

CodeChef March Challenge 2021

[比赛链接](#)

Paparazzi Gennady

题意

给定 n 条垂直的线段，其中第 i 条线段横坐标为 i 纵坐标范围为 $[0, h_i]$

求最大的 d 使得存在 i 使得线段 (i, h_i) to $(i+d, h_{i+d})$ 与其他所有给定垂线都不相交。

题解

首先构造 n 个点的凸包，设凸包的边界上的点(如果两点以上共线则忽略除两端点以外的点)为 $x_1, x_2, x_3, \dots$

对 $x_i \leq x_{i+1}$ 不难发现一定有 $x_{i+1} - x_i \leq d$ 所以 d 一定小于 $x_{i+1} - x_i$

于是答案为 $\max(x_{i+1} - x_i)$ 时间复杂度 $O(n)$

```
const int MAXN=5e5+5;
int h[MAXN],st[MAXN];
int main()
{
    int T=read_int();
    while(T--){
        int n=read_int();
        _for(i,0,n)h[i]=read_int();
        int top=0,ans=0;
        st[top]=0;
        _for(i,1,n){
            while(top){
                int x1=st[top]-st[top-1],x2=i-st[top-1];
                int y1=h[st[top]]-h[st[top-1]],y2=h[i]-h[st[top-1]];
                if(1LL*y2*x1>=1LL*y1*x2)top--;
                else break;
            }
            ans=max(ans,i-st[top]);
            st[++top]=i;
        }
        enter(ans);
    }
    return 0;
}
```

```
}
```

Consecutive Adding

题意

给定两个 $r \times c$ 的矩阵 A 和 B 。每次操作可以选择 A 中某行或某列中连续的 x 个元素同时加上 v 。

问能否经过有限次操作将 A 转化为 B 。

题解

首先我们证明题目所给操作与以下两种操作等效：

1. $a_i += v, a_{i+x} -= v$
2. 仅对原矩阵左上角 $x \times x$ 的矩阵进行原操作

首先不难发现可以通过题目原操作得到操作 1 和操作 2。接下来考虑怎么用操作 1 和操作 2 得到原操作。

假设原操作是对某行中连续的元素加上 v ，则可以先通过操作 1 对该行前 x 个元素加上 v ，再通过操作 2 转移到对应位置。

然后不难发现操作顺序与结果无关，于是不妨假设先只进行操作 2，然后再进行操作 1。

因为之后只能进行操作 1，于是必须先通过操作 2 将除左上角 $x \times x$ 的矩阵以外的位置全部变成 0。

接下来只考虑 $x \times x$ 的矩阵，将行操作和列操作视为 $2n$ 的结点 $a_{i,j}$ 相当于在行操作的第 i 个结点和列操作的第 j 个结点连一条权值 $a_{i,j}$ 的边。

于是最终只需要保证每条边的两个端点的数值和等于该边的权值即可。

另外我们发现如果存在合法解，可以对每个行操作加上 v ，每个列操作减去 v 仍然是合法解。

于是我们不妨假设行操作的第一个结点对应的值为 0，然后跑一遍 `dfs` 即可。时间复杂度 $O((rc+x^2))$ 。

```
const int MAXN=1e3+5;
LL a[MAXN][MAXN],s[2][MAXN];
bool vis[2][MAXN];
int x;
bool dfs(int t,int p,LL v){
    if(vis[t][p])
        return s[t][p]==v;
    vis[t][p]=true;
    s[t][p]=v;
```

```
bool flag=true;
_for(i,0,x){
    if(t==0)
        flag&=dfs(!t,i,a[p][i]-v);
    else
        flag&=dfs(!t,i,a[i][p]-v);
}
return flag;
}
int main()
{
    int T=read_int();
    while(T--){
        int r=read_int(),c=read_int();
        x=read_int();
        _for(i,0,x)_for(j,0,x)a[i][j]=0;
        _for(i,0,r)_for(j,0,c)a[i%x][j%x]+=read_int();
        _for(i,0,r)_for(j,0,c)a[i%x][j%x]-=read_int();
        _for(i,0,x)s[0][i]=s[1][i]=0,vis[0][i]=vis[1][i]=false;
        if(dfs(0,0,0))
            puts("YES");
        else
            puts("NO");
    }
    return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_mar21

Last update: 2021/03/29 21:22

