活动选择 (act.cpp)

【问题描述】

假设有一个需要使用某一资源的 n 个活动所组成的集合 S，S={1,…,n}。该资源一次只能被一个活动所占用，每一个活动有一个开始时间 bi 和结束时间 ei(bi<=ei)。若 bi>=ej 或 bj>=ei，则称活动 i 和活动 j 兼容（即两个活动在时间上不重叠）。

你的任务是：选择由互相兼容的活动所组成的最大集合（亦即，找出最多的、相容的活动）。

【输入】

输入文件名为 act.in，共 n+1 行，其中第 1 行为 n，第 2 行到第 n+1 行表示 n 个活动的开始时间和结束时间(中间用空格隔开)，格式为：

n

b1 e1

……

bn en

【输出】

输出文件名为 act.out，输出最多的、相容的活动数。

【样例输入】

```
11
3 5
1 4
12 14
8 12
0 6
8 11
6 10
5 7
3 8
5 9
2 13
```

【样例输出】

```
4
```

【参考程序】

```
#include <algorithm>
#include <iostream>
using namespace std;

struct node{
   int b;
   int e;
}act[100];

bool cmp(node a, node b)
{
   if (a.e !=b.e)
```

```
    return a.e < b.e;
  else
    return a.b > b.b;
}

int main()
{
  int n, i, j, num, last;

  //ios::sync_with_stdio(false);

  cin >> n;

  for (i=0; i<n; i++)
    cin >>act[i].b >> act[i].e;

  sort(act, act+n, cmp);

  num=1;  last=0;

  for (i=1; i<n; i++){
    if (act[last].e <= act[i].b) {
      last = i;
      num++;
    }
  }

  cout << num << endl;
  return 0;
}
```