

考试题型

- 单项选择题 (10%)
- 判断题 (10%)
- 多项选择题 (16%)
- 简答题 (10%)
- 综合分析 (44~48%)
- 算法编程 (6~10%)

仅供参考：

章节	Ch1	Ch2	Ch3	Ch6	Ch9	Ch10	Ch11
分值	~5%	~15%	~6%	~27%	~15%	~27%	~5%

如何衡量算法的有效性？

- **算法的增长率**: 当输入的值增长时，算法代价的增长速率。
- 对于大多数的算法而言，运行所花费的时间取决于输入的“规模”。
- **渐进分析**是指当输入规模很大时对一种算法进行的研究，对算法资源开销进行评估。
- **时间代价O**
- **最佳、平均、最差情况算法分析**

第一章

- 数据和数据结构的概念
- 数据结构的选择遵循的步骤
- 算法的概念、性质
- 重点掌握基本概念。
- 算法分析的基本概念及其法则
 - 渐进分析的相关概念及其应用, 增长率的概念及其应用
 - 时空代价的概念及分析方法

空间代价

- 分析时间代价 → 数据结构的算法
- 分析空间代价 → 数据结构
- 可以同样用渐进(时间)复杂度分析的方法分析空间代价。
- 结构性开销
- 牺牲空间 → 减少时间代价，反之亦然。
- 基于磁盘的空间/时间权衡原则：在磁盘上的存储代价越小，程序运行得越快。

第二、三章

- 重点。（要求能够算法编程）
 - 顺序表、链表
 - 栈、队列
- 概念、比较和应用
- 递归相关知识

特殊的线性表：队列

- 队列是指允许在一端进行插入、而在另一端进行删除的线性表。
- 尾(rear)指针总是指向最后被插入的元素；排头(front)指针指向排头元素的前一个位置。
- 最先插入的元素将最先能够被删除；最后插入的元素将最后才能被删除。
- 插入和删除元素只改变指针的位置，不移动别的元素。
- “先进先出”， First in first out, → 先来先服务。
- 实现方式：顺序队列、链式队列
顺序队列中，有一种队列称为循环队列。

特殊的线性表：栈

- 插入和删除元素的操作只在栈顶进行；栈底是不允许进行插入和删除操作的。
- 栈(Stack)是限定在一端进行插入和删除的线性表。
- 最先插入的元素最后被删除；最后插入的元素最先被删除。→ First in last out, last in first out。“先进后出”表。
- 顺序栈、链栈

第六章

- 重点。
 - 二叉树概念、遍历、重构
 - 二叉查找树概念、构建（第九章）
 - 堆概念、构建（第10章）
 - 哈夫曼编码概念、构建、期望代码长度

树的基本术语（含第六章的基本概念）

- 父结点(**parent**) 每一个结点的前件；
- 子结点(**children**) 每一个结点的后件；
- 根结点(**root**) 没有前件的结点；
- 结点的度 结点的分支数，即拥有后件的个数；
- 树的度 树中所有结点度的最大值；
- 叶子结点(**leaf**) 度为0的结点 (没有后件的结点)；
- 结点的层数(**level**) 树中根结点的层次为1，根结点子树的根为第2层，以此类推；
- 树的深度/高度(**height**) 树的最深结点的深度；

二叉树(Binary Tree)

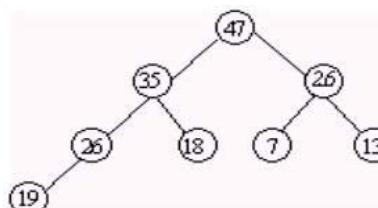
- 二叉树的定义
- 二叉树具有下列4个重要的性质。
- 完全二叉树、满二叉树、一般二叉树
- 满二叉树定理
- 二叉树的链表表示和数组表示
- 遍历：先序、中序、后序。
- 二叉树的重构，先序和中序，后序和中序
- 二叉查找树、堆

二叉查找树

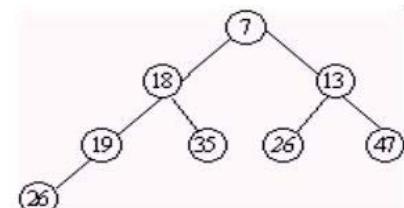
- 二叉查找树的构造方法
- 同一个无序序列，改变其数据元素的先后顺序，最后构造的二叉查找树将不一样；
- 中序遍历二叉查找树后可以得到有序序列
- 二叉查找树的插入、查找
- 平衡二叉查找树、完全不平衡二叉查找树
- 二叉查找树的删除

堆Heap的定义

- 若将堆看成是一棵以 k_1 为根的完全二叉树，则这棵完全二叉树中的每个非叶子结点的值均不大于（或不小于）其左、右孩子结点的值。由此可以看出，若一棵完全二叉树是堆，则根结点一定是这n个结点中的最小者或最大者。
- 下面给出两个堆的示例。



逆堆



正堆

堆、 Huffman编码树

- 建堆步骤：先写完全二叉树，然后调整堆。
- 堆的其它操作：插入、删除最大值、删除指定位置的值
- 堆的应用：优先队列
- 变长编码
- 建立Huffman编码树的步骤、对应的编码
- 反编码过程中的前缀特性
- 期望代码长度

第九章

- 静态查找表
- 重点：
- 动态查找表（二叉排序树）
- 散列的概念及其应用方法
冲突解决策略，开与闭散列方法，线性探查方法。

第十章

- 重点。
 - 各种排序技术的基本思想
 - 各种排序技术的基本过程
 - 各种排序技术的时空代价分析
 - 各种排序技术的性能比较

- 前三种为交换排序：只比较相邻的记录。
- 任何一种将比较限制在相邻两个元素之间进行的交换算法的平均时间代价都是 $O(n^2)$ 。
- 希尔排序 $O(n^{1.5})$ 、快速排序、归并排序、堆排序的具体排序步骤。
- 每种排序的时间代价、空间代价，稳定性
- 快速排序仿照二叉查找树、堆排序基于最大（小）堆。

各种内排序技术的代价比较

方 法	平均时间代价	最坏情况	最好情况	辅助空间	稳定性
Insert Sort	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	Yes
Bubble Sort	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	Yes
Select Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	No
Shell Sort		$O(n^{1.5})$		$O(1)$	No
Quick Sort	$O(n \log n)$	$O(n^2)$	$O(n \log n)$	$O(\log n)$	No
Merging Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$	Yes
Heap Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$	No
Radix Sort	$O(n)$	$O(n)$	$O(n)$	$O(n)$	Yes

第十一章

- 掌握基本概念。
磁盘结构、文件结构的相关概念
磁盘访问的步骤与代价分析
外排序的概念、基本方法及算法设计的规则、目的
- 问题: 待排序的数据量太大, 无法将其全部放置在主存中
 - 将数据驻留在磁盘上。
- 排序时, 一部分数据被读入主存中, 排序后返回磁盘。
- 外部排序算法的主要目标是尽量减少读、写磁盘的信息量。