

数据结构与算法 知识点复习

戴波

考查目标

- 掌握应用计算机解决问题的步骤（问题抽象与理解，算法分析与设计，算法实现与改进）；理解数据结构与算法的**基本概念**，掌握数据的**逻辑结构、存储结构及其差异**，各种基本操作的**实现**，以及算法的**最坏时间复杂度分析**。
- 在掌握基本的数据处理原理和方法的基础上，能够对算法进行基本的时间复杂度和空间复杂度的分析；能够选择合适的**数据结构和方法进行问题求解**，具备采用C或C++语言**设计程序与实现算法**的能力（实际应用能力的强调，但重点是算法设计能力，而不是单纯的编程能力的考查）

第一章 绪论

- 知识点1：数据结构课程的意义及重要性
- 知识点2：计算机问题求解过程（问题的理解即计算思维、数据结构选择和设计、算法选择和设计、算法分析、程序实现）
- 知识点3：《DS》主要研究内容
- 知识点4：数据结构概念及相关术语
- 知识点5：算法概念和特性（**有穷，可行，确定，功能**）
- 知识点6：算法分析基础（**运行时间，时空复杂度，ASL**）
- 重点1：数据结构作用认识和理解（知识点1、2）
- 重点2：数据结构概念及相关术语（知识点3、4）
- 重点3：算法概念及算法分析（知识点5、6）

第二章 线性表

- 知识点 2-1：存储结构——顺序存储和链式存储及其存储特点。
- 知识点 2-2：**线性表**的基本操作——查找、插入（入栈、入队）、删除（出栈、出队）在顺序存储及链式存储上的具体实现。
- 知识点 2-3：各种变形链表（循环链表、双向链表、带头结点的链表等）的含义及基本操作的实现。
- 知识点 2-4：栈和递归的关系、递归的特点及实现方法。
- 知识点 2-5：循环队列的判满、判空方法。

顺序与链式存储结构比较

顺序存储结构的特点

- 逻辑上相邻的元素，其物理位置也相邻；
- 可随机存取表中任一元素
- 必须按最大可能长度预分存储空间，存储空间利用率低，表的容量难以扩充，是一种静态存储结构
- 插入删除时，需移动大量元素，平均移动元素为 $n/2$

链式存储结构的特点

- 逻辑上相邻的元素，其物理位置不一定相邻；元素之间的邻接关系由指针域指示
- 是非随机存取存储结构；对链表的存取必须从头指针开始
- 是一种动态存储结构；
- 插入删除运算非常方便；只需修改相应指针值

• 顺序存储和链式存储各有优缺点？

- 顺序存储的优点是存储密度大 ($= 1$)，存储空间利用率高。缺点是插入或删除元素时不方便。
- 链式存储的优点是插入或删除元素时很方便，使用灵活。缺点是存储密度小 (< 1)，存储空间利用率低。
- 事实上，链表插入、删除运算的快捷是以空间代价来换取时间。



数组地址计算公式

则行优先存储时的地址公式为：

$$\text{Loc}[i,j] = \text{Loc}[c1,c2] + ((i-c1)*w2 + (j-c2))*L$$

• 二维数组列优先存储的通式为：

$$\text{Loc}[i,j] = \text{Loc}[c1,c2] + ((j-c2)*w1 + (i-c1))*L$$

考察目标要求

- 需要识记关键字、主关键字、次关键字的含义；静态查找与动态查找的含义及区别；**平均查找长度ASL**的概念及在各种查找算法中的计算方法和计算结果，特别是一些典型结构的ASL值，要会计算各种查找方法在**查找成功**和查找不成功时平均查找长度的计算

考察目标要求（续）

- Hash表相关的概念，冲突解决方法的选择和冲突处理过程的描述。
- 掌握**建立hash表**和查找hash表的过程。
- 要熟练掌握顺序表、链表、二叉树上的查找方法，特别要注意顺序查找、二分查找的适用条件(比如链表上用二分查找就不合适)和算法复杂度。

排序主要知识点

- 知识点7-1：排序的基本概念
- 知识点7-2：插入排序，直接插入排序、二分插入排序、希尔排序
- 知识点7-3：交换排序，冒泡排序、快速排序、
- 知识点7-4：选择排序，简单选择排序、堆排序
- 知识点7-5：2路归并排序
- 知识点7-6：基数排序
- 知识点7-7：各种排序方法的比较

内部排序章节大纲要求

- 内部排序既是重点，又是难点。排序算法众多，各种不同算法还有相应的一些概念定义需要记住。选择题常见的问题包括：**不同排序算法的复杂度，给定数列要求给出某种特定排序方法运行一轮后的排序结果，或者给出初始数列和一轮排序结果要求选择采用的排序算法，给定时间、空间复杂度要求以及数列特征要求选择合适的排序算法等等**。各种排序算法在有序输入情况下的时空复杂度比较。如果排序这一考点出现在综合应用题中则常与数组结合来考查。

- 哪些排序算法时间复杂度是 $O(n^2)$ ，哪些是 $O(n\log n)$ ？哪些空间复杂度是 $O(1)$ ，哪些是 $O(\log n)$ ，哪些是 $O(n)$ ？
- 哪些排序算法的时间复杂度和输入数据无关，哪些和输入数据有关？

第三章 树

- 知识点 3-1: 二叉树和树的递归定义、基本术语和概念、**基本性质**、基本操作, 满二叉树和完全二叉树的概念及特征。
- 知识点 3-2: 二叉树、树及森林的顺序存储及链式存储,
- 知识点 3-3; 各种**遍历方法**, **相互间的转换**。
- 知识点 3-4: 二叉排序树定义及基本操作,**AVL树**的基本操作
- 知识点 3-5: **哈夫曼树**的构建和哈夫曼编码
- 知识点 3-6: **堆**(优先队列)的存储结构, 建立堆与调整堆, 堆排序。

树

- 二叉树的基本概念
 - 3个结点的二叉树有五种形态
 - 节点的度、叶子, 孩子, 双亲, 子孙, 层次, 兄弟, 堂兄弟, 深度, 满二叉树, 完全二叉树等
 - 注意理解二叉树的度不一定是2。二叉树子树有序, 树的子树无序。3个结点的二叉树有5种形态, 树只有2种形态。
- 二叉树的性质
 - 二叉树的6个基本性质
 - 注意完全二叉树的树高, 叶子和结点数关系, 采用顺序存储结构的方法。

树

- 二叉树的存储
 - 顺序存储 (按层序存放, 存储空指针)
 - 链式存储
 - 二叉链表
 - 三叉链表
- 二叉树的遍历
 - 递归遍历
 - 先根序
 - 中根序
 - 后根序
 - 非递归遍历 (注意堆栈的控制和层次遍历 (用队列))

树

- 层次遍历
- 二叉树遍历算法的应用
 - 计算二叉树的叶子结点
 - 计算二叉树的高度
 - 根据遍历结果复原二叉树
- 二叉树的变形
 - 二叉排序树
 - **查找**
 - **插入**
 - 删除 (三种情况)
 - 会写代码完成各种操作

树

- 平衡二叉树
 - 4种失衡调整方法 (RR, LL, LR, RL)
- 哈夫曼树
 - 构建哈夫曼树
 - 哈夫曼编码
 - 注意哈夫曼树根权值固定, 但是各子树根的权值是否也是最小?
- 堆
 - 堆的建立, 调整
 - 堆排序
 - 堆的数据元素之间的关系
 - 会写代码: 判断一颗树是否堆; 堆的建立; 出/入优先队列
 - 时间复杂度分析

树

- 树和森林
 - 4种存储结构
 - 树, 森林, 二叉树的互转换 (孩子-兄弟法)

树

- 树的应用
 - 表达式树与算数表达式求值 (前, 中, 后缀表达式)
 - 括号配对的判断

掌握各种算法的递归实现!

第四章 图

- 知识点 4-1: 图的基本概念和术语
- 知识点 4-2: 基本操作和存储方法——邻接矩阵、邻接表、逆邻接表
- 知识点 4-3: 图的遍历方法——深度优先和广度优先,
- 知识点 4-4: 图的生成树和最小生成树 (普里姆和克鲁斯卡尔)。
- 知识点 4-5: 图中两结点以及所有结点间最短路径的求取方法——迪杰斯特拉和弗洛伊德方法,
- 知识点 4-6: 有向无环图的拓扑排序和关键路径求取。

图论 考查内容

- 掌握图以及基于图的各种**定义**，**存储方式**。
- 熟练掌握图的**深度遍历和广度遍历算法**（Prim, Dijkstra, topsort, critical path），这是用图来解决应用问题时常用的算法基础。
- 需要掌握基于图的多个算法，能够以**手工计算**的方式在一个给定的图上执行特定的算法求解问题。常见的应用问题直接给出或经过抽象，会成为下列问题：最小生成树求解(PRIM算法和KRUSKAL算法，两种方法思想都很简单，但要注意不要混淆这两种方法)，拓扑排序问题，关键路径问题(能做出表格找出关键路径)，最短路径问题

分治递归，贪心与动态规划算法

- 知识点5-1：递归的基本概念；递归方法及特点；分治算法特点及求解问题的方法。**主方法求解递归关系**
- 知识点5-2：贪心算法的性质和要素（贪心选择性质和最优子结构性质）
- 知识点5-3：动态规划定义和**求解的基本步骤**（寻找最优子结构、建立递归关系、自底向上求解、计算原问题解）
- 知识点5-4：贪心算法与动态规划的区别

分治递归，贪心与动态规划算法

- 比如给出1个递归问题，分析特点确定采用的方法。如果具有最优子结构性质，且子问题没有交叉，考虑分治算法。
- 如果明确要求用贪心算法，要分析是否具有贪心选择性质并进行证明。
- 如果子问题有交叉的最优子结构性质，采用动态规划或者备忘录法进行求解，并分析复杂度。

- 求解0-1背包问题。给定4种物品和一个背包。物品 i 的重量是 w_i ，其价值为 v_i ，背包的容量为 C 。已知 $V=(3,5,8,4)$ ， $W=(2,5,4,3)$ ， $C=10$ 如何选择装入背包的物品，使得装入背包中的物品总价值最大？画出求解图表并给出最优解与最优值。（6分）

31

定义递归式子

定义 $m(i, j)$ = 背包容量为 j , 由 $1, \dots, i$ 个物体装填背包问题的最优值

Case 1: $m(i, j)$ 不选择第 i 个物体.

$m(i, j) = m(i-1, j)$, 即当第 i 个物体不装填背包所产生的最大价值, 当

子问题有重复,
考虑动态规划解决

Case 2: $m(i, j)$

新的重量限制为 $j - w_i$

$m(i, j) = m(i-1, j - w_i)$, 即为新重量限制下, $\{1, \dots, i-1\}$ 个物体装填背包所产生的最大价值

$$m(i, j) = \begin{cases} 0 & \text{if } i = 0 \\ m(i-1, j) & \text{if } w_i > j \\ \max \{m(i-1, j), v_i + m(i-1, j - w_i)\} & \text{otherwise} \end{cases}$$

$V = (3, 5, 8, 4)$, $W = (2, 5, 4, 3)$, $C = 10$

价值 \rightarrow

物体 \downarrow

	0	1	2	3	4	5	6	7	8	9	10
NULL	0	0	0	0	0	0	0	0	0	0	0
{1}	0	0	3	3	3	3	3	3	3	3	3
{1,2}	0	0	3	3	5	5	8	8	8	8	8
{1,2,3}	0	0	3	3	8	8	11	11	11	13	13
{1,2,3,4}	0	0	3	4	8	8	11	12	12	15	15

最优解 1, 3, 4

最优值 15

- 给定数组 $a[1 \dots n]$, 设计一个算法, 在最坏情况下用 $n + \log n - 2$ 次比较找出 $a[1 \dots n]$ 中元素的最大值和次大值。 (5分)

锦标赛排序, $n-1$ 次比较建立锦标赛树, 找到最大值。

将最大值置为最小, 进行 $\log n(2n-1) = 1 + \log n$ 次比较, 总共: $n + \log n$ 次比较!

34

给定数组 $a[1 \dots n]$, 设计一个算法, 在最坏情况下用 $n + \log n - 2$ 次比较找出 $a[1 \dots n]$ 中元素的最大值和次大值

- (1) 首先将 $a[1 \dots n]$ 分为 2 组, $a[1 \dots k]$ 和 $a[k+1 \dots n]$, 其中 $k = n/2$ 。然后作 k 次比较: $a[i]$ 和 $a[i+k]$, 其中 $1 \leq i \leq k$ 。当 $a[i] > a[i+k]$ 时, 交换它们的位置。这样经过 k 次比较后我们有 $a[i] \leq a[i+k]$, 其中 $1 \leq i \leq k$ 。 (2分)
- (2) 至此, 再递归地在 $a[k+1 \dots n]$ 中找出其中的最大数 $a[p]$ 和次大数 $a[q]$ 。 (2分)
- (3) 容易看出, $a[p]$ 即为 $a[1 \dots n]$ 中的最大数。而 $a[1 \dots n]$ 中的次大数只能是 $a[q]$ (最后的同一组) 或 $a[p-k]$ (前面所有 $a[p]$ 所在的同一组)。 (1分)

复杂度: 最大数: $n/2 + n/4 + \dots + 1 = n$

次大数: 与 p 在同一组的 $\log n$ 个数做 $\log n - 2$ 次比较

35

(2) 给定数组 `int array[n]`, n 为偶数。设计一个算法, 在最坏情况下用 $3n/2-2$ 次比较找出 `array` 中元素的最大值和最小值。

N 分成 $n/2$ 组数据, 则 $n/2$ 次比较将大数放到一组, 在这一组共 $n/2$ 个数据, 进行 $n/2-1$ 次比较找到最大数, $n/2-1$ 次比较找到最小数

36

- 求解2个序列的最长公共子序列
- 求解一个序列的最长递增子序列

- 现有 n 个活动的集合 $\{a_1, a_2, a_3, \dots, a_n\}$, 其中每个活动都要求使用同一资源, 如演讲会场等, 而在同一时间内只有一个活动能使用这一资源, 每个活动 a_i 都有一个权重 w_i 和要求使用该资源的起始时间 s_i 和结束时间 f_i , 并且 $s_i \leq f_i$ 。如果选择了活动 a_i , 则它在半开时间区间 $[s_i, f_i)$ 内占用资源。若区间 $[s_i, f_i)$ 与区间 $[s_j, f_j)$ 不相交, 则称活动 a_i 与活动 a_j 是相容的, 也就是说, 当 $s_j \geq f_i$ 或 $s_i \geq f_j$ 时, 活动 a_i 与活动 a_j 相容。活动安排问题就是要在所给的活动集合中选出具有最大权重和的相容活动子集合。(9分)
- (a) 用动态规划方法求解该问题, 需要写出算法思想步骤和递归关系式。(7分)
- (b) 现给定活动集合 $\{b_1, b_2, b_3, \dots, b_8\}$, 对应的开始时间集 $\{0, 1, 3, 3, 4, 5, 6, 8\}$, 结束时间集 $\{6, 4, 5, 8, 7, 9, 10, 11\}$, 权重集 $\{23, 12, 20, 26, 13, 20, 11, 16\}$, 求解该问题的最优解和最优值。(2分)

38

最优子结构与子问题分析

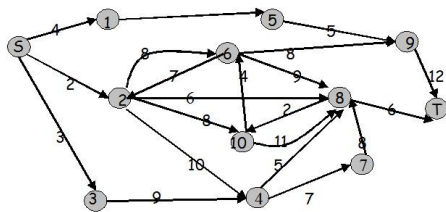
- 定义 $OPT(j)$ = 为 j 个活动 $1, 2, \dots, j$ 所获得的最大价值即最优解。

- 情况 1: OPT 中选择第 j 个活动。
 - 不能采用不兼容的活动
 - 必定包括剩余的 $1, 2, \dots, p(j)$ 个活动构成的最优解
- 情况 2: OPT 不选择第 j 个活动。
 - 必定包括活动 $1, 2, \dots, j-1$ 所形成的最优解

递归关系式

$$OPT(j) = \begin{cases} 0 & \text{if } j = 0 \\ \max \{ v_j + OPT(p(j)), OPT(j-1) \} & \text{otherwise} \end{cases}$$

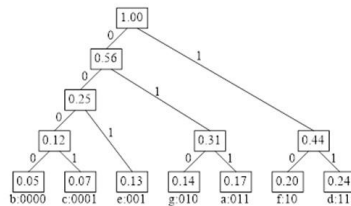
- 用Dijkstra算法求解下图中从S到T的最短路径，图中边旁边的数字代表该条边的长度。请写出Dijkstra算法求解步骤。求最短路径的计算中要求给出计算步骤和最后的最短路径以及它的长度。



41

- 给定一个包含100个字符的文本文件，文件中包含a,b,c,d,e,f,g七个字符，其中各个字符在文件中出现的频率如下： $f_a=0.17$, $f_b=0.05$, $f_c=0.07$, $f_d=0.24$, $f_e=0.13$, $f_f=0.20$, $f_g=0.14$ 。现在要求用哈夫曼编码对该文件进行编码使得文件长度达到最小。要求画出对应的哈夫曼树，给出每个字符的编码，并计算出编码后的文件长度。

43



- 编码后的文件长度为
- $100 * (0.17 * 3 + 0.05 * 4 + 0.07 * 4 + 0.24 * 2 + 0.13 * 3 + 0.20 * 2 + 0.14 * 3) = 268 \text{ bit}$

44

(1) 计算矩阵连乘积。给定 $n=4$ 个矩阵 $\{A, B, C, D\}$ 。要求以最小的计算量完成这 n 个矩阵的连乘积 $S = A_{35,40} \times B_{40,20} \times C_{20,10} \times D_{10,15}$ 。

(2) 0/1 背包问题。给定 N 中物品和一个背包。物品 i 的重量是 w_i ，其价值 V_i ，背包的容量为 C 。问应该如何选择装入背包的物品，使得装入背包的物品的总价值为最大？

(3) 有编号分别为 a, b, c, d, e 的五件物品；它们的重量分别是 2, 2, 6, 5, 4；它们的价值分别是 6, 3, 5, 4, 6。现在给你个承重为 10 的背包，如何让背包里装入的物品具有最大的价值总和。

(4) 旅行计划规划。某同学想利用悠长假中的连续 14 天安排一系列激动人心旅行计划。他拟出的旅行活动目的地有五个 $\{a, b, c, d, e\}$ ，以 14 天为单位，五项活动的启动日期分别是 $S = \{6, 1, 7, 6, 11\}$ ，结束日期分别是 $T = \{8, 5, 9, 13, 14\}$ ，将获得经验值 $V = \{2, 3, 5, 11, 7\}$ 。请安排其旅行计划，以使得获得经验值最多。要求写出动态规划的递归关系式，并求解其最优解。

46

- 小明去看望爷爷，准备了7种营养价值不同的米作为礼物。每种米的营养价值和最大重量如表所示。由于每个包装盒只能装1000g米，请问如何选择各种米使得组合米营养价值最大？

- 小马开发了一款名字叫“厨神”的app软件，吸引了上万人使用。每个人需要登记个人信息，爱好。他们可以申请厨师身份，通过审核成为厨师，也可以保持吃客的身份。每个人可以关注自己喜欢的菜式，喜欢的厨师，也可以取消关注。请问app使用者之间是什么关系？采用什么样的存储结构最合适？

- 如图有 n 个村庄，需要在一个村庄建医院，如何建可以使得其他村到医院的平均距离最近？

简答题

- (1) 描述递归方法及其特点
- (2) 描述分治方法及其特点
- (3) 描述动态规划法及其特点

分治法的适用条件

分治法所能解决的问题一般具有以下几个特征：

- 该问题的规模缩小到一定的程度就可以容易地解决；
- 该问题可以分解为若干个规模较小的相同问题，即该问题具有最优子结构性质
- 利用该问题分解出的子问题的解可以合并为该问题的解；
- 该问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子问题。

动态规划

- 动态规划算法与分治法类似，其基本思想也是将待求解问题分解成若干个子问题
- 但是经分解得到的子问题往往**不是互相独立的**。不同子问题的数目常常只有**多项式量级**。在用分治法求解时，有些子问题被重复计算了许多次。

用递归算法进行设计，很多子问题重复计算，无形的增加了算法计算量！

如何减少子问题的重复计算则是动态规划算法的关键解决思想！

动态规划

- 动态规划算法与分治法类似，其基本思想也是将待求解问题分解成若干个子问题
- 但是经分解得到的子问题往往**不是互相独立的**。不同子问题的数目常常只有**多项式量级**。在用分治法求解时，有些子问题被重复计算了许多次。
- 如果能够保存已解决的子问题的答案，而在需要时再找出已求得的答案，就可以避免大量重复计算，从而得到多项式时间算法。