Chapter 1 数据结构概论

数据

数据：信息的载体，是描述客观事物的数、字符，以及所有能输入到计算机中并被计算机程序识别和处理的符号的集合。

数据的基本单位是**数据元素**（data element），数据元素可由若干**数据项**（data item）组成

- 数据
  - 数据元素
    - **初等项**（不能分割的最小单位）
    - 组合项

数据结构

数据结构：由某一数据元素的集合和该集合中数据元素之间的关系组成

- 线性结构
  - 直接存取结构：数组
  - 顺序存取结构：链表
  - 字典结构：字典
- 非线性结构
  - 层次结构：树
  - 群结构：集合，图

讨论数据结构的三个方面：

- 逻辑结构：面向问题
  - 集合结构
  - 线性结构
  - 树形结构
  - 图形结构
- 存储结构：面向计算机，是逻辑结构的存储映像
  - 顺序储存：借助数组

    - 链接储存：借助指针
    - 索引储存：（关键码，地址）
        - 稠密索引：每个结点都有一个索引项
        - 稀疏索引：一组相邻的结点一个索引项
    - 散列储存：根据关键码通过函数计算得到地址
- 操作：算法

数据类型

类型：一组值的集合，可分为原子类型和结构类型

数据类型：一种类型以及定义于这个值集合上的一组操作的总称

抽象数据类型：由用户定义，用以表示应用问题的数据模型，由基本的数据类型组成，并包括一组相关的服务。使用与实现分离，实行封装和信息隐蔽

面向对象

Coad 和 Yourdon 给出定义：面向对象 = 对象 + 类 + 继承 + 消息通信

算法

算法：有穷指令集，为解决某一特定任务规定了一个运算序列

算法应具有如下特性

- 有输入：算法必须有 0 个或多个输入
- 有输出：算法必须有 1 个或多个输出
- 确定性：算法的每一步都应确切地、无歧义地定义
- 有穷性：算法无论在任何情况都将在执行有穷步后结束
- 能行性：算法的每一条运算都是足够基本的，原则上都能通过计算机指令精确执行。

算法不同于程序：程序可以不满足有穷性（操作系统等待用户指令）

算法的性能标准

- 正确性：算法要能正确地执行预定的功能和性能要求
- 可使用性：便于使用（API & 文档）
- 可读性：逻辑清晰、简单、结构化。有注释以及有意义的变量名/函数名
- 效率：空间开销以及时间开销

- 健壮性：对输入参数、打开文件、读文件记录、子程序调用状态等行为能够自动检错报错并通过用户对话来纠错（输入合法性检查）
- 简单性：采用数据结构和方法的简单程度

度量时间复杂度：程序步数

- 注释：0
- 声明语句：0
- 表达式：1，若包含函数调用，则总步数要包括函数调用的步数
- 赋值语句：等于表达式，若变量为数组或字符串，程序步数等于变量体积加上表达式步数
- 循环语句：仅考虑循环控制部分，while 与 do-while 结构为表达式的步数，for 循环第一次为初始化与判断语句的步数和，后续的为判断语句与递增语句的步数和
- switch 语句：首部为表达式的步数，case 子句为子句的步数加上前面条件判断的步数
- if-else：分别分配 if 语句块的步数和 else 语句块的步数
- 函数调用/执行：调用语句的步数为 0 ，开销计入执行语句，若参数为按值传递且值的大小与实例有关，则需加入这一部分的体积
- 动态存储管理语句：步数都为 1 ，需要注意 new 和 delete 隐式调用了对象的构造函数和析构函数，要用分析函数调用的方式分析其步数
- 转移语句：步数都为 1 ，注意 return 的为函数时要加入对函数调用步数的分析

渐近的时间复杂度

大 O 表示法