线段树 重复染色，离散

本题大意：给定一些海报，可能相互重叠，告诉你每个海报

的宽度（高度都一样的）和先后叠放顺序，问没有被完全盖住的有多少张？

海报最多10000张，但是墙有10000000块瓷砖长，海报不会落在瓷砖中间。

如果直接建树，就算不TLE,也会MLE。即单位区间长度太多。

其实10000张海报，有20000个点，最多有19999个区间。对各个区间编号，就是离散化。然后建数。

其实浮点数也是一样离散化的。

#include <iostream>

#include<cstdio>

#include<algorithm>

#define ll long long

using namespace std;

struct node

{

int l,r;

}poster[10010];

struct node1

{

int l,r;

bool cover;

}a[10010\*4];

int point[20010\*4];

int \_hash[10000010];

inline void build(int root,int l,int r)

{

a[root].l=l;

a[root].r=r;

a[root].cover=0;

if(l==r)return;

int mid=(l+r)>>1;

build(root<<1,l,mid);

build(root<<1|1,mid+1,r);

}

inline bool post(int root,int x,int y)

{

if(a[root].cover)

return 0;

if(a[root].l==x && a[root].r==y)

{

a[root].cover=1;

return 1;

}

int mid=(a[root].l+a[root].r)>>1;

bool cnt;

if(y<=mid)

cnt=post(root<<1,x,y);

else if(x>mid)

cnt=post(root<<1|1,x,y);

else

{

bool b1=post(root<<1,x,mid);

bool b2=post(root<<1|1,mid+1,y);

cnt=b1||b2;

//cnt=post(root<<1,x,mid)||post(root<<1|1,mid+1,y);

//注意这里一定不能这么写，否则一旦第一个post为真便不会去执行后面的post了

}

if(a[root<<1].cover && a[root<<1|1].cover)

a[root].cover=1;

return cnt;

}

int main()

{

//freopen("input.txt","r",stdin);

int T,n;

scanf("%d",&T);

while(T--)

{

scanf("%d",&n);

int ncount=0;

for(int i=1;i<=n;++i)

{

scanf("%d%d",&poster[i].l,&poster[i].r);

point[++ncount]=poster[i].l;

point[++ncount]=poster[i].r;

}

sort(point+1,point+1+ncount);

ncount=unique(point+1,point+1+ncount)-(point+1);

for(int i=1;i<=ncount;++i)

\_hash[point[i]]=i;

build(1,1,ncount);

int ans=0;

for(int i=n;i>=1;--i)

if(post(1,\_hash[poster[i].l],\_hash[poster[i].r]))

++ans;

printf("%d\n",ans);

}

return 0;

}