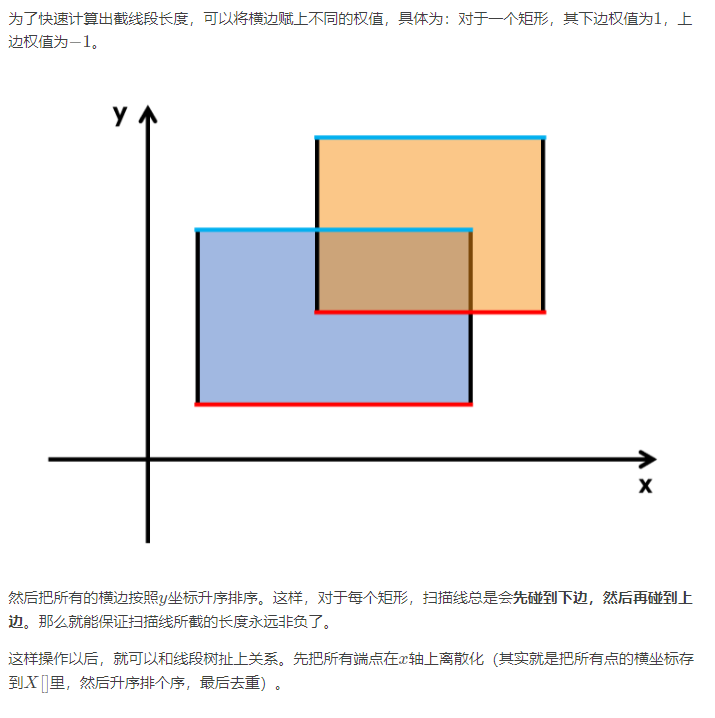
求 n 个矩形的面积并。

输入格式

第一行一个正整数 n。

接下来n行每行四个非负整数x1,y1,x2,y2，表示一个矩形的左下角坐标为(x1,y1)，右上角坐标为(x2,y2)。

**扫描线**：假设有一条扫描线从一个图形的下方扫向上方（或者左方扫到右方），那么通过分析扫描线被图形截得的线段就能获得所要的结果。该过程可以用**线段树**进行加速。



然后从下向上扫，把每一条横边加入线段树中，然后直接询问目前整个线段树中所有线段的总长度是多少，在乘以这一条横边与上一条横边之间的高，最后把所有的乘积相加就是矩形面积的并

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 200005

#define y1 y3

using namespace std;

ll b[mn],

ans=0,n,m;

struct w{ll a,b,s,len;}a[mn\*6];

//像下面这种建树的，线段树数组要开6倍

struct ww{ll l,r,h,mark;}c[mn];

bool cmp(ww a,ww b){return a.h<b.h;}

void up(ll v)

{

if(a[v].s)a[v].len=b[a[v].b]-b[a[v].a];

else a[v].len=a[v\*2].len+a[v\*2+1].len;

}

void bui(ll v,ll l,ll r)

{ll mid;

a[v].a=l;a[v].b=r;

a[v].len=0;a[v].s=0;

if(l+1==r)return ;

mid=(l+r)/2;

bui(v\*2,l,mid);

bui(v\*2+1,mid,r);

up(v);

}

void add(ll v,ll l,ll r,ll x)

{

if(b[a[v].b]<=l||b[a[v].a]>=r)return ;

if(l<=b[a[v].a]&&b[a[v].b]<=r){a[v].s+=x;up(v);return ;}

add(2\*v,l,r,x);

add(2\*v+1,l,r,x);

up(v);

}

int main()

{

ll x,y,z,i,j,k;

char ch;

cin>>n;

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

{

ll x1,x2,y1,y2;

scanf("%lld%lld%lld%lld",&x1,&y1,&x2,&y2);

b[2\*i-1]=x1;b[2\*i]=x2;

c[2\*i-1]=(ww){x1,x2,y1,1};

c[2\*i]=(ww){x1,x2,y2,-1};

}

n\*=2;

sort(c+1,c+n+1,cmp);

sort(b+1,b+n+1);

ll tot=unique(b+1,b+n+1)-b-1;

bui(1,1,tot);

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

{

add(1,c[i].l,c[i].r,c[i].mark);

ans+=a[1].len\*(c[i+1].h-c[i].h);

}

cout<<ans;

return 0;

}