

第一章

- 1、四类数据结构，集合、线性表、树、图，线性表、树、图元素之间的关系分别为一对一，一对多，多对多的关系。
- 2、算法的效率分为时间效率和空间效率。
- 2、计算某一条语句执行次数或某段程序段的时间复杂度。
- 3、时间复杂度计算，与后面的一些算法结合，比如排序算法、查找算法的时间效率及比较。

第二章 栈

- 1、 栈的 ADT（抽象数据类型）定义和基本操作。
- 2、 栈的特点（后进先出表）、性质（LIFO, overflow, underflow, push, pop 后栈的状态）
- 3、 栈的类定义及顺序实现（包括各个方法的具体实现）
- 4、 栈的应用：数据逆置算法、括号匹配的判断、后缀表达式求值算法
- 5、 根据中缀写出后缀表达式，前缀、中缀和后缀的含义

第三章 队列

- 1、 掌握队列的抽象数据类型定义和基本操作、扩展的队列操作。
- 2、 队列的特点、性质（先进先出表,队尾入队、队头出队后不改变原序列）
- 3、 队列的类定义及顺序实现，顺序队列产生的假溢出问题！如何解决？

- 4、 利用循环队列产生的问题？有哪些解决方案？
 - 5、 循环队列实现算法。（包括各个方法的具体实现）
- 利用栈和（或）队列的方法，编写算法。如判断回文等。

第四章 链栈和链队列

- 1、 链表结点类定义 p 123
- 2、 不使用safeguards的链栈（链队列，链表），可能会产生的一些问题剖析！如垃圾的累积，破坏封装特性等问题！举例说明？ P131-136
- 3、 链栈类定义、具体实现（包括各个方法的具体实现）
- 4、 链队列定义、具体实现（包括各个方法的具体实现）

第五章 递归

- 1、 递归概念，栈与函数调用和递归，用栈存放每一层函数的“调用记录”，简单递归算法实现

第六章

- 1、 线性表的概念和基本操作,为什么将线性表类设计成模板类？
- 2、 顺序线性表下元素的存储，类定义和算法。
- 3、 两种单链表下元素的存储，类定义和算法，插入和删除时需定位被删结点的前驱结点。

- 4、 双向链表下的算法。
- 5、 掌握串的概念和基本操作。空串？ P236-240,构造函数的实现，利用c串实现strcat， strncpy, strstr等操作。
- 6、 顺序表、链表下的算法设计，一般要求直接访问表元素，不调用其他方法。

第七章

- 1、 各查找(顺序查找/两种二分查找)算法的递归算法和非递归算法实现
- 2、 画比较树，利用比较树分析平均比较次数ASL
- 3、 查找算法效率分析和比较，

查找方法	存储结构要求	时间效率	优缺点	查找原理
顺序查找	顺序结构或链式结构的线性表	$O(n)$	当 n 较大时, 查找比较耗时	从线性表的头部对每个记录依次进行比较, 直至找到一个与目标关键字相同的记录, 或直到表尾都无法找到, 则为失败的查找。
二分查找	顺序存储结构、元素已有序	$O(\lg n)$	需要事先保持记录有序	按照逐渐缩小被查区间的方法进行查找。
二叉查找树查找	二叉链表结构, 记录按照二叉查找树的要求进行排序	在理想状态, $O(\lg n)$	二叉查找树的形态影响其查找性能	在二叉查找树下进行查找, 如与根结点值相同, 则查找成功; 若二叉树为空, 则查找失败; 否则按照值的大小到左子树或右子树上进行查找。
哈希查找	哈希表	理想状态为 $O(1)$, 实际与装载因子、解决冲突等因素有关。	需要解决哈希函数选择、冲突解决、装载因子的选择等问题	在哈希表下, 根据哈希函数的计算和冲突的解决方法, 进行关键字对应记录的存储和查找。

4、以关键字比较为基础的查找算法的最好性能 lower bounds

$O(\lg n)$

第八章 排序

1、 各种排序算法的手工执行过程、效率、稳定性分析及性能比较（重点是：插入排序、选择排序、归并排序、基数排序、快速排序、堆排序），根据实际情况选取合适的排序算法,lower bounds $O(n \lg n)$

排序方法	时间复杂度			辅助存储空间	数据表存储结构	适用场合	稳定性
	平均情况	最坏情况	最好情况				
插入排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	顺序/链式	数据基本有序	稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	顺序	每个数据记录较大，移动记录较花时间。性能与初始序列无关。	稳定
快速排序	$O(n \lg n)$	$O(n^2)$	$O(n \lg n)$	$O(\lg n)$	顺序	初始数据存放于顺序表，数据大小随机，当元素顺序或逆序，且选首（尾）元素为枢轴，产生最坏情况	不稳定
堆排序	$O(n \lg n)$	$O(n \lg n)$	$O(n \lg n)$	$O(1)$	顺序	性能比较均衡。可扩展使用在只需找出最大（最小）的几个数据或将已有数据按大小分成两个部分的场合。	不稳定
归并排序	$O(n \lg n)$	$O(n \lg n)$	$O(n \lg n)$	$O(1)/O(n)$	链式/顺序	初始数据存放于链表的最快速算法	稳定
基数排序	$O(nk)$				链式	关键字的长度基本相同，如位数相同的若干正整数的排序	稳定

(1) 选择排序的效率与初始状态无关。比较次数为 $n(n-1)/2$, 移动次数为 $3(n-1)$ 。

插入排序的效率与初始状态有关, 最好情况是当原始数据基本有序时, ($n-1$ 次比较, 0 次移动)。

(2) 由于减少了移动次数, 当记录元素较大时, 选择排序比插入排序更为合适。而记录元素较小, 移动花费时间不多时, 插入排序由于比较次数少, 会更好。

(3) 选择排序适用于顺序结构。

2、以关键字比较为基础的排序算法的最好性能 $O(n \log n)$

3、快速排序概念, 快速排序的过程, 计算总的比较次数。

3、堆的判别、堆的建立, 堆排序的过程, 与后面完全二叉树的内容有联系

4、归并排序的算法, 排序过程, 总的比较次数。

第九章 表格

1、 各种表格的存储 (普通二维数组, 特殊的矩阵), 二维表格映射到一维存储时的下标函数及访问数组的求法。

2、 基数排序过程及时间效率

3、 (重点) 哈希函数 (计算简单, 均匀散列), 解决冲突的几种方法: 开放定址法 (线性探测、二次探测等), 链表法。

如何构造哈希表, 哈希查找理想情况下的时间效率 ($O(1)$),

实际情况的效率与哪些因素有关(冲突解决策略、装载因子),
装载因子的概念, 平均查找长度的计算

第十章 二叉树

- 1、 二叉树概念
- 2、 求解二叉树的前序、中序、后序遍历,根据2个序列构造二叉树。
- 3、 二叉树的链式类定义及递归遍历算法、层次遍历等算法的实现
- 4、 二叉查找树的定义、判别、特点: 中序序列为递增序列、,
- 5、 二叉查找树下的查找算法实现、效率, 越矮的二叉树越有利于查找。
- 6、 二叉查找树下结点插入的方法
- 7、 二叉查找树下结点删除的方法
- 8、 二叉查找树的建立
- 9、AVL树的判定, 平衡因子。
- 11、补充的几个性质
 - (1) i 层上最多有 2^{i-1} 个结点
 - (2) 二叉树高度为 h , 至多 2^h-1 个结点, 至少为 h 个结点(设空二叉树高度为0, 仅含一个结点的二叉树高度为1)
 - (3) $n_0 = n_2 + 1$
 - (4) 具有 n 个结点的完全二叉树的高度为 $\lfloor \log_2 n \rfloor + 1$

(5) 若对含 n 个结点的完全二叉树从上到下且从左至右进行 1 至 n 的编号, 则对完全二叉树中任意一个编号为 i 的结点:

(1) 若 $i=1$, 则该结点是二叉树的根, 无双亲,

否则, 编号为 $\lfloor i/2 \rfloor$ 的结点为其双亲结点;

(2) 若 $2i > n$, 则该结点无左孩子,

否则, 编号为 $2i$ 的结点为其左孩子结点;

(3) 若 $2i+1 > n$, 则该结点无右孩子结点,

否则, 编号为 $2i+1$ 的结点为其右孩子结点。

补充:

满二叉树: 高度 h , 2^h-1 个结点的二叉树

完全二叉树: 满二叉树基础上从右下方开始连续地删除若干个结点。高度 h , 结点个数的范围: $2^{h-1} \text{ --- } 2^h-1$ 个

二叉树 (包括二叉查找树) 的递归算法的实现

第十一章 (多路) 树

1、 自由树的概念 (连通, 无环, n 个顶点, $n-1$ 条边), 介于不连通和有回路图的中间状态、森林的概念

2、 森林、树与二叉树之间的相互转换

3、 森林、树与二叉树遍历序列之间的关系

第十二章 图

1、 概念, 图, 有向图, 无向图, 连通图 (connected), 连通图的最少最多边数, 强连通图的最少最多边数, 带权图 (网) network, 顶点的入度, 顶点的出度, 所有顶点的度数之和是

边数的2倍。

- 2、 图的两种存储方案：邻接矩阵、邻接表表示，在这两种结构下求解图中顶点的出度、入度等的方法。
- 3、 图的深度优先遍历（树的先序遍历推广）和广度优先遍历（层次遍历推广，用队列辅助完成算法）算法过程的理解。
- 4、 拓扑排序的实际意义和深度优先拓扑序列和广度优先拓扑序列手工求解过程。
- 5、 什么是最短路径？求解方法
- 6、 生成树（含有无向连通图中全部的 n 个顶点， $n-1$ 条边，连通，无环的子图），最小生成树的概念，prim算法求解过程。

关于名词解释

- （1）数据结构：栈、队列、线性表、串、二叉树、树、森林、图
- （2）常见操作：查找（顺序查找、二分查找）、排序（快速排序、归并排序）、遍历(前序、中序、后序)
- （3）概念和术语：度、入度、出度、连通、强连通、邻接点、邻接矩阵、邻接表、哈希表、空串.....
- （4）图中的应用：生成树、最小生成树、最短路径、拓扑排序....
- （5）递归、抽象数据类型。。。。