

# 基本内容

2022年6月15日 23:00

数据结构：计算机组织和存储大量数据的方法

算法：处理数据的方法，和数据结构相辅相成

数据对象：具有相同性质的数据的集合（表、实体、数据集）

数据元素：数据的基本单位，通常看作一个整体来进行处理（记录、数据元）

数据元素由若干数据项组成

数据项：构成数据元素的最小单位，不可分割（数据域、列、字段、属性）

数据类型：一个值的集合和定义在此集合上的一组操作的总称

1、原子类型：其值不可再分的数据类型

2、结构类型：其值可以继续分解

顺序结构和链式结构既可以表示逻辑关系，也可以表示物理存储关系

数据结构三要素

1、逻辑结构：组织数据的方法，研究的是数据元素之间的逻辑关系

集合：任何数据元素之间没有关系，组织形式松散

线性结构：开始、终端结点均唯一，其中的每一个元素被称为结点，就像一条线一样将所有数据串联起来

树形结构：开始结点（也称根结点）唯一，终端结点不唯一，每个结点只有一个前驱结点

图形结构（网状结构）：没有开始、终端结点

2、物理结构（存储结构）：存储数据的方法，是具体在存储设备上的存储结构

顺序存储：排队占位，是连续的，查找迅速，但是删除数据等操作及其繁琐，时间复杂度高

链式存储：存储单元可以连续也可以不连续，只要有指针就能找到数据的位置，便于操作数据，如删除中间的数据，时间复杂度低，可以直接由前一个数据指向后一个数据，跳过中间的数据即可，但查找没有顺序结构快

索引存储：关键字+地址，通过建立索引表来查找元素，但是索引表也要占位置，且修改元素位置也要修改索引表

散列存储（哈希存储）：根据元素关键字（key）直接计算出该元素的地址

3、数据的运算（算法）：操作数据的方法

施加在数据上的运算包括运算的定义和运算的实现

运算的定义针对逻辑结构，指出运算的功能

运算的实现针对存储结构，指出具体的运算步骤

抽象数据类型（ADT）

一个数据模型及定义在该数据模型上的一组操作，如“整型”int就是一种ADT

ADT = 逻辑结构+数据运算的定义

是描述数据结构的一种理论工具，关注的是数据结构的特性，与物理结构、数据运算的实现均无关系，也与编程语言无关

可以用以下三元组表示：(D,S,P)

D表示数据对象，S是D中数据元素关系的集合，P是对D的基本操作的集合

算法：

有穷性

确定性

可行性

输入，0个或多个

输出，一个或多个

算法的效率

时间复杂度：执行算法所需要的计算工作量（关心数量级）

计算机运行一个算法时，程序代码被执行的总次数

空间复杂度：执行这个算法所需要的内存空间

算法消耗的内存空间，也是需要处理的数据量n的函数

后缀表达式：

我们一般使用的表达式都是中缀表达式，形象直观但是在计算顺序上可能存在歧义

后缀表达式原理：

ab+cde+\*+

从左往右开始扫描，一旦遇到运算符，就用运算符和运算符前面两个数进行运算

因此上方的表达式等价于中缀表达式：(a+b)+(d+e)\*c

运用后缀表达式，就算没有界限符如小括号，也能清晰地表达运算顺序

# 逻辑结构

2022年6月16日 9:48

## 集合

数组：一维数组、二维数组、多维数组.....

矩阵：可以用二维数组进行存储，对元素进行随机存取，其中每个元素也称为格点

## 线性结构

顺序结构

是最简单、最常用的一种数据结构

优点：可随机存取，存储密度高

缺点：要求一段连续的内存空间，扩容不方便，需要删除或移动其他元素

线性表是具有相同特征的数据元素的有限序列， $n$ 为表长， $n=0$ 时空表

注意序列不能理解为已排序

关键字（主键）：可用于区分不同元素的数据项

顺序表：

静态实现：数组

动态实现：动态分配内存

顺序表的合并：归并算法

栈（stack），是操作受限的线性表，是线性表操作的一个子集。普通的线性表可以在任何位置插入和删除，但栈仅限在一段进行插入（入栈）和删除（出栈）。只有在栈顶可以进行操作，栈底无法操作

其特点为后进先出（因为像瓶子一样只有一个口子），可以想象成瓶子里装石块，最上面的肯定先倒出来

注意有些特殊情况，比如一系列元素没有全部进栈，而是进一个出一个，那么之后元素的排列还是和原来一样

队（queue），是另一种操作受限的线性表，一段进行插入（入队），另一端进行删除（出队），特点是先进先出

**栈是桶，队是管子**

## 链式结构

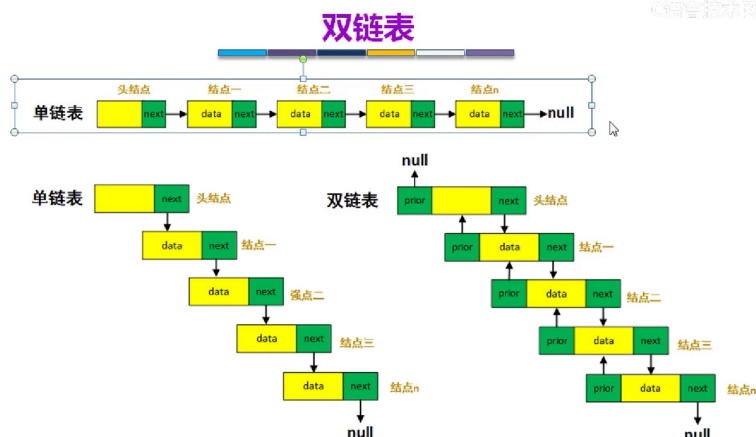
优点：不要求连续的内存空间，扩容方便

缺点：指针也需要消耗内存，且不能随机存取

单链表和双链表

头节点：只有指针而没有data的数据

带头结点和不带头结点（下图是双向但不循环链表）



prior (先前的)

双链表每个结点都有前置后置结点的指针

一般的单链表的最后一个结点的指针指向是空的，而循环单链表，最后一个结点的指针指向第一个结点的数据，循环双列表也有类似的原理

串 (字符串等) 的匹配: 判断是否相等

BP算法 (Brute Force): 普通的模式匹配算法, 对于两个串, 要判定它们是否相等, 或模式串是否包含在目标串中。是一种暴力算法, 其核心是“逐个比较”

KMP算法

## 树形结构

是非线性的数据结构, 是以分支关系的层级结构, 也就是有明显的层次结构

只有一个根节点 (最上面)

结点的度 (叉): 某个结点拥有的分支的数目

树的度 (叉): 所有结点的度的最大值, 以此命名为x叉树, 如树的度为3, 就叫三叉树, 表示树的度为3

父节点、子节点、兄弟节点。子节点还有左右之分

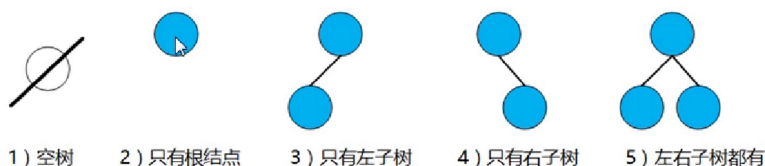
边: 一个由n个结点的树有n-1条边

树的层次: 多少层, 也称为高度

树的总结点数等于每个结点的度+1

二叉树

二叉树五种基本形态



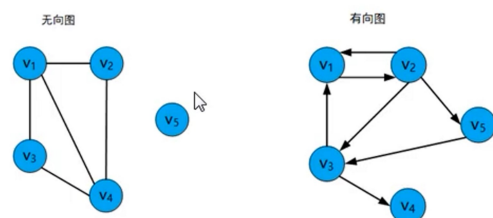
满二叉树: 有h层且结点个数为 $2^h - 1$

线索二叉树、二叉排列树 (BST) .....

哈夫曼树

## 图形结构

分为有向图和无向图



是一种比数据表和树更复杂的线性结构

链表中元素是一对一的关系; 树形结构中元素是一对多的关系; 而图形结构则是多对多的关系

图中每个元素称为顶点 (vertex), 两个元素之间的关系称为边 (必须有顶点, 但可能没边)

入度: 某顶点的入度即以该顶点为终点的有向边的个数

出度: 某顶点的出度即以该顶点为起点的有向边的个数

顶点的度: 该顶点入度与出度之和

全部入度之和=全部出度之和

路径：某个顶点到另一个顶点的序列

回路：路径终点与起点相同

路径长度：路径上边的个数

顶点与顶点的距离：最短路径长度，若没有路径则记为 $\infty$

无向图中，若两点之间有路径存在（不一定要直接相连），就说明两元素是连通的

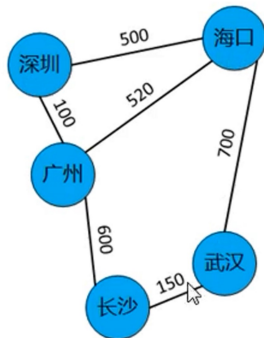
有向图中，若两点之间双向都有路径，就称为强连通

连通图：无向图中任意两点都是连通的，也就是没有孤立点

强连通图：有向图中任意两点都是强连通的

在一个图中，边可以表示某种含义的数值，如顶点之间的距离，该数值称为该边的权值。

图的边如果带上了权值，该图称为带权图，或网。



带权图中，某条路径上全部的边的权值之和，称为该路径的带权路径长度。

无向完全图：任意两个顶点都直接相连

有向完全图：任意两个顶点都直接双向连接

#### 图的存储

邻接矩阵

邻接表

十字链表

邻接多重表

#### 图的遍历

广度优先搜索BFS

深度优先搜索DFS

# 查找与存储结构

2022年6月16日 16:35

查找表：我们查找的对象，从中查找我们需要的元素

静态查找表：表中元素不会变化

动态查找表：表中元素会发生修改、删除、新增等变化

关键字key：数据唯一的标识，和id性质类似

查找：查找元素的过程，一般通过对比关键字来查找

查找长度：需要对比的关键字的次数

## 查找方式

顺序查找（顺序查找）：针对顺序表、链表，就是一个个对比

折半查找（二分查找）：针对有序的顺序表，其最坏情况时间复杂度最低

分块查找

## 哈希（Hash）表（散列表）

特点：数据元素的关键字和其在表中的存储位置直接相关

哈希函数：关键字与存储地址之间的对应关系  $Address = Hash(key)$

哈希函数是一种映射关系，很灵活

对于不同的关键字，可能得到相同的哈希地址，这种现象称为冲突。哈希地址相同的关键字称为同义字

处理冲突的方法

- 1、链地址法
- 2、开放定址法
- 3、再散列法
- 4、建立公共溢出区

哈希表的装填因子（表数据/表长）越大，冲突可能就越多，同义词越多，查找效率就更低

# 排序算法

2022年6月16日 16:53

## 以下经典排序算法均可用代码实现

### 交换类排序法：

冒泡排序法：从左往右依次比较相邻的两个数的大小，将大的放右边，直到将最大的数找到并放在最右边。然后再从左重复这一过程，即可从小到大排序，最坏需要比较  $n(n-1)/2$  次

快速排序法（有概率快，但最慢和上面的相同）：三个指针，L指向最左边的数，P指向最右边的数，R指向倒数第二个数。从L开始，比较L与P所指的数的大小，L比P小就不动（因为小的本来就要放左边）。直到L比P大时，改成判断P与R的大小，当R大于P时就不动，直到R比P小。这个时候，将L与R所指的数交换位置，然后继续重复以上步骤。当R与L指向同一个数时，将这个数和P指的数换位置，这个时候中间值就被找到了。然后分别处理左边和右边的数据

### 插入类排序法：

插入排序法：将第二个数拉出列，然后依次比较。如果比左小就往左走，然后再比左小就继续往左，直到比左大，然后插入正确的位置，以此类推。与冒泡不同的是，这个是以某个数为导向的

希尔排序法：最坏排序次数 $n$ 的1.5次方

### 选择类排序法：

选择排序法：扫描整个序列，找到最小值，然后和第一个元素交换，再接着找次小值，将其移到第二个位置，依次类推

堆排序法：最坏排序次数 $n \cdot \log_2 n$ （最快）

此外，还有桶排序、归并排序、基数排序、计数排序等排序方法

除了希尔排序和堆排序法，其他四种排序最坏次数完全一致，均为  $n(n-1)/2$