数据结构概述

[TOC]

# # 1 基本概念

**数据结构**：相互之间存在一种或多种特定关系的数据元素的集合。包括三方面：**逻辑结构**、**存储结构**和**数据运算**。

# # 2 数据结构“三要素”

## ## 2.1 逻辑结构

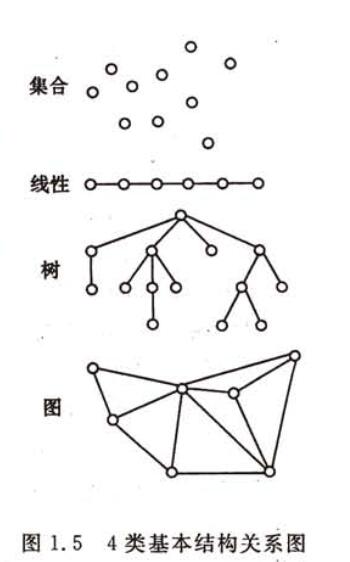
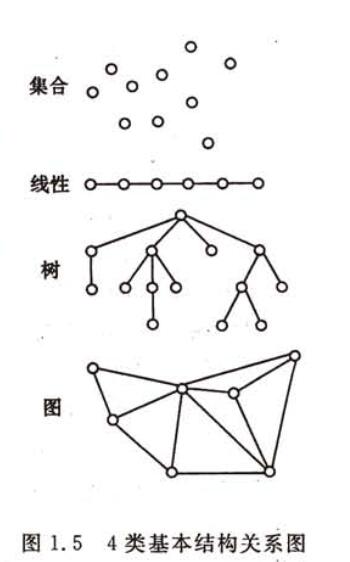
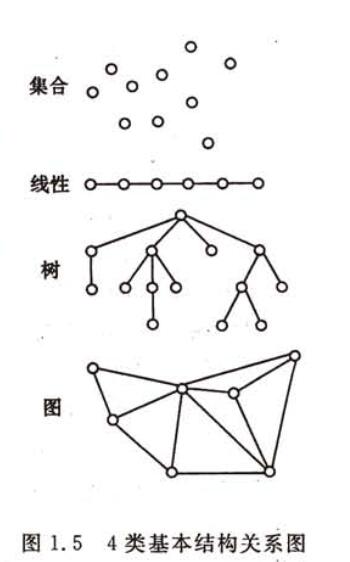
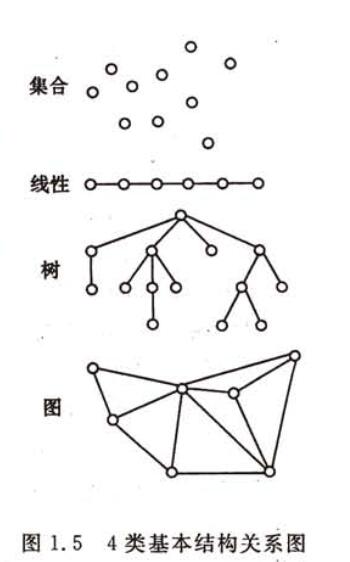
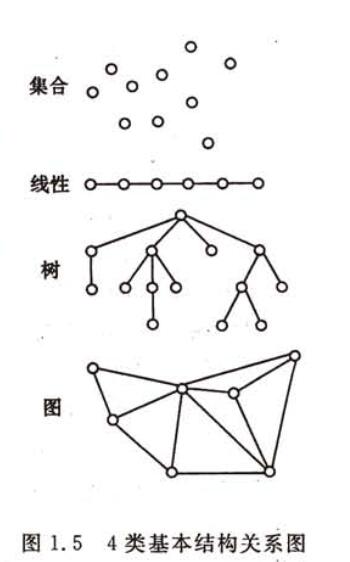
**逻辑结构**是描述数据元素之间的逻辑关系，它与数据的存储无关，是独立于计算机的。数据的逻辑结构分为**线性结构**和**非线性结构**，**线性表**是典型的线性结构；**树**和**图**是典型的非线性结构。数据的逻辑结构分类如下：

**线性结构**：数据元素之间存在“**一对一**”的关系。

**树形结构**：数据元素之间存在“**一对多**”的关系。

**图形结构**：数据元素之间存在“**多对多**”的关系。

**集合关系**：数据元素之间除了“**同属于一个集合**”的关系外，无其他关系。

****

## ## 2.2 存储结构

**存储结构**又称**物理结构**，是指数据结构在计算机的实际表示方式，用计算机语言的实现，依赖于计算机语言。数据的存储结构分类如下：

**顺序存储**：逻辑上相邻的结点存储在物理位置也相邻。**优点**：可以随机存储，每个结点占用较少空间。**缺点**：要占用相邻一整块存储空间，会产生外部碎片。

**链式存储**：不要求逻辑上相邻的结点存储在物理位置也相邻。**优点**：能充分利用存储空间。**缺点**：只能顺序存储，每个结点占用较多空间。

**索引存储**：存储结点要建立附加的**索引表**，索引表中的每一项叫**索引顶（关键字，地址）**。**优点**：检索速度快。**缺点**：占用额外的空间。

**散列存储**：也叫哈希存储（Hash），由结点的关键字通过**散列函数**直接计算该结点的地址。**优点**：检索速度很快。**缺点**：如果散列函数设计不好会出现存储单元冲突，而解决冲突会带来额外的时空开销。

## ## 2.3 数据运算

施加在数据上的运算包括运算的**定义**和**实现**。**运算的定义**是针对逻辑结构的，指出运算功能；**运算的实现**是针对存储结构的，指出运算的具体操作步骤。

# # 3 算法和算法评价

## ## 3.1 算法

**算法（algorithm）**是对特定问题的求解步骤的一种描述，它是指令的有限步骤。

### ### 3.1.1算法的特性

**有穷性**：执行有穷步骤后结束。区别于程序的死循环。

**确定性**：了叫无二义性，对于相同的输入只能得到相同的输出。

**可行性**：可以通过基本运算实现。

**输入性**：有0个或多个输入。

**输出性**：有1个或多个输出。

### ### 3.1.2算法设计要求

正确性，易读性(readability)，健壮性(robustness)，时空效率。

## ## 3.2 算法评价

算法效率的度量是通过算法的**时间复杂度**和**空间复杂度**来描述的。

### ### 3.2.1 时间复杂度

**语句的频度**是该语句在算法中被重复执行的次数。算法所有语句的频度之和记作**T(n)**，它是该算法问题规模n的函数，**时间复杂度**主要分析T(n)数量级。算法的基本运算（最深层循环内的语句）的频度与T(n)同数量级，所以采用算法中基本运算的频度**f(n)**来分析算法的时间复杂度。因此，算法的时间复杂度也记为：**T(n) = O(f(n))**。

上式中“**O**”的含义是T(n)的**数量级**，其数学定义：若T(n)和f(n)是定义在正整数集合的两个函数，则存在正常数C和n0，使得当n >= n0时，都满足0 <= T(n) < C\*f(n)。

算法的时间复杂度不仅依赖于**问题的规模n**，也取决于**输入数据的初始状态**（如各种排序算法）。

三类时间复杂度：

**平均时间复杂度**：所有可能输入实例在等概率的情况下，算法的期望运行时间。

**最好时间复杂度**：最好情况下，算法的时间复杂度。

**最坏时间复杂度**：最坏情况下，算法的时间复杂度。一般考虑最坏情况的时间复杂度，以保证运行时间不会比它更长。

分析一个程序时间复杂度的两个规则：

**加法规则**：T(n) = T1(n)+T2(n) = O(f(n))+O(g(n)) = O(max(f(n),g(n)))

**乘法规则**：T(n) = T1(n)\*T2(n) = O(f(n))\*O(g(n)) = O((f(n)\*g(n))

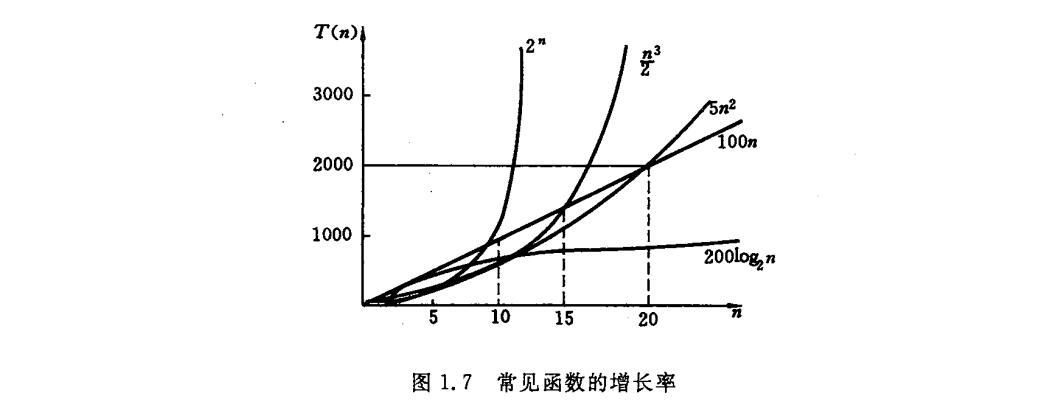
### ### 3.2.2 空间复杂度

**空间复杂度S(n)**：该算法所耗费的存储空间，它是问题规模n的函数。渐进空间复杂度也常简称为空间复杂度，记作**S(n) = O(g(n))**。

算法原地工作是指算法所需辅助空间是常量，即**O(1)**。

### ### 3.2.3 数量级比较

**O(1) < O(log2 n) < O(n) < O(n\*log n) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)**

****