**题目描述 Description**

数轴上有n条线段，线段的两端都是整数坐标，坐标范围在0~1000000，每条线段有一个价值，请从n条线段中挑出若干条线段，使得这些线段两两不覆盖（端点可以重合）且线段价值之和最大。

n<=1000

**输入描述 Input Description**

第一行一个整数n，表示有多少条线段。

接下来n行每行三个整数, ai bi ci，分别代表第i条线段的左端点ai，右端点bi（保证左端点<右端点）和价值ci。

**输出描述 Output Description**

输出能够获得的最大价值

**样例输入 Sample Input**

3

1 2 1

2 3 2

1 3 4

**样例输出 Sample Output**

4

**数据范围及提示 Data Size & Hint**

数据范围

对于40%的数据，n≤10；

对于100%的数据，n≤1000；

0<=ai,bi<=1000000

0<=ci<=1000000

#include<bits/stdc++.h>

using namespace std;

struct node

{

int l,r,c;

};

bool cmp(node x,node y)

{

return x.r<y.r;

}

int main()

{

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

node a[1010];

int n,i,j;

cin>>n;

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

{

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

if(a[i].l>a[i].r)swap(a[i].l,a[i].r);

}

sort(a,a+n,cmp);  //根据线段的右端点进行排序

int ans(0);int max=a[0].c;

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

{

max=0;

for(int j=0;j<i;j++) //找出当前线段的前面线段中最大的合法线段。

if(a[j].r<=a[i].l&&a[j].c>max)

max=a[j].c;

a[i].c+=max;

if(a[i].c>ans)ans=a[i].c;

}

cout<<ans;

return 0;

}

a[j].r<=a[i].l

注意是小于等于