


# CS153

A Mars rover, likely a Curiosity rover, is shown on a rocky, sandy surface under a blue sky. The rover is equipped with various cameras, sensors, and a large antenna. Its shadow is cast on the ground to the left. The entire image has a blue color overlay.

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>

# 浮点数危机

```
1 #include<iostream>
2 using namespace std;
3 int main() {
4     cout<<(7.2==7.2)<<endl;
5     cout<<(2.2+5==7.2)<<endl;
6     cout<<(12.2-5==7.2)<<endl;
7     cout<<(2.2+5==12.2-5)<<endl;
8     cout<<(2.2+5>12.2-5)<<endl;
9     cout<<(2.2+5<12.2-5)<<endl;
10    return 0;
11 }
```

# 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

# 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

# 区间模型3

## 最少区间全覆盖

有 $n$ 个区间，选择最少的区间数，  
完全覆盖某给定线段 $[S, T]$

# 思考题：车位危机

你作为保密单位的保安负责停车位安排的工作。你们单位只有 $m$ 个访客车位，在你值班时间内共有 $n$ 辆车需要停车，你提前得知了第 $i$ 辆车的停车申请时间为时刻 $s_i$ 到 $t_i$ 。对于这些车辆，你有权利同意停车申请或者拒绝，请问最少会有几辆车停不了车？

## 贪心算法I：先到先得

不断循环重复选择活动：

每次在停车申请中选**最早开始停**的车，安排在**最晚空闲**的车位

## 贪心算法II：

不断循环重复选择活动：

每次在停车申请中选**结束最早**的车，安排在**最晚空闲**的车位

## 贪心算法III：

不断循环重复选择活动：

每次在停车申请中选**结束最早**的车，安排在**任意可以停**的车位

# 区间模型4

## 区间重叠最厚层数

有 $n$ 个区间，求最厚重叠的层数

# 修正带

小明写作文时，每次写错一句就马上会用修正带涂一次错误的地方。已知他错了 $n$ 次，第 $i$ 次错误是从第 $s_i$ 个字错到第 $t_i$ 个字。小明的错误罄竹难书，写着写着他发现有些地方的修正带变得越来越厚，小明想知道最厚的地方有几层修正带？

输入第一行为正整数 $n$ ， $n \leq 1000$ ，之后 $n$ 行每行两个正整数 $s_i$ 和 $t_i$ ，保证都不超过1000。输出最厚的地方有几层。

## 样例输入

4

1 3

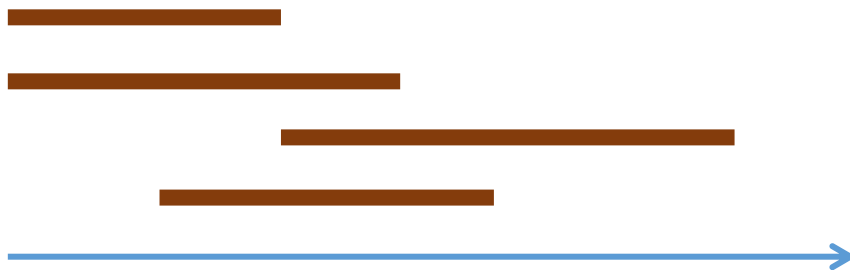
1 4

2 5

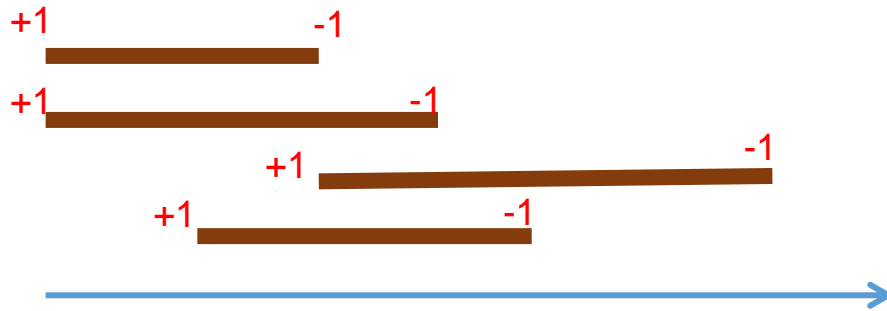
3 7

## 样例输出

3



# 扫描算法



## 算法：依次扫描

1. 将 $n$ 区间拆分成 $2n$ 个端点，每个区间左端点标记+1，右端点标记-1
2. 将 $2n$ 个端点按照从左到右依次排序（位置相同时，+1优先）
3. 从左到右扫描查看每个端点，并累计端点标记的数字（加1或减1）
4. 扫描过程中出现的最大累计值就是最厚层数



# 扫描算法

```
1 #include<iostream>
2 #include<algorithm>
3 #define N 2005
4 using namespace std;
5 struct point{int pos,tag;}; //tag 正负1 代表左右端点
6 bool cmp(const point&a,const point&b){
7     return a.pos<b.pos || a.pos==b.pos&&a.tag>b.tag;
8 }
9 point d[N];
```

易错点：需要 $N \geq 2n$

易错点：

位置相同，+1优先

其他问题可能要求不同

```
11 int n,cnt=0,ans=0;
12 cin>>n;
13 for(int i=0;i<n+n;i+=2){
14     cin>>d[i].pos>>d[i+1].pos;
15     d[i].tag=1;
16     d[i+1].tag=-1;
17 }
```

# 扫描算法

```
11 int n,cnt=0,ans=0;  
12 cin>>n;  
13 for(int i=0;i<n+n;i+=2){  
14     cin>>d[i].pos>>d[i+1].pos;  
15     d[i].tag=1;  
16     d[i+1].tag=-1;  
17 }
```

```
18 sort(d,d+n+n,cmp);  
19 for(int i=0;i<n+n;i++){  
20     cnt+=d[i].tag;  
21     ans=max(ans,cnt);  
22 }  
23 cout<<ans<<endl;
```

# 讨论题：模型2 和 模型4

## 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

## 区间模型4

区间重叠最厚层数

有 $n$ 个区间

求最厚重叠的层数

# 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 综合练习