时间限制：1秒

空间限制：32768K

平面内有n个矩形, 第i个矩形的左下角坐标为(x1[i], y1[i]), 右上角坐标为(x2[i], y2[i])。

如果两个或者多个矩形有公共区域则认为它们是相互重叠的(不考虑边界和角落)。

请你计算出平面内重叠矩形数量最多的地方,有多少个矩形相互重叠。

**输入描述:**

输入包括五行。  
第一行包括一个整数n(2 <= n <= 50), 表示矩形的个数。  
第二行包括n个整数x1[i](-10^9 <= x1[i] <= 10^9),表示左下角的横坐标。  
第三行包括n个整数y1[i](-10^9 <= y1[i] <= 10^9),表示左下角的纵坐标。  
第四行包括n个整数x2[i](-10^9 <= x2[i] <= 10^9),表示右上角的横坐标。  
第五行包括n个整数y2[i](-10^9 <= y2[i] <= 10^9),表示右上角的纵坐标。

**输出描述:**

输出一个正整数, 表示最多的地方有多少个矩形相互重叠,如果矩形都不互相重叠,输出1。

**输入例子1:**

2

0 90

0 90

100 200

100 200

**输出例子1:**

2

暴力你也不会暴力！！！

暴力做法，枚举每个横纵坐标，计算矩阵内的个数，找最大

最多50\*50\*50嘛

#include<stdio.h>

#include<iostream>

#include<cstring>

//#include<cmath>

#include<set>

typedef long long ll;

using namespace std;

ll x1[100];

ll x2[100];

ll y1[100];

ll y2[100];

set<ll> xx,yy;

set<ll>::iterator it1,it2;

int main()

{

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

int n;

cin>>n;

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

{

cin>>x1[i];

xx.insert(x1[i]);

}

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

{

cin>>y1[i];

yy.insert(y1[i]);

}

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

{

cin>>x2[i];

xx.insert(x2[i]);

}

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

{

cin>>y2[i];

yy.insert(y2[i]);

}

ll ans(0);

for(it1=xx.begin();it1!=xx.end();it1++)

for(it2=yy.begin();it2!=yy.end();it2++)

{

ll cnt(0);

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

if(x1[i]<=(\*it1) && y1[i]<=(\*it2)&&x2[i]>(\*it1) &&y2[i]>(\*it2))

//注意这个，是定义点在这个矩形中是在内部或者左边和下边上，而不能包含右边和上边

//一般都是四个边中选两个，否则如果4条边都可以 的话，如果一个点位于边上，则它可能会被左边的矩形和右边的矩形算两次

{

cnt++;

}

ans=max(ans,cnt);

}

cout<<ans<<"\n";

return 0;

}