1. Reference: 《数据结构》（C语言版）严蔚敏吴伟民编著清华大学出版社
2. Web course: <https://www.bilibili.com/video/av6239731?from=search&seid=15262918976300524527>
3. // video lecture;
4. Ep 01: 数据结构讨论的范畴; algorithm + data structure = program; 为计算机处理问题编制一组指令集; 数据结构, 问题的数学模型; example, 线性代数方程组, 环流模式方程; 算法, 模型; 数据; 数据元素; 数据项; 数据结构; 线性结构, 树形结构, 图状结构, 集合结构; 数据元素的有限集, D上关系的有限集; 存储结构, 逻辑结构在存储器中的映像; 关系的映像方法, 顺序映像, 链式映像; 数据类型; 抽象数据类型, 数据抽象, 数据封装; example, 复数; 算法, 解决问题的有限长的操作序列; (2018-4-8)
5. Ep 02: 算法, 有穷性, 确定性, 可行性, 有输入, 有输出; 算法设计, 正确性, 可读性(易于理解), 健壮性(数据非法进行相应处理), 高效率低存储量; 衡量算法效率, 事后统计, 事前分析估算; 算法执行时间相关, 策略, 规模, 语言, 编译, 机器; 算法的渐进时间复杂度; 执行时间与执行次数成正比; O(n^3); 冒泡排序法; 算法存储量, 输入数据, 程序本身, 辅助变量; 通常按最坏情况考虑; (2018-4-8)
6. Ep 03: 线性结构, 数据元素的有序集; 基本特征, 第一元素, 最后元素, 唯一的后继, 唯一的前驱; 数据对象, 数据关系; ListEmpty; ListLength; PriorElement; GetElement; LocateElement; ClearList; PutElement; ListInsert; ListDelete; 集合并; 归并两个线性表; 线性表的实现, 顺序映像; 用一组地址连续的存储单元依次存放线性表; (2018-4-8)
7. Ep 04: 起始地址(基地址); 线性表的初始化; LocateElement; ListInsert; ListDelete; 线性表链式映像; 单链表; 元素+指针=结点; 数据域, 指针域; 单链表操作的实现; GetElement; ListInsert, 修改指向后继的指针; ListDelete, 修改指向后继的指针; (2018-3-10)
8. Ep 05: CreateList; 顺序映像和链式映像的时间复杂度相同; 单链表存在的问题; 带头结点的线性链表; InitList, DestoryList; LocatePos, LocateElement, ListTraverse; ClearList, SetCurrentElement, Append, InsertAfter, DeleteAfter; (2018-4-8)
9. Ep 06: 归并链表; 双向链表; 循环链表; example, 一元多项式的表示; 如果结构不常变, 则数组好, 如果经常变, 则链表好; 栈的类型定义; InitStack, destroyStack, stackEmpty, stackLength, GetTop, ClearStack, Push, Pop; (2018-4-9)
10. Ep 07: example, 数制转换; example, 括号匹配的检验; example, 行编辑程序问题; example, 迷宫求解; 若通, 则纳入路径继续前进, 若不可通, 则后退换向, 若四周都不同, 则删除路径; (2018-4-9)
11. Ep 08: 表达式求值; 前缀式, 中缀式, 后缀式; ; 中缀式和后缀式的相互转换; 实现递归, 后调用先返回; (2018-4-9)
12. Ep 09: 汉诺塔; 顺序栈; 链栈; 队列; InitQueue, DestoryQueue, QueueEmpty, QueueLength, GetHead, ClearQueue, EnQueue, DeQueue; 链队列; 循环队列, 顺序映像; 浪费一个空间, 用来区分队满和队空; 栈是先进后出, 队列是先进先出; (2018-4-9)
13. Ep 10: 串的定义; StringAssign, StringCopy, StringLength, StringEmpty, StringCompare, Concat, SubString, Index, replace, StringDelete; 串类型的最小操作子集; 利用Compare, Length, Sub实现Index; 串的定长数据存储表示; 截断; (2018-4-10)
14. ep 11: 串的堆分配存储表示; malloc, free; 串的块链存储表示, 每个结点放一个串; 串的模式匹配; 简单算法index; 首尾匹配算法; KMP算法, 时间复杂度O(m+n), 因为解决了指针回溯问题; 把串比较转化成自比较; 模式串的next函数; (2018-4-11)
15. Ep 12: 寻找模式串; review; (2018-4-14)
16. Ep 13: 数组和广义表; 数组的类型定义; InitArray, DestroyArray, Value, Assign; 数组的顺序表示和实现; 行序为主序; n维数组的映像函数; 稀疏矩阵的压缩存储; 稀疏因子; 特殊矩阵的压缩存储(三角矩阵, 对角矩阵); 三元组顺序表, 类似于多项式; 求转置矩阵, 需要移动元素; (2018-4-14)
17. Ep 14: FastTransposeMatrix; 统计每一列非零元; 行逻辑连接的顺序表; 矩阵乘法的经典算法; 三元顺序表的矩阵相乘, 时间复杂度大大降低; (2018-4-18)
18. ep 15: 十字链表; review, (1) 了解数组的两种存储表示方法, 并掌握以行为主的地址计算; (2)掌握特殊矩阵压缩表示的坐标变换; (3)了解稀疏矩阵的两种压缩存储方法, 领会三元组进行矩阵运算; 树; 空树; 子树; 有向树; Node; 结点的度; 叶子结点; 分支结点; 孩子父亲兄弟祖先子孙; 结点的层次; 树的深度; 树的查找; 树的插入; 树的删除; 有向树; 有序树, 子树间存在次序关系; 线性结构和树结构的比较; 二叉树; 二叉树的5种基本形态; (2018-4-18)
19. ep 16: 习题课; example 2.4, 比较两个顺序表大小(类似于字符串比较); 删除有序表中区间(a, b)元素; example 2.8, ABC都是有序表, 从A链表中删除B表和C表共有的数据元素, 谁小谁后移; example 2.9, 双向循环链表, 增加访问频度; (2018-4-18)
20. Ep 17: example 3.2, 给出证明栈操作是否合法的准则; 第一个子列s>=x, 整个序列s=x; example 3.6, 用while和stack代替递归; 识别一个读入的反对称序列, 用stack; example 3.15, 判断是否回文, 比较栈顶和队头; example 4.7, 从串S中删除所有和串T相同的子串; example 4.11, 求最长重复子串; (2018-4-20)
21. Ep 18: 又讲了一遍KMP算法的next; 时间复杂度O(m+n); 性质1, 二叉树第i层最多有2^(i-1)结点; 性质2, 深度为k的二叉树上至多有2^k-1个结点; 性质3, n0 = n2 + 1; 性质4, 具有n个结点的二叉树深度为log2n+1; (2018-4-20)
22. ep 19: 满二叉树; 完全二叉树; (1) i/2是i的parent; (2) 若2i>n, 则i无左孩, 若2i=n, 则2i为i左孩; (3)若2i+1>n, 则i无右孩, 若2i+1=n, 则2i+1为i右孩; 二叉树的顺序存储表示; 用完全二叉树规模的数组存储普通二叉树; 二叉树的链式存储; 二叉链表; 三叉链表; 二叉树的遍历; 每个结点经过3次; 先序遍历, 第1次访问时送报; 中序遍历, 第2次访问时送报; 后序遍历, 第3次访问时送报; Preorder递归; 用栈来实现遍历回退; 中序遍历的非递归实现; (2018-4-20)
23. Ep 20: 统计二叉树中叶子结点的个数(先序遍历); 求二叉树的深度(后序遍历); 复制二叉树(后序遍历); 建立二叉树的存储结构; 按给定的先序序列建立二叉链表; 按给定的表达式建相应二叉树, 由先缀表达式建树; (2018-4-27)
24. Ep 21: 由先序序列和中序序列建树; 线索链表, 线索二叉树; 先序线索化二叉树; 线索链表的遍历算法; 如何建立线索链表; 在线性链表上遍历不需要栈; (2018-4-27)
25. ep 22: 树的双亲表示法; 孩子链表表示法; 二叉树和森林的转化; 先根遍历; 中序遍历; 森林组成, 森林中第一棵树的根节点, 森林中第一棵树的子树森林, 森林中其他树构成的森林; 求树的深度的算法; (2018-4-28)
26. Ep 23: 孩子链表表示法; 输出树中所有从根到叶子的路径的算法; 建树的存储结构的算法; 哈夫曼树和哈夫曼编码; 结点的路径长度, 树的路径长度; (2018-4-28)
27. ep 24: 前缀编码; 等长码; 不等长码, 防止二义性任意码不能是另一个码的前缀; 哈夫曼编码是最优前缀编码; review; 图; 顶点集, 弧; 有向图; 子图; 邻接点, 关联, 度; 路径长度, 简单路径; 回路, 简单回路; 生成树; (2018-4-28)
28. Ep 25: 结构的建立和销毁, 对顶点的访问操作, 对邻接点的操作, 插入或删除顶点, 插入和删除弧, 遍历; 图的数组(邻接矩阵)存储表示; 邻接表; 有向图的十字链表; 无向图的邻接多重表存储; 图的遍历; (2018-4-28)
29. Ep 26: DFSTraverse, 对图作深度优先遍历; 深度优先生成树; 广度优先搜索, BFSTraverse; (2018-4-29)
30. ep 27: 求一条顶点i到顶点s的简单路径; 求两个顶点之间最短路径; 图的最小生成树; Prim算法, 选择连通两区权值最小的边; Kruskal算法, 添加不产生回路的最小边; Prim适合稠密图, Kruskal适合稀疏图; (2018-4-29)
31. ep 28: 重(双)连通图, 关节点; 深度优先生成树; 若根节点有两个及以上的分支, 则必为关节点; 对任意顶点, 若某棵子树的根或子树中的其他顶点没有和其祖先相同的回边, 则该顶点必为关节点; 如何求关节点; 两点之间的最短路径; 从源点到其余各点的最短路径, 依路径长度递增的次序求得各条路径; (2018-4-29)
32. Ep 29: 拓扑排序; 选取一个没有前驱的顶点, 删去此顶点以及所有以它为尾的弧; 关键路径; 求关键活动; (2018-4-30)
33. ep 30: review; 广义表的类型定义; 广义表是多层次的线性结构; 广义表的数据元素有相对次序, 长度定义为最外层包含的元素个数, 深度定义为括弧的重数, 可以共享, 可以递归, 非空广义表可以分解表头表尾; 基本操作, 结构的创建和销毁, 状态函数, 插入和删除操作, 遍历; 广义表操作的递归函数; Divide and conquer; 求广义表的深度; (2018-4-30)
34. Ep 31: 复制广义表; 创建广义表的存储结构, CreateGList; 删除单链表中所有值为x的数据元素, void Delete\_GL; (2018-5-1)
35. Ep 32: 一个递归函数必定能改写成利用栈实现的非递归函数; 尾递归; review; 遍历是所有非线性结构操作的基础; 深度优先搜索, 二叉树(前序, 中序, 后序), 树(先根, 后根), 图(先访问顶点, 后访问顶点), 广义表(先遍历表头, 先遍历表尾); example 6.12, 判别结点u是否是v的子孙; example 6.19, 删除二叉树中所有值为x的结点的子树, 并释放空间; 必须用后序遍历; example 6.22, 输出二叉树表示的算数表达式; 因为中缀表达式, 所以中序遍历; 表达式求值应该后序遍历; (2018-5-1)
36. Ep 33: example 6.24, 无重复地输出以孩子兄弟链表存储的树T中所有的边; example 7.7, 编写算法求有向图从u到v的所有简单路径; example 6.15, 在二叉链表增加parent和mark, 编写不用栈的后序遍历; 深度优先算法的非递归形式; 广度优先搜索遍历的算法; 编写按层次顺序遍历二叉树的算法; 查找表; 查找某元素是否在表中, 检索属性, 插入元素, 删除元素; (2018-5-2)
37. Ep 34: 静态查找表, 只做查询和检索; 动态查找表; 关键字; 主关键字, 次关键字; 查找成功, 查找不成功; 查找表的结构; 基本操作, create, destory, search, traverse; 顺序查找表; 回顾顺序表的查找算法; location; 平均查找长度, 进行比较的关键字个数的期望值; 有序查找表; 折半查找; (2018-5-2)
38. Ep 35: 分析折半查找的平均查找长度; 表长为n的折半查找判定树的深度和含有n个结点的完全二叉树的深度相同; 静态查找树表, 不等概率下, 折半查找不是最好; 最优查找二叉树; 次优查找二叉树; 构造次优二叉树的算法; 索引顺序表; 顺序表和有序表的比较; 查找方法, 由索引确定记录所在的区间, 在顺序表的某个区间进行查找; 索引表的平均查找长度; (2018-5-3)
39. Ep 36: 动态查找树表; InitDSTable; SearchDSTable; InsertDSTable; DeleteDSTable; 几种查找表的特性; 二叉排序树(二叉查找树); SearchBST; 二叉排序树的插入算法, InsertBST; 二叉排序树的删除算法, DeleteBST; (2018-5-4)
40. Ep 37: 二叉排序树定义; 二叉排序树的查找算法; 查找性能分析; 二叉平衡树; B树, 平衡的多路查找树; B树结构的C语言描述; 查找过程; (2018-5-4)
41. Ep 38: B树的查找, 插入, 删除; (2018-5-4)
42. Ep 39: nothing; (2018-5-4)
43. Ep 40: 哈希表; 开放定址法; 增量d有三种取法, 线性探测再散列, 平方探测再散列, 随机探测再散列; 增量di应具有完备性; 链地址法; 哈希表的查找; 开放定址哈希表的存储结构; 决定哈希表查找的ASL的因素, 选用的哈希函数, 选用的处理冲突的方法, 哈希表饱和的程度(装载因子值的大小); 线性探测再散列, 随机探测再散列, 链地址法; 哈希表的删除操作; 理想的哈希函数; review, (1)顺序表和有序表的查找方法及其平均查找长度的计算方法, (2)静态查找树的构造方法和查找算法, 理解静态查找树和折半查找的关系, (3)熟练掌握二叉排序树的构造和查找方法; (4)理解B树B+树和键树的特点以及它们的建树过程; (5) 熟练掌握哈希表的构造方法, 深刻理解哈希表与其它结构的表的实质性的差别; (6)掌握按定义计算各种查找方法在等概率情况下查找成功时的平均查找长度; (2018-5-5)
44. Ep 41: 排序的定义; 内部排序, 不需要访问外存, 否则就是外部排序; 内部排序的方法, 逐步扩大记录的有序序列长度; 插入类, 将无序子序列中的一个或几个记录插入到有序序列中, 从而增加记录的有序子序列的长度; 交换类; 选择类; 归并类; 一趟插入排序; 直接插入排序; 实现排序的基本操作, 比较和移动; 对于直接插入排序, 最好的情况n-1, 最坏的情况n^2; 折半插入排序; 表插入排序, 表插入的过程; (2018-5-6)
45. Ep 42: void LInsertSort(); void Arrange(); 希尔排序; void ShellInsert(); void ShellSort(); 快速排序; 冒泡排序; void BubbleSort(); 比较n^2, 移动n^2; 一趟快速排序; int Partition(); 快速排序是递归; 简单选择排序; (2018-5-6)
46. Ep 43: 堆排序; 堆的定义; 堆排序是利用堆的特性对记录序列进行排序; 筛选, 对左右均为堆的完全二叉树, 调整根节点使整个二叉树为堆; void HeapSort(); 堆排序的时间复杂度, nlogn; 归并排序; void Merge(); 2路归并排序; void Msort(); void MergeSort(); 多关键字排序; 实现多关键字排序, 最高位优先MSD法, 最低位优先LSD法; (2018-5-6)
47. Ep 44: 链式基数排序; 分配-收集的方法; review; O(nlogn), 快速排序, 堆排序, 归并排序; O(n^2), 直接插入排序, 冒泡排序, 简单选择排序; O(n), 基数排序; 当排序记录序列按关键字有序, 直接排序和起泡排序O(n), 快速排序为O(n^2); 空间性能; 所有的简单排序方法和堆排序的空间复杂度为O(1), 快速排序为O(logn), 归并排序为O(n), 链式基数排序为O(rd); 排序方法的稳定性能, 两个关键字相等的记录在排序后顺序不变; 多关键字LSD, 必须采用稳定; 快速排序, 堆排序, 希尔排序不稳定; 基于比较关键字的排序算法时间下限是O(nlogn), 从判定树可以证明; review; 外部排序, 记录数量很大内存不够; 外部排序的基本过程, 构造若干有序子序列, 通过归并, 逐步扩大有序子序列的长度; (2018-5-6)
48. Ep 45: m段k路, 则次数logk(m); 文件, 记录的集合; 操作系统的文件, 数据库文件; 文件的逻辑结构, 物理结构; 检索, 顺序存取, 直接存取; 顺序文件的组织形式, 连续文件, 串联文件; 便于顺序存取, 不便于直接存取, 插入新的记录只能在末尾, 删除记录只作标记, 更新必须生成新的文件; 批处理方式, 主文件, 事务文件; 每个操作判别合法性, 对于一个记录存在多个操作; 索引文件; 主文件和多级索引; 关键字和指针; 主文件无序索引有序; 索引在输入数据建立文件自动生成; 操作的特点; 索引文件的结构; 多级静态索引, 动态索引; 索引表采用查找树表或哈希表; (2018-5-10)
49. Ep 46: 索引顺序文件; ISAM文件; 文件的组织方式, 磁道索引, 柱面索引, 主索引; 检索, 顺序存取, 按关键字存取; 删除, 作删除标记; VSAM文件; 控制区间(逻辑磁道), 控制区域(逻辑柱面); 顺序集是单链表, 索引集的结点是B+树的非叶结点; 文件的操作, 检索, 插入, 删除; VSAM文件通常作为大型索引顺序文件的标准组织方式, 优点动态分配和释放, 缺点占用较多存储空间; 直接存取文件, 直接得到映像地址; 哈希文件的结构; 文件的操作, 检索, 插入, 删除; 优点, 无需排序; (2018-5-10)
50. Ep 47: 多关键字文件; 主索引, 次索引; 次索引的组织方法, 多重链表文件; 倒排文件; review; example 9.15; example 9.5, 二叉排序树及其平均查找长度; example 9.18; example 9.13; (2018-5-10)
51. Ep 48: example 10.4, 快速排序最好情况; example 10.9; example 10.17; example 11.1; (2018-5-10)
52. // text book;
53. Chapter 1;
54. Ch 1.2: 数据, 数据元素, 数据对象, 数据结构; 集合, 线性结构, 树形结构, 图状结构或网状结构; 存储结构, 元素或结点, 数据域; 顺序存储, 链式存储; 虚拟存储结构; 数据类型; 原子类型, 结构类型; 抽象数据类型; 原子类型, 固定聚合类型, 可变聚合类型; D(数据对象), S(关系集), P(基本操作集); 多形数据类型; (2018-5-11)
55. Ch 1.3: 类C语言; (2018-5-11)
56. Ch 1.4: 算法, 有穷性, 确定性, 可行性, 输入输出; 正确性, 可读性, 健壮性, 效率与低存储量需求; 事后统计的方法, 事先分析估算的方法; 时间复杂度, 空间复杂度; (2018-5-11)
57. Chapter 2;
58. Ch 2.1: 线性表; 数据项, 记录, 文件; example 2.1, 集合的并; example 2.2, 有序表的并; (2018-5-11)
59. Ch 2.2: 线性表的顺序存储结构; 线性表的插入和删除; (2018-5-11)
60. Ch 2.3: 结点, 数据域, 指针域, 链表; 单链表GetElem; 单链表ListInsert, 单链表ListDelete; 建立单链表CreateList; 合并两个有序链表MergeList; 静态链表, 用指针描述的链表; 静态链表的定位函数LocateElem; 有向图的十字链表; 无向图的邻接多重表; 循环链表; 双向链表; (2018-5-12)
61. Ch 2.4: 一元多项式的表示及相加; (2018-5-14)
62. Chapter 3;
63. Ch 3.1: 栈; 栈顶; 栈底; 后进先出; (2018-5-14)
64. Ch 3.2: 栈的应用; 数制转换; 括号匹配的检验; 行编辑程序; 迷宫求解; 表达式求值, 算符间的优先关系; example 3.1, 对算术表达式求值; (2018-5-15)
65. Ch 3.3: 栈和递归的实现; example 3.2, n阶汉诺塔问题; (2018-5-15)
66. Ch 3.4: 队列; 队尾, 队头; 双端队列; 链队列; 循环队列; (2018-5-15)
67. Ch 3.5: 离散事件模拟; example 3.3, 银行客户模型; (2018-5-15)
68. Chapter 4;
69. Ch 4.1: 串类型的定义; 串的长度, 空串; 子串, 主串, 位置; 串的相等; 空格串; (2018-5-15)
70. Ch 4.2: 串的表示和实现; 定长顺序存储表示; 串联接Concat; 求子串SubString; 堆分配的存储表示; 串的块链存储表示, 块链结构; (2018-5-15)
71. Ch 4.3: 串的模式匹配算法;
72. Chapter 6: 树和二叉树;
73. Ch 6.1: 子树; 结点, degree; leaf, child, parent, sibling; 祖先, 子孙; 层次, 堂兄弟, 深度; 有序树, 无序树; 森林; (2018-5-12)
74. Ch 6.2: 二叉树; 二叉树的性质; 二叉树的顺序存储; 二叉树的链式存储; (2018-5-13)
75. Ch 6.3: 遍历二叉树; Preorder, Inorder, Postorder; 前缀表示, 中缀表示, 后缀表示; 中序遍历的非递归算法; 按先序序列建立二叉链表; 遍历二叉树时间复杂度O(n), 空间复杂度O(n); 线索链表, 线索, 线索二叉树, 线索化; (2018-5-13)
76. Ch 6.4: 树的存储结构, 双亲表示法, 孩子表示法, 孩子兄弟表示法; 树和二叉树的对应关系; 先根遍历树, 后根遍历树; 先序遍历森林, 中序遍历森林; (2018-5-14)
77. Ch 6.5: 等价关系, 等价类; example 6.1, 求等价类; (2018-5-14)
78. Ch 6.6: 路径长度, 带权路径长度; 最优二叉树, 赫夫曼树; 构造赫夫曼树, 赫夫曼算法; 赫夫曼编码; 出现次数较多的字符采用尽可能短的编码; example 6.2, 设计赫夫曼编码; (2018-5-15)
79. Ch 6.7: 回溯法和树的遍历; example 6.3, 求幂集; example 6.4, 求4皇后问题的所有布局; (2018-5-15)
80. Ch 6.8: 树的计数; example 6.5, 从前序序列和中序序列构造二叉树; (2018-5-15)
81. Chapter 7;
82. Ch 7.1: 顶点, 弧尾, 弧头, 有向图, 边, 无向图; 完全图, 有向完全图; 稀疏图, 稠密图; 权, 网; 邻接点, 依附, 关联; 度, 入度, 出度; 路径, 回路或环, 简单路径; 连通, 连通图, 连通分量; 强连通图, 强连通分量; 生成树; 有向树, 生成森林; (2018-5-12)
83. Ch 7.2: 邻接矩阵; 利用邻接矩阵求顶点的度; 网的邻接矩阵; 邻接表, 每个顶点与之相邻的顶点; 逆邻接表; (2018-5-12)
84. Ch 7.3: 图的遍历; 深度优先搜索, 类似于先根遍历; 广度优先搜索, 类似于按层次遍历; (2018-5-12)
85. Ch 7.4: 图的连通性; 非连通图, 需从多个顶点出发进行搜索; 极小连通子图, 深度优先生成树, 广度优先生成树; 非连通图的生成森林; 深度优先生成森林; 利用十字链表, 求有向图的强连通分量; 最小生成树; 最小生成树的MST性质; Prim算法; Kruskal算法; 关节点(割点), 重连通图, 连通度; (2018-5-12)
86. Ch 7.5: 有向无环图; 拓扑排序; 选一个没有前驱的结点并输出, 从图中删除该顶点和所有以它为尾的弧; 顶点表示活动的网(AOV网); 边表示活动的网(AOE网); 关键路径; (2018-5-12)
87. Ch 7.6: 最短路径; Dijkstra算法; D[i], 当前所找到的从始点v到终点vi的最短路径长度; Floyd算法; (2018-5-12)
88. Chapter 9;
89. 查找表, 静态查找表, 动态查找表; 关键字, 主关键字, 次关键字; 查找, 查找成功, 查找不成功; (2018-5-18)
90. Ch 9.1: 静态查找表; 顺序查找; 平均查找长度; 有序表的查找; 折半查找; 折半查找的性能分析; 静态树表的查找; ;;;;;
91. Ch 9.3: 哈希函数; 冲突, 同义词; 哈希表, 散列, 哈希地址或散列地址; 哈希函数的构造方法; 均匀的哈希函数; 直接定址法; 数字分析法; 平方取中法; 折叠法; 除留余数法; 随机数法; 处理冲突的方法; 开放定址法; 再哈希法; 链地址法, example 9-3, 链地址法构造的哈希表; 建立一个公共溢出区; 哈希表的查找及其分析; 哈希表的装填因子; (2018-5-19)
92. Chapter 10;
93. Ch 10.1: 排序方法是稳定的, 排序方法是不稳定的; 内部排序, 外部排序; 简单的排序, 先进的排序, 基数排序; 比较两个关键字的大小, 将记录葱一个位置移动到另一个位置; 线性表, 链表排序, 地址排序; (2018-5-21)
94. Ch 10.2: 直接插入排序; ;;;;;;;;;;;;;;;;;;
95. 王道论坛考研数据结构;
96. Chapter 1: 绪论;
97. 分析算法的时间复杂度和空间复杂度; (2018-9-11)
98. Node 1.1: 数据结构的基本概念;
99. List 1.1.1: 基本概念和术语; 数据; 一个数据元素可由若干个数据项组成; 数据对象, 具有相同性质的数据元素的集合; 数据类型, 原子类型, 结构类型, 抽象数据类型; 抽象数据类型, 数据对象, 数据关系, 基本操作集; 数据结构, 逻辑结构, 存储结构, 数据的运算; (2018-9-11)
100. List 1.1.2: 数据结构的三要素; 数据的逻辑结构; 线性结构, 线性表; 非线性结构, 集合, 树, 图; 数据的存储结构; 顺序存储; 链式存储; 索引存储; 散列存储; 数据的运算; (2018-9-11)
101. List 1.1.3: 选择4, 循环队列是顺序表表示的; 选择7, 结点内一定连续;
102. Node 1.2: 算法和算法评价;
103. List 1.2.1: 算法的基本概念; 有穷性, 确定性, 可行性, 输入, 输出; 正确性, 可读性, 健壮性, 效率与低存储量需求; (2018-9-11)
104. List 1.2.2: 算法效率的度量; 时间复杂度; 最坏时间复杂度, 平均时间复杂度, 最好时间复杂度; 空间复杂度; (2018-9-11)
105. List 1.2.3: 选择6; 选择12;
106. Chapter 2: 线性表;
107. Node 2.1: 线性表的定义和基本操作;
108. List 2.1.1: 线性表是具有相同数据类型的数据元素的有限序列; 每个元素有且仅有一个前驱, 一个后继; (2018-9-11)
109. List 2.1.2: 线性表的基本操作; (2018-9-11)
110. Node 2.2: 线性表的顺序表示;
111. List 2.2.1: 顺序表的定义, 线性表的顺序存储; 一维数组可以是静态分配的, 也可以是动态分配的; (2018-9-11)
112. List 2.2.2: 顺序表上基本操作的实现; 插入操作; 删除操作; 按值查找; 顺序表存储空间=表长\*sizeof(元素类型); (2018-9-15)
113. page 18, question 12; (2018-9-15)
114. Node 2.3: 线性表的链式表示;
115. List 2.3.1: 单链表的定义;
116. 非随机存取; 头指针; 头结点; (2018-9-15)
117. List 2.3.2: 单链表上基本操作的实现;
118. 采用头插法建立单链表; 采用尾插法建立单链表; 按序号查找结点; 按值查找结点; 插入结点操作; 删除结点操作; 求表长操作; (2018-9-15)
119. List 2.3.3: 双链表;
120. 双链表的插入操作; 双链表的删除操作; (2018-9-15)
121. List 2.3.4: 循环链表; 循环双链表; (2018-9-15)
122. List 2.3.4: 循环双链表;
123. List 2.3.5: 静态链表;
124. List 2.3.6: 顺序表和链表的比较;
125. 存取方式; 逻辑结构与物理结构; 插入查找删除; 空间分配; 基于存储的考虑; 基于运算的考虑; 基于环境的考虑; (2018-9-17)
126. Page 37, question 6; end at 16; //////////////////////////////////////////////////////
127. Chapter 3: 栈和队列;
128. Node 3.1: 栈;
129. List 3.1.1: 栈的基本概念;
130. 栈的定义; 栈; 栈顶; 栈底; 空栈; 栈的基本操作; (2018-9-19)
131. List 3.1.2: 栈的顺序存储结构;
132. 顺序栈的实现; 顺序栈的基本运算; 共享栈; (2018-9-19)
133. List 3.1.3: 栈的链式存储结构;
134. List 3.1.4: 试题精选;
135. 选择题11, 卡特兰数; (2018-12-6)
136. Node 3.2: 队列;
137. List 3.2.1: 队列的基本概念;
138. 队列的定义; 队列常见的基本操作; (2018-9-21)
139. Lis 3.2.2: 队列的顺序存储结构;
140. 队列的顺序存储; 循环队列; 牺牲一个单元来区分队空和队满; 循环队列的操作; (2018-9-21)
141. List 3.2.3: 队列的链式存储结构;
142. 队列的链式存储; (2018-9-21)
143. List 3.2.4: 双端队列;
144. List 3.2.5: 试题精选;
145. 练习题8; 练习题10; 练习题18; ;;;;;;;(2018-12-7)
146. Node 3.3: 栈和队列的应用;
147. List 3.3.1: 栈在括号匹配中的应用;
148. List 3.3.2: 栈在表达式求值中的应用;
149. List 3.3.3: 栈在递归中的应用;
150. List 3.3.4: 队列在层次遍历中的应用;
151. List 3.3.5: 队列在计算机系统中的应用;
152. Node 3.4: 特殊矩阵的压缩存储;
153. List 3.4.1: 数组的定义;
154. List 3.4.2: 数组的存储结构;
155. List 3.4.3: 矩阵的压缩存储;
156. 对称矩阵; 三角矩阵; 三对角矩阵; (2018-9-25)
157. List 3.4.4: 稀疏矩阵;
158. List 3.4.5: 本节试题精选;
159. 选择6, 还要再减1; 选择9; (2018-12-8)
160. Chapter 4: 树与二叉树;
161. Node 4.1: 树的基本概念;
162. List 4.1.1: 树的定义;
163. 空树; 根; 子树; (2018-9-26)
164. List 4.1.2: 基本术语;
165. 祖先结点; 子孙结点; 双亲结点; 孩子结点; 兄弟结点; 结点的度; 树的度; 分支结点; 叶子结点; 结点的层次; 结点的深度; 结点的高度; 树的高度; 有序树; 无序树; 路径, 路径长度; 森林; (2018-9-26)
166. List 4.1.3: 树的性质;
167. Node 4.2: 二叉树的概念;
168. List 4.2.1: 二叉树的定义及其主要特性;
169. 二叉树的定义; 几个特殊的二叉树; 满二叉树; 完全二叉树; 二叉排序树; 平衡二叉树; 二叉树的性质; (2018-9-26)
170. List 4.2.2: 二叉树的存储结构;
171. 顺序存储结构; 链式存储结构; (2018-9-27)
172. List 4.2.3: 本节试题精选;
173. 选择14; 选择16; (2018-12-9)
174. Node 4.3: 二叉树的遍历和线索二叉树;
175. List 4.3.1: 二叉树的遍历;
176. 先序遍历; 中序遍历; 后序遍历; 层次遍历; 由遍历序列构造二叉树; (2018-9-28)
177. List 4.3.2: 线索二叉树;
178. 线索二叉树的基本概念; 线索二叉树的构造; 线索二叉树的构造; 线索二叉树的遍历; (2018-9-28)
179. Node 4.4: 树, 森林;
180. List 4.4.1: 树的存储结构;
181. 双亲表示法; 孩子表示法; 孩子兄弟表示法; (2018-10-6)
182. List 4.4.2: 树森林与二叉树的转换;
183. List 4.4.3: 树和森林的遍历;
184. List 4.4.4: 树的应用, 并查集;
185. List 4.4.5: 本节试题精选;
186. 应用题4, 这奇葩的形状; (2018-12-10)
187. Node 4.5: 树与二叉树的应用;
188. List 4.5.1: 二叉排序树;
189. 二叉排序树的定义; 二叉排序树的查找; 二叉排序树的插入; 二叉排序树的构造; 二叉排序树的删除; (2018-10-8)
190. List 4.5.2: 平衡二叉树;
191. 平衡二叉树的定义; 平衡二叉树的插入; 平衡二叉树的查找; (2018-10-8)
192. List 4.5.3: 哈夫曼树和哈夫曼编码;
193. 哈夫曼树的定义; 哈夫曼树的构造; 哈夫曼编码; (2018-10-13)
194. List 4.5.5: 本节试题精选;
195. 选择17, Nh=1+Nh-1+Nh-2; 选择27; 选择29, 把树补全; 解答13, 不要忘了-1; (2018-12-10)
196. Chapter 5: 图;
197. Node 5.1: 图的基本概念;
198. List 5.1.1: 图的定义;
199. 有向图; 无向图; 简单图, 不存在重复边, 不存在顶点到自身的边; 多重图， 允许重边, 允许自环; 完全图, 任意两个顶点之间都存在方向相反的两条弧; 子图; 连通, 连通图和连通分量; 连通分量, 无向图的极大连通子图; 强连通图， 有向图中, 从顶点v到顶点w和从顶点w到顶点v都有路径; 强连通分量; 生成树, 生成森林; 顶点的度, 入度和出度; 边的权和网; 稠密图, 稀疏图; 路径, 路径长度和回路; 简单路径, 简单回路; 距离; 有向树; (2018-10-13)
200. Node 5.2: 图的存储及基本操作;
201. List 5.2.1: 邻接矩阵法;
202. List 5.2.2: 邻接表法;
203. List 5.2.3: 十字链表;
204. List 5.2.4: 邻接多重表;
205. List 5.2.5: 图的基本操作;
206. List 5.2.6: 本节试题精选;
207. 解答题5; (2018-12-11)
208. Node 5.3: 图的遍历;
209. List 5.3.1: 广度优先搜索BFS;
210. BFS算法性能分析; 邻接矩阵O(|V|^2); 邻接表O(|V|+|E|); BFS算法求解单源最短路径问题; 广度优先生成树; 邻接矩阵唯一; 邻接表不唯一; (2018-10-17)
211. List 5.3.2: 深度优先搜索DFS;
212. 类似于先序遍历; DFS算法的性能分析; 深度优先的生成树和生成森林; (2018-10-17)
213. List 5.3.3: 图的遍历与图的连通性;
214. List 5.3.4: 本节试题精选;
215. 选择5, 深度优先回退; 选择6, 深度回退; 选择17, 有向图; (2018-12-11)
216. Node 5.4: 图的应用;
217. List 5.4.1: 最小生成树;
218. 最小生成树不是惟一的; Prim算法; Kruskal算法; (2018-10-24)
219. List 5.4.2: 最短路径;
220. Dijkstra算法求单源最短路径问题; (2018-10-28)
221. List 5.4.3: 拓扑排序;
222. 有向无环图; AOV网; 拓扑排序; (2018-10-28)
223. List 5.4.4: 关键路径;
224. AOE网; 关键路径, 具有最大路径长度的路径; 关键活动; 完成工程的最短时间, 关键路径的长度; 事件vk的最早发生事件;ve(source)=0, ve(k)=max{ve(j)+weight(vj, vk)}; 事件vk的最迟发生时间; vl(dest)=ve(dest), vl(j)=min{vl(k)-weight(vj, vk)}; 活动ai的最早开始时间e(i); e(i)=ve(k); 活动ai的最迟开始时间; l(i)=vl(j)-weight(vk, vj); 一个活动ai的最迟开始时间l(i)和最早开始时间e(i)的差额d(i)=l(i)-e(i); 该活动可以拖延的时间; 关键活动, l(i)-e(i)=0; (2018-11-1)
225. List 5.4.5: 本节试题精选;
226. 应用题11; (2018-12-12)
227. Chapter 6: 查找;
228. Node 6.1: 查找的基本概念;
229. 查找; 查找表(查找结构); 静态查找表, 动态查找表; 关键字; 平均查找长度; (2018-11-1)
230. Node 6.2: 顺序查找和折半查找;
231. List 6.2.1: 顺序查找;
232. 一般线性表的顺序查找; 通过引入哨兵, 可以避免很多不必要的判断语句; 有序表的顺序查找; (2018-11-2)
233. List 6.2.2: 折半查找;
234. 仅适合线性表的顺序存储结构; (2018-11-2)
235. List 6.2.3: 分块查找;
236. List 6.2.4: 本节试题精选;
237. 选择8, ceil(log2(n+1)); 选择15, 别忘了+1-1; (2018-12-13)
238. Node 6.3: B树和B+树;
239. List 6.3.1: B树及其基本操作;
240. B树的高度, 磁盘存取次数; B树的查找; (2018-11-12)
241. List 6.3.3: 本节试题精选;
242. 选择16, 文件系统; 解答5, 查找失败的长度; (2018-12-14)
243. Node 6.4: 散列(Hash)表;
244. List 6.4.1: 散列表的基本概念;
245. 散列函数; 冲突; 同义词; 散列表; (2018-11-12)
246. List 6.4.2: 散列函数的构造方法;
247. 直接定址法; H(key)=a\*key+b; H(key)=key%p; 数字分析法; 数字分析法; 平方取中法; 折叠法; (2018-11-12)
248. List 6.4.2: 处理冲突的方法;
249. 开放定址法; 线性探测法; 平方探测法; 再散列法; 伪随机序列法; 拉链法; (2018-11-12)
250. List 6.4.4: 散列表查找及性能分析;
251. 查找效率取决于散列函数, 处理冲突和装填因子; (2018-11-13)
252. Node 6.5: 字符串模式匹配;
253. List 6.5.1: 简单的模式匹配算法;
254. List 6.5.2: 改进的模式匹配算法--KMP算法;
255. Chapter 7: 排序;
256. Node 7.1: 排序的基本概念;
257. List 7.1.1: 排序的定义;
258. 排序; 算法的稳定性; 比较和移动; (2018-11-14)
259. Node 7.2: 插入排序;
260. List 7.2.1: 直接插入排序;
261. 稳定的排序方法; 适用于顺序存储和链式存储的顺序表; (2018-11-14)
262. List 7.2.2: 折半插入排序;
263. 稳定的排序方法; (2018-11-14)
264. List 7.2.3: 希尔排序;
265. 不稳定; (2018-11-14)
266. Node 7.3: 交换排序;
267. List 7.3.1: 冒泡排序;
268. 最小的元素如气泡一般逐渐往上漂浮直到水面; 稳定的排序算法; (2018-11-15)
269. List 7.3.2: 快速排序;
270. 不稳定; (2018-11-15)
271. Node 7.4: 选择排序;
272. 每一趟选出关键字最小的元素; (2018-11-16)
273. List 7.4.1: 简单选择排序;
274. 简单选择排序是一个不稳定的排序方法; (2018-11-16)
275. List 7.4.2: 堆排序;
276. 小根堆; 大根堆; 交换n和n/2; 练习题1, 实在搞不定就行特殊值; (2018-11-18)
277. Node 7.5: 归并排序和基数排序;
278. List 7.5.1: 归并排序;
279. 稳定; (2018-11-19)
280. List 7.5.2: 基数排序;
281. 不是基于比较进行排序的; 稳定; (2018-11-18)
282. Node 7.6: 各种内部排序算法的比较及应用;
283. List 7.6.1: 内部排序算法的比较;
284. List 7.6.2: 内部排序算法的应用;
285. Node 7.7: 外部排序;
286. List 7.7.1: 外部排序的基本概念;
287. List 7.7.2: 外部排序的方法;
288. List 7.7.3: 多路归并与败者树;
289. List 7.7.4: 置换-选择排序(生成初始归并段);
290. List 7.7.5: 最佳归并树;
291. ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
292. 2018真题
293. 填空1, 注意是循环; 填空5, 注意A[4][5]含义; 填空8, 递增顺序; ;;;
294. -