

昆明理工大学硕士研究生入学考试《数据结构与算法分析》考试大纲

第一部分 考试形式与试卷结构

一、试卷满分及考试时间

试卷满分为 150 分，考试时间为 180 分钟。

二、答题方式

答题方式为闭卷、笔试。

三、试卷内容结构

基本概念、基本知识、基本方法约占 40%~50%;

综合应用、算法和程序设计与算法分析约占 50%~60%。

四、试卷题型结构

试卷共 150 分,基本的考试题型为:

- (1)单项选择题和多项选择题;
- (2)填空题(基本概念、基本知识、基本方法);
- (3)画图题;
- (4)简答题;
- (5)应用题(求解问题);
- (6)算法和程序设计填空题;
- (7)算法和程序设计与分析题;
- (8)其它题型。

五、特别说明

用 C 语言(或 C++)描述算法和程序设计。

第二部分 考察的知识及范围

1. 数据结构和算法

数据结构、存储结构的概念；数据类型与抽象数据类型；算法的概念，用 C/C++ 描述算法和程序设计。

2. 线性表

线性表的定义和基本操作；线性表的抽象数据类型；线性表的顺序存储结构，

应用举例；线性表的链式存储结构(单链表, 双链表, 循环链表), 应用举例。

3. 栈

栈的定义和基本操作；栈的抽象数据类型；顺序栈，链式栈；栈和递归算法，算术表达式求值, 其它应用。

4. 队列

队列的定义和基本操作；队列的抽象数据类型；顺序队列，链式队列；双端队列的定义和基本操作；应用举例。

5. 数组和广义表

(1) 数组

数组的定义和基本操作；数组的顺序存储结构，应用举例；特殊矩阵和稀疏矩阵的压缩存储。

(2) 广义表

广义表的定义和基本操作, 广义表的抽象数据类型, 广义表的存储结构。

*广义表运算的实现举例。

6. 字符串

字符串的定义和基本操作, 字符串的存储结构, 字符串操作的实现举例, 字符串和模式匹配。

7. 树和二叉树

(1) 树的基本概念和基本操作, 树的抽象数据类型。

(2) 二叉树的概念和性质, 特殊二叉树；二叉树的存储结构；

(3) 二叉树的生成与建立。

(4) 遍历二叉树：前序遍历, 中序遍历, 后序遍历, 层次遍历。

(5) 二叉树其它操作实现举例。

(6) 线索二叉树的概念和存储结构, 二叉树的线索化, 线索二叉树的遍历。

(7) 树的存储结构, 树与二叉树之间的转换, 森林与二叉树之间的转换, 树和森林的遍历。

(8) 树的路径长度和带权路径长度, 哈夫曼树 (Huffman) 的概念, 哈夫曼算法, 哈夫曼编码树。

(9) 二叉排序树的概念和基本操作, 二叉排序树的建立, 二叉排序树其它操作实现举例。

8. 图

(1) 图的基本概念和基本操作, 图的抽象数据类型。

(2) 图的存储结构：数组表示法 (邻接矩阵); 邻接表, 逆邻接表, 十字链表; 邻接多重表。

(3) 图的遍历：深度优先搜索法, 宽度优先搜索法, 求图的连通分量。

(4) 生成树、最小生成树的概念; 克鲁斯卡尔 (Kruskal) 算法, 普里姆 (Prim) 算法。

* (5) 从一个顶点到其余各顶点的最短路径, 每对顶点之间的最短路径。

* (6) 拓扑排序和关键路径

9. 查找

(1) 查找的概念, 关键字比较次数, 平均查找长度。

(2) 顺序表的查找：顺序查找, 折半查找, 分块查找。

(3) 树表的查找：二叉排序树, 平衡二叉树。

(4) 哈希 (Hash) 表的查找：哈希表的概念, 哈希函数构造方法, 哈希表的建立和查找, 冲突处理方法。

10. 排序

(1) 排序的概念; 排序的稳定性; 比较关键字次数, 移动记录次数; 顺序表的排序, 链表表 (单链表) 的排序。

(2) 内排序方法与算法

(a) 交换排序:冒泡排序, 快速排序。

(b) 插入排序:直接插入排序, 2 路插入排序, 折半插入排序, 希尔排序。

(c) 选择排序:直接选择排序, 锦标赛排序, 堆排序。

(d) 归并排序。

(e) 基数排序。

(3) 各种排序算法的评价和应用。

11. 文件

(1) 文件的基本概念, 文件的基本操作。

(2) 文件的物理结构: 顺序文件, 索引文件与索引顺序文件, 直接存取文件,

链接文件和多重链表文件, 倒排文件。

*12. 外排序

外排序的基本过程, 初始归并段的生成, 多路平衡归并排序, 最佳归并树。

13. 算法分析

(1) 算法分析基础

(a) 熟悉渐近表示法, 掌握渐近符号 O 等的定义, 能判断一个较复杂的函数属于哪个渐近增长阶;

(b) 熟悉一些算法复杂度分析的方法, 比如说主定理法等, 能对结构复杂的算法进行分析。

(2) 算法设计基础

(a) 熟悉算法设计的三大技巧: 贪心算法、分而治之, 动态规划。

(b) 能证明各种算法的正确性。

(c) 能用这三大技巧设计相应的算法。

(3) NP 完备性理论及近似算法

(a) 了解并掌握 NP 完备性理论及其实际意义;

(b) 熟悉多项式规约。掌握证明一个问题 NP 完全性的基本方法和思路;

(c) 熟悉最小点覆盖、最大独立集等问题的 NP 完备性证明;

(d) 了解并掌握近似算法的设计步骤与技巧,掌握点覆盖等问题的近似算法的设计。

说明:带“*”号的章节为一般考查内容,其余为重点考查内容。