

K题解题报告：线段树套线段树

题意：维护矩阵区域的最大最小值，修改单点的值

先来演示俩错误做法：~~多么痛的领悟~~

1. 把矩阵拼接成一维数组，用线段树维护区间最值和修改。题中的区域查询即为同时查询多个无交叉区间的最值（时间爆表）
2. 分块。单纯的分块过不了，极容易被卡；可能有其他分块的方法能过这道题吧，反正我是过不了。

正确做法：线段树套线段树：

外层线段树的每个节点里面都含有另一个线段树；

外层线段树可看作是维护x轴，内层线段树可看作是维护y轴；

```
struct Node{//这样存节点信息
    int mmax;
    int mmin;
}node[maxn<<2][maxn<<2];
```

线段树维护最值；

```
void build2(int out,int cnt,int out_l,int out_r,int l,int r){
    if(l==r){
        int mid=(l+r)/2;
        build2(out,cnt<<1,out_l,out_r,l,mid);
        build2(out,(cnt<<1)+1,out_l,out_r,mid+1,r);
        node[out][cnt].mmax=max(node[out][cnt<<1].mmax,node[out][(cnt<<1)+1].mmax);
        node[out][cnt].mmin=min(node[out][cnt<<1].mmin,node[out][(cnt<<1)+1].mmin);
    }
    else{
        if(out_l==out_r) {
            node[out][cnt].mmax=arr[out_l][l];
            node[out][cnt].mmin=arr[out_r][r];
        }
        else{
            node[out][cnt].mmax=max(node[out<<1][cnt].mmax,node[(out<<1)+1][cnt].mmax);
            node[out][cnt].mmin=min(node[out<<1][cnt].mmin,node[(out<<1)+1][cnt].mmin);
        } //注意这个else里面的语句很耐人寻味
    } //递归、回溯的顺序决定了这些语句的正确实现
}
```

查询某区域最值时，先锁定x轴的区域，再锁定y轴的区域；

```
int find1(int cnt,int l,int r,int x1,int y1,int x2,int y2){
    if(l>=x1&&r<=x2){
        find2(cnt,1,1,n,x1,y1,x2,y2);
    }//x区间锁定了一处，开始锁定y区间
    else{
        int mid=(l+r)/2;
        if(x1<=mid) find1(cnt<<1,l,mid,x1,y1,x2,y2);
        if(x2>mid) find1((cnt<<1)+1,mid+1,r,x1,y1,x2,y2);
    }
}
```

修改时注意回溯时候更新最值；（修改和构建其实很像的）

注意以上几点，线段树套线段树应该就没啥大问题了。

时间复杂度 $O(n\log n\log n)$:

因为是嵌套的，所以单次查询和修改 $\log n * \log n$