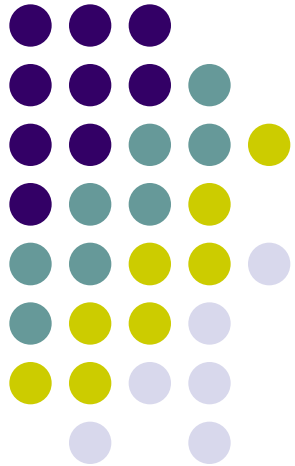
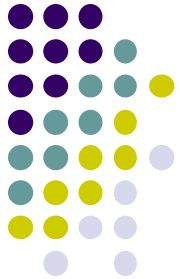
L1:数据结构与算法

谷方明
fmgu2002@sina.com

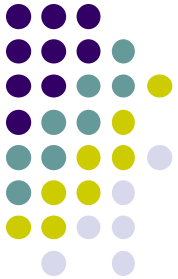


学习目标

- 数据结构的研究内容
- 数据结构的定义
- 算法的定义和描述
- 算法评价基本准则
- 算法的正确性证明

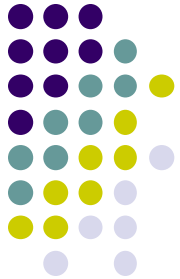


例1：排序问题（引入）



- 排序问题
  - ✓ 输入：n 个数的一个序列 $a_1, a_2, \ldots, a_n$
  - ✓ 输出：输入序列的一个排列 $a_{i1}, a_{i2}, \ldots, a_{in}$，满足$a_{i1} \le a_{i2} \le \ldots \le a_{in}$
- 解题方法
  - ✓ ……
  - ✓ 运行（演示）

数据结构的研究内容



- 演示分析
  - ✓ 编写一个能运行的程序
  - ✓ 编写一个有效(efficient)的程序（满足资源限制）

- 数据结构研究：组织数据（结构化信息）的方法，以支持有效的处理。

课程地位

- 计算机科学的核心课
  - ✓ 算法 + 数据结构 = 程序
- 重要的专业基础课
  - ✓ 编译原理：栈、语法树
  - ✓ 操作系统：队列、目录树、死锁（环）
  - ✓ ……
- 计算机相关专业的考研课

教材和参考书

- 教材：《数据结构》第3版，刘大有等编著，高等教育出版社，2017年3月；

- 参考书
  - ✓ 《算法导论》，CLRS，潘金贵，机械工业出版社
  - ✓ 《计算机编程艺术》，Donald E. Knuth
  - ✓ 《数据结构与算法分析》，Clifford A. Shaffer
  - ✓ 《数据结构与算法分析》，Mark Allen Weiss
  - ✓ STL网上教程（搜索引擎：STL 教程）
  - ✓ 《STL源码剖析》，侯捷，华中科技大学出版社

在线课程

- 数据结构(学习通)
- 参考在线课程
  - MIT 6.006: (Introduction to Algorithms) https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/
  - Stanford cs166: http://web.stanford.edu/class/cs166/
  - 清华大学 数据结构 上/下 https://www.xuetangx.com/course/THU08091000384/5883586?channel=search_result
  - 浙江大学 数据结构 https://www.icourse163.org/course/zju0901-93001

课程设置

- 学　　时：理论课（48） ＋ 实验课（32）
- 课程特点
  - ✓ 理论：难（抽象层级、课时）
  - ✓ 实践：多（代码行数、实验种类）
- 学习方法
  - ✓ 动手编程（大量）；
  - ✓ 阅读经典（参考书、论文等）；
- 考核方式：平时（学习通：在线课程+作业+课堂）；实验；期末考试；

课程要求

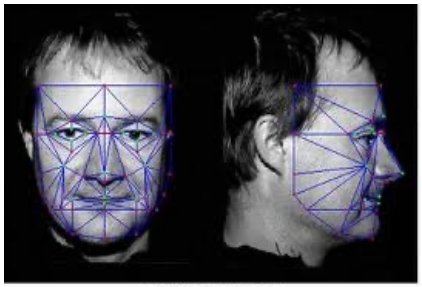
- **算法竞赛（ACM/CCSP/天梯赛, CSP，…）**
  - ✓ 课程偏理论,课时少；竞赛偏应用,投入时间多
  - ✓ 竞赛选手一般擅长实验，但理论（笔试）未必擅长
- **课程免修**
  - ✓ 达到规定条件
  - ✓ 任课教师同意（支持 VS 反对）
- **出勤（签到；学院、安全责任；安静自学）**

1. 数据

- **数据是对象的表示。**

  即按照适合于通信、解释或处理（借助人或自动装置）的方式所形成的关于事实、概念或指令的表示（百科）。

- **数据**是**计算机程序**要处理**的“原料”，**是所有**被计算机识别、存储和加工处理的符号的总称。**



人脸特征点分布图示例

上证指数 (000001) 2421.117 ↑0.49% +11.69 实时行情 加入自选股
2010-07-07 15:03

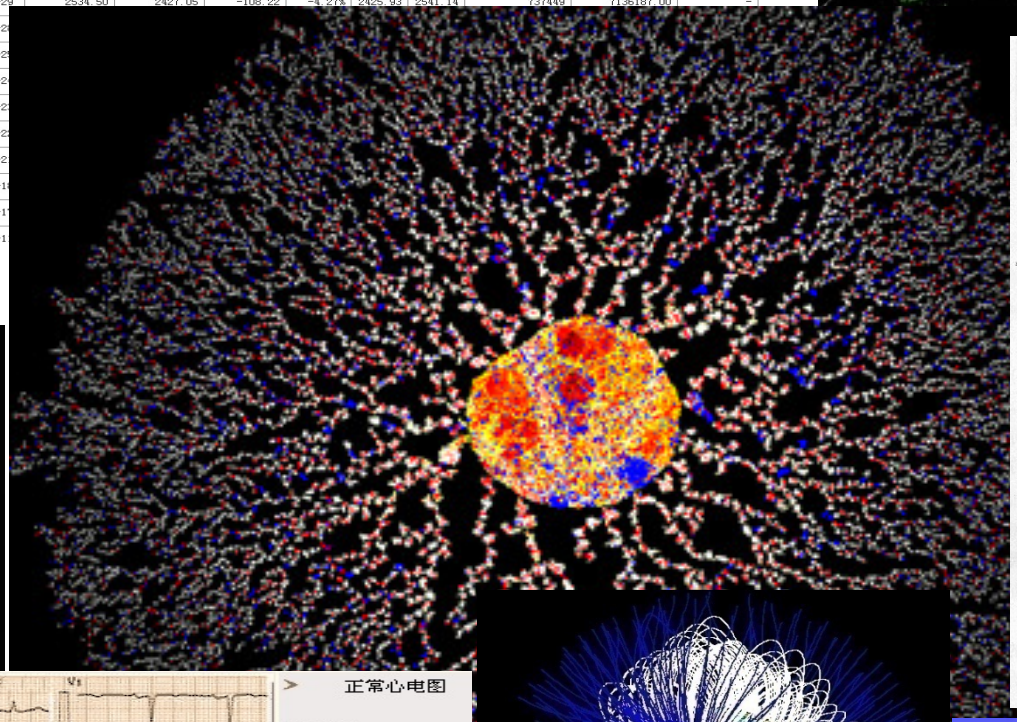
查看其他证券“历史行情” 代码/拼音缩写/中文 确定

实时行情 成交明细 历史行情

历史行情

从 2010-05-21 至 2010-07-06 按日 按周 按月 查询

日期	开盘	收盘	涨跌额	涨跌幅	最低	最高	成交量(手)	成交金额(万)	换手率
累计：	2010-05-21至2010-07-06		-174.10	-6.74%	2319.74	2686.54	20652674	217033504.00	-
2010-07-06	2358.28	2409.42	45.48	1.92%	2356.55	2409.78	613643	5561825.50	-
2010-07-05	2358.76	2363.95	-18.95	-0.80%	2335.57	2378.09	479965	4299967.00	-
2010-07-02	2371.32	2382.90	9.11	0.38%	2319.74	2386.40	683296	6154527.00	-
2010-07-01	2393.95	2373.79	-24.58	-1.02%	2371.78	2410.77	499677	4641516.00	-
2010-06-30	2409.54	2398.37	-28.68	-1.18%	2382.36	2411.44	552106	5115435.00	-
2010-06-29	2534.50	2427.05	-108.22	-4.27%	2425.93	2541.14	737449	7136187.00	-
2010-06-28									
2010-06-25									
2010-06-24									
2010-06-23									
2010-06-22									
2010-06-21									
2010-06-18									
2010-06-17									
2010-06-1									



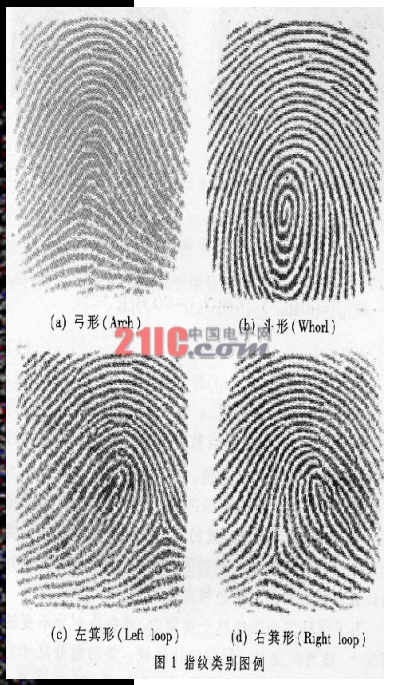
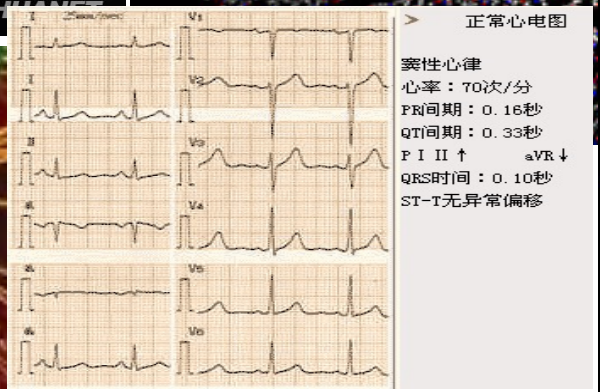
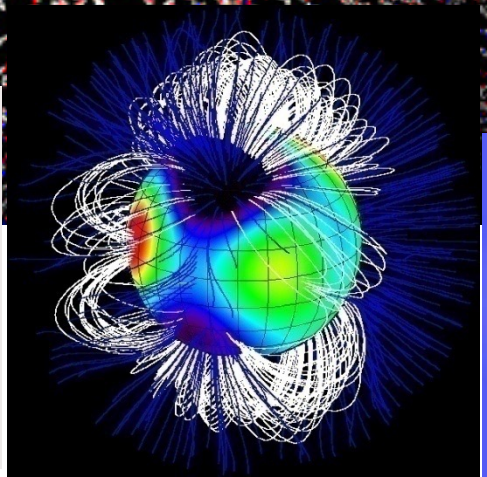
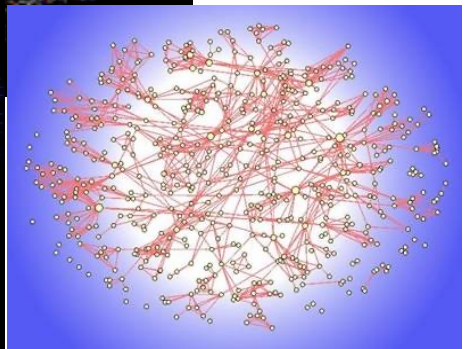


图 I 指纹类别图例





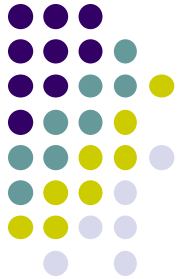


数据元素

- 数据元素是组成数据的基本单位，也称数据成分、元素、结点等 。数据元素可大可小，在程序中通常把一个数据元素作为一个整体来考虑和处理

- 数据元素可由若干数据项（域或字段）组成。

例2　学生信息表（元素和数据项）

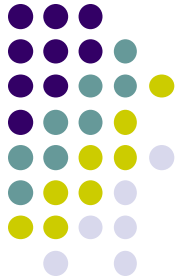


学号	姓名	学院	班级	性别
12210904	赵一	计算机学院	212116	男
12210906	钱二	软件学院	552114	女
12210912	孙三	计算机学院	212116	女
12210914	李四	软件学院	552114	男
19211613	周五	计算机学院	212116	男

2. 数据结构

- 数据结构由三部分组成
  - ✓ 数据的逻辑结构
  - ✓ 数据的存储结构
  - ✓ 运算（操作）

2.1 数据的逻辑结构



- 数据的逻辑结构是指数据元素及其间的逻辑关系。

- 可形式定义为一个二元组：**L = (N, R)**
  - ✓ **N** 是有限结点集合，
  - ✓ **R** 是 **N** 上的二元关系 **r** 的集合

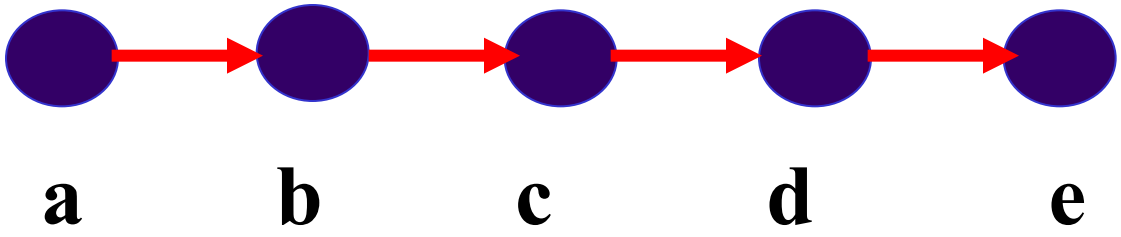
- 图形化表示
  - ✓ 将数据元素抽象为结点
  - ✓ 数据元素间的关系抽象为连接结点的边。

例3 L=(N,R)，

N={a, b, c, d, e} ，

R={r} ，

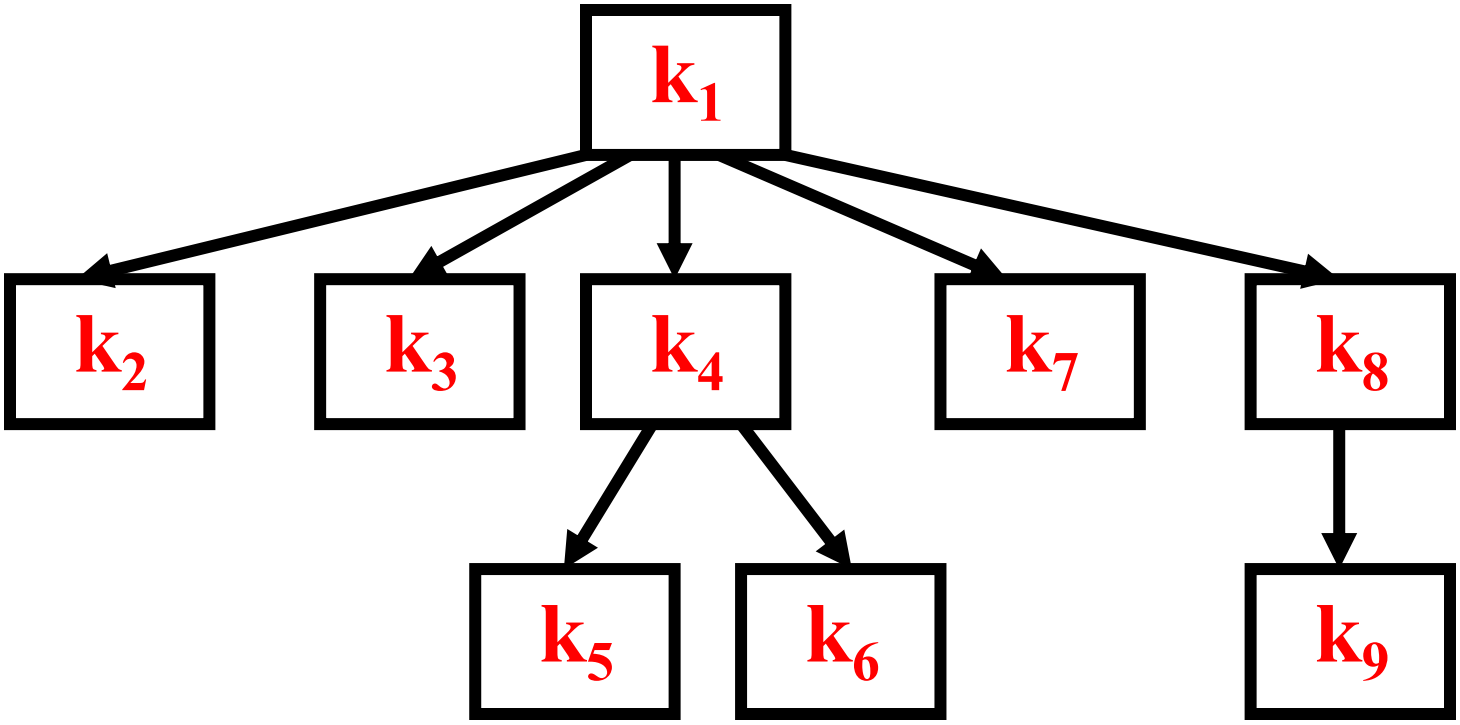
r={<a,b>, <b,c>, <c,d>, <d,e>}



例4 $L=(N,R)$, $N=\{k_1,k_2,\ldots,k_9\}$, $R=\{r\}$,

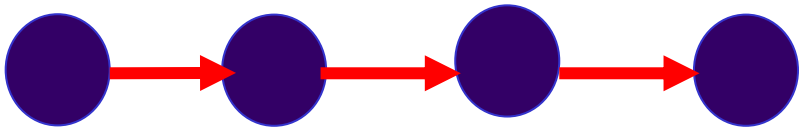
$r=\{<k_1,k_2>, <k_1,k_3>, <k_1,k_4>, <k_1,k_7>,$

$<k_1,k_8>, <k_4,k_5>, <k_4,k_6>, <k_8,k_9>\}$



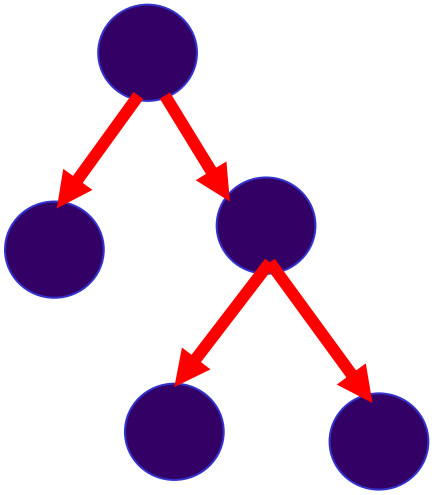
逻辑结构

线性结构



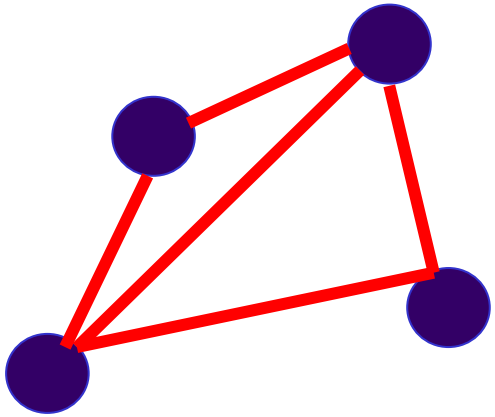
一对一

树



一对多

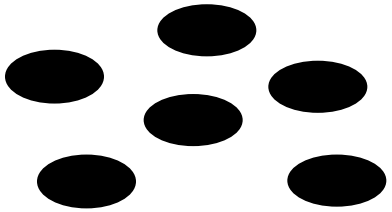
图



多对多

分类准则：图形表示中的前驱后继个数

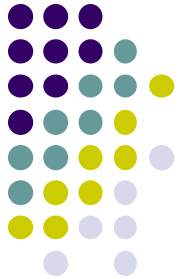
逻辑结构说明



集合

- 这种分类不是绝对的。
  - ✓ 有些书将数据的逻辑结构分为四种：集合、线性结构、树和图（集合：数据元素之间除了同属于一个集合的关系之外，别无其他关系）。
  - ✓ 未涉及复杂关系（如广义表）。
- 数据的逻辑结构有时被直接称为数据结构。可根据语境区分。

2.2 数据的存储结构（物理结构）

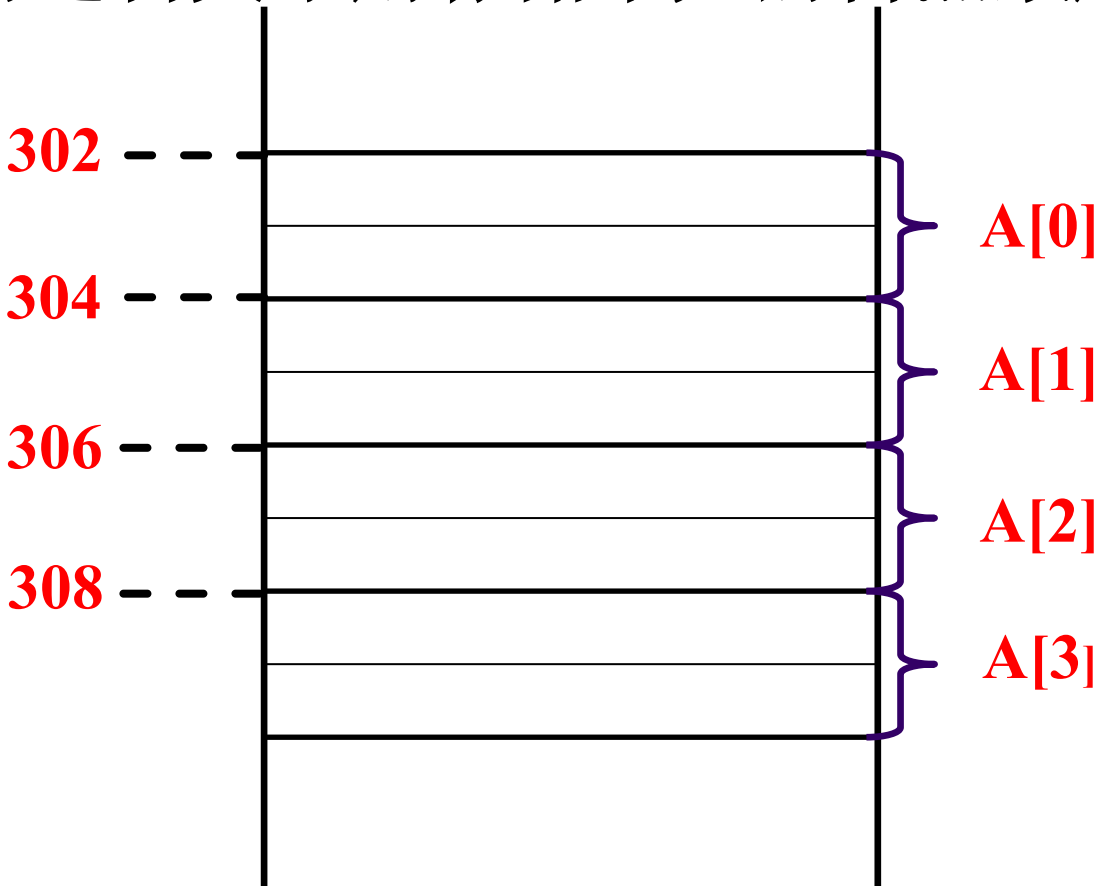


- 数据的存储结构是指：数据的逻辑结构在计算机中所需的存储空间、空间的构成结构及对该存储结构的访问方式等的总称。

- 数据的存储结构是建立一种由逻辑结构到存储空间的映射。
  - ✓ 例：学生信息表

1）顺序存储结构

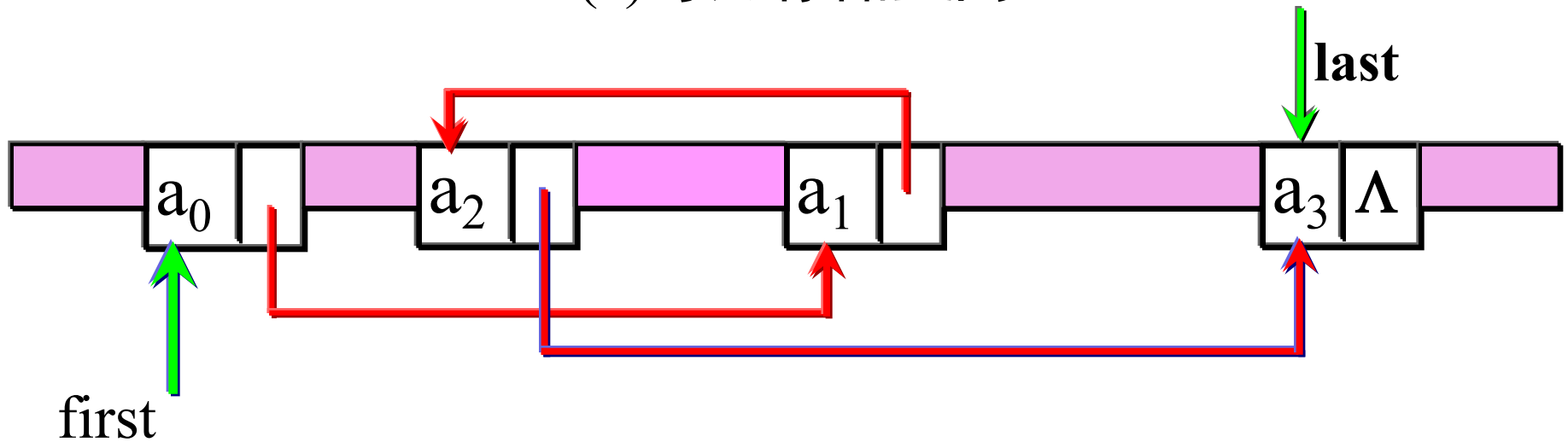
- 把一组结点存放在地址相邻的存储单元里，结点间的逻辑关系用存储单元的自然顺序关系表达



2）链接存储结构

- 在结点的存储结构中附加指针字段，两个结点的逻辑后继关系用指针的指向来表达。

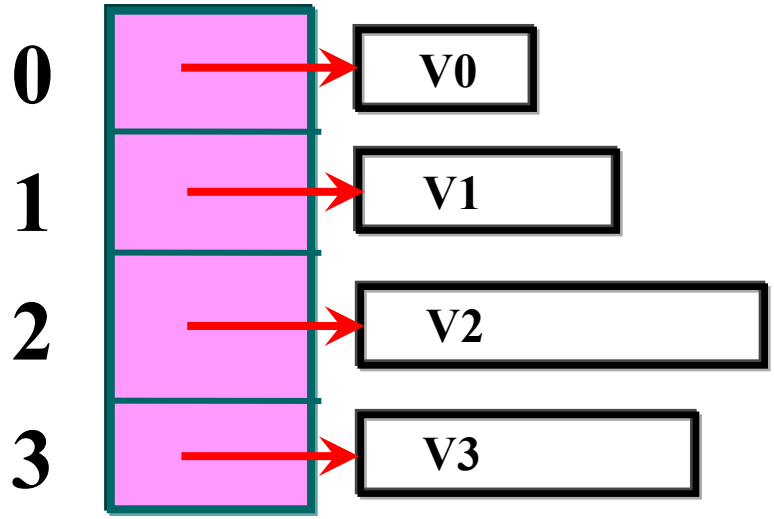
(a)可用存储空间



(b)创建若干个结点后

3）索引存储结构

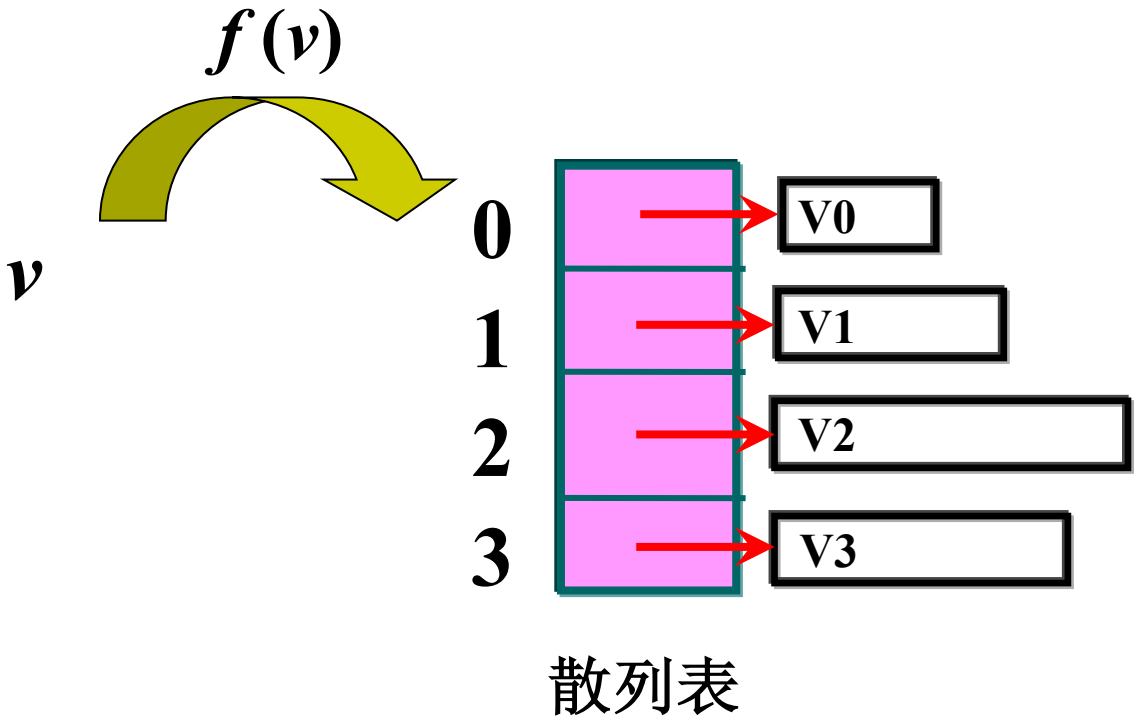
- 索引表把整数索引值映射到结点的存储地址。索引表存储一串指针，每个指针指向存储区域的一个数据结点。
- 查找时，先查索引表，再找数据结点。



索引表

4）散列存储结构

- 利用散列函数进行索引值的计算，并通过索引表（散列）表求出结点的地址。可看作索引存储的一种延伸。

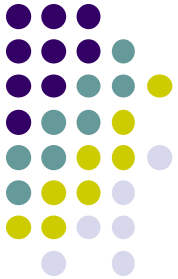


小结

- 四种存储结构
  - ✓ 顺序
  - ✓ 链接
  - ✓ 索引
  - ✓ 散列

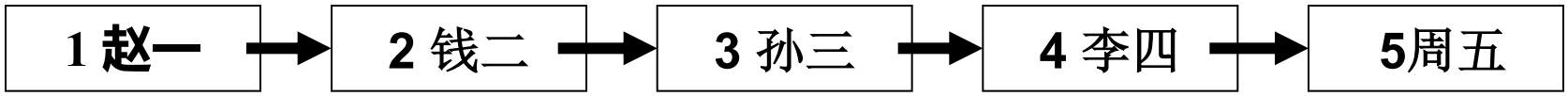
- 存储结构要正确反映逻辑结构，并便于操作。
- 存储结构可单独使用，也可组合使用。

2.3 对数据的操作（运算）

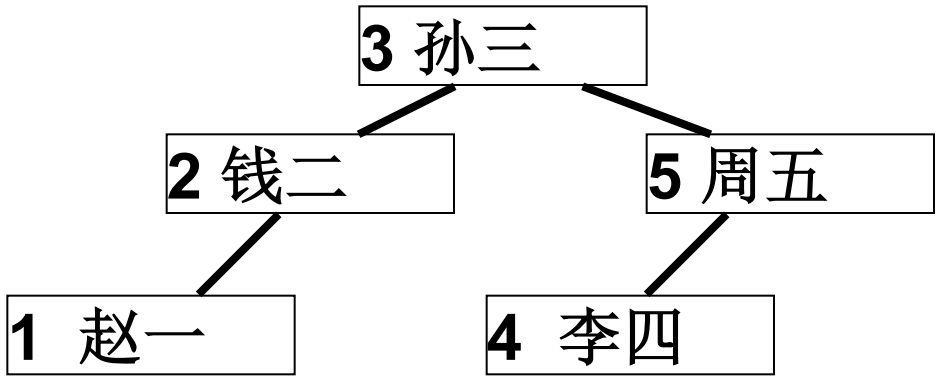


- 查找
- 排序
- 插入
- 删除
- 修改
- ……

例3　学生信息表的数据结构示例



序号	姓名
1	赵一
2	钱二
3	孙三
4	李四
5	周五

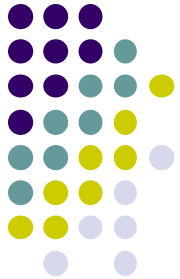


2.4 数据结构的定义

- 数据结构包括数据的逻辑结构、数据的存储结构和对数据的操作三方面内容

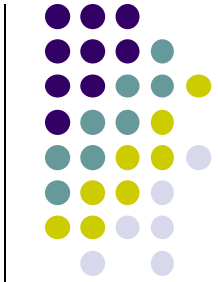
- 三者紧密相关，构成有机整体
  - ✓ 逻辑结构的不同会产生不同的数据结构。
  - ✓ 存储结构不同，即使逻辑结构相同，也会产生不同的数据结构。线性表是一种逻辑结构，顺序存储则为顺序表；链接存储则为链表；散列存储则为散列表。
  - ✓ 对数据的操作不同，即使其逻辑结构和存储结构相同，也可能对应着不同的数据结构。栈和队列。

相近概念



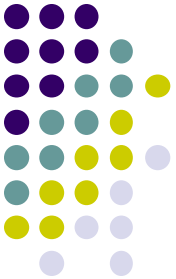
- **数据类型**
  - ✓ 性质相同的**数据元素的集合**及其上定义的一组**操作**
  - ✓ 例：int
  - ✓ 实现了的数据结构；实现细节对用户透明(不可见)
- **抽象数据类型（ADT）**
  - ✓ 一组操作的集合；
  - ✓ 不涉及如何实现，用于设计决策阶段
  - ✓ 例: 学生信息表

2.5 数据结构简史



- 20世纪40年代
  - ✓ 电子计算机、数值计算，不存在数据结构
- 20世纪50年代
  - ✓ 数据类型增多（数组、记录、串和层次表 ）
- 20世纪60年代
  - ✓ **信息结构，Knuth 《计算机编程艺术》**
- 20世纪70年代
  - ✓ 数据库（文件组织、存储 等）
  - ✓ **N. Wirth 《算法＋数据结构＝程序》**
- 20世纪80年代：**抽象数据类型（面向对象）**
- 20世纪90年代：**网络（图模型）**
- **21世纪：？ 数据科学**

3 算法(Algorithm)



- 著名例子：Euclid‘s Algorithm（Euclid’s Elements，第VII卷，命题i和ii）。
  - ✓ 用于求两个正整数最大公约数。
  - ✓ 也称为辗转相处法

例：求 15 和 9 的最大公约数gcd(15,9)

算法的定义

- 算法是有穷规则的集合，规定了解决某一特定类型问题的运算序列（教材）。
- 通常，一个算法有5个重要特性（Knuth）：

  1.有限性：算法必须在执行有限步后结束；

  2.确定性：算法描述必须无歧义，通常结果确定；

  3.可行性：算法中的运算必须是可行的；

  4.输入：一个算法有零个或多个输入；

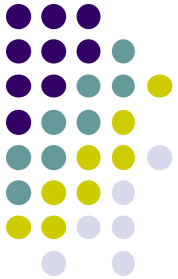
  5.输出：一个算法有一个或多个输出；

算法的描述

- 自然语言
- 流程图（盒图、**PAD**图）
- 程序（**C\C++**、**Java**等）
- 自定义（伪代码）

- 教材上用**PASCAL**风格的算法描述语言**ADL**；代码使用的是**C++**语言（扫描二维码）。

例 计算以任意正实数为半径的圆的面积。（自然语言描述）



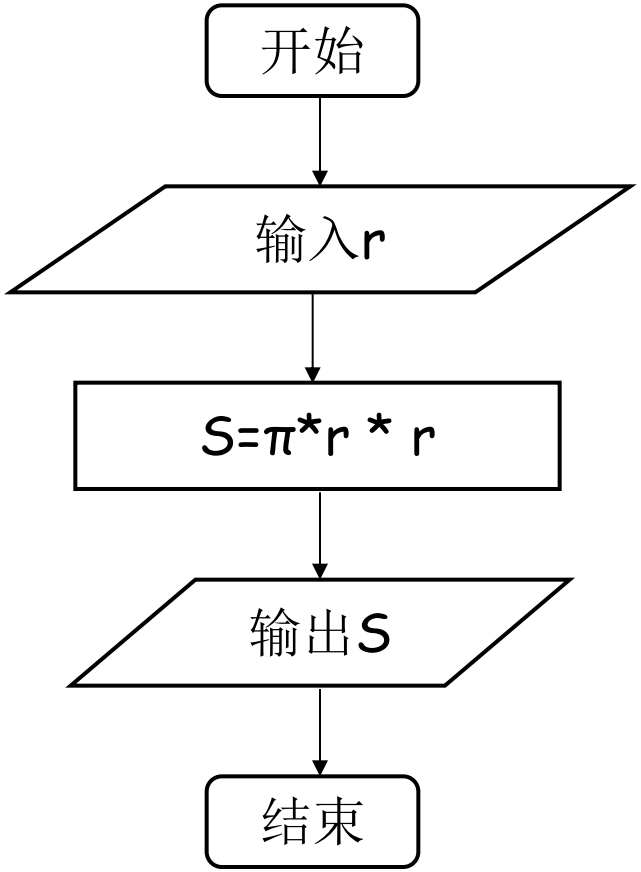
算法:

S1. 输入半径**r.**

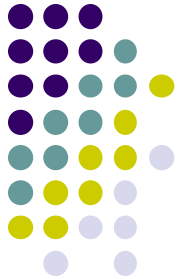
S2. 计算圆的面积**S=π*r * r**

S3. 输出**S.**

例 计算以任意正实数为半径的圆的面积。（流程图描述）

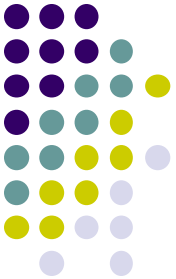


例 计算以任意正实数为半径的圆的面积。（C程序描述）



```
int main()
{
    double r,s,pi=3.14159265;
    scanf("%lf", &r);
    s=pi*r*r;
    printf("%lf\n",s);
}
```

ADL描述



算法 Area (*r* . *s*)

/* 给定圆的半径r ， 算法Area求圆的面积s*/

A1. [初始化]

READ(r).

pi ←3.14159265

A2. [计算] s ←pi*r*r.

A3. [输出] WRITE(s). ▌

ADL格式

算法<算法名>(变量i_1, …,变量i_m .变量j_1, …,变量j_n)

//<算法的概括说明> 或者 /*<算法的概括说明>*/

<步骤名><步骤号>. [<本步骤的概括说明>]

<操作1>

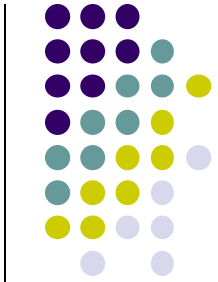
…

<操作J>

<步骤名><步骤号>. [<本步骤的概括说明>]

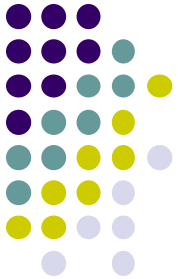
… ▌

ADL格式说明



- **<算法名>是由字母和数字组成的有限字符串，且串中第一个符号必须是字母。（通常要短一些）**
- **<算法的概括说明>：算法的功能、前提条件、重要参数说明等，通常必须有**
- **用符号“▌”作为算法描述完毕的结束符。**
- 操作
  - ✓ 表达式：可使用数学表示
  - ✓ 语句：类PASCAL；语句结束符号使用“.”

算法描述小结

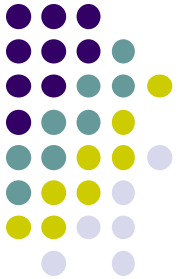


- 自然语言描述一般文字较多，适合复杂问题或顶层描述；可根据夹杂数学运算；
- 图形描述法需要画图，适合入门或特殊用途；
- 伪代码描述能专注计算和逻辑（忽略数据类型、直接使用数学运算和符号），适合函数级算法。
- 代码描述能直接运行，但关注存储细节，导致描述较长、不利于设计层面的思考；

算法描述建议

- 函数级算法建议使用伪代码
  - ✓ 能读懂教材上的**ADL**描述；
  - ✓ 描述算法时，可用**C/C++**风格的指令替换**PASCAL**指令，**ADL**是种约定。
- 问题复杂或抽象程度较高时，建议使用自然语言。
- 程序设计语言描述不推荐。

ADL-C/C++ pseudocode描述



算法 Area (*r* . *s*)

/* 给定圆的半径r ， 算法Area求圆的面积 s*/

A1. [初始化]

cin>>r

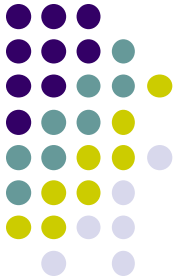
pi =3.14159265

A2. [计算] s = pi*r*r

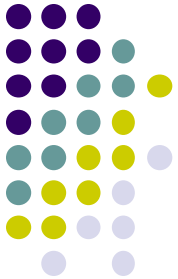
A3. [输出] cout << s

```
INSERTION-SORT(A)
1  for j ← 2 to length[A]
2       do key ← A[j]
3          ▷ Insert A[j] into the sorted sequence A[1..j−1].
4          i ← j−1
5          while i>0 and A[i]>key
6               do A[i+1] ← A[i]
7                  i ← i−1
8          A[i+1] ← key
```

```
DAG-SHORTEST-PATHS(G, w, s)
1  topologically sort the vertices of G
2  INITIALIZE-SINGLE-SOURCE(G, s)
3  for each vertex u, taken in topologically sorted order
4       do for each vertex v ∈ Adj[u]
5            do RELAX(u, v, w)
```



算法定义的说明



- **算法和程序**
  - ✓ 算法可看作是程序的思想；
  - ✓ 程序可看作是算法的实例（不是所有的程序都是算法的实例）
  - ✓ 有时不加区分的使用两者

- **算法的一个经典定义：算法就是定义良好的计算过程，它取一个或一组值作为输入，并产生一个或一组值作为输出。——《算法导论》**
  - ✓ 定义良好指满足前述算法的5个特性。

算法与数据结构的关系

- **算法与数据结构相互依存**
  - ✓ 算法依附于具体的数据结构，数据结构直接关系到算法的选择和效率。
  - ✓ 对数据结构的操作由计算机来完成，这就要设计相应的插入、删除和修改等算法。

- **算法+数据结构=程序**

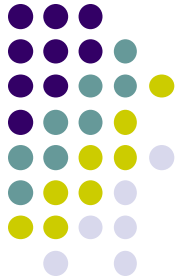
算法的评价准则

- 一个好的算法通常考虑以下5个方面
  - ✓ 正确性
  - ✓ 时间效率：运行时间
  - ✓ 空间效率：占用内存
  - ✓ 可读性
  - ✓ 鲁棒性（ROBUSTNESS）
- 算法设计的基本要求
  - ✓ 正确性
  - ✓ 效率

4 算法的正确性证明

- 一个算法是正确的，是指对于一切合法的输入数据，该算法都会在有限时间产生正确的结果。
- 算法的正确性一般通过数学方法证明，例如直接证明法、数学归纳法等
- 对于一个具体算法，证明步骤如下
  - ✓ 算法有限步终止；
  - ✓ 算法终止时，得到的结果是正确的；
- 通用算法的正确性证明和停机问题都是不可计算问题（冰雹猜想/角谷猜想/3x+1）

数学归纳法



- 数学归纳法，通常被用于证明某个给定命题（定理）在整个（或者局部）自然数范围内成立。

- 数学归纳法依靠假设的事实来证明定理，是用“有限”步骤解决“无限”步骤问题的一种严格证明方法。

- 例 证明由如下递归关系式

$$T(n) = 0 \qquad \text{当 } n = 1$$

$$T(n) = T(n-1) + 1 \qquad \text{当 } n > 1$$

可推出 $T(n) = n - 1$，其中 $n \geq 1$.

证明（数学归纳法）

1. 基础情形： $n=1$ 时，$T(1)=1-1=0$，结论成立。

2. 归纳步骤：

假设$T(n-1)=n-2$ 成立，往证 $T(n)=n-1$ 成立。

由递归定义知：

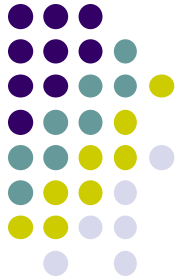
$n>1$ 时，有$T(n)=T(n-1)+1$

再由归纳假设：

$$T(n)=T(n-1)+1=n-2+1=n-1$$

由数学归纳法推出$T(n)=n-1$成立。 ∎

第一数学归纳法



设T是一个定理，n 是T中的正整数参数。数学归纳法表明，如果下面两个条件为真，则对于正整数参数 n 的任何值，T都是正确的：

①基础情形：$n = c$ 时，T成立。

②归纳步骤：若 $n = k\text{-}1$ 时T 成立, 则 $n=k$ 时T也成立。

其中，c 是一个较小的正整数常量，$n \geq c$.

第二数学归纳法 (强归纳法)

①基础情形：$n = c$ 时，T成立。

②归纳步骤：若对于所有的 $n < k$ 时T 都成立，

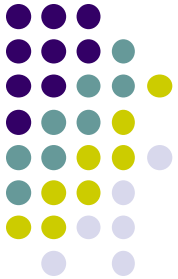
则 $n = k$ 时T也成立。

第一数学归纳法的归纳步骤：

若 $n = k\text{-}1$ 时T 成立, 则 $n=k$ 时T也成立。

- 强归纳法的假设条件更强；
- 两者等价；根据问题的情况选用；

例：欧几里得算法



算法 E (m , n . n)

/* 给定两个正整数m和n， 算法E求它们的最大公因数，输出结果在n中 */

E1. [求余数] $r \leftarrow m$ **MOD** n .

E2. [余数为零？] IF $r = 0$ **THEN RETURN** n.

E3. [辗转] $m \leftarrow n$. $n \leftarrow r$. **GOTO E1.** ▌

证明算法E的正确性

1. 算法E有限步终止

   r最大是min(m,n)，每次辗转相除，r至少减1

2. 终止时得到正确结果。即证明：经过辗转相除，当$r = 0$时，得到的变量n的值n'是m和n的最大公因子。

引理1.1

- 若*m*，*n*为正整数，*r* = *m* MOD *n*，则*m*和*n*最大公因子 等于 *n*和r 的最大公因子。

引理1.1证明

- 设***r*** 为***m***除以***n***所得的余数，则***m***可以表示为：***m = kn + r***。其中，***k***是正整数。
- 设***m***和***n***的公因子集为***F***，***n***和***r*** 的公因子集为***G*** 。
  - ✓ 任取$c \in F$，则 c 整除m，c整除n。而 $r = m - kn$，因此，c整除r，即 $c \in G$, $F \subseteq G$。
  - ✓ 任取$d \in G$，则 d 整除n，d整除r。而$m = kn + r$，因此，d整除m，即 $d \in F$, $G \subseteq F$。
- 因此，***F = G***。即***m***和***n***的公因子等于***n***和***r***的公因子。特别地，***m***和***n***的最大公因子等于***n***和***r***的最大公因子。

算法E的正确性证明：对n归纳

1.基础情形：$n=1$ 时， m和n的最大公因数为1，此时，$n'=1$. 结论成立。

2.归纳步骤：假设 $n<k$ 时，n'是m和n的最大公因子，结论成立。往证$n=k$ 时，n'是m和n的最大公因子；

- 设第一次运行算法时，$m=dn+r$，$m\leftarrow n$，$n\leftarrow r$。由于$r<k$，由假设知：n'即为n和r的最大公因子。
- 由引理1.1，m和n的最大公因子等于 n和r的最大公因子。因此，n'是m和n的最大公因子。证毕。

总结

- 数据结构的定义
- 算法的定义（**5**个特性）
- 算法的描述
- 算法的正确性证明（两步）

课后任务

- 慕课
  - ✓ 在线学习/预习 第 2 章 视频
- 作业
  - ✓ P48：2-1，2-2，2-5，
    2-6，2-9，2-10，2-11
  - ✓ 在线提交