绪论

「定义:数据结构就是**关系**,就是数据元素相互之间存在的一种或多种特定关系的集合。

分类:

图形结构: 如图

定义:解决特定问题求解步骤的描述,在计算机中表现为指令的有限序列,并且每条指令

表示一个或多个操作。

描述方法: ADT 算法

特点: ①输入(至少0个)②输出(至少1个)③有穷性④可行性⑤确定性

衡量一个算法: 复杂度(时间复杂度、空间复杂度) O(f(n))

线性表

顺序表: 空间固定, 用数组储存(物理结构和逻辑结构一致)

链式表: 用动态分配储存(物理结构和逻辑结构不 静态链表:通过数组的索引操作

(忧点: 访问、查找方便 (缺点: 增删复杂(牵一发而动全身) 链式表 (忧点: 增删方便,对稀疏性数据节省空间 缺点: 访问复杂(从表头开始一直往后找)

本章核心: 对指针操作需十分熟悉, 对动态链表操作时, 需想办法记住上一个结点, 故顺序

创建链表

队列

栈和队列(本质:一种操作受限的线性表)

'先进后出(滞后思想),只能对栈顶元素进行操作

常用操作: Get Top(), Pop(), Push()

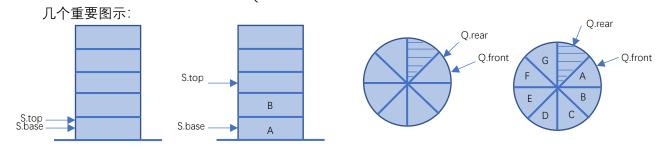
判断栈空: S.top == S.base

【注意:S.top 指向的不是栈顶元素,而是下一个可存放的位置

〔先进先出,队尾增加,队头删除

问题的产生:假溢出现象 ====== 循环队列

、 取模(如同钟表一样,12 点即为 0 点,即 time%12) Q.front == Q.rear 表示空还是满? ⇒ 空一格做标识



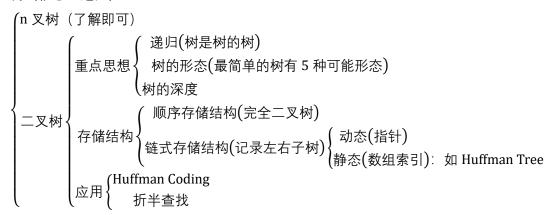
数组和广义表 (不做重点)

①n 维数组的索引: 第一维*n₁+第二维*n₂+······+第 n 维*m

注意: 在计算机中不存在维度的说法,任何数组均为线性存储结构,N 维为逻辑结构。

②广义表(不考)

树(核心: 递归)



Q、为什么 Huffman Tree 使用静态链表存储而不用动态链表?

A: 由于 Huffman Tree 节点个数已知,故用表格形式存储更加方便。

冬

查找

「静态查找:只读 动态查找: 读写

(哈希函数: 关键字与地址间——对应的关系 f(key) (哈希冲突: f(key1) = f(key2)

(条件:建立在有序表的基础上查找

排序

二路归并排序: 会画图

冒泡排序: 会画图 插入排序: 会画图

【快速排序(略):分堆思想