


# CS153

A Mars rover, likely a Curiosity rover, is shown on a rocky, sandy surface under a clear sky. The rover is equipped with various scientific instruments and cameras. The entire image has a blue color overlay. A white triangle is positioned on the right side of the image, containing the text '算法建模'.

算法  
建模

---

课件下载链接:

<https://pan.baidu.com/s/1htbqTfA>

密码: imfv

作业网站:

<http://120.132.18.213:8080/thrall-web/main#home>

# 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， $n \leq 100$ ）

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 $s_i$ 开始，到时刻 $t_i$ 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， $n \leq 100$ )

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

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

输出样例

3



# 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



# 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

```
5 struct movie{int s,t};
```

定义区间类型

```
6 bool cmp(const movie& a,const movie& b){
```

```
7     return a.t<b.t;
```

定义区间顺序比较规则

```
8 }
```

```
9 movie d[N];
```

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

# 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。

# 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

# 区间模型1

## 综合练习