1.定义

大量复杂的问题以特定的数据类型(单个数据怎样存储)和特定的存储结构(个体关系),保存到内存中,

数据结构就是(数据类型加存储结构)

在此基础上,为实现某个功能,(对数据进行增删改查的操作,排序)而执行的操作,就叫做算法,

理解:15个人一个数组,就可以搞定,10000个.内存中没有这么多连续的空间,所以就改用链表.认识管理系统的存储,这时就需要树,又比如交通图,就需要图,图就是每个及诶点都可以和其他节点产生联系.

数据结构=个体与个体的关系

算法=对存储数据的操作

算法的标准:

时间复杂度,大概程序要执行的次数,而非执行的时间;

空间复杂度,算法执行过程中所占用的最大内存;

难易程度,

健壮性

结构体,自定义的符合数据类型,(理解常用的model类),

java中局部变量必须初始化,这些变量都是垃圾值,,属性不必须初始化,因为已经默认初始化为0;

动态分配的内存一定要手动释放,否则会造成内存泄漏.

-------------------------

连续存储:数组(元素类型相同,大小相等),读取速度快;事先必须知道数组的长度,插入删除速度很慢,空间有限制,需要大块连续的内存块,插入删除元素的效率低,

离散存储:链表,

定义:n个节点离散分配,彼此通过指针相连,,每个节点只有一个前驱节点,每个节点只有一个后续节点,

首节点,尾节点,头结点(没有存放有效数据只是方便对链表的操作),

头指针,指向头结点的指针变量.

头结点有可能很大,占的内存可能大,假设做一个函数输出所有链表的值,如果不用头指针作为形参,由于不同链表的头及诶单不一样大,没办法找出形参.

确定一个链表只需要一个头指针,可以推出链表的所有信息.

单链表:

双链表:每个节点有两个指针域

循环链表:能通过一个节点找到所有的节点.

非循环链表:

链表:空间没有限制,插入删除云元素很快,缺点,读取速度很慢,

算法:遍历,查找,清空,销毁,求长度,排序,删除节点,插入节点,

狭义的的算法,与数据的存储方式密切相关

广义的算法与数据的存储方式无关,

泛型:不同的存储方式,执行的操作是一样的

线性结构的两种常见应用之一,

栈:(存储数据的结构)静态栈(类似于数组实现)动态栈(类似于链表实现)

树:有且只有一个称为根的节点,有若干个互不相交的子树,这些子树本身也是一根树

父节点,子节点,节点,深度(从根节点到最底层节点的层数)

叶子节点(最底层的节点),非终端节点(非叶子节点),度(子节点的个数)

树的分类:

一般树:任意一个节点的子节点的个数都不受限制

二叉树(有序树):任意一个节点的子节点最多两个,且子节点的位置不可更改

分类:一般二叉树:

满二叉树:在不增加层数的前提下,不能在增加任何节点,

完全二叉树:如果只删除了满二叉树最体层最右边的连续的若干节点,这样的二叉树就是完全二叉树,

森林:n个不交叉的树的集合

一般的二叉树要以数组的方式存储,要先转化成完全二叉树,因为如果只存有效节点(无论先序,中序,后序)则无法直到这个树的组成方式是什么样的.

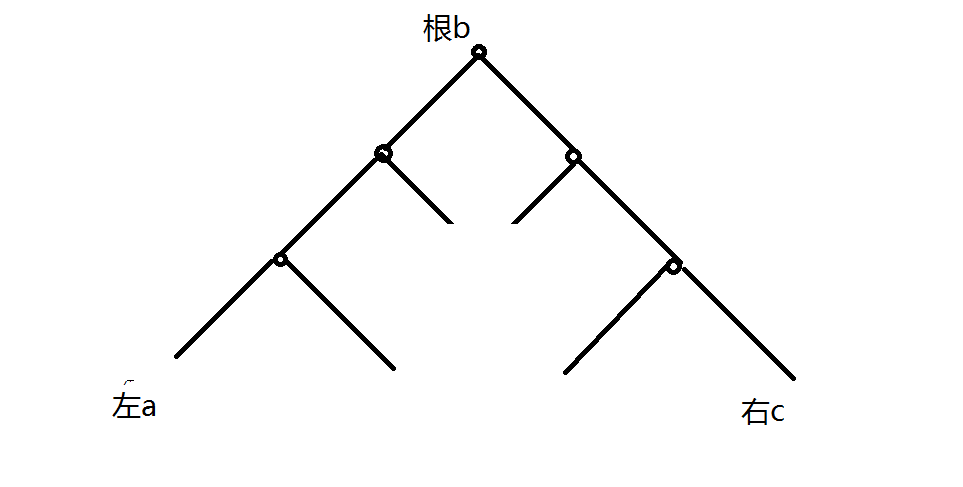
树的存储一般都是转化成二叉树来存储的

二叉树的存储:

连续存储:(完全二叉树)优点:查找判断很快,但是耗用内存空间大,

链式存储:

二叉树遍历



先序:先根节点,后左子树,最后右子树BAC

中序:先左,后根,最后右ABC

后序:先左后右最后根,ACB

知道中序和其他任意一个可以求出整个二叉树

数据结构研究的是,数据的存储和数据的操作

数据的存储分为两个部分:个体的存储和个体关系的存储.