**题目描述 Description**

    给定x轴上的N（0<N<100）条线段，每个线段由它的二个端点a\_I和b\_I确定，I=1,2,……N.这些坐标都是区间（－999，999）的整数。有些线段之间会相互交叠或覆盖。请你编写一个程序，从给出的线段中去掉尽量少的线段，使得剩下的线段两两之间没有内部公共点。所谓的内部公共点是指一个点同时属于两条线段且至少在其中一条线段的内部（即除去端点的部分）。

**输入描述 Input Description**

    输入第一行是一个整数N。接下来有N行，每行有二个空格隔开的整数，表示一条线段的二个端点的坐标。

**输出描述 Output Description**

    输出第一行是一个整数表示最多剩下的线段数。

**样例输入 Sample Input**

3

6  3

1  3

2  5

**样例输出 Sample Output**

2

**数据范围及提示 Data Size & Hint**

0<N<100

根据右端点排序

先结束的一定开始时间在前面

#include<bits/stdc++.h>

using namespace std;

struct node

{

int l,r;

};

bool cmp(node x,node y) //想用sort依据线段的右端进行升序排序

{

return x.r<y.r;

}

int main()

{

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

int n;

cin>>n;

node a[n];

for(int i=0;i<n;i++)

{

cin>>a[i].l>>a[i].r;

if(a[i].l>a[i].r)swap(a[i].l,a[i].r); //线段，都把小的设置为左端

}

sort(a,a+n,cmp); //已经按照右端进行排序了。先结束的一定开始时间在前面。

int right=a[0].r;

int cnt(1);

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

if(a[i].l>=right)

{

right=a[i].r;

cnt++;

}

cout<<cnt;

}