数算期末复习笔记

1. 概述

计算：基于有穷观点的能行办法

数据抽象和过程抽象

抽象的概念

封装 继承 面向对象的方法

抽象数据类型 用形式化的过程描述封装

抽象数据类型 ADT 栈（列表 链式） 队列

算法及衡量算法的方法

三种基本控制流： 顺序 选择 循环

基本python语法及实用技巧（如内置数据类型、特殊方法）

1. 算法分析

程序与算法的关系：算法是一个数学上的模型 步骤的描述

程序在设计时与程序设计语言有关 实现时与编译环境有关

算法分析的概念：

计算资源及资源消耗指标，运行时间检测方法

算法复杂度的衡量指标 大O表示法

对算法的实现代码进行分析，以得到大O数量级

理解常见python数据类型中操作的大O数量级

（列表 元组 字典各种操作的大O数量级！）

（排序的算法复杂度！）

1. 基本数据结构

线性数据结构

理解ADT的不同实现方案及其复杂度分析

线性 两种方案 1、挨个排 2、链式

栈的概念、特性和ADT Stack

（栈在括号匹配 进制转换 前缀后缀表达式转换及其求值问题的算法）

队列的概念、特性和ADT Queue

队列在热土豆问题、打印任务问题求解的算法

双端队列的概念、特性和ADT Deque

双端队列在回文词判定问题上的算法

列表的概念、特性和ADT List、ADT OrderedList

无序表的链表实现、单链表、双链表实现方案的特点

有序表的实现、及python sort的扩展应用 （\_\_lt\_\_自己定义自己的数据结构排序规则）

（查找 二分查找）

1. 递归

递归的概念及初步例子

缩小问题规模

将大问题分解成小问题（最简单的问题 基本结束条件）

调用自身解决规模更小的问题

递归“三定律”：

用递归解决进制转换问题

递归调用的内部实现：与栈相关 递归的深度不能超过调用栈的容量

递归与自相似图形，理解绘制自相似图形的递归算法 二叉树

用递归解决河内塔问题和探索迷宫问题

动态规划算法策略

从兑换硬币问题对比递归算法和动态规划算法 如何避免递归爆炸（函数值缓存 直接查表用，降低算法复杂度）

1. 排序与搜索

顺序搜索算法 以及在无序表和有序表数据结构中的不同实现

二分搜索（有序表） 必须在能比大小或能排成一维顺序的对象上才能实现

分而治之

常数级别的查找

散列的概念 散列冲突概念 完美散列函数

（通过散列函数返回地址）

散列函数设计的几种方法

散列冲突解决方案

开放定址法:线性探索

数据链法

抽象数据类型ADT Map及实现的算法分析

排序算法

根据数据特征来选择排序算法

1. 树及其算法

非线性数据结构 不是简单的前驱后继的关系 还有一对多 多对多

树的概念及例子，树的两种定义 一种是递归定义

实现树的方法：嵌套列表 节点链接

树的应用：解析树（语法树和表达式树） 表达式树的建立算法 利用表达式解析树求值

树的遍历：前序 中序 后序 遍历

优先队列的概念，实现优先队列的经典方案 二叉堆

二叉搜索树BST及平衡树AVL树的概念及实现

七 图及其算法

图的概念，用图来表示的网络

熟悉图的术语及定义 ADT Graph

有向无圈图

图的实现方法：邻接矩阵 邻接链表法

词梯问题及广度优先搜索BFS

骑士周游问题及深度优先搜索DFS “启发式规则”

通用的深度优先搜索算法

DFS用于解决拓扑排序和强连通分支问题

路由选择 最短路径及Oijkstra算法

信息广播 最小生成树问题及Prim算法