

全国计算机二级考试（C++）复习提纲

第1章 数据结构与算法基础

1.1 数据结构概述

1. 数据结构的概念

数据结构（Data Structure，简称DS）是指相互之间存在一种或多种特定关系的数据元素（也称结点或节点）的集合，可用二元组： $DS=\{D,R\}$ 表示，其中D表示数据元素的信息，R表示各数据元素之间的关系。

2. 数据结构的研究内容

（1）数据的逻辑结构

数据的逻辑结构是指数据元素之间的逻辑关系，即数据元素之间的前后关系，包括：

- ①**集合**：数据结构中的元素之间除了“同属一个集合”的相互关系外，别无其他关系。
- ②**线性结构**：数据结构中的元素存在一对一的相互关系。
- ③**树形结构**：数据结构中的元素存在一对多的相互关系。
- ④**图形结构**：数据结构中的元素存在多对多的相互关系。

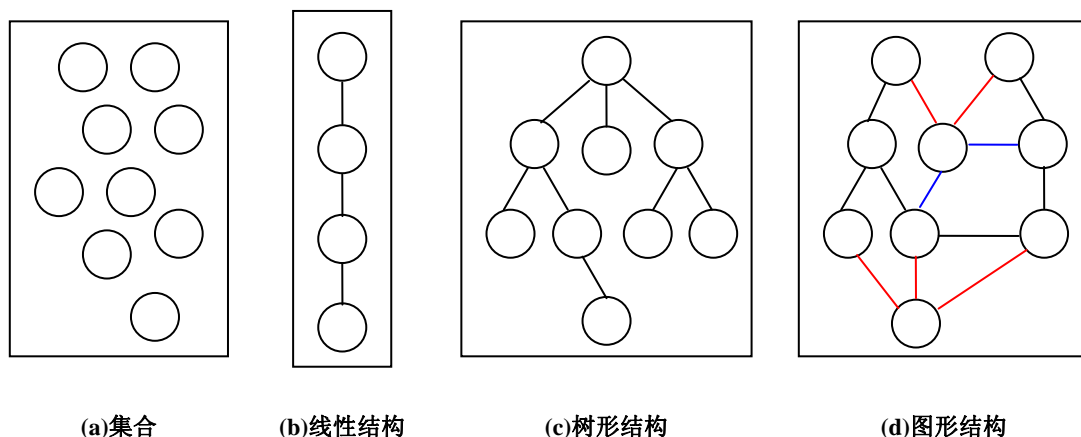


图 1-1 数据的逻辑结构

（2）数据的存储结构

数据的存储结构也称数据的物理结构，是指数据元素在计算机中的存储关系，即数据的逻辑结构在计算机存储空间中的存放形式，有顺序、链接、索引、散列等存储形式。

①**顺序存储**：把逻辑上相邻的结点存储在物理位置相邻的存储单元里，结点间的逻辑关系由存储单元的邻接关系来体现，即用存储关系表示逻辑关系。

②**链接存储**：逻辑上相邻的结点在物理位置上并不一定相邻，结点间的逻辑关系由附加的指针域表示，即用附加的指针表示逻辑关系。

③**索引存储**：除建立存储结点信息外，还建立附加的索引表来标识结点的地址，即用附加的索引表表示逻辑关系。

同一种逻辑结构的数据可以采用不同的存储结构，其数据处理效率可能不同。

（3）数据结点的运算

与数据的逻辑结构相对应的运算集合，主要包括输入、输出、查找（检索）、排序、插入、删除及更新等。

3. 数据结构的分类

数据的逻辑结构分为线性结构和非线性结构两大类型。

(1) 线性结构

非空的线性结构满足下列条件：

- ①有且只有一个根结点。
- ②每一个结点最多有一个前趋（驱）结点，也最多有一个后继结点。

通常情况下，第 1 个结点（也称首结点或根结点）没有前趋结点，有一个后继结点；最后 1 个结点（尾结点）没有后继结点，有一个前趋结点；其他结点有一个前趋结点和一个后继结点。常见的线性结构有线性表、栈、队列等。

(2) 非线性结构

不满足线性结构条件的数据结构，常见的非线性结构有树、二叉树和图等。

4. 顺序表

(1) 线性表的概念

线性表是由 $n(n \geq 0)$ 个数据元素组成的一个有限序列，线性表中数据元素的个数（ n ）称为线性表的长度，线性表可以为空表（ n 的值为 0）。

线性结构是一种逻辑结构，如 $(a_1, a_2, \dots, a_{n-1}, a_n)$ ；线性表是一种存储结构，有顺序存储和链接存储两种方式，其中：

- ①采用顺序存储结构的线性表称为顺序线性表，简称顺序表，如数组、串等，会溢出。
- ②采用链接存储结构的线性表称为线性链表，简称链表，只会下溢、不会上溢。

(2) 顺序表的基本特点

- ①顺序表中所有的数据元素所占的存储空间是连续的。
- ②顺序表中各数据元素在存储空间中是按逻辑顺序依次存放的。

$LOC(a_i) = LOC(a_1) + (i-1) * L$ ，其中 $LOC(a_i)$ 表示第 i 个数据元素的存储地址， $LOC(a_1)$ 表示第 1 个数据元素的存储地址， L 是每个数据元素占用存储单元的长度。通过地址可直接存取任意一个数据元素，此谓顺序存储，随机存取。

(3) 顺序表的运算

①插入运算：一般情况下，要在第 i ($1 \leq i \leq n$) 个元素之前插入一个新元素时，首先要从最后一个（即第 n 个）元素开始，直到第 i 个元素之间共 $n-i+1$ 个元素依次向后移动一个位置；移动结束后，第 i 个位置就被空出，然后将新元素插入到第 i 项。插入结束后，顺序表的长度增加了 1；在等概率情况下，插入运算平均需要移动 $n/2$ 个元素。

②删除运算：一般情况下，要删除第 i ($1 \leq i \leq n$) 个元素时，则要从第 $i+1$ 个元素开始，直到第 n 个元素之间共 $n-i$ 个元素依次向前移动一个位置。删除结束后，顺序表的长度减小了 1；在等概率情况下，删除运算平均需要移动 $(n-1)/2$ 个元素。

(4) 顺序表的缺点

- ①顺序表插入或删除数据元素时需要移动大量的数据元素，运算效率较低。
- ②顺序表的存储空间是在定义时确定的，极易造成浪费。
- ③顺序表的存储空间不是动态分配的，不便于扩充或减少。

5. 链表

(1) 链表的概念

线性表的链式存储结构称为线性链表，是一种物理存储单元上非连续、非顺序的存储结构。链表中，每个数据元素（结点）由数据域和指针域两部分组成，如图 1-2 所示；数据元素的逻辑顺序是通过链表中的指针链接来实现，如图 1-3 所示。

- ①数据域：用于存放数据元素的值。
- ②指针域：用于存放指针，指向该结点的前趋或后继结点。

数据域	指针域
<i>data</i>	<i>next</i>

图 1-2 链表的结点结构

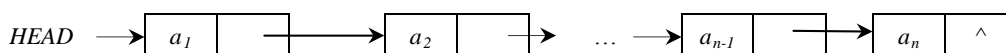


图 1-3 非空链表的逻辑与存储结构示意图

(2) 链表的类型

链表通常分为单链表、双向链表和循环链表三种类型。

①单链表：只有一个指针，如图 1-2 和图 1-3 所示。

②双向链表：每个结点的指针域包含两个指针，一个指向其前趋结点，另一个指向其后继结点。如图 1-4 和图 1-5 所示。

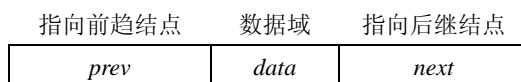


图 1-4 双链表的结点结构



图 1-5 非空双链表示意图

③循环链表：最后一个结点的指针域不为空，而是指向表头结点，图 1-6 所示。

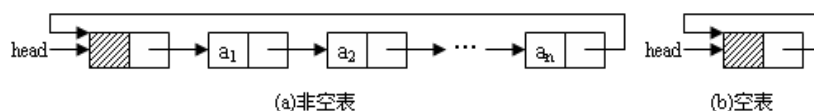


图 1-6 循环单链表示意图

(3) 链表的运算

①基本运算

插入结点：不需要移动数据元素，只需要修改相关结点指针，不会出现“上溢”现象。

删除结点：不需要移动数据元素，只需要修改相关结点指针。

链表的运算还包括查找、排序、合并、逆序等，但单链表的所有运算都必须从第 1 个（头）结点开始，依次向后找到要操作的结点，然后进行运算，此谓**链式存储，顺序存取**。

②双链表的运算

双链表的运算与单链表相比，每次运算都要修改指针域中两个指针的值；同时，双链表允许从后向前运算。

③循环链表的运算

为了实现空表与非空表运算的统一，循环链表中**增加了一个表头结点**（不是第 1 个数据元素），其指针域指向第 1 个数据元素，而循环链表的头指针指向表头结点。循环链表的优点主要体现在两个方面：只要指出表中任何一个结点的位置，就可以从它出发访问到表中其他所有的结点；任何情况下，循环链表中至少有一个结点（空表）。