

数据结构-第一章-简述

Created by Yusheng Huang

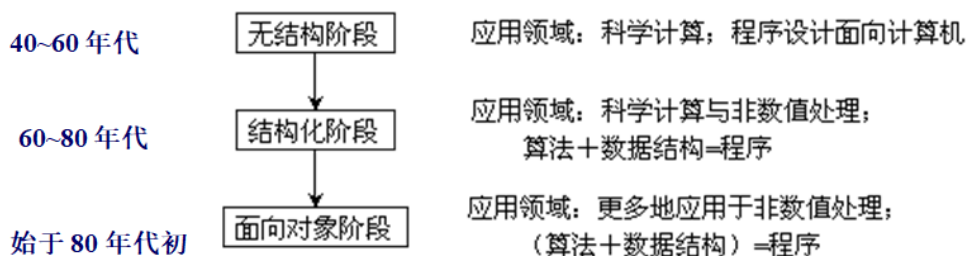
数据结构-第一章-简述

1. 数据结构发展三阶段
2. 数据结构
 - 2.1 数据
 - 2.2 数据元素
 - 2.3 数据项
 - 2.4 数据对象
 - 2.5 数据结构
3. 逻辑结构与物理结构
 - 3.1 逻辑结构
 - 四种基本逻辑结构
 - 3.2 物理结构（存储结构）
4. 抽象数据类型
 - 4.1 数据类型
 - 4.2 抽象数据结构（ADT）
 - 4.3 ADT三种实现方法
5. 算法
 - 5.1 算法的基本概念
 - 5.2 时间复杂度简述
 - 5.3 数据结构与算法简述
6. C语言定义结构体

1. 数据结构发展三阶段

二、数据结构随着程序设计的发展而发展

程序设计经历了三个阶段：无结构阶段、结构化阶段和面向对象阶段，相应地，数据结构的发展也经历了三个阶段：



2. 数据结构

2.1 数据

- 描述事物的符号，信息的载体，能够被计算机识别、存储和加工处理的对象。
- 数据，其实就是具备以下两个条件的符号：
 - ① 可以输入到计算机；
 - ② 能被计算机程序处理。

2.2 数据元素

- 描述特定事物不同属性的相关信息的总和
- 数据元素是一个整体

2.3 数据项

- 具有独立含义的最小单位
- 数据元素由数据项组成，一个数据项代表事物的一类属性

2.4 数据对象

- 相同性质的数据元素的集合
- 数据元素是数据对象的一个实例

2.5 数据结构

1) 数据结构两要素：

- 元素的集合
- 关系的集合

2) 数据元素的三部分：

- 数据
- 数据的关系
- 数据集合上的操作

3) 数据结构的基本操作：增删查改排

- 插入
- 删除
- 查找
- 修改
- 排序

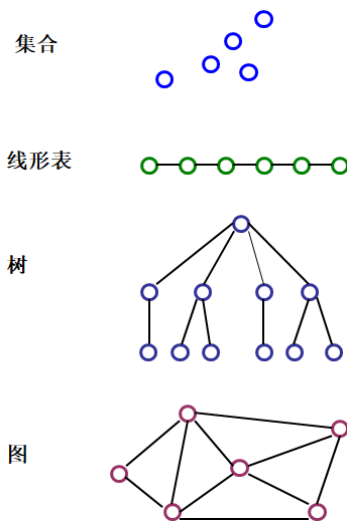
3.逻辑结构与物理结构

3.1 逻辑结构

- 数据的逻辑结构是指数据元素之间的逻辑关系。
- 逻辑结构有两个要素：数据元素集合、关系的集合。
- 在形式上，逻辑结构通常可以采用一个二元组来表示：
 - $\text{Data_Structure}=(D, R)$
 - 其中，D是数据元素的有限集，R是D上关系的有限集。

四种基本逻辑结构

- 集合-属于不属于
- 线性结构-一对一关系
- 树形结构-一对多关系
- 图状结构-多对多关系



3.2 物理结构（存储结构）

只有两种：

- 顺序存储
 - 顺序存储结构是把数据元素存放在**地址连续的存储单元中**，其**数据元素之间的逻辑关系和物理位置一致**。
- 链式存储
 - 链式存储结构是把数据元素存放在**任意的存储单元中**，这组存储单元可以连续也可以不连续。其**数据元素之间的物理位置不能反映其逻辑关系**，用指针来反映数据元素之间的逻辑关系。

4. 抽象数据类型

4.1 数据类型

- 数据类型是指一个值的集合和定义在这个值集上的一组操作的总称。
- 数据类型决定了数据占内存的字节数、数据的取值范围、可进行的操作。

4.2 抽象数据结构（ADT）

- 抽象数据类型(Abstract Data Type, ADT)是指一个数学模型以及定义在此数学模型上的一组操作。
- 是将“数据”、“结构”、“处理操作”封装在一起而形成的复合体。抽象数据类型实际上就是对数据结构的逻辑定义。

4.3 ADT三种实现方法

- 封装法：类--数据及其操作封装成一个整体，比如C++ 中的类。
- 分散法：**完全不封装**--将数据和处理数据的函数各自分开。无法从程序的物理结构(即代码的物理次序)上区分哪些数据和函数属于哪个ADT。
- 半封装法：**结构体封装数据，函数不封装**--将数据和为处理数据需要而定义的相关变量封装在一起形成一个结构，有关处理函数定义在结构之外。这种方法仅做到了对数据存储结构的封装，其特点介于封装法和分散法之间。

5. 算法

5.1 算法的基本概念

- 算法是对特定问题求解步骤的一种描述
- 特点：
 - (1) 有穷性：有限步骤之内完成,不能形成无穷循环。
 - (2) 确定性：每一个步骤必须有确定含义,无二义性。
 - (3) 可行性：操作可通过已实现的基本运算执行有限次完成。
 - (4) 输入： 有多个或0个输入。
 - (5) 输出： 至少有一个或多个输出。
- 要求：
 - 正确：没有语法错误，对所有可能的输入均能得到预期的结果
 - 可读
 - 健壮：不合法数据应被正确处理
 - 高效率、低存储：执行时间短，需要的存储空间小

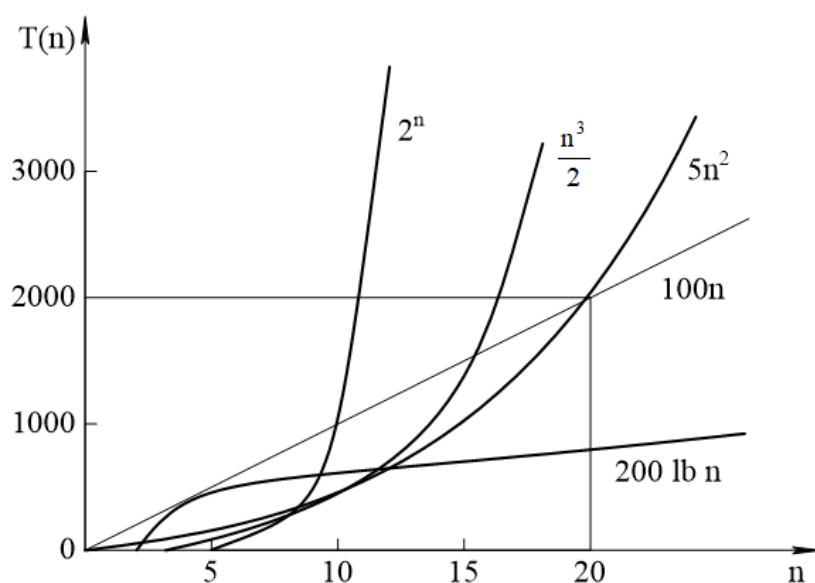
算法与程序的区别：

- 一个程序不一定满足有穷性。例如操作系统，只要整个系统不遭破坏，它将永远不会停止，即使没有作业需要处理，它仍处于动态等待中，因此操作系统不是一个算法。
- 程序中的指令必须是机器可执行的，而算法中的指令则无此限制
- 一个算法若用程序设计语言来描述，则它就是一个程序。

5.2 时间复杂度简述

常用的时间复杂度：常量阶 $O(1)$ 、线性阶 $O(n)$ 、平方阶 $O(n^2)$ 、对数阶 $O(\lg n)$ 。此外，算法还能呈现的时间复杂度有二维阶 $O(n \lg n)$ 、立方阶 $O(n^3)$ 、指数阶 $O(2^n)$ 、阶乘阶 $O(n!)$ 等，数据结构中常用的时间复杂度关系如下：

$$O(1) < O(\lg n) < O(n) < O(n \lg n) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)$$



5.3 数据结构与算法简述

2. “数据结构与算法设计”课程的主要内容

名称	特 点	存 储 结 构
线性表	数据元素之间的关系是一对一	顺序存储结构、链式存储结构
栈	插入操作和删除操作位置在线性表的一端进行，具有先进后出的特点	顺序栈、链栈
队列	插入操作在线性表的一端、删除操作在线性表的另一端进行，具有后进先出的特点	顺序队列、链队列、循环队列
串	数据元素是字符的线性表，操作对象是数据元素的序列	顺序串、块链结构、堆存储串
数组	线性表的推广，数据元素可以是一个表，如二维数组、三维数组.....	以行为主序或者以列为主序的顺序存储。特殊矩阵的压缩存储；稀疏矩阵的三元组表存储、十字链表存储
广义表	线性表的推广，数据元素可以是一个表，也可以是一个元素	广义表链式存储
树	数据元素之间存在一对多的关系，具有清晰的层次关系	双亲表示法、孩子表示法、孩子链表表示法、孩子兄弟表示法
二叉树	数据元素之间存在一对多的关系，具有清晰的层次关系。每个结点最多有两个孩子结点，而且有序	顺序存储结构、二叉链表存储结构、三叉链表存储结构
图	数据元素之间存在多对多的关系，也称网状结构	邻接矩阵、邻接表、十字链表、多重邻接表

(2) 查找与排序。

查找方法包含有：**静态查找表的查找方法(顺序查找、折半查找、分块查找)、动态查找表的查找方法(二叉排序树、平衡二叉树、B树)、哈希表的查找。**

排序方法包括：**简单的排序方法(简单选择排序、直接插入排序、希尔排序、冒泡排序)、先进的排序方法(快速排序、归并排序、堆排序)、基数排序方法。**

(3) 经典算法。

经典算法包括**分治法、贪婪法、回溯法、动态规划法。**

6. C语言定义结构体

C语言的结构体由一组称为结构体成员的项组成。
定义一个结构体类型变量由两步组成。第一步，定义结构体类型；第二步，定义结构体类型变量。

例如：

```
struct student
{ char name[20];
  float score[5];
  float average; };
struct student stu1,
stu2[100], *p;
```

或写作：

```
typedef struct
{ char name[20];
  float score[5];
  float average;
} studenttype ;
studenttype stu1,
stu2[100], *p;
```