信奥算法

区间模型1
最多不重叠区间

有n个区间，选择尽量多互相不重叠的区间

看最多电影

电影节上有n项电影放映活动可以观看，第i部电影分别从时刻s_i开始，时刻t_i结束。最多可以完整观看几部电影？（不允许结束时刻和开始时刻重叠，n<=100）

输入样例
2
1 0
4 1

输出样例
1

输入样例
5
1 2 4 6 8
3 5 7 9 10

输出样例
3

吃最多美食

美食节上有n个美食摊位，排队都很火爆，要想吃到第i个美食必须从时刻si开始，到时刻ti结束一直排队在这个摊位前。请问最多可以吃到几个美食？（允许结束时刻和开始时刻重叠，n<=100）

输入样例
2
1 0
4 1

输出样例
2

输入样例
5
1 2 4 6 8
3 5 7 9 10

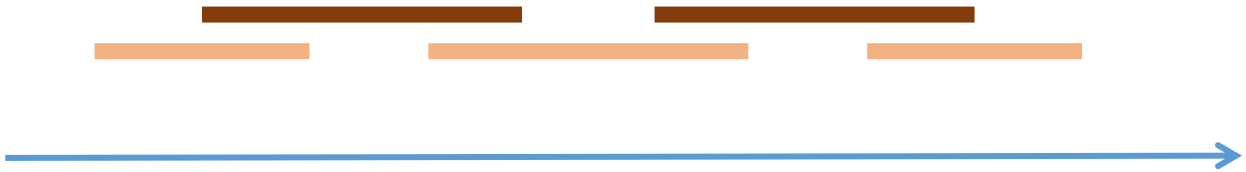
输出样例
3

每场电影：抽象成一个区间

输入样例
5
1 2 4 6 8
3 5 7 9 10

输出样例
3

每场电影对应一个区间
区间左端点为 开始时间
区间右端点为 结束时间



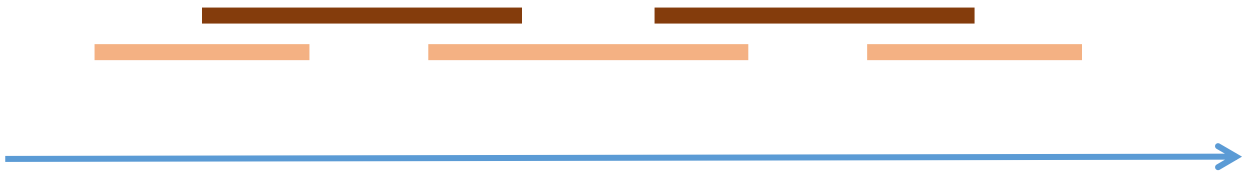
"时间"信息可以用数轴表示

每个美食摊位：抽象成一个区间

输入样例
5
1 2 4 6 8
3 5 7 9 10

每个美食对应一个区间
区间左端点为 开始排队时间
区间右端点为 结束排队时间

输出样例
3



"时间"信息可以用数轴表示

最大不重叠区间数

错误的贪心算法：

不断循环重复选择活动：

每次在可选活动中，选择**用时最少**的活动

能否举出反例？

最大不重叠区间数

错误的贪心算法：

不断循环重复选择活动：

每次在可选活动中，选择**最早开始**的活动

能否举出反例？

最大不重叠区间数

正确的贪心算法：

不断循环重复选择活动：

每次在可选活动中，选择**最早结束**的活动

可以先对活动按照结束时间从小到大排序

这个贪心法
为什么正确

```
5 struct movie{int s,t;};
6 bool cmp(const movie& a,const movie& b){
7     return a.t<b.t;
8 }
9 movie d[N];
```

定义区间类型

定义区间顺序比较规则

定义包含N个区间的数组

```
15     sort(d,d+n,cmp);
16     x=-1; ans=0;
17     for(i=0;i<n;i++)
18         if(d[i].s>x) {
19             ans++;
20             x=d[i].t;
21         }
22     cout<<ans<<endl;
```

按区间右端点从小到大排序

初始化: x记录当前右端位置

按区间右端点从小到大查看

如第i个区间左端>x

多安排一个区间
x保持为右端位置

现场挑战

请在电脑上
完成"看最多电影"
对每一行写注释
限时10分钟

自编题挑战

仿照课堂例题，请自编一道编程题

要求以“**最多不重叠区间**”的算法为核心求解步骤。

鼓励加入各类算法元素，构成原问题的变种形式。

区间问题难在哪

两个区间的"先后"关系不太显然

谁先?
谁后?

可以按照左端点排序

也可以按照右端点排序

区间模型2

不重叠区间最少分组数

有n个区间分成若干组，求最少分组数量，
能保证每组内区间互相不重叠

教室安排

共有n门课要安排教室，每门课i都有一个开始时间si和结束时间ti。如果两门课不重叠（结束时间和开始时间也不能重叠）就可以安排在同一个教室。至少需要几间教室？
0<=n<=100，其他输入都小于10000

样例输入
4
1 2
5 7
1 4
3 6

样例输出
2

样例输入
3
1 3
3 4
1 4

样例输出
3

世界杯

世界杯小组赛如火如荼地开展，今天共有n场比赛，每场比赛i都有一个开始时间ti（以分钟为单位），如果两场比赛的开始时间相差大于等于200分钟，那么这两场球赛可以安排在一个足球场进行。否则，由于时间间隔太短容易引起骚乱，必须将两场球赛分开到不同球场。输入第一行为n，第二行为n个正整数分别代表每场比赛的开赛时间。输出至少需要几个足球场。

0<=n<=16，其他输入都小于10000

样例输入

5

30 10 20 200 210

样例输出

4

样例输入

2

30 30

样例输出

2

每门课：抽象成一个区间

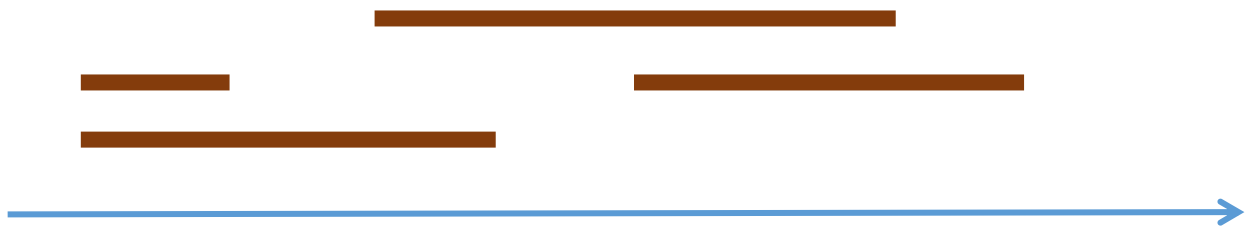
样例输入

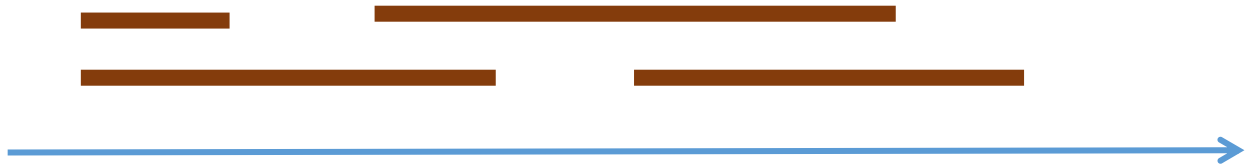
```
4
1 2
5 7
1 4
3 6
```

样例输出

```
2
```

每门课对应一个区间
区间左端点为 开课时间
区间右端点为 结束时间



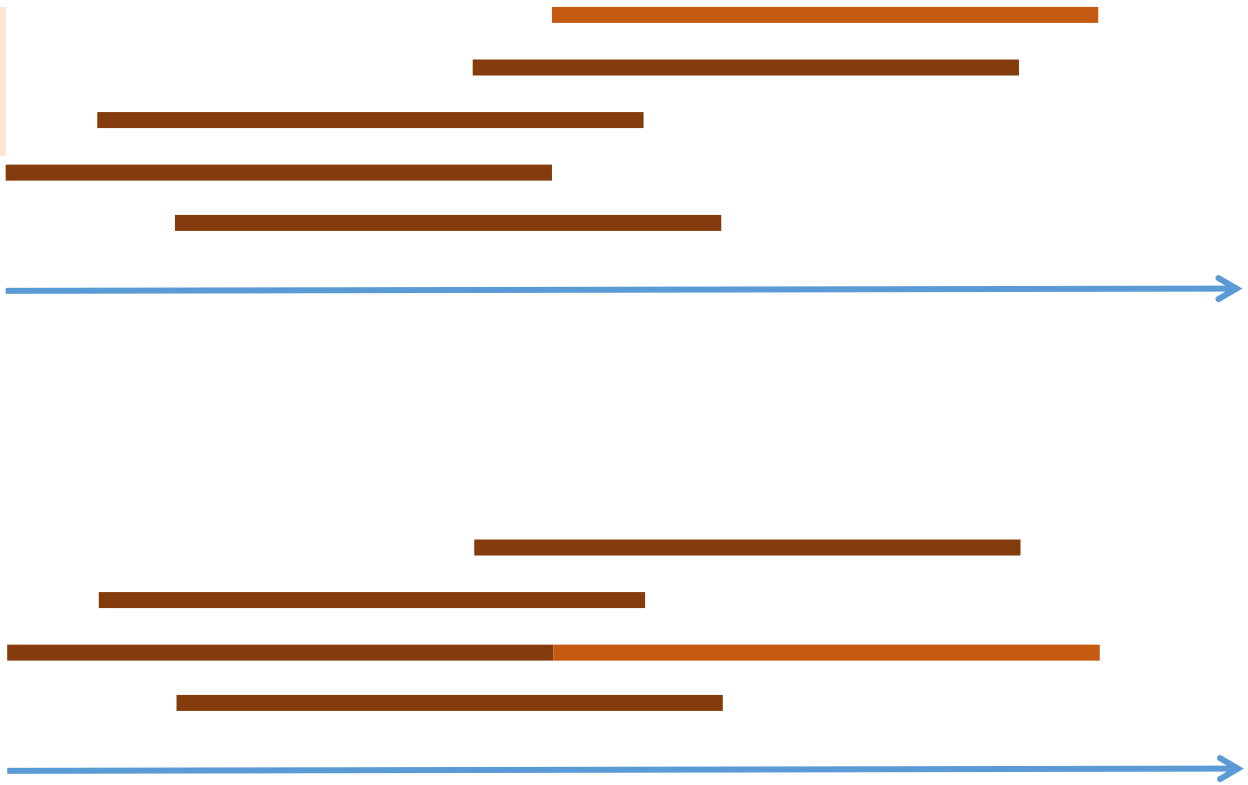


每场比赛：抽象成一个区间

样例输入
5
30 10 20 200 210

每场比赛对应一个区间
区间左端点为 开赛时间
区间右端点为 开赛时间+200分钟

样例输出
4



现场算法讨论

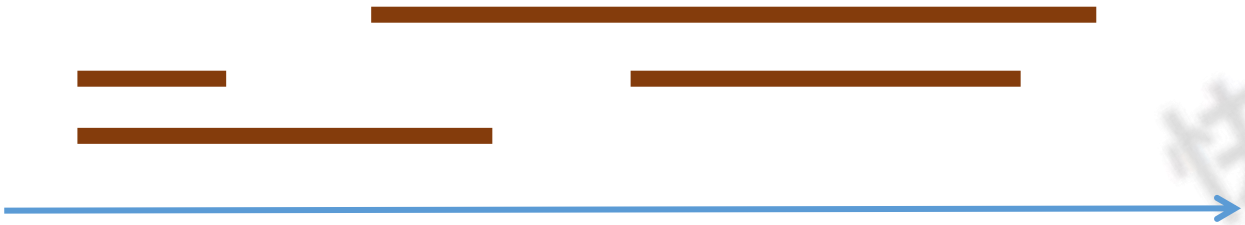
以下算法是否正确？
如果正确，请证明
如果不正确，请找出反例

教室安排 算法1

初始化教室数量为零

逐个开放教室直到所有课程都被安排完：

对于每个教室，**尽量排最多门课**



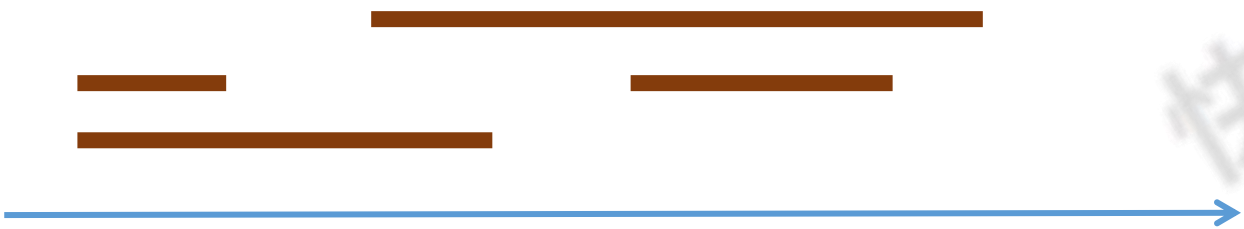
教室安排 算法2

初始化教室数量为零

按照**结束时间先后**
依次查看每堂课：

如果所有目前教室都无法上这堂课
就新增一间教室来上这堂课

否则就在可以使用的教室里选**最早空闲**的教室上这堂课



教室安排 算法3

初始化教室数量为零

按照**结束时间先后**
依次查看每堂课：

如果所有目前教室都无法上这堂课
就新增一间教室来上这堂课

否则就在可以使用的教室里选**最晚空闲**的教室上这堂课

教室安排 算法4

初始化教室数量为零

按照**开始时间先后**
依次查看每堂课：

如果所有目前教室都无法上这堂课
就新增一间教室来上这堂课

否则就在可以使用的教室里**选任意一间**的教室上这堂课

教室安排：算法3

```
5  struct ke{int s,t;};
6  bool cmp(const ke& a,const ke& b){
7      return a.t<b.t||a.t==b.t&&a.s<b.s;
8  }
9  ke d[N];
```

定义ke类型:课的开始和结束时间

定义两课间比较顺序

定义d数组包含N门课程

```
11     int n,i,j,k,x[N],cnt=0;
12     cin>>n;
13     for(i=0;i<n;i++) cin>>d[i].s>>d[i].t;
14     sort(d,d+n,cmp);
15     x[n]=-1;
16     for(i=0;i<n;i++){
17         for(j=0,k=n;j<cnt;j++)
18             if(d[i].s>x[j]&&x[j]>x[k]) k=j;
19         if(k<n) x[k]=d[i].t;
20         else x[cnt++]=d[i].t;
21     }
22     cout<<cnt<<endl;
```

x[i]为教室i里目前课程结束时间

cnt为教室总数

按照课程**结束时间从小到大**排序

设置临时教室

按照结束时间 安排每门课程

查看已有教室,**找出最晚空闲的教室k**

如果找到空教室,就在教室k上课i

否则增加新空教室, 上课i

教室安排：算法4

```
5  struct ke{int s,t;};
6  bool cmp(const ke& a,const ke& b){
7      return a.s<b.s||a.s==b.s&&a.t<b.t;
8  }
9  ke d[N];
```

定义ke类型:课的开始和结束时间

定义两课间比较顺序

定义d数组包含N门课程

```
11     int n,i,j,x[N],cnt=0;
12     cin>>n;
13     for(i=0;i<n;i++) cin>>d[i].s>>d[i].t;
14     sort(d,d+n,cmp);
15     for(i=0;i<n;i++){
16         for(j=0;j<cnt;j++)
17             if(d[i].s>x[j]) break;
18         if(j<cnt) x[j]=d[i].t;
19         else x[cnt++]=d[i].t;
20     }
21     cout<<cnt<<endl;
```

x[i]为教室i里目前课程结束时间

cnt为教室总数

按照课程开始时间从小到大排序

按照开始时间 安排每门课程

查看已有教室,找出任意空闲教室j

如果找到空教室,就在教室j上课i

否则增加新空教室, 上课i

世界杯:算法4

定义game类型:
开赛和结束时间

定义cmp比较规则:
开赛时间越早越靠前

定义一个数组
game d[N];
存放所有比赛信息

定义整数数组x[N]:
x[i]记录i号足球场里
目前比赛结束时间

数据结构

按照**开赛时间先后**循环查看每场比赛i:

循环查看目前所有已有的球场,寻找空闲球场

如果找到任意空闲球场j，就在球场j举办比赛i

否则新增一个球场举办比赛i

算法步骤

```cpp
#include<iostream>
#include<algorithm>
#define N 20
using namespace std;
struct game{int s,t;};
bool cmp(const game& a,const game& b){
    return a.s<b.s||a.s==b.s&&a.t<b.t;
}
game d[N];
int main(){
    int n,i,j,x[N],cnt=0;
    cin>>n;
    for(i=0;i<n;i++) {
        cin>>d[i].s;
        d[i].t=d[i].s+200;
    }
    sort(d,d+n,cmp);
    for(i=0;i<n;i++){
        for(j=0;j<cnt;j++)
            if(d[i].s>=x[j]) break;
        if(j<cnt) x[j]=d[i].t;
        else x[cnt++]=d[i].t;
    }
    cout<<cnt<<endl;
    return 0;
}
```

自编题

仿照课堂例题，请自编一道编程题

要求以“**不重叠区间最少分组数**”的算法为核心求解步骤。

鼓励加入各类算法元素，构成原问题的变种形式。

参考资料

周小博，**浅谈信息学竞赛中的区间问题**
2008年信息学国家集训队论文

快快编程作业

361

369

拓展题	362,370