



- [首页](#)
- [关于](#)
- [友情链接](#)
- [文章目录](#)

## [最后的题解]NOIP2011提高组复赛 Day1 解题报告

十一  
2011  
13

本文由 [浅栖](#) 发表在 [OI](#)

[16 条评论](#)

//已更新

恩，CCF果然是欠吐槽了。两天之内的6道题目，无数据结构，无动态规划，无图论。

既然是真真正正的比赛，那我就真真正正的写一次解题报告吧。

最前面先贴上WJMZBMR神犇的[神题解](#)，涨RP。

完整题目和代码稍后放上。

代码已经放上，[附在文末](#)，两天的前两题经CCF测评AC。

篇幅问题，分两篇写了。

### Day1

#### 铺地毯

##### 【问题描述】

为了准备一个独特的颁奖典礼，组织者在会场的一片矩形区域（可看做是平面直角坐标系的第一象限）铺上一些矩形地毯。一共有 $n$ 张地毯，编号从1到 $n$ 。现在将这些地毯按照编号从小到大的顺序平行于坐标轴先后铺设，后铺的地毯覆盖在前面已经铺好的地毯之上。地毯铺设完成后，组织者想知道覆盖地面某个点的最上面的那张地毯的编号。注意：在矩形地毯边界和四个顶点上的点也算被地毯覆盖。

##### 【输入】

输入文件名为 carpet.in。

输入共  $n+2$  行。

第一行，一个整数  $n$ ，表示总共有 $n$ 张地毯。

接下来的  $n$  行中，第  $i+1$  行表示编号  $i$  的地毯的信息，包含四个正整数  $a, b, g, k$ ，每两个整数之间用一个空格隔开，分别表示铺设地毯的左下角的坐标  $(a, b)$  以及地毯在  $x$  轴和  $y$  轴方向的长度。

第  $n+2$  行包含两个正整数  $x$  和  $y$ ，表示所求的地面的点的坐标  $(x, y)$ 。

#### 【输出】

输出文件名为 `carpet.out`。

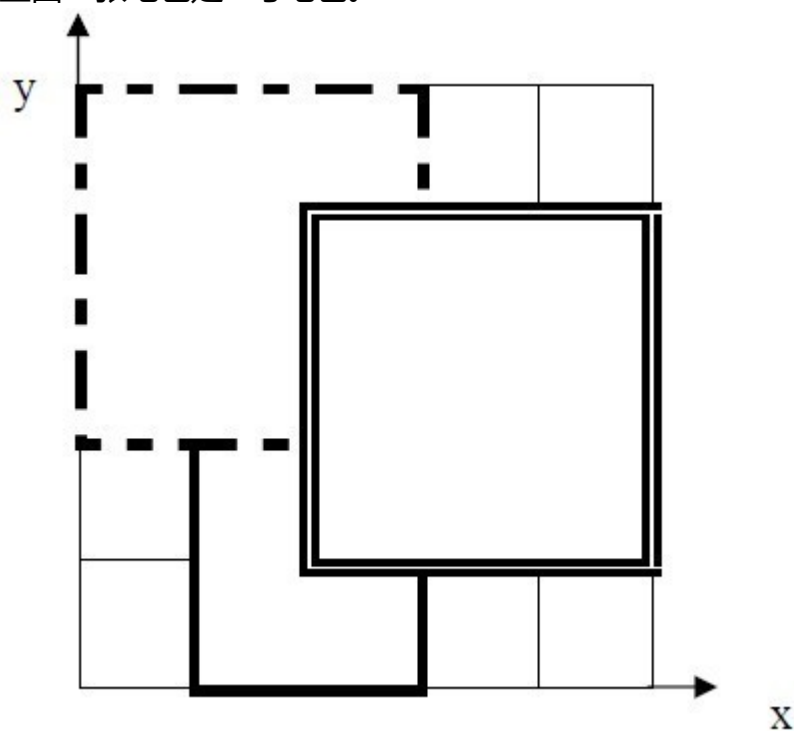
输出共 1 行，一个整数，表示所求的地毯的编号；若此处没有被地毯覆盖则输出 -1。

#### 【输入输出样例 1】

carpet.in	carpet.out
3	3
1 0 2 3	
0 2 3 3	
2 1 3 3	
2 2	

#### 【输入输出样例说明】

如下图，1 号地毯用实线表示，2 号地毯用虚线表示，3 号用双实线表示，覆盖点  $(2, 2)$  的最上面一张地毯是 3 号地毯。



#### 【输入输出样例 2】

carpet.in	carpet.out
3	-1
1 0 2 3	
0 2 3 3	
2 1 3 3	
4 5	

**【输入输出样例说明】**

如上图，1 号地毯用实线表示，2 号地毯用虚线表示，3 号用双实线表示，点 (4, 5) 没有被地毯覆盖，所以输出-1。

**【数据范围】**

对于 30%的数据，有 $n \leq 2$ ；

对于 50%的数据， $0 \leq a, b, g, k \leq 100$ ；

对于 100%的数据，有 $0 \leq n \leq 10,000$ ， $0 \leq a, b, g, k \leq 100,000$ 。

**【一句话题意】**

给定 $n$ 个按顺序覆盖的矩形，求某个点最上方的矩形编号。

**【考察知识点】**

枚举

**【思路】**

好吧我承认看到图片的一瞬间想到过二维树状数组和二维线段树。

置答案 $ans = -1$ ，按顺序枚举所有矩形，如果点在矩形内则更新 $ans$ 。注意题中给出的不是对角坐标，实际上是 $(a, b)$ 与 $(a+g, b+k)$ 。

还有一种办法可以从 $n$ 到1枚举矩形，一旦在矩形内就直接输出，可能会快一点。不过对于这题的范围什么都是浮云。。

恩，写完这题一看还没过8:30

**【时间复杂度】**

$O(n)$

**选择客栈****【问题描述】**

丽江河边有  $n$  家很有特色的客栈，客栈按照其位置顺序从1 到 $n$  编号。每家客栈都按照某一种色调进行装饰（总共 $k$  种，用整数 $0 \sim k-1$  表示），且每家客栈都设有一家咖啡店，每家咖啡店均有各自的最低消费。

两位游客一起去丽江旅游，他们喜欢相同的色调，又想尝试两个不同的客栈，因此决定分别住在色调相同的两家客栈中。晚上，他们打算选择一家咖啡店喝咖啡，要求咖啡店位于两人住的两家客栈之间（包括他们住的客栈），且咖啡店的最低消费不超过 $p$ 。

他们想知道总共有多少种选择住宿的方案，保证晚上可以找到一家最低消费不超过 $p$ 元的咖啡店小聚。

**【输入】**

输入文件 hotel.in，共 $n+1$  行。

第一行三个整数  $n, k, p$ ，每两个整数之间用一个空格隔开，分别表示客栈的个数，色调的数目和能接受的最低消费的最高值；接下来的  $n$  行，第 $i+1$  行两个整数，之间用一个空格隔开，分别表示 $i$  号客栈的装饰色调和 $i$  号客栈的咖啡店的最低消费。

**【输出】**

输出文件名为 hotel.out。

输出只有一行，一个整数，表示可选的住宿方案的总数。

**【输入输出样例 1】**

hotel.in	hotel.out
----------	-----------

5 2 3	3
0 5	
1 3	
0 2	
1 4	
1 5	

## 【输入输出样例说明】

客栈编号	①	②	③	④	⑤
色调	0	1	0	1	1
最低消费	5	3	2	4	5

2 人要住同样色调的客栈，所有可选的住宿方案包括：住客栈①③，②④，②⑤，④⑤，但是若选择住4、5 号客栈的话，4、5 号客栈之间的咖啡店的最低消费是4，而两人能承受的最低消费是3 元，所以不满足要求。因此只有前3 种方案可选。

## 【数据范围】

对于 30%的数据，有 $n \leq 100$ ；

对于 50%的数据，有 $n \leq 1,000$ ；

对于 100%的数据，有 $2 \leq n \leq 200,000$ ， $0 < k \leq 50$ ， $0 \leq p \leq 100$ ， $0 \leq \text{最低消费} \leq 100$ 。

## 【一句话题意】

合法区间 $[l, r]$ 定义： $l, r$ 的色调相同，且 $[l, r]$ 之间存在一个最低消费不超过 $p$ 。求合法区间总数。

## 【考察知识点】

二分查找/枚举

## 【思路】

贴吧神吐槽：CCF收了丽江多少钱？

①： $O(n^3)$ ：

看完题目后不知所云，再多看几遍，一个 $O(n^3)$ 的算法有了一点雏形。

用两层循环枚举区间的左右端点 $l, r$ ，再用一层循环判断区间内是否有可行的咖啡店，累计即可。这个算法思维难度和编程难度都非常低，但是只能过30%的数据，可以作为对拍程序备份。

②： $O(nk)$ 

再仔细思考，发现题中合法区间的限制条件其实很强。首先区间端点的色调必须相同，其次区间内必须要存在一个咖啡店最低消费不超过 $P$ 。

因此，如果我们用一层循环枚举左端点，并很快找到右端点的可行数，那么题目就能解决了。这里置答案为变量 $ans$ ，**千万注意类型要为int64**，昨天我就手抽打模板时直接打了 $ans:longint$ ，超级大杯具！！

这里首先要用到区间部分和优化。设 $sum[i, j]$ 为前 $i$ 个客栈中，色调为 $j$ 的客栈总数，那么：

$sum[i, j] = sum[i-1, j] \quad (color[i] \neq j)$

$sum[i, j] = sum[i-1, j] + 1 \quad (color[i] = j)$

这里要用 $O(NK)$ 的复杂度，是算法的瓶颈所在，不过对于题中的数据范围已经足够了。并且具体实现可以先用数组赋值 $sum[i] = sum[i-1]$ ，然后再为 $sum[i, color[i]] + 1$ ，应该会快很多。

我们还需要解决的问题就是，已知了L，如何快速找到R的可行范围？

再次注意区间内必须要存在一个咖啡店最低消费不超过P。

因此，如果L就是一个最低消费不超过P的咖啡店，那么R可以取到 $[L+1, n]$ 中所有色调为color[L]的客栈，即 $ans = ans + \text{sum}[n, \text{color}[L]] - \text{sum}[L, \text{color}[L]]$ ；

如果L是一个最低消费超过P的咖啡店，那么我们要找到一个 $T \in [L+1, n]$ ，且咖啡店T的最低消费不超过P，那么R就可以取到 $[T, n]$ 中所有色调为color[L]的客栈，即

$ans = ans + \text{sum}[n, \text{color}[L]] - \text{sum}[T-1, \text{color}[L]]$ 。

问题是我们如何找到这个T，其实很简单，二分查找即可。再次预设一个数组，保存所有最低消费不超过P的咖啡店序号，二分查找L即可。注意这里L一定不存在这个数组中，因此找到的应该是最靠近L且大于L的序号，细节处理很重要。找不到返回-1，不用累加ans就是了。

目测写完这题已经只剩1:30时间，且未对拍，第三题鸭梨巨大。

### ③ : $O(n \log n)$

这个办法比②更优一些。来自Clarkok的做法。

用list[i,j]表示颜色为i的第j个客栈，也就是将客栈按照颜色紧缩存储。另用pos[i]表示第i个旅馆在list[color[i]]中的位置。用线段树/ST算法(推荐)预处理出区间消费的最小值，也就是 $\min\{\text{cost}[i..j]\}$ ，易得到 $\min[k, i]$ 是非增的，注意这是后面二分的关键。

然后枚举第二个人，在list[color[i]]中用二分找到一个j满足 $\min[j, i] \leq P$ ，那么 $ans = ans + j$ ，因为list[color[i], 1..j]中必然都是颜色为color[i]，且区间最小值也都 $\leq P$ 。

### ④ : $O(n)$

这应该是最优算法了，无论从空间、时间、编程复杂度方面来说。

这个算法转自上善若水

记f(i)为第1~i的客栈中，编号最大的最低消费小于p的旅馆编号；r(i)为1~i-1号旅馆中，编号最大的与第i号旅馆色调相同的旅馆编号，count1(i)为第1~i-1号旅馆中与第i号旅馆色调相同旅馆数目，count2(i)为第1~i-1号旅馆中与第i号旅馆色调相同，且到第i号旅馆的路上存在最低消费不大于p的旅馆的旅馆数目。

(I)若 $f(i) < r(i)$ ，那么必然有 $f(i) = f(r(i))$ ，故 $\text{count2}(i) = \text{count2}(r(i))$ 。

(II)若 $f(i) \geq r(i)$ ，那么第1~r(i)号旅馆中，所有与第i号旅馆色调相同的旅馆到第i号旅馆的路上必然存在一个旅馆的最低消费不大于p。故此时 $\text{count2}(i) = \text{count1}(i)$ 。

从1到n扫描一次即可，时间复杂度 $O(n)$ 。具体实现时可以将数组压缩，空间复杂度 $O(k)$ 。

#### 【时间复杂度】

最低 $O(n)$

## mayan 游戏

#### 【问题描述】

Mayan puzzle 是最近流行起来的一个游戏。游戏界面是一个7行5列的棋盘，上面堆放着一些

方块，方块不能悬空堆放，即方块必须放在最下面一行，或者放在其他方块之上。游戏通关是指在规定的步数内消除所有的方块，消除方块的规则如下：

1、每步移动可以且仅可以沿横向（即向左或向右）拖动某一方块一格：当拖动这一方块时，如果拖动后到达的位置（以下称目标位置）也有方块，那么这两个方块将交换位置（参见输入输出样例说明中的图6 到图7）；如果目标位置上没有方块，那么被拖动的方块将从原来的竖列中抽出，并从目标位置上掉落（直到不悬空，参见下面图1 和图2）；

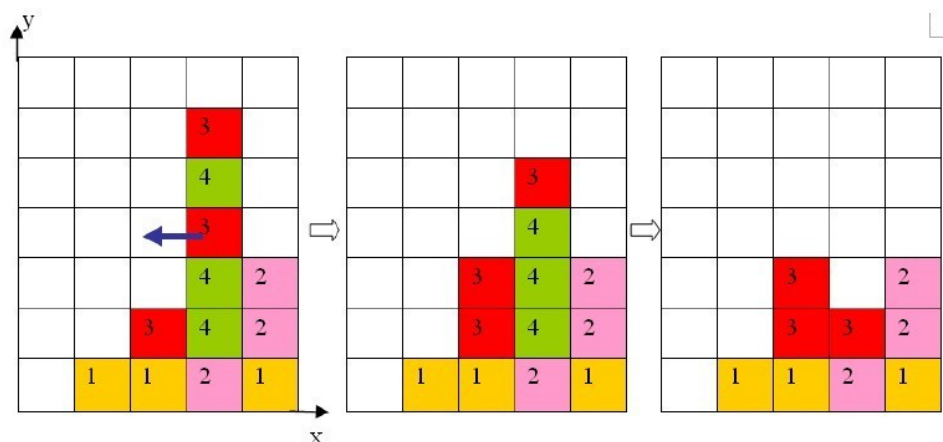


图 1

图 2

图 3

2、任一时刻，如果在一横行或者竖列上有连续三个或者三个以上相同颜色的方块，则它们将立即被消除（参见图1 到图3）。

注意：

- 如果同时有多组方块满足消除条件，几组方块会同时被消除（例如下面图4，三个颜色为1 的方块和三个颜色为2 的方块会同时被消除，最后剩下一个颜色为2 的方块）。
- 当出现行和列都满足消除条件且行列共享某个方块时，行和列上满足消除条件的所有方块会被同时消除（例如下面图5 所示的情形，5 个方块会同时被消除）。

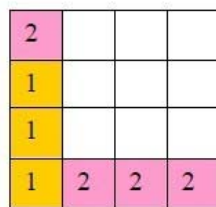


图 4

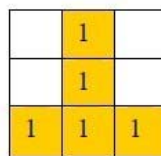


图 5

3、方块消除之后，消除位置之上的方块将掉落，掉落可能会引起新的方块消除。注意：掉落的过程中将不会有方块的消除。上面图 1 到图3 给出了在棋盘上移动一块方块之后棋盘的变化。棋盘的左下角方块的坐标为  $(0, 0)$ ，将位于  $(3, 3)$  的方块向左移动之后，游戏界面从图1 变成图2 所示的状态，此时在一竖列上有连续三块颜色为4 的方块，满足消除条件，消除连续3 块颜色为4 的方块后，上方的颜色为3 的方块掉落，形成图3 所示的局面。

【输入】

输入文件 mayan.in，共6 行。

第一行为一个正整数  $n$ ，表示要求游戏通关的步数。

接下来的 5 行，描述  $7 \times 5$  的游戏界面。每行若干个整数，每两个整数之间用一个空格隔开，每行以一个0 结束，自下向上表示每竖列方块的颜色编号（颜色不多于10 种，从1 开始顺序编号，相同数字表示相同颜色）。输入数据保证初始棋盘中没有可以消除的方块。

【输出】

输出文件名为 mayan.out。

如果有解决方案，输出  $n$  行，每行包含3 个整数  $x, y, g$ ，表示一次移动，每两个整数之间用一个空格隔开，其中  $(x, y)$  表示要移动的方块的坐标， $g$  表示移动的方向，1 表示向右移动，-1 表

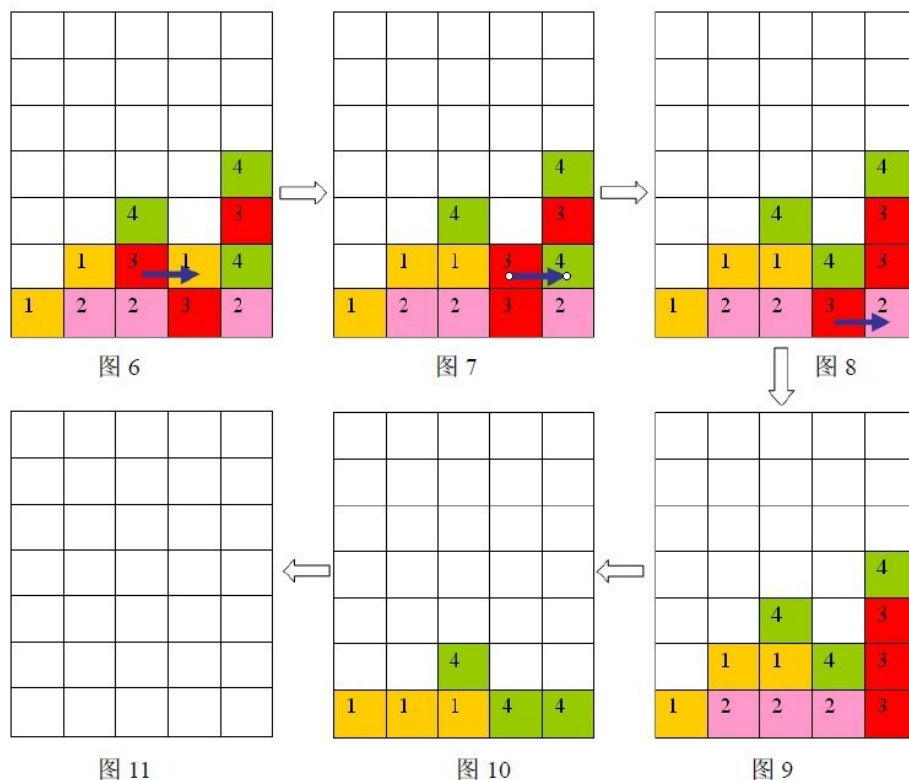
示向左移动。注意：多组解时，按照 $x$ 为第一关键字， $y$ 为第二关键字，1优先于-1，给出一组字典序最小的解。游戏界面左下角的坐标为 $(0, 0)$ 。如果没有解决方案，输出一行，包含一个整数-1。

【输入输出样例 1】

mayan.in	mayan.out
3	
1 0	
2 1 0	2 1 1
2 3 4 0	3 1 1
3 1 0	3 0 1
2 4 3 4 0	

【输入输出样例说明】

按箭头方向的顺序分别为图 6 到图11



样例输入的游戏局面如上面第一个图片所示，依次移动的三步是： $(2, 1)$ 处的方格向右移动， $(3, 1)$ 处的方格向右移动， $(3, 0)$ 处的方格向右移动，最后可以将棋盘上所有方块消除。

【数据范围】

对于 30%的数据，初始棋盘上的方块都在棋盘的最下面一行；

对于 100%的数据， $0 < n \leq 5$ 。

【一句话题意】

给定一个存在重力的矩阵，每次只能向左或右交换方块，连续3个或以上的方块群会被消除。求操作次数为 $N$ 时的操作步骤。

【考察知识点】

DFS

【思路】

我实在是忍不住吐槽了，第三题竟然是iPhone上Mayan Puzzle游戏的完全复制。[见贴吧此帖](#)



子。

[威锋上果然有完全题解啊](#)，第14关果然是样例啊有木有！！！人类智力威武，Orz.....  
是的，[我找到了PC版](#)，各位尽情虐吧，人脑过不去就用自己的程序跑。。。



对于这种完全裸搜索，固定搜索顺序，掐死搜索深度，基本无任何剪枝的题目彻底无语。。。。  
恩，吐槽完毕。

注意注意，第一页描述这题每个测试点时间限制是3S，我当时就震惊了！

首先马上确定这是搜索题目，抛弃所有动规贪心等等。然后锁定DFS，因为题中已经限定搜索深度，BFS是自找MLE。

再确定状态存储方式，我是将输入逆时针转90°后用数组保存，因此坐标之类的要特别注意。

然后，然后就没什么特别的了，搜索顺序题目已经给了。

设过程down(i)，表示将第i列的所有方块下沉。设函数clear，分行列清空后返回是否可以清空，这里用到连续区间指针l、r优化，类似前向星分组时的操作。于是：

```
while clear do
  for i:=1 to 5 do
    down(i);
```

注意在此之前如果有方块一交换就往下掉，那么就down交换的那两列。

表示码了200多行代码用了半小时，调试几乎用了1小时，那叫一个蛋疼。。。。

根据我的CCF测评分数可以看出，爆搜只能过50分，若常数小一点可以过80分。回家后我稍微修改了一下代码，用上了WJMZBMR的前两个剪枝，最后一个点在家里的电脑上跑了2s，其他点轻松通过了。

下面附加几个剪枝(from WJMZBMR)：

- 左右交换是等价的，根据题中的顺序，只需向右交换即可
- 某个颜色方块的数量 $\leq 2$ ，则很显然不能被消除
- 掉落不能改变方块的列，因此某列 $l_1$ 上某颜色方块数量 $\in [1, 2]$ ，则必须通过交换来从其他列 $l_2$ 得到方块。取一个 $l_1$ ，且 $l_2$ 最远，设这个距离为 $D_i$ ，那么必须要把多有颜色的 $D_i$ 消到0。而一次操作最多减少两个颜色的 $D_i$ ，因此最少操作次数为： $\max\{\max\{D_i\}(1 \leq i \leq 10), (\sum(D_i)(1 \leq i \leq 10) + 1)/2\}$
- 设个变量掐时，大概快超时的时候直接输出-1

【时间复杂度】



$O(k(4*7)^5)$ 

## 代码/标程

### 铺地毯

```
program carpet;
//uses sysutils;
const
    prname='carpet';
type
    juxing=record
        x1,y1,x2,y2:longint;
    end;
var
    fin,fout:text;
    i,j,k,l,r,m,n,x,y,s,t,ans:longint;
    a:array[0..11000]of juxing;
    t1,t2,t3,t4:longint;

procedure pin;
var
    i,j,k:longint;
begin
    readln(fin,n);
    fillchar(a,sizeof(a),0);
    for i:=1 to n do
        begin
            readln(fin,t1,t2,t3,t4);
            a[i].x1:=t1;
            a[i].y1:=t2;
            a[i].x2:=t1+t3;
            a[i].y2:=t2+t4;
        end;
    readln(fin,x,y);
end;

function ok(i:longint):boolean;
begin
    if (a[i].x1<=x)and(x<=a[i].x2) then
        if (a[i].y1<=y)and(y<=a[i].y2) then
            exit(true);
    exit(false);
end;

procedure main;
var
    i,j,k:longint;
begin
    ans:=-1;
    for i:=1 to n do
        if ok(i) then
            ans:=i;
end;

procedure pout;
```

```

var
    i,j,k:longint;
begin
    writeln(fout,ans);
end;

begin
    assign(fin,praname+'.in');
    assign(fout,praname+'.out');
    reset(fin);
    rewrite(fout);
    //time:=now;
    pin;
    main;
    pout;
    //writeln(fout,(now-time)*24*3600*1000:0:0);
    close(fin);
    close(fout);
end.

```

## 选择客栈

```

//O(nk)
program hotel;
//uses sysutils;
const
    praname='hotel';
type
    kezhan=record
        color,cost:longint;
    end;
var
    fin,fout:text;
    i,j,k,l,r,m,n,x,y,s,t:longint;
    a:array[-10..200100]of kezhan;
    sum:array[-10..200100,-10..60]of longint;
    colorsum,mincost:longint;
    cafe:array[-10..200100]of longint;
    v:array[-10..200100]of boolean;
    ans:int64;

procedure pin;
var
    i,j,k:longint;
begin
    readln(fin,n,colorsum,mincost);
    fillchar(a,sizeof(a),0);
    for i:=1 to n do
        readln(fin,a[i].color,a[i].cost);
end;

function find(x:longint):longint;
var
    i,j,k,mid:longint;
begin
    l:=1;
    r:=s;

```

```
find:=-1;
repeat
    mid:=(l+r) shr 1;
    if x<=cafe[mid] then
        begin
            r:=mid-1;
            find:=cafe[mid];
        end
    else
        l:=mid+1;
    until l>r;
end;

procedure main;
var
    i,j,k:longint;
begin
    for i:=0 to colorsum-1 do
        sum[0,i]:=0;
    for i:=1 to n do
        begin
            sum[i]:=sum[i-1];
            sum[i,a[i].color]:=sum[i-1,a[i].color]+1;
        end;
    s:=0;
    fillchar(cafe,sizeof(cafe),0);
    fillchar(v,sizeof(v),false);
    for i:=1 to n do
        if a[i].cost<=mincost then
            begin
                inc(s);
                cafe[s]:=i;
                v[i]:=true;
            end;
    ans:=0;
    for i:=1 to n-1 do //mei ju di yi ge ren
        begin
            if v[i] then
                begin
                    ans:=ans+(sum[n,a[i].color]-sum[i,a[i].color]);
                end
            else
                begin
                    j:=find(i);
                    if j=-1 then exit;
                    ans:=ans+(sum[n,a[i].color]-sum[j-1,a[i].color]);
                end;
        end;
    end;

    procedure pout;
    var
        i,j,k:longint;
    begin
        writeln(fout,ans);
    end;

begin
```

```

    assign(fin,praname+'.in');
    assign(fout,praname+'.out');
    reset(fin);
    rewrite(fout);
    //time:=now;
    pin;
    main;
    pout;
    //writeln(fout,(now-time)*24*3600*1000:0:0);
    close(fin);
    close(fout);
end.

```

## mayan 游戏

```

program mayan;
//uses sysutils;
const
    praname='mayan';
type
    game=record
        data:array[0..8,0..6]of longint;
    end;
var
    fin,fout:text;
    i,j,k,l,r,m,n,x,y,s,t,ans:longint;
    first:game;
    b:array[0..10]of game;
    caozuo:array[0..10,0..3]of longint;
    tt:array[0..8,0..6]of longint;
    cut:array[0..11]of longint;

procedure pin;
var
    i,j,k:longint;
begin
    readln(fin,n);
    fillchar(first,sizeof(first),0);
    for j:=1 to 5 do
        begin
            i:=8;
            repeat
                read(fin,k);
                if k=0 then break;
                dec(i);
                first.data[i,j]:=k;
            until false;
            readln(fin);
        end;
    end;

procedure swap(x,x1,y1,x2,y2:longint);
var
    t:longint;
begin
    t:=b[x].data[x1,y1];
    b[x].data[x1,y1]:=b[x].data[x2,y2];

```

```
    b[x].data[x2,y2]:=t;
end;

function ok(x:longint):boolean;
var
    i,j,k:longint;
begin
    ok:=true;
    for i:=1 to 5 do
        if b[x].data[7,i]<>0 then
            exit(false);
    end;

procedure down(x,lie:longint);
var
    i,j,k:longint;
begin
    for i:=6 downto 1 do
        if (b[x].data[i,lie]<>0)and(b[x].data[i+1,lie]=0) then
            begin
                for k:=7 downto 1 do
                    if b[x].data[k,lie]=0 then
                        break;
                    swap(x,i,lie,k,lie);
                end;
            end;
    end;

function clear(x:longint):boolean;
var
    i,j,k:longint;
    l,r:longint;
    step:longint;
begin
    fillchar(tt,sizeof(tt),0);
    l:=0;
    r:=0;
    step:=x;
    clear:=false;
    for i:=1 to 7 do
        begin
            for j:=1 to 5 do
                if b[x].data[i,j]<>b[x].data[i,j-1] then
                    begin
                        r:=j-1;
                        if (l<>0)and(r-l+1>=3)and(b[x].data[i,l]<>0) then
                            begin
                                for k:=1 to r do
                                    tt[i,k]:=1;
                                clear:=true;
                            end;
                        l:=j;
                    end;
            end;
            if (5-l+1>=3)and(b[x].data[i,5]<>0) then
                begin
                    for k:=1 to 5 do
                        tt[i,k]:=1;
                    clear:=true;
                end;
        end;
```

```

    end;
l:=0;
r:=0;
for i:=1 to 5 do
    begin
        for j:=1 to 7 do
            if b[x].data[j,i]<>b[x].data[j-1,i] then
                begin
                    r:=j-1;
                    if (l<>0)and(r-l+1>=3)and(b[x].data[l,i]<>0) then
                        begin
                            for k:=1 to r do
                                tt[k,i]:=1;
                                clear:=true;
                            end;
                            l:=j;
                        end;
                    if (7-l+1>=3)and(b[x].data[l,i]<>0)and(b[x].data[7,i]<>0) then
                        begin
                            for k:=1 to 7 do
                                tt[k,i]:=1;
                                clear:=true;
                            end;
                        end;
                    end;
                end;
            for i:=1 to 7 do
                for j:=1 to 5 do
                    if tt[i,j]=1 then
                        b[x].data[i,j]:=0;
                    end;
                end;
            end;

procedure dfs(step:longint);
var
    i,j,k:longint;
begin
    if step=n+1 then
        begin
            if ok(n+1) then
                begin
                    for i:=1 to n do
                        writeln(fout,caozuo[i,1], ' ',caozuo[i,2], ' ',caozuo[i,
                    close(fin);
                    close(fout);
                    halt;
                end;
            exit;
        end;
    //剪枝2
    fillchar(cut,sizeof(cut),0);
    for i:=1 to 5 do
        for j:=1 to 7 do
            inc(cut[b[step].data[j,i]]);
    for i:=1 to 10 do
        if (cut[i]=1)or(cut[i]=2) then
            exit;
    b[step+1]:=b[step];
    for i:=1 to 4 do
        for j:=7 downto 1 do
            //剪枝1 这里有一个问题，即第一层为01101且操作数为1时，必须要向右移

```



```
//因此这个剪枝需要稍加修改，这段程序不能AC!!!
    if (b[step+1].data[j,i]<>b[step+1].data[j,i+1]) then
        begin
            swap(step+1,j,i,j,i+1);
            down(step+1,i+1);
            down(step+1,i);
            while clear(step+1) do
                for k:=1 to 5 do
                    down(step+1,k);
            caozuo[step,1]:=i-1;
            caozuo[step,2]:=7-j;
            caozuo[step,3]:=1;
            dfs(step+1);
            b[step+1]:=b[step];
        end;
end;

procedure main;
var
    i,j,k:longint;
begin
    fillchar(b,sizeof(b),0);
    fillchar(caozuo,sizeof(caozuo),0);
    b[1]:=first;
    dfs(1);
end;

procedure pout;
var
    i,j,k:longint;
begin
    writeln(fout,-1);
end;


begin
    assign(fin,praname+'.in');
    assign(fout,praname+'.out');
    reset(fin);
    rewrite(fout);
    //time:=now;
    pin;
    main;
    pout;
    //writeln(fout,(now-time)*24*3600*1000:0:0);
    close(fin);
    close(fout);
end.
```

## 相关文章

- [\[最后的题解\]NOIP2011提高组复赛 Day2 解题报告](#) (10)
- [赛前全真模拟 Day1 经验教训&解题报告](#) (5)
- [常州八记 \(三\) : 动规](#) (4)
- [常州八记 \(二\) : 寻觅](#) (1)
- [再见!OI!](#) (26)

标签：[NOIP](#), [OI](#), [Pascal](#), [代码](#), [随记](#)

上一篇:[赛前全真模拟 Day1 经验教训&解题报告](#) 下一篇:[\[最后的题解\]NOIP2011提高组复赛 Day2 解题报告](#)

 本站遵行[知识共享署名-非商业性使用-相同方式共享 3.0 Unported许可协议](#)进行许可，转载请注明来自[qian-qi.com](#)。

## 16 条评论

1.  [NerdDaniel](#)

[2011 年 11 月 13 日 at 20:19](#)

# 1

第一题我也是开始想多了。考试不是8:30开始吗？你1min内写完的？拜！  
第二题看错题，本来写个线段树可以的。巨大悲剧！  
第三题1h差一点调出来。悲剧！

[回复](#)

○  浅栖

[2011 年 11 月 13 日 at 20:22](#)

# 2

。。。当然是提前就进了教室啊，做好了一切准备工作还没到8:30，密码也是提早公布的。  
第二题写代码前我真的深思熟虑了好久好久。。  
第三题200多行代码调试到最后已经完全混乱了啊魂淡。。。。。。

杯具握手。

[回复](#)

2.  sevenkplus

[2011 年 11 月 14 日 at 21:20](#)

# 3

第三题需要剪枝。

[回复](#)



浅栖

[2011 年 11 月 15 日 at 14:07](#)

# 4

的确需要剪枝，但是光调试就花了我1个小时了，实在没什么时间去写剪枝了。

[回复](#)

sevenkplus

[2011 年 11 月 15 日 at 14:29](#)

# 5

您这是解题报告而不是考场过程记录啊.....

[回复](#)

浅栖

[2011 年 11 月 15 日 at 14:32](#)

# 6

说实话我一开始就想写成考场过程记录的，因为就是OI总结吧。  
但是后来热心的大牛把此文发到贴吧上，所以看的人就多了。。。。  
现在还在学校，回家后我会去看看神牛的题解，补充这两篇文章的。



路过

[2011 年 11 月 15 日 at 23:37](#)

# 7

第二题完全可以 $O(N)$ 的楼主想过没有，枚举右边的人，同时更新各颜色的数量，40行搞定

[回复](#)

浅栖

[2011 年 11 月 19 日 at 15:56](#)

# 8

恩，在WJMZBMR的题解里看到了

[回复](#)



浅栖

[2011年11月19日 at 18:45](#)

# 9

求详细解释，WJMZBMR里就一句 $O(L)$ 得出没了。。。

[回复](#)

路过

[2011年11月20日 at 13:00](#)

# 10

开个数组记录到当前第J种颜色的个数，枚举右边的人位置，从1到N，若该位置小于P那么更新数组，否则不更新。大概代码A:颜色B:费用T:个数R:上次更新位置...for i=1 to n do begin if bi<=p then begin for j=r+1 to i do inc(t[a[j]])..dec ans.r=i.end.ans=ans+t[a[i]].end

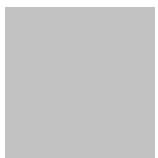
[回复](#)

路过

[2011年11月20日 at 13:00](#)

# 11

开个数组记录到当前第J种颜色的个数，枚举右边的人位置，从1到N，若该位置小于P那么更新数组，否则不更新。大概代码A:颜色B:费用T:个数R:上次更新位置...for i=1 to n do begin if bi<=p then begin for j=r+1 to i do inc(t[a[j]])..dec ans.r=i.end.ans=ans+t[a[i]].end手机打好累

[回复](#)

路过

4.

[2011年11月15日 at 23:40](#)

# 12

第三题裸的也就150行额...思路清晰的话都不用调...我悲剧看成1秒加的卡点...

[回复](#)

浅栖

[2011 年 11 月 19 日 at 16:06](#)

# 13

问题是考试那时候，都已经写了很多，不可能删了重写的，尽管思路已经乱了

[回复](#)



5. [redswallow](#)

[2011 年 11 月 21 日 at 14:47](#)

# 14

现在noip考2天了阿orz

[回复](#)



○ [浅栖](#)

[2011 年 11 月 26 日 at 19:19](#)

# 15

恩，刚今年改了赛制

[回复](#)



6. [qiao](#)

[2012 年 5 月 14 日 at 17:39](#)

# 16

第三题是怎么用变量计时的？？

[回复](#)

## 引用

目前还没有关于这篇文章的引用。

## 留下评论

姓名

电子邮件

站点

确认

## 搜索

搜索：

搜索

## 文章分类

- [常道](#) (12)
- [OI](#) (21)
- [呖语](#) (8)
- [痴言](#) (5)
- [无辻](#) (5)
- [言停](#) (8)
- [浮光](#) (3)
- [希音](#) (4)

## 标签云

## 随机文章



- [Happy Ending - Mika](#)
- [常州八记 \(二\) : 寻觅](#)
- [2011年 暑假生活拾遗](#)
- [\[福利\] 新型Fan Q1ang利器](#)
- [赛前全真模拟 Day1 经验教训&解题报告](#)

## 近期评论

- [exp618 : 七拐八拐翻过来的, 希望能加个友情链接, 谢谢~](#)
- [林思 : 祝保送考试顺利](#)
- [Makeecat : 楼盖得好混乱啊](#)
- [qiao : 第三题是怎么用变量计时的??](#)
- [lyic : 很久没过来看了; 居然惊奇的在列表里看到我的名字, 有点小激动](#)
- [lyic : 我也入了一个32GB Wi-Fi 其实你别对原包抱太大希望, 很少水客会这样冒险, 深圳海关抓的很严, ...](#)

## 功能

- [注册](#)
- [登录](#)
- [文章 RSS](#)
- [评论 RSS](#)
- [WordPress.org](#)

## 链接表

- [91OI.com 论坛](#) | OI学习论坛
- [bangbang93](#) | 技术宅的革命根据地
- [Clarkok](#) | Pre大牛ing
- [Dick Wu](#) | 很谦虚的大牛
- [Evan](#) | OIer
- [Extreme's Blog](#) | 特色OIer
- [Indeed](#) | 稳重的Blogger
- [lsdsjy](#) | 初中生OIer
- [Marylandeny-Chu的OI世界](#) | OIer
- [Matrix67](#) | 神级OIer的神级Blog
- [NerdDaniel](#) | 靠谱OIer
- [oCourse](#) | 公开课字幕组
- [OI](#) | 难以更新的题解分站
- [ParabolaWorld](#) | 一个新的Blog
- [wyj](#) | OIer!
- [宇宙的心弦](#) | 物理大神牛
- [木遥的窗子](#) | 我最喜欢的博客
- [煎蛋](#) | 无聊必备
- [笔锋网](#) | 用文字记录时间
- [西子](#) | 我老妈的博客 欢迎~

