7

数据结构基本概念

Data Structures

郝家胜

hao@uestc.edu.cn

自动化工程学院

内容回顾：计算机求解问题

- 明确需求
  - 建立问题模型，解决方法的前提和出发点：问题抽象
- 建立抽象模型
  - 分析和简化，去伪存真：数据抽象
- 设计算法
  - 以伪代码、流程图、N-S图等方式表示：过程抽象
- 编码实现
  - **ASM/C/C++/PSCAL/Java/Python/…**
- 测试验证

内容提要

- 数据结构的基本概念
  - 什么是数据结构
  - 为什么要研究数据结构
  - 数据结构的基本概念
- 抽象数据类型
  - 数据类型的概念
  - 什么是抽象数据类型
  - 抽象数据类型的用处

内容提要

- 什么是数据结构
- 为什么要研究数据结构
- 数据结构的基本概念
- 抽象数据类型

计算机查找的到底是什么？

- 计算问题
  - 查找
- 算法设计
  - 顺序查找
  - 二分查找
- 数据对象
  - 逻辑有序性
  - 物理连续性
- 如何对数据对象进行组织和存储？
- ——数据结构的研究范畴

什么是数据

- 数据：信息的载体，泛指能被计算机识别和处理的对象的全体
  - 整数、有理数、实数
  - 文本编辑器的处理对象
  - 编译器的处理对象
  - 媒体播放器的处理对象
- 数据的基本单位是数据元素
  - 程序中作为一个整体进行处理
  - 数据元素可由若干数据项（数据域）组成
  - 举例：花名册

数据类型 (Data Type)

- 数据类型是数据的一种抽象属性，它描述了一组值的集合以及定义在这些值上的操作
  - a **data type** defines a set of values and the allowable operations on those values
  - 限定了允许的取值集合 (模型)
  - 限定了允许的操作方法 (使用)
- 原子类型（基本类型）
  - C: int, short, long, char, float, double
- 结构类型（复合类型）
  - array，struct

数据类型及其表示

- 示例：**int**
  - 值域：**-32768 ~ 32767**
  - 操作：**+, -, *, /, %, sqrt**
  - 实现（内部表示）？
    - 不知道
    - 不需要知道
    - 不应该知道
- 类型的含义
  - 抽象出逻辑特性，设计合适的接口提供逻辑操作
  - 采用合适的表示方式来实现这些逻辑操作
  - 客户仅通过类型提供的接口来识别和操纵数据

复杂数据的表示

- 如何表示有理数、复数？
- 如何表示多项式？
  - $f(x) = x^n + a_{n-1}x^{(n-1)} + \ldots + a_1x + a_0$
- 编译器如何计算复合表达式的？
  - a + 2 * (b + c / 4)
- 如何表示棋局？
- 如何表示迷宫问题？

组合数据的基本方法

- 数组
  - 同一类型的数据元素组成的序列，每个元素表示的意义相同
- 记录
  - 若干数据项组合而成的一个整体，每个数据项分别表示不同的含义
  - 数据项的类型可以不同
- 复合数据的结构
  - 从宏观上研究数据之间的关系
  - 有效组织复杂的数据
    - 提高执行效率
    - 提高抽象层次

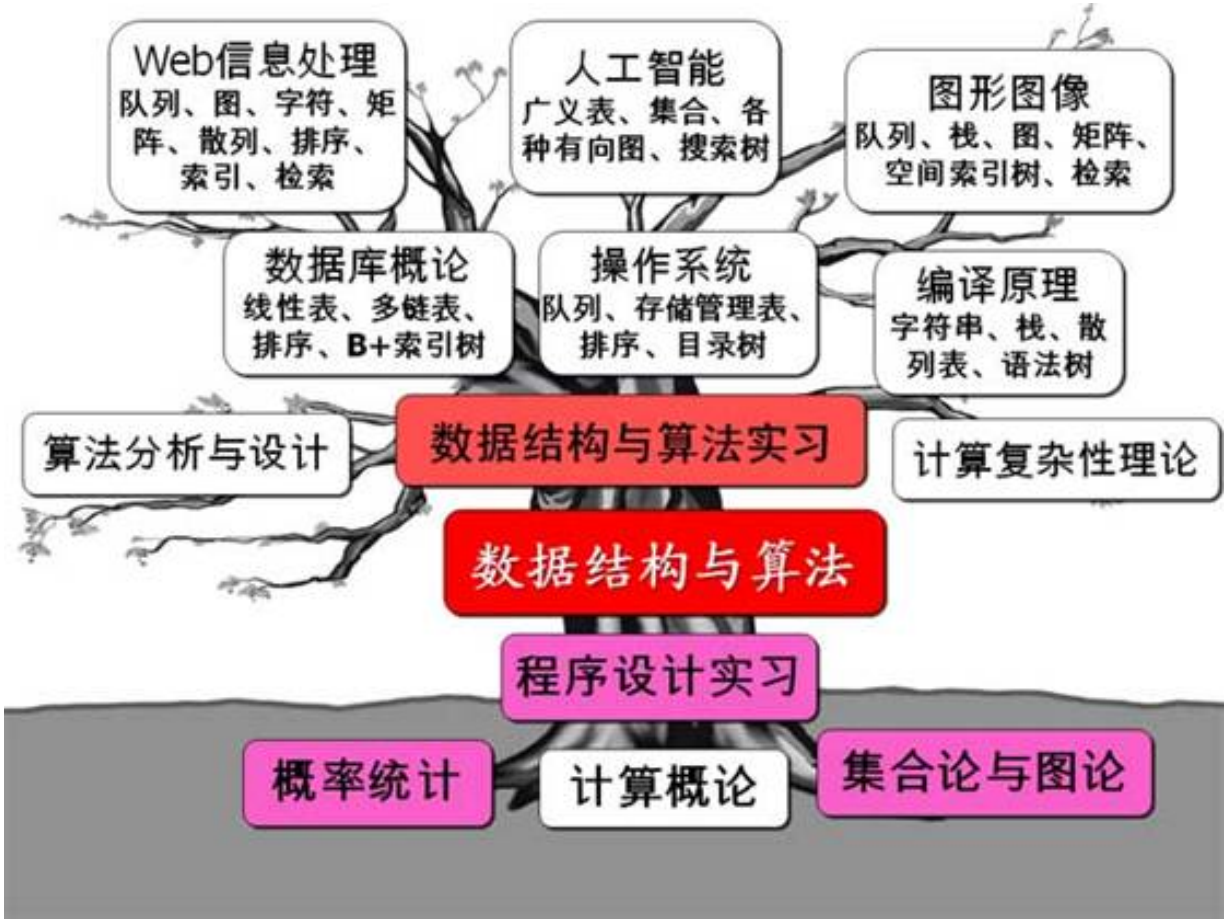
数据结构的基本概念

什么是数据结构

- 1968年，Donald E. Knuth开创了数据结构的最初体系
- 数据结构是指计算机系统中数据的组织方式，包括数据的逻辑结构、数据的逻辑运算及存储结构三个方面
  - The term ***data structure*** is used to describe the way data is *orgnized*
  - the term ***algorithm*** is used to describe the way data is *processed*
- 一类按照一定逻辑关系组织起来的数据的表示及其相关操作

数据结构的意义

- 为复杂数据的表示提供了有力手段
- 数据结构是编译器和操作系统的重要基础



数据结构中的结点

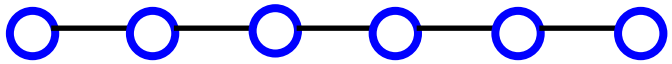
- 数据的基本单位称为结点（记录）
  - 关注数据元素之间的关系
  - 忽略数据元素的类型
- 结点的构成
  - 基本类型
  - 复合类型
  - 结点具有相同的尺寸
- 结点之间的关系
  - 逻辑关系
  - 物理关系

数据的逻辑结构

- 结点之间的相互关系（逻辑关系）
- 两类逻辑结构
  - 线性结构
  - 非线性结构

线性结构

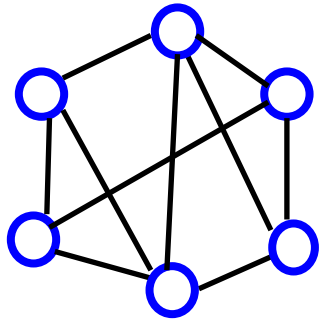
- 有且仅有一个开始结点，它最多只有一个直接后继
- 有且仅有一个终端结点，它最多只有一个直接前驱
- 其它所有结点（内部结点）都有且仅有一个直接前驱，且有且仅有一个直接后继

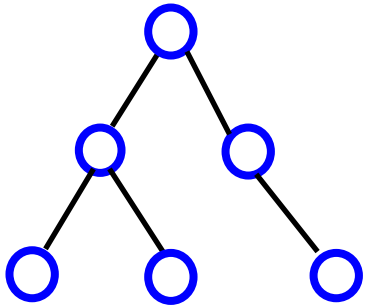


- 结点之间存在“一对一”的关系
- 实例：线性表、栈、队列、数组、串

非线性结构

- 结点可能有多个直接前趋和多个直接后继
  - 图结构（网状结构）
    - 结点的直接前驱和直接后继的个数没有限制
    - “多对多”
  - 树结构
    - 有且仅有一个称为根的结点，它没有直接前驱
    - 其它结点都有且仅有一个直接前驱
    - 所有结点都存在一条从根到该结点的路径
    - “一对多”
- 结点之间存在一对多或多对多的关系





数据结构的操作运算

- 向外提供行为特征
  - 定义在逻辑结构上的运算（做什么）
- 常用的几种操作运算
  - 遍历
  - 插入
  - 更新
  - 删除
  - 查找
  - 排序（线性结构）

数据的存储结构

- 数据结构在计算机中的表示
  - 数据元素的存储
  - 逻辑关系的实现
- 四种主要方式
  - 顺序存储
  - 链接存储
  - 索引存储
  - 散列存储

顺序存储

- 在连续的地址空间中，顺序存储各结点
  - 逻辑上相邻的结点在物理位置上也相邻
  - 逻辑关系由存储单元的邻接关系体现
  - 可实现随机访问
- 顺序存储一般借助数组来实现
- 主要用于线性数据结构

链接存储

- 把结点的数据和反映结点间关系的地址数据一并存储在计算机中
  - 存储单元可以不连续
  - 结点可以存储在任意的位置
- 每个结点所占存储单元分成两部分
  - 结点本身的数据
  - 地址数据，指出其后继或前趋元素的存储地址
  - 形成链状结构
- 结点间的逻辑关系由附加的地址数据体现
  - 一般借助指针来实现
  - 多用于动态数据结构的存储

索引存储

- 在存储元素信息的同时，还建立附加的索引表
- 索引表中的每一项为索引项，索引项一般形式是：（关键字、地址）
- 关键字是能唯一标识一个元素的数据项

散列存储

- 根据元素的关键字直接计算出该元素的存储地址
- 即在数据元素的字段中有一个或几个字段的值，通过一散列函数唯一地确定该元素的存储地址

三者之间的关系

- 逻辑结构抽象地反映了结点之间的关系
  - 同一逻辑结构可以采用不同的存储结构来实现
  - 不同逻辑结构可以采用相同的存储结构来实现
  - 逻辑结构决定操作运算
- 操作运算
  - 定义依赖于逻辑结构（做什么）
  - 实现依赖于存储结构（怎么做）
- 存储结构
  - 节点的内部表示
  - 逻辑结构的物理实现
  - 操作运算的实现基础

算法与数据结构

- 程序 = 算法 + 数据结构
- 算法与数据结构密切相关
  - 算法的设计取决于数据结构的选择
  - 数据结构的设计取决于算法的选择
    - 逻辑结构和操作取决于问题
    - 存储结构取决于算法

小结

- 数据结构的作用
- 数据与数据类型
- 数据结构的基本概念