

数据结构

笔记本： 我的第一个笔记本

创建时间： 2020/7/16 14:30

更新时间： 2020/9/9 19:53

作者： 1272209351@qq.com

牛客网; leetCode (力扣) ----- 题库

数据结构 (DS)

list 链表

seqlist 顺序表

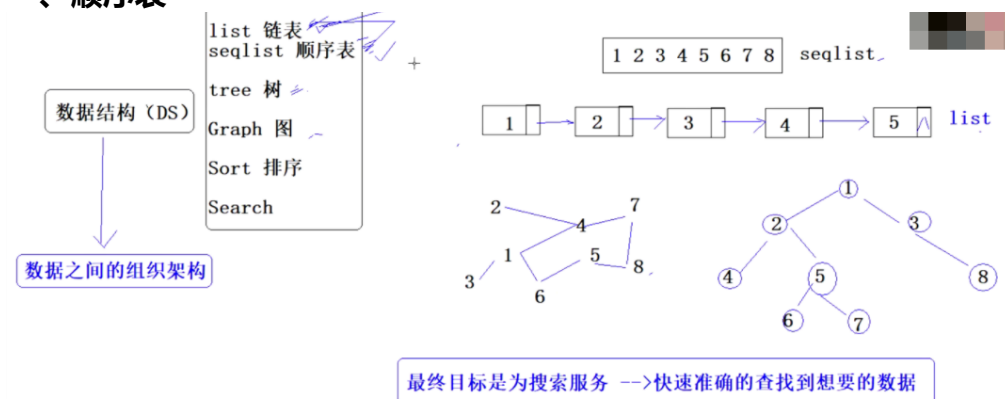
tree 树

Graph 图

Sort 排序

Search 查找

一、顺序表



数据结构----->

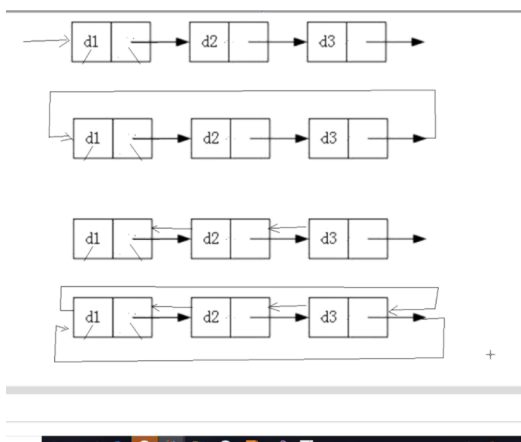
线性（顺序）结构：seqlist (顺序表)、list(链表)、stack(栈)、queue(队列)

非线性结构：tree <binary tree> (树)、graph (图)

数组（顺序表）：连续的内存空间

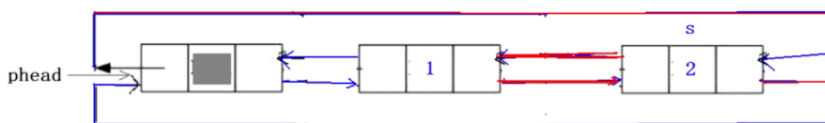
链表：逻辑连续但是物理不连续的内存

二、链表



$$O(1) > O(\log) > O(n) > O(n^2)$$

双向循环链表：



3 4 5

```
s->prev = head->prev;
s->next = head;
head->prev->next = s; //s->prev->next = s
head->prev = s;
```

顺序表

空间连续 访问速度快 插入和删除数据速度慢

链表

空间不连续 访问速度慢 插入和删除数据速度快

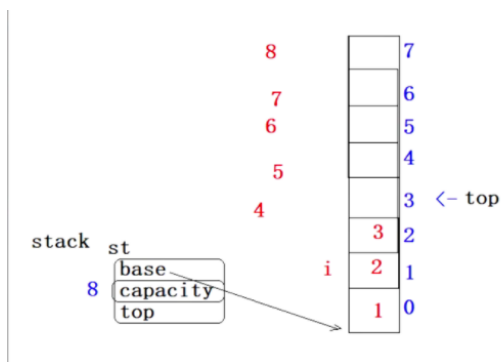
三、栈和队列

队列：先进先出（队尾：插入；队头：删除）

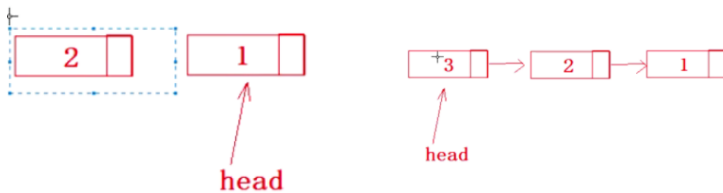
栈：栈顶（插入删除）、栈底

后进先出

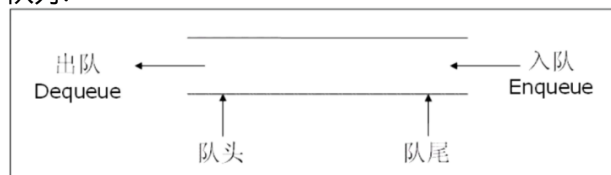
顺序栈：



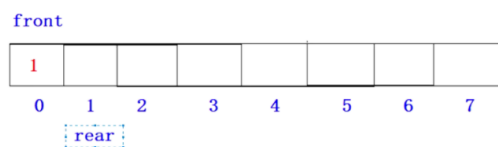
链栈：
从头部删除插入



队列：

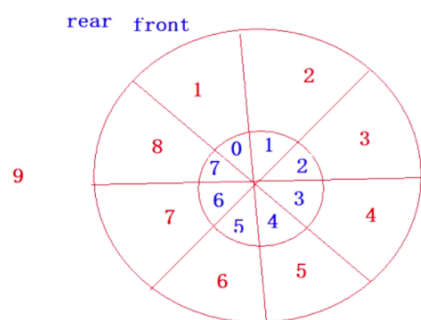


顺序队：



2 3 4 5 6 7 8 9

循环队列：



复习：

数据结构+算法

DS=? 研究数据之间的组织架构

查找：线性结构、非线性结构、排序

学会计算时间复杂度和空间复杂度===》大O渐近法

$O(1) > O(\log n) > O(n) > O(n^2)$

$O(n^3)$ $O(n!)$ 重构代码

顺序表 VS 链表:

- 1、写代码实现顺序表和链表
- 2、<单链表、单循环链表、双链表>不带头结点
插入和删除的时候需要单独考虑第一个结点
<双循环链表>带头结点
- 3、**理解顺序表和链表的优缺点，异同点-----考点**
顺序表空间连续、链表空间不连续
- 4、课件上OJ 题

栈和队列:

- 1、掌握概念
- 2、能够代码实现栈和队列-----顺序、链式，循环队列
- 3、课件上OJ 题

二叉树:

- 1、理解概念及二叉树的性质（五个性质）
- 2、代码实现二叉树（创建、遍历（递归、非递归）、求解二叉树）
先序遍历：根左右
中序遍历：左根右
后序遍历：左右根
层次遍历、广度深度遍历
- 3、二叉树的恢复
- 4、左孩子右兄弟
- 5、堆的理解和堆的实现，重点在于堆的调整（向上和向下调整）
- 6、OJ题

二叉树具有下列重要特性。

性质 1 在二叉树的第 i 层上至多有 2^{i-1} 个结点 ($i \geq 1$)。

性质 2 深度为 k 的二叉树至多有 $2^k - 1$ 个结点, ($k \geq 1$)。

由性质 1 可见,深度为 k 的二叉树的最大结点数为

$$\sum_{i=1}^k (\text{第 } i \text{ 层上的最大结点数}) = \sum_{i=1}^k 2^{i-1} = 2^k - 1$$

性质 3 对任何一棵二叉树 T , 如果其终端结点数为 n_0 , 度为 2 的结点数为 n_2 , 则 $n_0 = n_2 + 1$ 。

设 n_1 为二叉树 T 中度为 1 的结点数。因为二叉树中所有结点的度均小于或等于 2, 所以其结点总数为

$$n = n_0 + n_1 + n_2 \quad (6-1)$$

完全二叉树将在很多场合下出现, 下面介绍完全二叉树的 i

性质 4 具有 n 个结点的完全二叉树的深度为 $\lfloor \log_2 n \rfloor + 1$

性质 5 如果对一棵有 n 个结点的完全二叉树(其深度为 $\lfloor \log_2 n \rfloor + 1$)的结点按层序编号(从第 1 层到第 $\lfloor \log_2 n \rfloor + 1$ 层,每层从左到右),则对任一结点 $i (1 \leq i \leq n)$,有

(1) 如果 $i=1$,则结点 i 是二叉树的根,无双亲;如果 $i>1$,则其双亲 $\text{PARENT}(i)$ 是结点 $\lfloor i/2 \rfloor$ 。

(2) 如果 $2i>n$,则结点 i 无左孩子(结点 i 为叶子结点);否则其左孩子 $\text{LCHILD}(i)$

排序:

1、排序的分类

时间的复杂度: 简单、先进

根据重复元素: 稳定、不稳定

手段: 插入、交换、选择、基数

shell (希尔排序) quick (快速排序) heap 二路、分发回收

2、徒手写出具体的某种排序