



中國人民大學

RENMIN UNIVERSITY OF CHINA

信息学院

SCHOOL OF INFORMATION

程序设计1荣誉课程

4. 算法1——穷举法

授课教师：游伟 副教授

授课时间：周一08:00 – 09:30, 周四16:00 – 17:30 (明德新闻楼0201)

上机时间：周四18:00 – 21:00 (理工配楼二层5号机房)

课程主页：<https://rucsesec.github.io/programming>

引子：21级图灵班选拔卷第一、二题

一、设有限集合 $A = \{a_1, a_2, \dots, a_n\}$, $\sum_{i=1}^n a_i$ 叫做集合 **A** 的和, 记作 S_A .

- 若集合 $P = \{x | x = 2n - 1, n \text{ 是正整数}, n \leq 4\}$, P 的含有3个元素的全体子集分别为 P_1, P_2, \dots, P_k , 求 $\sum_{i=1}^k S_{P_i}$.
- 若集合 $Q = \{x | x = 2n - 1, n \text{ 是正整数}, n \leq 20\}$, 定义 Q^{20} 为 Q 的所有非空子集中满足集合的和不大于20的子集构成的集合 (即 $Q^{20} = \{X | X \subseteq Q \text{ and } X \neq \emptyset \text{ and } S_X \leq 20\}$). 例如: $\{1, 5, 9\} \in Q^{20}$, 而 $\{1, 5, 7, 9\} \notin Q^{20}$. 求 $|Q^{20}|$ (即 Q^{20} 的元素个数).

二、Rachel is a deceitful child. She has a funny pattern of telling lies. She tells lies six days a week and will always tell the truth on a certain day of a week.

Rachel made the following statements on three consecutive days:

- Day 1: "I lie on Monday and Tuesday."
- Day 2: "Today is either Thursday, Saturday, or Sunday."
- Day 3: "I lie on Wednesday and Friday."

Which day does Rachel tell the truth?

目录

1. 用穷举法解决逻辑问题
2. 用穷举法解决数值问题

穷举法

- 基本思想：将问题的所有可能的答案一一列举，然后根据条件判断此答案是否合适，保留合适的，丢弃不合适的
- 使用枚举算法解题的基本思路：
 - 确定枚举对象、枚举范围和判定条件
 - 逐一枚举可能的解，验证每个解是否是问题的解
- 枚举算法一般按照如下3个步骤进行
 - 题解的可能范围，不能遗漏任何一个真正解，也要避免有重复
 - 判断是否是真正解的方法
 - 使可能解的范围降至最小，以便提高解决问题的效率

4.1 用穷举法解决逻辑问题

■ 示例1：谁做的好事

人大附中有四位同学中的一位做了好事，不留名，表扬信来了之后，校长问这四位是谁做的好事。

A说：不是我。

B说：是C。

C说：是D。

D说：他胡说。

已知三个人说的是真话，一个人说的是假话。现在要根据这些信息，找出做了好事的人

示例1：谁做的好事

■ 将四个人说的四句话写成关系表达式

- 在声明变量时，我们让 thisman 表示要寻找的做了好事的人，定义它是字符变量，其可能的取值范围为{'A', 'B', 'C', 'D'}，分别代表四个人
- 让 “==” 的含义为“是”，让 “!=” 的含义为“不是”

`char thisman = “”; // 定义字符变量并初始化为空`

说话人	说的话	写成关系表达式
A	“不是我”	thisman!=‘A’
B	“是C”	thisman==‘C’
C	“是D”	thisman==‘D’
D	“他胡说”	thisman!=‘D’

示例1：谁做的好事

■ 穷举所有可能的状态

- A、B、C、D四个人，只有一位是做好事者。令做好事者为1，未做好事者为0，可以有如下4种状态（情况）
- 第一种状态是假定A是做好事者，第二种状态是假定B是做好事者，...
- 所谓穷举是按照这四种假定**逐一去测试**四个人的话有几句是真话，如果不满足三句为真，就否定掉这一假定，换下一个状态再试

状态	A	B	C	D	赋值表达式
1	1	0	0	0	thisman='A'
2	0	1	0	0	thisman='B'
3	0	0	1	0	thisman='C'
4	0	0	0	1	thisman='D'

示例1：谁做的好事

- 情况1：假定让thisman='A'代入四句话中

说话人	说的话	关系表达式	值
A	thisman!='A';	'A'!='A'	0
B	thisman=='C';	'A'=='C'	0
C	thisman=='D';	'A'=='D'	0
D	thisman!='D';	'A'!='D'	1

四个关系表达式的值的和为1，不满足3句话为真，**假设不成立**，因此显然不是'A'做的好事。

示例1：谁做的好事

- 情况2：假定让thisman='B'代入四句话中

说话人	说的话	关系表达式	值
A	thisman!='A';	'A'!='A'	1
B	thisman=='C';	'A'=='C'	0
C	thisman=='D';	'A'=='D'	0
D	thisman!='D';	'A'!='D'	1

四个关系表达式的值的和为2，不满足3句话为真，**假设不成立**，因此显然不是'B'做的好事。

示例1：谁做的好事

- 情况3：假定让thisman='C'代入四句话中

说话人	说的话	关系表达式	值
A	thisman!='A';	'A'!='A'	1
B	thisman=='C';	'A'=='C'	1
C	thisman=='D';	'A'=='D'	0
D	thisman!='D';	'A'!='D'	1

四个关系表达式的值的和为3，**假设成立**，因此就是'C'做的好事。

示例1：谁做的好事

■ 用流程图表示

第一块
循环结构

for (k=1; k<=4; k=k+1)

被试者 **thisman = 64+k;**
sum = (被试者 thisman != 'A')+
(被试者 thisman == 'C')+
(被试者 thisman == 'D')+
(被试者 thisman != 'D');

真

sum == 3

假

输出该被试者;
有解标志 **g=1;**

第二块
分支结构

真

g != 1

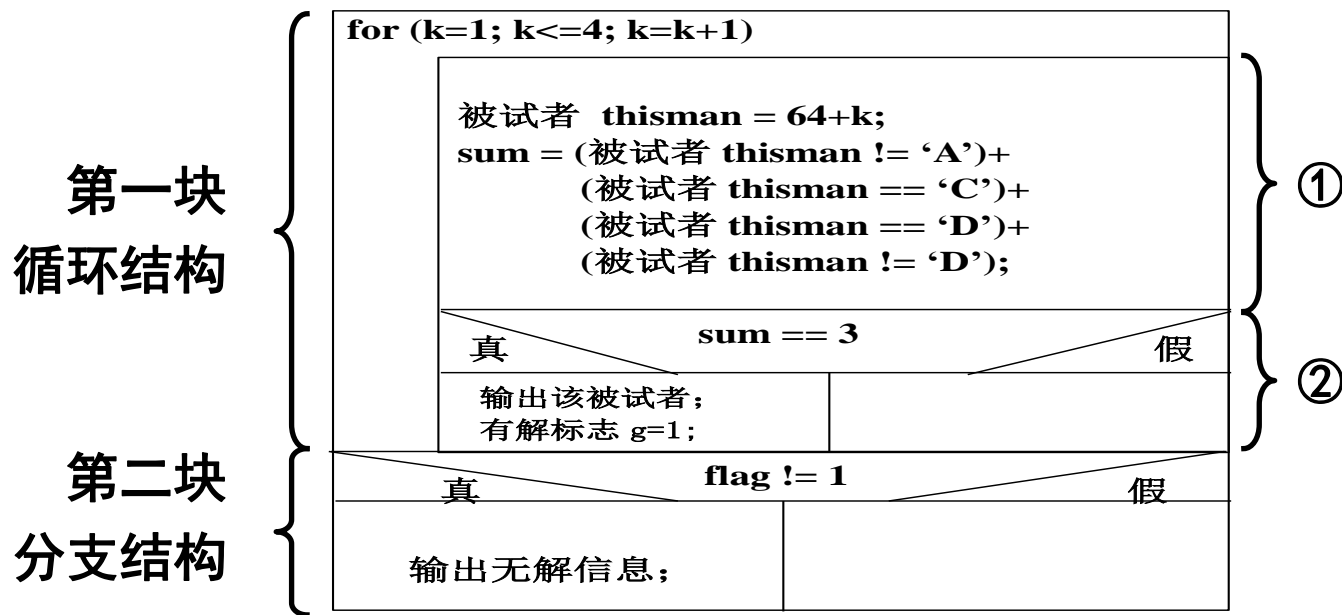
假

输出无解信息;

示例1：谁做的好事

■ 用流程图表示

- 第一块是循环结构，功能是产生被试对象，依次为A、B、C、D。并测试四句话有多少句为真，如有3句为真，则可确定做好事者，同时置有解标志为1（①中含两条赋值语句，②中含一条分支语句）
- 第二块为分支结构，功能是判断是否无解，如为真，则输出无解信息



示例1：谁做的好事

■ 程序源代码

```
1. #include <stdio.h>

2. int main(int argc, char **argv) {
3.     int k=0, sum=0, flag=0;
4.     char thisman = ' ';

5.     for (k=1; k<=4; k++) {
6.         thisman = 64+k; //A的ASCII码是65
7.         sum = (thisman!='A') + (thisman=='C') + (thisman=='D') + (thisman!='D');
8.         if (sum == 3) {
9.             printf("%c did the good thing.\n", thisman);
10.            flag = 1;
11.        }
12.    }

13.    if (flag != 1) printf("Can't find who did the good thing.\n");
14.}
```

4.1 用穷举法解决逻辑问题

■ 示例2：破案

某地刑侦大队对涉及六个嫌疑人的一桩疑案进行分析：

- A、B 至少有一人作案；
- A、E、F 三人中至少有两人参与作案；
- A、D 不可能是同案犯；
- B、C 或同时作案，或与本案无关；
- C、D 中有且仅有一人作案；
- 如果 D 没有参与作案，则 E 也不可能参与作案

试编一程序，将作案人找出来

示例2：破案

- 将案情的每一条写成逻辑表达式
 - CC1: A和B至少有一人作案
 - CC2: A和D不可能是同案犯
 - CC3: A、E、F 中至少有两人涉嫌作案
 - CC4: B和C或同时作案，或都与本案无关
 - CC5: C、D中有且仅有一人作案
 - CC6: 如果D没有参与作案，则E也不可能参与作案
- 将案情分析的6条归纳成一个破案综合判断条件CC

CC = CC1 && CC2 && CC3 && CC4 && CC5 && CC6

示例2：破案

- CC1: A和B至少有一人作案

- 令 A 变量表示 A 作案, B 变量表示 B 作案

- 显然这是或的关系, 有 $CC1 = (A \parallel B)$

A	B	CC1
0	0	0
1	0	1
0	1	1
1	1	1

示例2：破案

■ CC2: A和D不可能是同案犯

■ $CC2 = \neg (A \ \&\& \ D)$

A	D	A&&D	CC2
1	0	0	1
1	1	1	0
0	0	0	1
0	1	0	1

示例2：破案

■ CC3: A、E、F 中至少有两人涉嫌作案

■ 有3种不同的情况

- 情况1: A 和 E 作案, $(A \ \&\& \ E)$
- 情况2: A 和 F 作案, $(A \ \&\& \ F)$
- 情况3: E和 E 作案, $(E \ \&\& \ F)$

■ 这三种可能性是 或 的关系, $CC3 = (A \ \&\& \ E) \parallel (A \ \&\& \ F) \parallel (E \ \&\& \ F)$

■ CC4: B和C或同时作案, 或都与本案无关

■ 有2种不同的情况

- 情况1: 同时作案, $(B \ \&\& \ C)$
- 情况2: 都与本案无关, $(!B \ \&\& \ !C)$

■ 这两种可能性是 或 的关系, $CC4 = (B \ \&\& \ C) \parallel (!B \ \&\& \ !C)$

■ CC5: C、D中有且仅有一人作案

■ $CC5 = (C \ \&\& \ !D) \parallel (!C \ \&\& \ D)$

示例2：破案

■ CC6: 如果D没有参与作案，则E也不可能参与作案

■ 分析这一条比较麻烦一些，可以列出真值表再归纳

■ $CC6 = D \parallel !E$ （实际上是蕴含关系： $!D \rightarrow !E$ 等价于 $D \rightarrow E$ 等价于 $D \parallel !E$ ）

D	E	!E	CC6	含 义	
1	1	0	1	D作案，E也作案	可能
1	0	1	1	D作案，E不作案	可能
0	0	1	1	D不作案，E也不可能作案	可能
0	1	0	0	D不作案，E却作案	不可能

示例2：破案

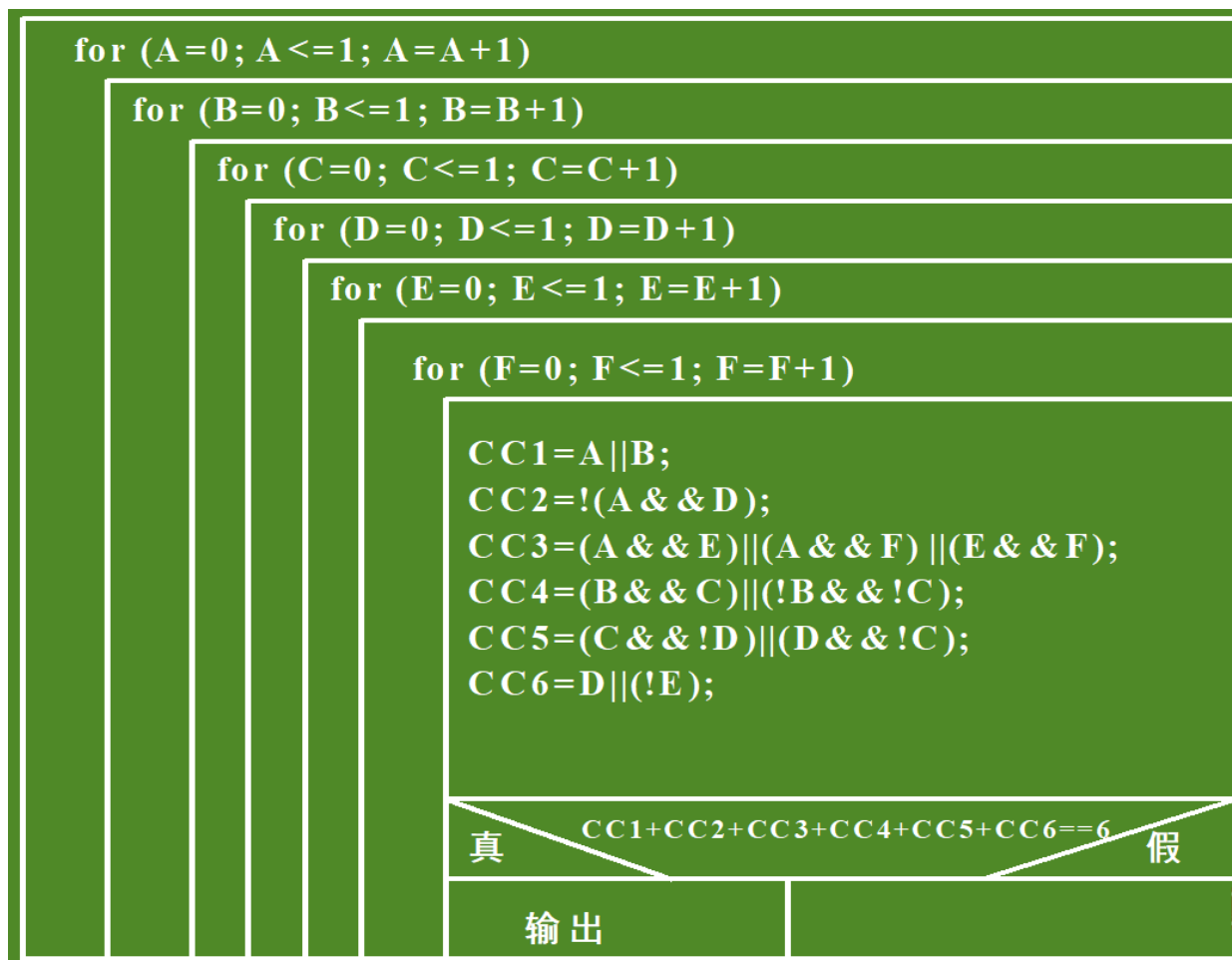
■ 采取穷举方法，穷举什么呢？穷举组合

- 6个人每个人都有作案或不作案两种可能，故有 2^6 种组合，从这些组合中挑出符合6条分析的作案者
- 定义 6 个整数变量，分别表示 6 个人A, B, C, D, E, F
- 枚举每个人的可能性：让 0 表示不是罪犯，让 1 表示是罪犯

A	B	C	D	E	F
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0
.....
1	1	1	1	1	1

示例2：破案

■ 用流程图表示



示例2：破案

■ 程序源代码

```
1. #include <stdio.h>
2. int main(int argc, char **argv) {
3.     int cc1, cc2, cc3, cc4, cc5, cc6;
4.     int A, B, C, D, E, F;
5.     for (A=0; A<=1; A++)
6.         for (A=0; A<=1; A++)
7.             for (A=0; A<=1; A++)
8.                 for (A=0; A<=1; A++)
9.                     for (A=0; A<=1; A++)
10.                        for (A=0; A<=1; A++)
11.                            {
12.                                cc1 = A||B; cc2 = !(A&&D); cc3 = (A&&E)|| (A&&F)|| (E&&F);
13.                                cc4 = (B&&C)|| (!B&&!C); cc5 = (C&&!D)|| (!C&&D); cc6 = D|| !E;
14.                                if (cc1+cc2+cc3+cc4+cc5+cc6 == 6) {
15.                                    printf("A:%d B:%d C:%d D:%d E:%d F:%d\n", A, B, C, D, E, F);
16.                                    break;
17.                                }
18.}
```

4.2 用穷举法解决数值问题

■ 示例3：百钱买百鸡

公元前五世纪，我国古代数学家张丘建在《算经》一书中提出了“百鸡问题”：鸡翁一值钱五，鸡母一值钱三，鸡雏三值钱一。百钱买百鸡，问鸡翁、母、雏各几何？

■ 解题的基本思路：

- 用变量 n_cocks , n_hens , n_chicks 分别表示鸡翁、母鸡、雏鸡的数量，则：
- $n_cocks + n_hens + n_chicks = 100$ (百鸡)
- $5 * n_cocks + 3 * n_hens + n_chicks / 3 = 100$ (百钱)

4.2 用穷举法解决数值问题

■ 示例4：搬砖问题

36块砖，36人搬；男搬4，女搬3，两个小孩抬一砖。要求一次全搬完，问男、女、小孩各若干？

■ 解题的基本思路：

- $4 * \text{men} + 3 * \text{women} + \text{children} / 2 = 36$
- $\text{men} + \text{women} + \text{children} = 36$
- men取值范围：0~9
- women取值范围：0~12
- children取值范围：0~36

思考

用穷举法求解选拔题一、二