题目大意：

给你n根木棍的长度和重量。根据要求求出制作该木棍的最短时间。建立第一个木棍需要1分钟，如果接着制作的木棍比这个木棍的长度长，重量要重，那么接着制作的木棍不需要花费时间！然后如果再继续接着制作，则下一个木棍要比上一个木棍的长度长，重量大，则这个木棍也不需要花费时间！依次类推，反之，则需要花费一分钟，然后让你求出制作这一批木棍花费的最少的时间是多少！

解题思路：

其实大概明白了题意，便可以知道，这题考的是贪心！然后关键就在于贪心策略的选择！题目说了求最小时间，那就肯定要求长度和重量都要相对来说最小，我的策略如下：

1.首先按照木棍的长度进行排序，如果长度相等则重量小的放在前面。

2.当然排序完后，不一定都是下一根木棍重量和长度都大于前一根的，但是我们可以根据排序建立的数组再进行多次扫描，比较。

3.根据建立的数组依次比较他们的重量，如果大于标记数组跳过，否则ans加1。

注意

不能将判断条件随便拿到for循环里面去

#include<iostream>

#include<stdio.h>

#include<map>

#include<cstring>

#include<queue>

#include<algorithm>

#include<iomanip>

using namespace std;

struct node

{

int l,w;

};

node a[5010];

bool b[5010];

bool cmp(node a,node b)

{

if(a.l==b.l)

return a.w<b.w;

else return a.l<b.l;

}

int main()

{

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

int T,n;

scanf("%d",&T);

while(T--)

{

scanf("%d",&n);

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

scanf("%d%d",&a[i].l,&a[i].w);

sort(a,a+n,cmp);

memset(b,0,sizeof(b));

int cnt=0;

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

if(!b[i])

{

b[i]=1;

cnt++;

int weight=a[i].w;

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

if(!b[j] && a[j].w>=weight)

{

b[j]=1;

weight=a[j].w;

}

}

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

}

return 0;

}