


# 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 $[S, T]$

# 武林秘籍

有一本武林秘籍失传已久，江湖上存有 $n$ 本手抄本片段。每一个片段 $i$ ，抄写的是秘籍第 $s_i$ 页到第 $t_i$ 页。原秘籍共100页，请问至少要多少个片段能复原秘籍内容？如果无法复原输出sorry。  $s_i, t_i$ 都是正整数。  $n \leq 100$

样例输入

```
4
1 19
10 50
20 100
30 60
```

样例输出

```
2
```

样例输入

```
2
1 99
1 80
```

样例输出

```
sorry
```

# 探险

你要走一条探险之路，长L公里，从来没有人走完全程。共有n个向导，每个向导i只熟悉其中第si公里到第ti公里。请问需要几个向导才能覆盖整条探险之路。如果无法覆盖输出sorry。  
 $L \leq 100$ ， $n \leq 10$ ， $s_i, t_i$ 都是浮点数

样例输入

50 4  
0.0 30.5  
12.2 22.2  
25.5 45.5  
36.2 50.0

样例输出

3

样例输入

10 3  
1.0 30.5  
12.2 22.2  
25.5 45.5

样例输出

sorry

# 代课老师

这学期编程课共**16**节课，但编程老师要去拯救地球没空上第**5**到第**10**课，于是需要寻找同事来代课。共有**n**个同事，每个同事*i*可以上第**si**到**ti**节课。请问至少需要找几个代课老师？  
如果无法找到代课方案，输出**-1**。  $n \leq 10$

样例输入

4  
5 5  
8 16  
4 9  
1 3

样例输出

2

样例输入

3  
5 5  
6 6  
7 9

样例输出

-1

# 每个秘籍片段：抽象成一个区间

样例输入

5

1 90

10 50

20 100

30 60

每个片段对应一个区间  
区间左端点为 片段起始页数  
区间右端点为 片段结束页数

样例输出

2

整数  
覆盖  
问题

# 每个向导：抽象成一个区间

样例输入

50  
0.0 30.5  
12.2 22.2  
25.5 45.5  
36.2 50.0

样例输出

3

每个向导对应一个区间

区间左端点为 向导熟悉的最近公里数

区间右端点为 向导熟悉的最远公里数

浮点数  
覆盖  
问题



# 每个老师：抽象成一个区间

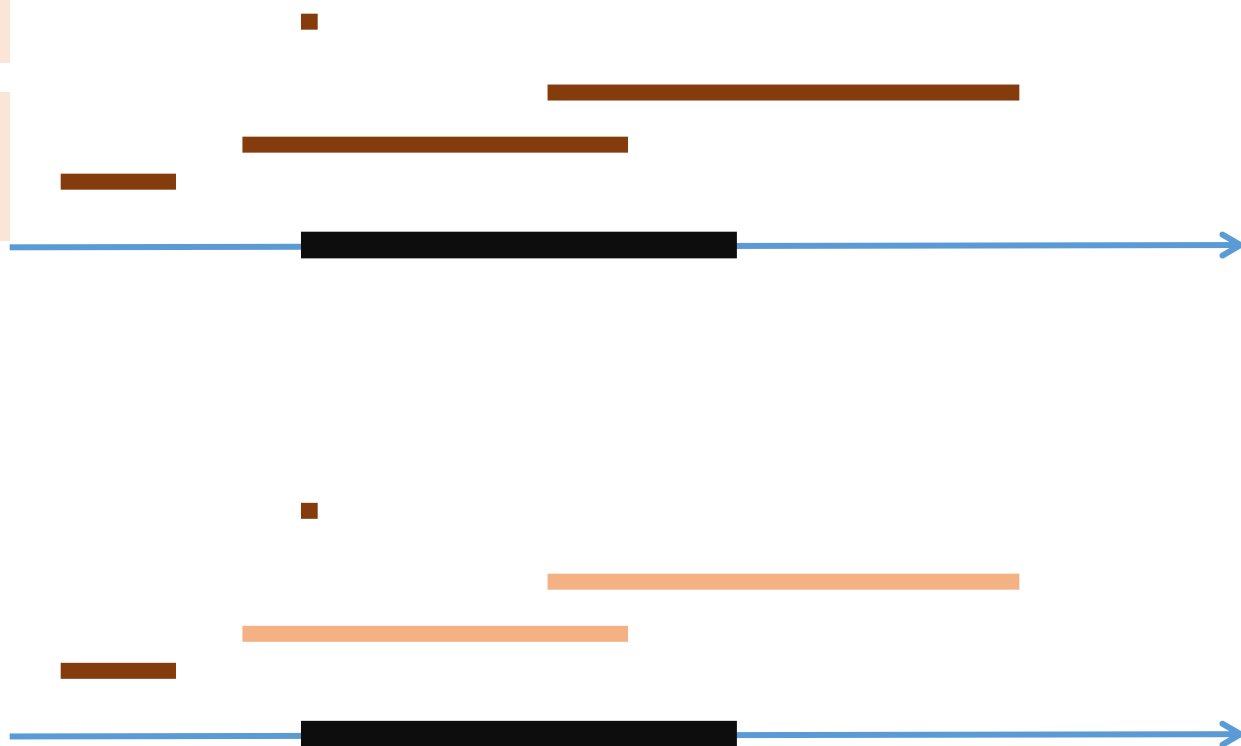
样例输入

4  
5 5  
8 16  
4 9  
1 3

样例输出

2

每个代课老师对应一个区间  
区间左端点为 最早上课编号  
区间右端点为 最晚上课编号



整数  
覆盖  
问题

# 代课老师 算法

贪心算法描述：

从需要首次代课的那一课开始，找能代那一课并且那课之后能坚持上课最久的老师。

反复这样地找老师，直到全部覆盖，或者发现无解



# 代课老师 算法

待覆盖区间  $[S,T]$  初始化为  $[5,10]$

将所有区间按**左端点从小到大排序**

循环处理每个区间，直到区间被全覆盖：

整数  
覆盖  
记得  
+1

每次选覆盖点  $S$  的区间中右端点最大的区间

将  $S$  更新为该区间的右端点位置+1



# 探险 算法

待覆盖区间  $[S, T]$  初始化为  $[0, L]$

将所有区间按**左端点从小到大排序**  
循环处理每个区间，直到区间被全覆盖：

每次选覆盖点  $S$  的区间中右端点最大的区间

将  $S$  更新为该区间的右端点位置

浮点数  
覆盖  
无需  
+1



# 武林秘籍：整数覆盖

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

定义片段类型:起始页和结束页

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

```
7     return a.s<b.s||a.s==b.s&&a.t<b.t;
```

```
8 }
```

定义两个片段比较顺序

```
9 piece d[N];
```

定义d数组包含N个片段

```
14 sort(d,d+n,cmp);
```

按照片段**起始页从小到大**排序

```
15 int S=1,T=100,cnt=0;
```

cnt为片段总数, [S,T]为待覆盖区间

```
16 for(i=0;i<n&&S<=T;i++){
```

当还有片段并且仍有未覆盖时循环

```
17     for(j=i;j<n&&d[j].s<=S;j++)
```

寻找**能覆盖S点**  
**并且右端最大的区间**

```
18         if(d[j].t>d[i].t) i=j;
```

```
19     if(d[i].s>S) break;
```

如无法覆盖, 就结束循环宣布无解

```
20     S=d[i].t+1; cnt++;
```

将S更新为该区间的右端点位置**+1**

```
21 }
```

```
22 if(S<=T) cout<<"sorry"<<endl;
```

```
23 else cout<<cnt<<endl;
```

# 探险：浮点数覆盖

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

定义向导类型:起始和结束公里数

定义两个向导比较顺序

定义d数组包含N个向导

cnt为向导总数，[S,T]为待覆盖区间

按照片段起始公里数从小到大排序

当还有向导并且仍有未覆盖时循环

寻找能覆盖S点  
并且右端最大的区间

如无法覆盖，就结束循环宣布无解

将S更新为该区间的右端点位置

```
15 sort(d,d+n,cmp);
16 for(i=0;i<n&&S<T;i++){
17     for(j=i;j<n&&d[j].s<=S;j++)
18         if(d[j].t>d[i].t) i=j;
19     if(d[i].s>S) break;
20     S=d[i].t; cnt++;
21 }
22 if(S<T) cout<<"sorry"<<endl;
23 else cout<<cnt<<endl;
```

# 代码细节讨论

---

请观察以上代码  
找出代码细节不同之处  
说明区别背后的原因

---

# 易错点

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

定义片段类型:double还是int?

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

```
7     return a.s<b.s||a.s==b.s&&a.t<b.t;
```

```
8 }
```

定义比较顺序:比左端还是右端?

```
9 piece d[N];
```

```
14 sort(d,d+n,cmp);
```

```
15 int S=1,T=100,cnt=0;
```

```
16 for(i=0;i<n&&S<=T;i++){
```

循环条件 $S \leq T$ 还是 $S < T$ ?

```
17     for(j=i;j<n&&d[j].s<=S;j++)
```

```
18         if(d[j].t>d[i].t) i=j;
```

```
19     if(d[i].s>S) break;
```

```
20     S=d[i].t+1; cnt++;
```

$S=x$ 还是 $S=x+1$ ?

```
21 }
```

```
22 if(S<=T) cout<<"sorry"<<endl;
```

```
23 else cout<<cnt<<endl;
```

无解条件 $S \leq T$ 还是 $S < T$ ?



# 自编题

仿照课堂例题，请自编一道编程题

要求以“最少区间全覆盖”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3

## 综合练习

# 作业七

## HOMEWORK 7



### 第一题 长城 （网站第 371 题）

万里长城，是伟大的军事防御工程，用来抵御怪兽“饕餮”的入侵。长城总长为  $L$  米，一端标记为 0，另一端标记为  $L$ ，其中有  $m$  个防御塔。每个防御塔  $i$  的位置标记为  $a_i$ ，防御半径为  $r_i$ ，也就是能防御  $[a_i - r_i, a_i + r_i]$  的范围。作为军师，你要覆盖长城全程的防御。请问你需要至少部署几个防御塔？

输入第一行为正整数  $L$  和  $m$ ，之后  $n$  行每行两个正整数  $a_i, r_i$ 。输入数字都不超过 10000。输出一个正整数，表示需要至少部署几个防御塔。如果无法完成全程覆盖，输出 Impossible

#### 输入样例：

10 3

2 4

5 4

9 3

#### 输出样例：

2

#### 输入样例：

100 2

0 50

76 25

#### 输出样例：

Impossible

# 作业七

## HOMEWORK 7



### 第二题 消防喷淋（网站第 372 题）

有一间室内游乐场的房顶需要安装消防喷淋系统，游乐场长  $L$  宽  $W$ 。房顶的喷头可以安装在中心线上（平行于长边，总长也是  $L$ ）不同位置处有  $n$  个点状的喷水装置可以安装。每个喷水装置  $i$  喷水的效果是让以它为中心半径为  $R_i$  的圆都被喷到水。请在给出的喷水装置中选择尽量少的喷水装置，把整个游乐场覆盖。

输入第一行有三个整数  $n, L, W$ 。随后的  $n$  行，都有两个整数  $x_i$  和  $r_i$ ,  $x_i$  表示第  $i$  个喷水装置到中心线一个端点的距离， $r_i$  表示该喷水装置能覆盖的圆的半径。 $n \leq 100$ ，其他输入数据不超过 1000。

输出一个正整数，表示共需要多少个喷水装置。 如果不存在一种能够把整个游乐场覆盖的方案，请输出-1。

样例输入

2 8 6

1 1

4 5

样例输出

1

样例输入

2 10 6

4 5

6 5

样例输出

2

# 作业七

## HOMEWORK 7

---



### 第三题 自编题

仿照课堂例题，请自编一道编程题，要求以“最少区间全覆盖”的算法为核心求解步骤。本作业题的提交方式为：一个 word 文档发到课程微信群

需要提供：

1. 题目描述
2. 输入数据的范围
3. 输入格式
4. 输出格式
5. 输入样例至少 1 组
6. 输出样例至少 1 组
7. 标准答案程序 1 份

优秀自编题会在整理后发布到网站向所有人开放。

# 作业七

## HOMEWORK 7

---



# 作业七

## HOMEWORK 7



### 第二题 Hanoi 双塔 （网站第 182 题） NOIP2007 普及组

给定 A、B、C 三根足够长的细柱，在 A 柱上放有  $2n$  个中间有孔的圆盘，共有  $n$  个不同的尺寸，每个尺寸都有两个相同的圆盘，注意这两个圆盘是不加区分的（下图为  $n=3$  的情形）。现要将这些圆盘移到 C 柱上，在移动过程中可放在 B 柱上暂存。要求：

（1）每次只能移动一个圆盘；

（2）A、B、C 三根细柱上的圆盘都要保持上小下大的顺序；

任务：设  $A_n$  为  $2n$  个圆盘完成上述任务所需的最少移动次数，对于输入的  $n$ ，输出  $A_n$ 。

#### 输入格式：

一个正整数  $n$ ，表示在 A 柱上放有  $2n$  个圆盘

#### 输出格式：

一个正整数，为完成上述任务所需的最少移动次数

#### 输入样例#1：

2

#### 输出样例#1：

6

#### 说明

对于 50% 的数据， $1 \leq n \leq 25$

对于 100% 的数据， $1 \leq n \leq 200$



# 作业七

## HOMEWORK 7



### 第三题 闯关（网站第 357 题）

有一个闯关游戏，在一条直线上，从左到右共有  $2n$  个游戏道具： $n$  个炸弹和  $n$  个补血袋，以某个顺序依次排列。你需要从最左边闯关到最右边，并且体力始终不为负数。每次碰到炸弹你的体力减一，每次遇到补血袋体力加一。初始时你的体力为零，请问道具有多少种排列方式能使你闯关成功？

输入一个正整数  $n$ ， $n \leq 100$ 。输出一个正整数代表能闯关成功的道具排列数量。

输入样例：

3

输出样例：

5

说明： $n$  为 3 时，共 5 种排列：血血血炸炸炸炸，血血炸血炸炸，血血炸炸血炸，血炸血炸血炸，血炸血血炸炸。