1;
  while (x[j]>a) or ((x[j]=a) and (y[j]>b)) do j:=j-1;
  if i<=j then
    begin
      c:=x[i]; x[i]:=x[j]; x[j]:=c;
      c:=y[i]; y[i]:=y[j]; y[j]:=c;
      inc(i); dec(j);
    end;
  until i>j;
  if j>l then sort(l,j);
  if i<r then sort(i,r);
end;

```

```

procedure sort_y(l,r:longint);
var
  i,j:longint;
begin
  a:=y[(l+r) div 2]; i:=l; j:=r;
  repeat
    while (y[i]<a) do i:=i+1;
    while (y[j]>a) do j:=j-1;
    if i<=j then
      begin
        c:=y[i]; y[i]:=y[j]; y[j]:=c;
        inc(i); dec(j);
      end;
    until i>j;
    if j>l then sort_y(l,j);
    if i<r then sort_y(i,r);
  end;

```

```

procedure max(a:longint);
begin
  if a>best then best:=a;
end;

```

```

begin
  assign(f,'happy.in');
  reset(f);
  readln(f,l,w);
  readln(f,n);
  for i:=1 to n do
    readln(f,x[i],y[i]);

```

```

close(f);
inc(n); x[n]:=l; y[n]:=0;
inc(n); x[n]:=0; y[n]:=w;
sort(1,n);
best:=0;
for i:=1 to n do
begin
  high:=w; low:=0; maxl:=l-x[i];
  for j:=i+1 to n do
    if (y[j]<=high) and (y[j]>=low) then
      begin
        if maxl*(high-low)<=best then break;
        max((x[j]-x[i])*(high-low));
        if y[j]=y[i] then break
        else if y[j]>y[i] then
          if y[j]<high then high:=y[j]
          else
            else if y[j]>low then low:=y[j];
        end;
      high:=w; low:=0; maxl:=l-x[i];
      for j:=i-1 downto 1 do
        if (y[j]<=high) and (y[j]>=low) then
          begin
            if maxl*(high-low)<=best then break;
            max((x[i]-x[j])*(high-low));
            if y[j]=y[i] then break
            else if y[j]>y[i] then
              if y[j]<high then high:=y[j]
              else
                else if y[j]>low then low:=y[j];
            end;
          end;
        sort_y(1,n);
        for i:=1 to n-1 do
          max((y[i+1]-y[i])*l);
        writeln(best);
      end.

```

2、Candy 的程序

```

program candy;
const
  maxn=1000;
var

```

```

left,right,high:array[1..maxn] of longint;
s:array[0..maxn,0..maxn] of longint;
now,res,leftmost,rightmost,i,j,k,n,m:longint;
f:text;
begin
  assign(f,'candy.in');
  reset(f);
  readln(f,n,m);
  fillchar(s,sizeof(s),0);
  for i:=1 to m do
    begin
      left[i]:=1; right[i]:=m; high[i]:=0;
    end;
  res:=0;
  for i:=1 to n do
    begin
      k:=0; leftmost:=1;
      for j:=1 to m do
        begin
          read(f,now); k:=k+now;
          s[i,j]:=s[i-1,j]+k;
          if now=0 then
            begin
              high[j]:=0; left[j]:=1; right[j]:=m;
              leftmost:=j+1;
            end
          else
            begin
              high[j]:=high[j]+1;
              if leftmost>left[j] then left[j]:=leftmost;
            end;
          end;
        rightmost:=m;
        for j:=m downto 1 do
          begin
            if high[j]=0 then
              begin
                rightmost:=j-1;
              end
            else
              begin
                if right[j]>rightmost then right[j]:=rightmost;
                now:=s[i,right[j]]+s[i-high[j],left[j]-1]-s[i-high[j],right[j]]-s[i,left[j]-1];
                if now>res then res:=now;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        end;
    end;
end;
writeln(res);
end.

```

3、 **Big Barn** 用第二种算法解的程序

```

program BigBarn;
var
    d:array[1..1000,1..1000] of longint;
    height,left,right:array[1..1000] of longint;
    leftmost,rightmost,res,i,j,k,t,n:longint;
    f:text;
begin
    assign(f,'bigbrn.in');
    reset(f);
    readln(f,n,t);
    fillchar(d,sizeof(d),0);
    for i:=1 to t do
        begin
            readln(f,j,k);
            d[j,k]:=1;
        end;
    close(f);
    for i:=1 to n do
        begin
            height[i]:=0; left[i]:=1; right[i]:=n;
        end;
    res:=0;
    for i:=1 to n do
        begin
            leftmost:=1;
            for j:=1 to n do
                if d[i,j]=1 then
                    begin
                        height[j]:=0; left[j]:=1; right[j]:=n;
                        leftmost:=j+1;
                    end
                else
                    begin
                        height[j]:=height[j]+1;
                        if leftmost>left[j] then left[j]:=leftmost;

```



```
    end;
    rightmost:=n;
    for j:=n downto 1 do
        if d[i,j]=1 then rightmost:=j-1
        else
            begin
                if rightmost<right[j] then right[j]:=rightmost;
                k:=height[j];
                if right[j]-left[j]+1<k then k:=right[j]-left[j]+1;
                if k>res then res:=k;
            end;
        end;
    assign(f,'bigbrn.out');
    rewrite(f);
    writeln(f,res);
    close(f);
end.
```