

2019 年数据结构复习提纲 2019.12

- 考试形式：闭卷

- 考试题型及分值分布：

- 1、选择（10 分）、2、判断（10 分）、3、简答（36 分）、4、程序填空（10 分）、5、程序题（程序阅读、编程）（34 分）

(一)线性部分：要求掌握到会编程

1. 列表：

- a. 基于数组（静态数组，动态数组）实现的列表；基于链表（单向链表）实现的列表
- b. 列表上的操作：添加元素（在指定位置插入）、删除元素（删除指定位置的元素）、访问元素、查找元素（无序：顺序查找；有序：折半查找/二分查找，不支持链表存储模型）
----各种操作的原理，复杂度，程序实现，并能对一些衍生的应用需求进行编程

c.链表有变形：带头结点（哨兵）的链表、单向循环链表、双向链表、双向循环链表、带尾指针的链表----各种链表的样子，会描述（画），知道各种链表的特点，优点。这部分了解。

d.基于列表的排序：**冒泡，选择，插入，希尔，基数，快速，归并，非线性结构中掌握堆排序（依赖线性存储-数组实现）**，其中**粗体**为历年考查频率比较高的----排序原理（会模拟排序的全过程），复杂度，程序实现(数组、单向链表)

2. 栈：

- a. 基于数组/链的实现；结构特点（先进后出），操作受限（添加/删除 均在栈顶）
- b. 栈的应用：表达式计算（中缀转换成后缀；计算后缀表达式）----理解栈在其中扮演的角色，会模拟全过程。

3. 队列：基于数组/链的实现，特别考虑循环队列，结构特点（先进先出），操作受限（添加在队尾；删除在队首---考虑循环队列的处理），循环队列空满的判断（**牺牲一个元素法**）

4. 线性结构在 STL 中：vector/stack/queue 能做到读代码，在编程中使用，即明白有哪些成员操作，怎么进行声明和初始化构造。

(二)非线性部分：要求掌握到能读懂代码，部分简单的实现会编程。

1. 树/二叉树：

- a. 森林/一般树和二叉树之间的转换关系；
- b. 二叉树是有序树；递归定义的结构；
- c. 二叉树的术语/几种特例（满二叉（树），完全二叉（树），二叉查找树，AVL 树，堆（最大堆））；二叉树的一些结论（**满二叉：叶子节点数=中间节点数+1；完全二叉：最适合用数组存储**；第 i 层最多存放的节点个数？n 层二叉树最多存放的节点个数？n 个节点的二叉树最高/最矮？3 个节点的二叉树的形态？）；
- d. 二叉树的存储表示（数组，链节点）；二叉树的遍历（4 种：前，中，后，层次），其中前、中、后序遍历掌握递归代码的实现；
- e. 二叉查找树 BST 的操作：遍历（中序遍历为有序）-掌握到代码；元素查找-掌握到代码；

元素添加（总是成为叶子）-掌握到代码；元素删除（3 种情况，其中有两个子树的要特别注意）-----掌握各种操作的复杂度，原理，能模拟，能读懂代码

- f. 二叉查找树的平衡→AVL 树：平衡因子--会计算每个节点的平衡因子；在添加节点过程中进行平衡调整—4 种调整方法（左旋/右旋/左右旋/右左旋）；保持稳定的树高($\log_2 n$)，因此平衡树上的所有操作的复杂度均稳定。

2. 优先队列：

- 优先队列的结构特点；不属于线性结构
- 堆的概念；堆是天然的优先队列；（最大）堆的特点；
- 堆上的操作：元素添加（向上调整）；元素删除（向下调整）；建堆（两种方法：从空堆开始逐个插入；调整已存在的数据集合--从最后的子树开始调整）；**堆排序**----掌握所有操作的原理，复杂度，会模拟，**掌握堆排序的代码（基于数组实现）**
- 霍夫曼树：构造，编码，期望长度。

3. 散列：

- 散列的概念；散列函数；冲突
- 散列冲突解决的两种策略：线性探测/桶散列—链地址法；----理解冲突解决的原理，会模拟，能计算查找长度

4. 图：

- 图的一些基本概念（无向，有向，有向带权）
- 图的表示，两种方法：邻接矩阵；邻接表 ---如何表示（无向，有向，有向带权图），图的遍历/可达/连通：两种搜索策略-深度优先搜索/广度优先搜索---会识别/模拟 warshall 算法了解
- 最短路径：掌握 dijkstra 算法

（三）辅助

- 复杂度的概念；简单代码段复杂度的计算；常用结构操作的复杂度
- ADT 的概念；ADT 和具体的程序数据类型之间的关系/区别
- 递归程序能理解程序运行的机制（使用栈，自己调用自己），对于简单递归定义的需求能写递归程序，也要会读，书中几个例程-求阶乘、求指数、数制转换、回文检测、二分查找等。