



# 离散数学

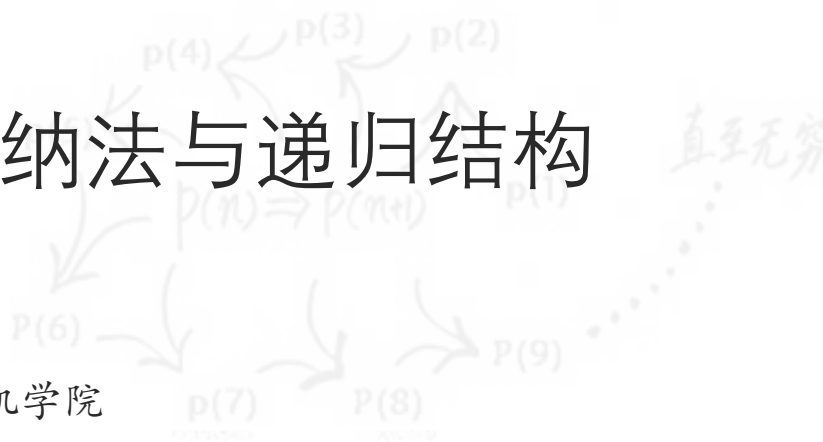
## Discrete Mathematics

### 第十讲：数学归纳法与递归结构

吴楠

南京大学计算机学院

2025年3月28日





# 前情提要



- 整数的性质
- 整数的基本运算
- 素 数
- Euler 函数与 Euler 定理





# 本讲主要内容



- 数学归纳法
- 强数学归纳法
- 递归的定义
- 函数的递归结构与结构归纳法





# 什么是数学归纳法？



- **数学归纳法** (mathematical induction, **MI**) 是利用归纳原理进行定理证明的一种逻辑方法
- 数学归纳法的理论基础源自两个公理系统：一是在自然数的公理化系统中的**无穷公理**，二是存在于**ZFC**系统中的**选择公理**（等价于**良序公理**）
- 数学归纳法常用于证明有关正整数或自然数的命题



# 数学归纳法的逻辑基础



- 数学归纳法是建立在公理系统中的一个逻辑**演绎**推理过程：

- I. 基于皮亚诺自然数公理系统的（第一）数学归纳法

$$P(1), \forall k(P(k) \rightarrow P(k+1)) \Rightarrow \forall xP(x)$$

- II. (部分) 基于选择公理的强数学归纳法（完全归纳法）

$$P(1), \forall k(\forall y < k, P(y) \rightarrow P(k)) \Rightarrow \forall xP(x)$$



# 数学归纳法的逻辑基础\* (续)



- 在二阶逻辑中，数学归纳法以“归纳公理”（注意不是Peano算术中的归纳公理）的形式出现：对谓词 $P$ ,

$$(\text{Ax. Ind}) \quad \forall P \left( P(0) \wedge \forall k (P(k) \rightarrow P(k+1)) \rightarrow \forall x (P(x)) \right)$$

- 一阶集合论(ZFC)不允许遍历谓词，但可通过遍历集合的方式绕过上述限制，在集合论中描述MI：对集合 $A$ ,

$$\forall A (0 \in A \wedge (\forall k \in \mathbb{N}) (k \in A \rightarrow (k+1) \in A) \rightarrow \mathbb{N} \subseteq A)$$



# 数学归纳法的基本证明框架



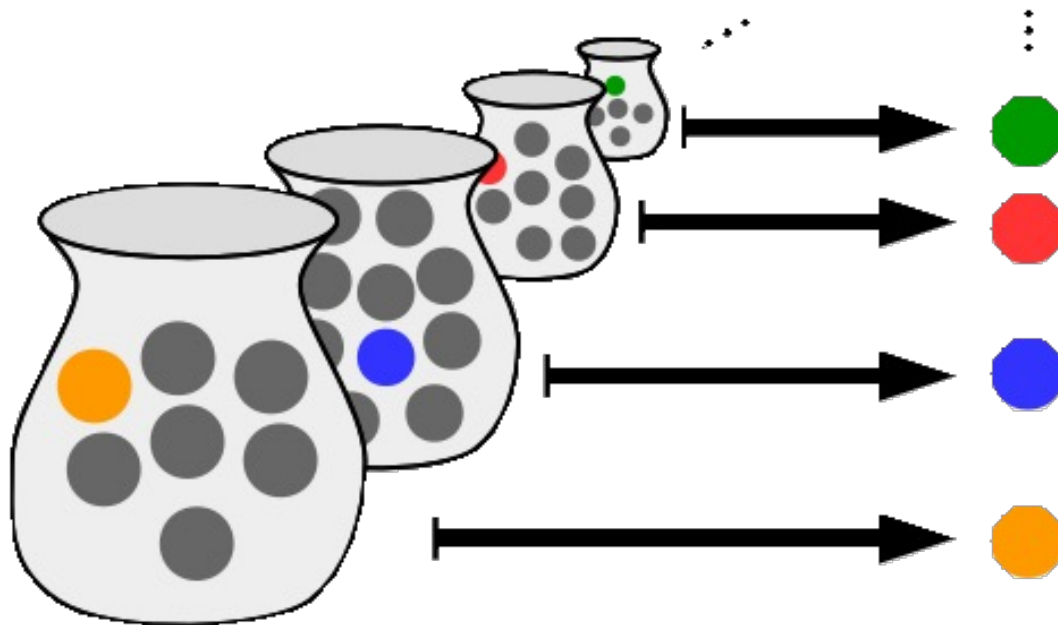
- 理论依据:  $P(1), \forall k(P(k) \rightarrow P(k+1)) \Rightarrow \forall xP(x)$
- 证明目标:
  - $\forall nP(n)$ , 其中 $n$ 的论域为正整数集或自然数集
- 证明框架:
  - 奠基 (Basis) : 证明 $P(1)$ 为真;
  - 归纳假设 (Inductive hypothesis, I.H.) : 假设对任意正整数 $k$ ,  $P(k)$ 为真;
  - 归纳步骤 (Inductive step) : 证明 $P(k) \Rightarrow P(k+1)$ , 即证明  $\forall k(P(k) \rightarrow P(k+1))$ ;
  - 由数学归纳法, 命题对任意论域中的元素均成立



# 选择公理\*



- ZFC (Zermelo-Fraenkel set theory with Axiom of Choice) 中的公理9即为选择公理



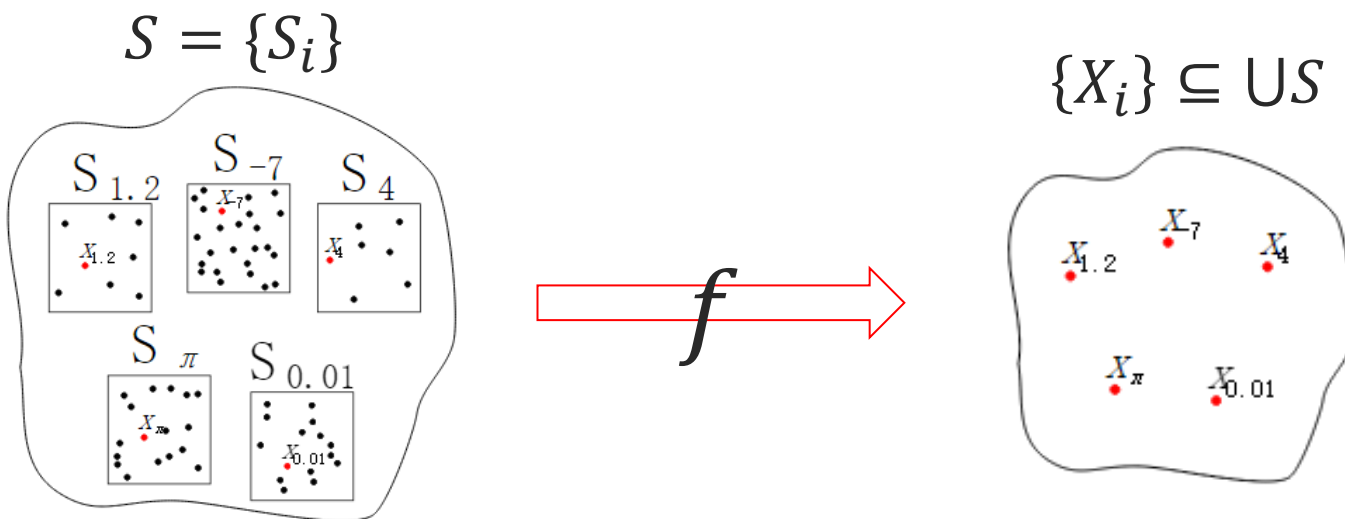


# 选择公理\*



- ZFC (Zermelo-Fraenkel set theory with Axiom of Choice) 中的公理9即为选择公理

**Ax. 9 (AC):**  $\forall S \left( \emptyset \notin S \rightarrow (\exists f: S \rightarrow \cup S) \left( (\forall S_i \in S) (f(S_i) \in S_i) \right) \right)$





# 选择公理\*



- ZFC (Zermelo-Fraenkel set theory with Axiom of Choice) 中的公理9即为选择公理
- $\mathbf{ZF} + \mathbf{AC}$  (即  $\mathbf{ZF}$  与  $\mathbf{AC}$  逻辑相容:  $\neg(\mathbf{ZF} \rightarrow \neg\mathbf{AC})$ , K. Gödel, 1939) 即, **ZFC逻辑相容**
- $\mathbf{ZF} + \neg\mathbf{AC}$  (若  $\mathbf{ZF}$  是逻辑相容的, 则:  $\mathbf{ZF} \rightarrow \neg\mathbf{AC}$ , P. Cohen, 1963) 即, **ZF¬C逻辑相容**
- 因此: **选择公理AC独立于ZF系统**



# 数学归纳法有效性的逻辑证明



## ■ 超限归纳法与ZFC 集合系统

- ZFC的系统公理中包含良序公理（等价于选择公理）：

对任一集合 $X$ ，总存在一个表序的二元关系 $R$ 。即：

**Ax. 9 (AC<sup>\*</sup>):  $\forall X \exists R (R \text{ well-orders } X)$**

- 正整数集合上的良序公理：正整数集合上的任一非空子集皆含有一个最小元（即最小的整数）  
$$\forall i \in I, A_i \neq \emptyset \Rightarrow \prod_{i \in I} A_i \neq \emptyset$$
- 良序公理普遍被认为是良序集合存在的逻辑基础，因此若不承认良序公理一般应避免使用（超限）归纳法



# 数学归纳法有效性的逻辑证明（续）



- 现用归谬法证明ZF+AC中数学归纳法依然有效：
- **证明：**假设在数学归纳法框架下 $\forall xP(x)$ 不成立，即 $\exists n(\neg P(n))$ 。令集合 $S = \{n \in \mathbb{Z}^+ | \neg P(n)\}$ ， $S$ 是非空子集。根据ZFC中的AC，集合 $S$ 存在最小元，记为 $m$ ； $\because P(m)$ 不成立，由数学归纳法的逻辑框架 $m \neq 1$ ，但由于 $(m-1) \notin S$ ，故 $P(m-1)$ 成立。根据数学归纳法逻辑框架中的归纳过程， $P(m)$ 成立，矛盾！由归谬法， $\forall xP(x)$ 成立。  $\square$



# 用数学归纳法证明命题



■ 例：令  $H_k = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}$ ,  $k \in \mathbb{Z}^+$ ,

证明：  $H_{2^n} \geq 1 + \frac{n}{2}$ ,  $n \in \mathbb{Z}^+$ .

$P(1)$  成立

■ 证明：奠基：对  $n = 1$ ,  $H_{2^1} = 1 + \frac{1}{2} \geq 1 + \frac{1}{2}$  成立；

I.H.: 假设对任意正整数  $k$ , 有  $H_{2^k} \geq 1 + \frac{k}{2}$ , 则：

假设  $P(k)$  成立

$P(k) \Rightarrow P(k+1)$

归纳步骤：  $H_{2^{k+1}} = H_{2^k} + \frac{1}{2^{k+1}} + \frac{1}{2^{k+2}} + \cdots + \frac{1}{2^{2^{k+1}}} \geq$

$\left(1 + \frac{k}{2}\right) + 2^k \cdot \frac{1}{2^{k+1}} = 1 + \frac{k+1}{2}$ . 由数学归纳法，命题得证.  $\square$



# 数学归纳法证明时的常见错误



- **例1:** 任意 $n$ 个人必定都在同一天出生.
- **错误证明:**
  - Basis: 当 $n = 1$ 时, 只有一个人, 命题显然成立;
  - I.H.: 假设任意  $k$  ( $k > 1$ ) 个人, 他们全部在同一天出生;
  - I.S.: 当有 $k + 1$ 个人时 (编号为 $1, 2, \dots, k, k + 1$ ), 根据归纳假设, 第1人至第 $k$ 人 (共 $k$ 个人) 一定在同一天出生; 第2至第 $k + 1$ 人 (共 $k$ 个人) 也一定在同一天出生。因此, 这 $k + 1$ 人全部在同一天出生。根据数学归纳法, 命题成立.  $\square$
  - 归纳基础错误: Basis应为 $P(2)$ , 但 $P(2)$ 非永真



# 数学归纳法证明时的常见错误 (续)



- **例2:** 证明  $\sum_{i=1}^n (2i - 1) = n^2$ .
- **错误证明:**
  - Basis: 当  $n = 1$  时,  $\sum_{i=1}^1 (2i - 1) = 1^2$  命题成立;
  - I.H.: 假设当  $n = k$  时  $\sum_{i=1}^k (2i - 1) = k^2$  成立, 则:
  - I.S.: 据等差求和公式,  $\sum_{i=1}^{k+1} (2i - 1) = 1 + 3 + 5 + \dots + 2(k + 1) - 1 = \frac{[1+2(k+1)-1](k+1)}{2} = (k + 1)^2$ 。根据数学归纳法, 命题成立.  $\square$
  - 归纳过程错误: 未证明  $P(k) \rightarrow P(k + 1)$



# 强数学归纳法的基本证明框架



- 强数学归纳法依赖的公理体系：**ZFC** 集合系统
- 理论依据： $P(1), \forall k(\forall y < k, P(y) \rightarrow P(k)) \Rightarrow \forall x P(x)$
- 证明目标： $\forall n P(n)$ ，其中 $n$ 的论域为正整数集
- 证明框架：
  - 奠基：证明 $P(1)$ 为真；
  - 归纳假设与归纳步骤：给定任意正整数 $k$ ，证明当 $P(1), P(2), \dots, P(k)$ 为真时 $P(k+1)$ 也为真，即证明 $\forall k(P(1) \wedge P(2) \wedge \dots \wedge P(k) \rightarrow P(k+1))$ ；
  - 由强数学归纳法，命题对任意论域中的元素均成立



# 强数学归纳法的简化形式



- 设 $P(n)$ 是与正整数 $n$ 有关的陈述， $x$ 和 $y$ 是两个给定的整数，且 $y \leq x$ 。若能够证明以下陈述：
  - (1)  $P(y), P(y+1), \dots, P(x)$ 皆为真
  - (2) 对任意 $k \geq x$ ,  $P(y) \wedge P(y+1) \wedge \dots \wedge P(k) \rightarrow P(k+1)$
- 则以下陈述成立： $\forall n \geq y, P(n)$
- 显然，强数学归纳法的成立依赖于整数集的良好序性



# 用强数学归纳法证明命题



- **例：**证明仅用面值4分和5分的两种邮票就可以组成12分及以上的每种邮资。

- **证明：**

$P(y), P(y+1), \dots, P(x)$  皆为真

- **奠基：**经验证，当  $n = 12, 13, 14, 15$  时上述命题皆成立；
- **归纳假设：**对任意自然数  $k > 15$ ，假设  $P(k)$  为真，则：
- **归纳步骤：**当邮资为  $n = k + 1$  时，可由邮资为  $k - 3$  的邮票组合方法加上一张4分的邮票构成。 $\because k - 3 > 12$ ， $\therefore$ 根据归纳基础和归纳假设，邮资为  $k - 3$  的邮票可由4分和5分邮票组合构成。根据强数学归纳法，命题得证。  $\square$



# 递归结构



从前有座山，山里有座庙，  
庙里有个老和尚  
给小和尚讲故事，讲的那是  
从前有座山，  
山里有座庙，  
庙里有个老和尚  
给小和尚讲故事  
讲的那是  
从前有座山...





# 递归结构（续）



- 命名中递归的例子：

Linux  
Is  
Not  
Unix



# 递归结构（续）



## ■ 艺术作品中递归的例子：

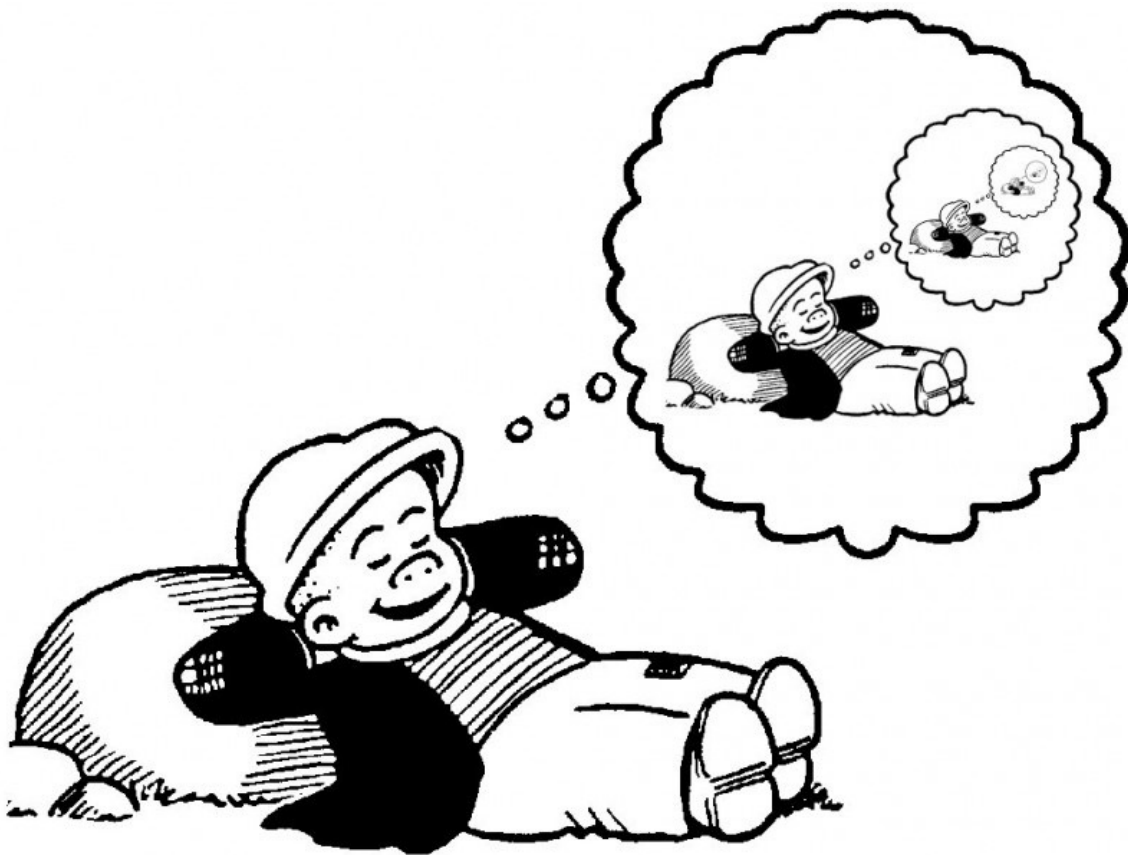




# 递归结构（续）



## ■ 艺术作品中递归的例子（续）：





# 函数的递归定义



- **定义（递归）**：在计算机科学中，递归指在函数的定义中使用函数自身的方法。良定义的递归函数一般要保证其能被还原为定义中的基础情况
- 阶乘的递归定义：
  - $\text{Fac}(0) = 1$ ;
  - $\text{Fac}(n) = n \cdot \text{Fac}(n - 1)$
- Fibonacci序列 $\{f_n\}$ 的递归定义：
  - $f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2}$





# 集合的递归定义



## ■ 递归地定义集合：

- (1) **奠基**：指定一些初始元素；
- (2) **递归步骤**：给出由集合中已有元素构造新元素的规则；
- (3) **排斥规则**：限制集合中的元素仅限由步骤(1)和(2)生成

## ■ **例1**：正整数集合的子集 $S$ 可以如下递归构造

- **奠基**：给定正整数 $x \in S$ ；
- **递归步骤**：若 $x \in S \wedge y \in S$ ，则 $x + y \in S$



# 集合的递归定义 (续)



- **例2:** 递归定义命题公式 (即合式公式  $w.f.f.$ )
  - **奠基:** 命题变元 (包括  $T/F$ ) 为命题公式;
  - **递归步骤:** 若  $X$  和  $Y$  为命题公式, 则  $(\neg X)$ 、 $(X \wedge Y)$ 、 $(X \vee Y)$ 、 $(X \rightarrow Y)$  和  $(X \leftrightarrow Y)$  也都为命题公式;
  - **排斥规则:** 命题公式仅限于此
- **例3:** 递归生成字母表  $\Sigma$  上的字符串集合  $\Sigma^*$ 
  - **奠基:** 空串  $\lambda \in \Sigma^*$ ;
  - **递归步骤:** 若  $\omega \in \Sigma^* \wedge x \in \Sigma$ , 则  $\omega x \in \Sigma^*$ ;
  - **排斥规则:**  $\Sigma^*$  仅由此生成



# 集合的递归定义 (续)



- **例4:** 递归定义字符串的长度函数  $l: \Sigma^* \rightarrow \mathbb{N}$ 
  - 奠基:  $l(\lambda) = 0$ ;
  - 递归步骤:  $l(\omega x) = l(\omega) + 1$ , 其中  $\omega \in \Sigma^* \wedge x \in \Sigma$
- **例5:** 递归定义  $\Sigma^*$  上的字符串连接运算 “ $\cdot$ ”
  - 奠基:  $\omega \in \Sigma^*$ , 则  $\omega \cdot \lambda = \omega$ ;
  - 递归步骤: 若  $\omega_1 \in \Sigma^* \wedge \omega_2 \in \Sigma^* \wedge x \in \Sigma$ , 则  $\omega_1 \cdot (\omega_2 x) = (\omega_1 \cdot \omega_2)x$



# 集合的递归定义（续）



## ■ 例6：递归定义Peano算术中的加法（+）运算

- 奠基： $a + 0 = a$
- 递归步骤： $a + b^+ = (a + b)^+$ ，其中 $\square^+$ 为后继运算

## ■ 例7：递归定义Peano算术中的乘法（ $\times$ ）运算

- 奠基： $a \times 1 = a$
- 递归步骤： $a \times b^+ = a + (a \times b)$ ，其中 $\square^+$ 为后继运算
- 特别地， $a \times 0 = 0$



# 结构归纳法



- **结构归纳法** (structural induction) 一般用于证明由递归构造的集合中的元素所具有的性质，或者用于证明与递归定义集合相关的命题，证明框架如下：
  - **奠基**：对于集合中的初始元素，证明命题成立；
  - **归纳步骤**：针对产生集合中新元素的规则，证明若已有元素满足命题，则该规则产生的新元素也满足命题
- 结构归纳法的有效性来源于递归定义的集合结构



# 结构归纳法 (续)



- **例1:** 根据 $\Sigma^*$ 与 $l$ 和 $\cdot$ 的定义(见例3—例5), 用结构归纳法证明:  $l(x \cdot y) = l(x) + l(y)$ , 其中 $x, y \in \Sigma^*$

- **证明:**

初始元素成立

- **奠基:** 对 $x \in \Sigma^*$ , 显然有 $l(x \cdot \lambda) = l(x) + l(\lambda)$ ;
- **I.H.:** 令 $P(y)$ 表示: 对任意 $y \in \Sigma^*$ , 有 $l(x \cdot y) = l(x) + l(y)$ 。假设 $P(y)$ 成立, 则:
- **归纳步骤:** (以下证明对于任一 $a \in \Sigma$ ,  $P(y) \rightarrow P(ya)$ , 即当 $x \in \Sigma^*$ , 有 $l(x \cdot ya) = l(x) + l(ya)$ ) 由归纳假设,  $P(y)$ 成立, 即 $l(x \cdot y) = l(x) + l(y)$ , 故 $l(x \cdot ya) = l(x \cdot y) + 1 = l(x) + l(y) + 1 = l(x) + l(ya)$ 。由结构归纳法, 命题得证。 □

新元素产生规则(例3):  $y \Rightarrow ya$

规则产生的新元素亦满足命题



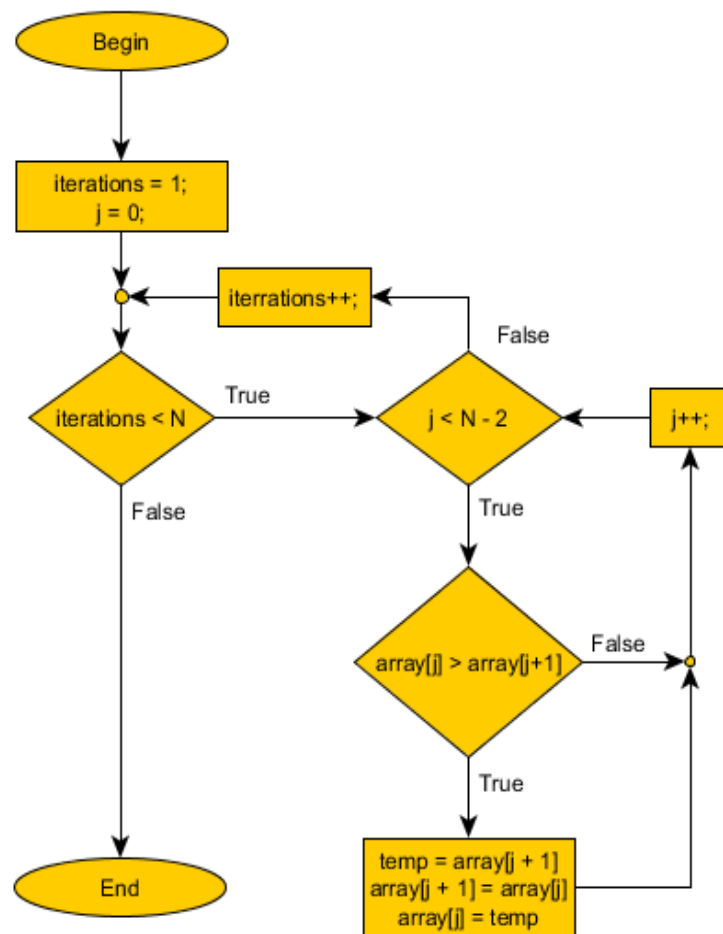
# 结构归纳法用于证明



自学内容  
(pp. 28—30)

## ■ 例2\*：用结构归纳法证明 冒泡排序算法的正确性

```
1 void bubbleSortBasic(int array[], int size)
2 {
3     int iterations = 0;
4     int temp;
5     for(int pass = 1; pass < size; ++pass)
6     {
7         for(int i = 0; i < size - 1; ++i)
8         {
9             ++iterations;
10            if(array[i] > array[i+1])
11            {
12                temp = array[i+1];
13                array[i+1] = array[i];
14                array[i] = temp;
15            }
16        }
17    }
18    printf("Basic bubble sort finished, using %d iterations.\n", iterations);
19 }
```





# 结构归纳法用于证明 (续)



- **例2\*:** 用结构归纳法证明冒泡排序算法的正确性
- **证明:** 设 $P(n)$ 为冒泡排序对于长度为 $n$ 的序列能正确排序, 其中 $n$ 是一个非负整数;
  - **奠基:** 当 $n = 1$ 时, 序列已经有序, 故 $P(1)$ 成立;
  - **I.H.:** 假设对于任意长度为 $k$ 的序列,  $P(k)$ 成立, 其中 $k$ 是一个任意的非负整数;
    - 说明算法对初始元素成立
  - **归纳步骤:** 考虑长度为 $k + 1$ 的序列。冒泡排序的过程是通过多次遍历序列, 不断交换相邻的元素, 将较大的元素逐渐交换到序列的最后。在每一次遍历中, 至少有一个元素会移动到其最终的位置上。因此, 经过 $k$ 次遍历之后, 长度为 $k + 1$ 的序列中最大的元素一定会被移动到正确的位置上。
    - 描述算法产生新元素的过程

(续后)



# 结构归纳法用于证明（续）



- **例2\*:** 用结构归纳法证明冒泡排序算法的正确性
- **证明:** 设 $P(n)$ 为冒泡排序对于长度为 $n$ 的序列能正确排序, 其中 $n$ 是一个非负整数;
  - **证明规则产生的新元素亦满足命题**
  - **归纳步骤（续）:** 根据归纳假设, 对于长度为 $k$ 的序列, 冒泡排序能正确排序。而在长度为 $k+1$ 的序列中, 最大的元素已经处于正确的位置上。因此, 我们可以将该元素剔除, 然后对剩余的长度为 $k$ 的序列进行冒泡排序。根据归纳假设, 剩余的 $k$ 个元素会被正确排序。因此, 长度为 $k+1$ 的序列也会被正确排序。即证明了 $P(k) \rightarrow P(k+1)$ 。由结构归纳法, 算法正确性得证.  $\square$



# 本次课后作业



- 教材内容：[Rosen] 5.1—5.3节，5.4节（自学）
- 课后习题：
  - Problem Set 10
- 提交时间：4月1日 10:00前

