**首都经济贸易大学**

**硕士研究生入学考试《数据结构》（804）考试大纲**

**第一部分 考试说明**

一、考试目的

《数据结构》考试是为首都经济贸易大学招收计算机科学与技术专业研究生而设置的具有选拔性质的全国统一入学考试科目，其目的是科学、公平、有效地测试学生掌握大学本科阶段对本专业核心课程的掌握情况。

二、考试范围

主要考核考生对《数据结构》课程的掌握情况，具体包括：数据结构的基本概念、基本数据结构（线性结构、树形结构、图结构）、查找与排序算法、递归算法与非递归算法的设计、算法时间复杂度分析等方面。

三、考试基本要求

1. 掌握数据结构各主要章节的基本概念和具体算法，领会数据结构在分析与解决实际问题过程中的基本思想和具体算法。

2. 理解不同数据结构的特性及其在实际问题中的应用，能够运用所学知识设计合理的数据结构，并解决具体问题。

四、考试形式与试卷结构

（一）答卷方式：闭卷，笔试

（二）答题时间：180分钟

（三）题型及分值

本试卷满分为150分，具体题型及分值如下：

1.选择题：一般10题，每小题约2分，约占20分左右。

2.填空题：一般10题，每小题约2分，约占20分左右。

3.判断题：一般10题，每小题约2分，约占20分左右。

4.简答与应用题：一般5题，每小题约6分，约占30分左右。

5.算法分析与程序设计题：一般5-8题，每题8-12分左右，约占60分左右。此部分要求有算法设计的思路以及具体的代码，按过程分段评定分数。

五、参考书目

《数据结构（C语言版）》（第二版），人民邮电出版社，2021.12，严蔚敏、李冬梅、吴伟民编著。

**第二部分 考试内容**

1.数据结构的基本概念

了解数据结构的基本概念，如数据、数据元素、数据项等；理解算法描述和分析方法，以及数据结构在软件系统中的重要性；掌握数据的逻辑结构和存储结构，以及它们之间的关系；掌握算法的时间复杂度和空间复杂度，并能分析一般算法的性能。

2．线性表

了解循环链表的应用；了解顺序表和链表在性能上的优劣以选择最优存储结构；理解线性表的逻辑结构和存储方法；掌握如何在线性表结构上实现基本运算；能够利用这些知识设计算法并评估其时间复杂度，以解决实际问题。

3.栈和队列

了解递归工作栈的使用；理解栈、队列的逻辑结构及其运算；掌握栈与线性表差异，实现栈的基本操作；掌握队列与线性表差异，实现队列基本操作，包括循环队列；掌握队列上溢、下溢概念及判别；掌握循环队列优势及边界处理方法；能够运用栈、队列设计算法解决问题。

4. 串

了解串与线性表的关系；理解串的逻辑结构、存储结构及其串上的基本运算；掌握串实现的模式匹配算法，串的两种存储表示，用串实现的模式匹配算法及其时间性能分析；能够使用串操作函数构造与串相关的算法解决简单的应用问题。

5. 数组和广义表

了解广义表的有关概念及其与线性表的关系；广义表的括号表示和图形表示之间的转换；能够求给定的非空广义表的表头和表尾运算；理解多维数组的逻辑结构特征，多维数组的顺序存储结构及地址计算方式；掌握多维数组的存储方式、矩阵的压缩存储方式、广义表的定义及其求表头和表尾的运算；能够应用稀疏矩阵的压缩存储表示实现矩阵的转置和矩阵的乘法运算。

6. 树和二叉树

了解树、森林与二叉树的转换；了解森林存储结构及特点；理解树逻辑结构、术语及遍历；熟悉二叉树定义、性质及存储；掌握二叉树遍历算法及序列确定；能够设计基于遍历的算法；理解二叉树线索化；能够查找中序线索树中的结点；掌握最优二叉树和最优前缀码的概念和特点，并能够构造最优二叉树及对应的哈夫曼编码。

7.图

理解最短路径；掌握Dijkstra算法求单源最短路径，并能根据Dijkstra算法画出求单源最短路径的过程示意图；了解图的逻辑结构和术语；能够选择合适的图存储结构；掌握图的深度优先和广度优先遍历；掌握Prim和Kruskal算法求最小生成树；能够图示深度优先和广度优先生成树；掌握拓扑排序思想和步骤；能够分析拓扑排序失败原因；能够为给定有向图写出一个或多个拓扑序列。

8.查找

了解散列表及其相关概念；掌握散列函数选取原则、冲突原因及常用构造方法；理解查找重要性及查找算法效率评判标准；掌握顺序查找、二分查找、分块查找算法及其效率分析，并能根据实际问题选择适合的查找算法；掌握二叉查找树和B-树。

9.排序

能够通过比较排序的多个方面（如记录数、信息量、关键字结构、稳定性、空间需求、时间性能）来选择合适排序方法；理解排序在数据处理中的重要性及其“稳定”性含义；掌握五种内部排序方法的基本思想、算法实现和性能分析；掌握直接插入排序、冒泡排序、快速排序、归并排序算法基本思想和性能分析；掌握堆、小顶椎 、大顶堆、堆顶；掌握直接选择排序和堆排序（包括小顶堆、大顶堆）的基本思想、算法实现和性能分析。

**第三部分 题型示例**

一、选择题

若栈的输入序列为1、2、3、4，则不可能出现的输出序列为（ ）。

A.1、2、3、4 B. 4、3、2、1

C.4、3、1、2 D. 3、4、2、1

答案：C

二、填空题

假设三维数组A[10][9][8]按行优先顺序存储，若每个元素占3个存储单元，且首地址为100，则元素A[8][8][7]的存储地址是 。

答案：2041

三、判断题

某无向连通网只有唯一的一棵最小生成树，则该无向连通网各条边上的权值互不相同。（ ）

答案： X

四、简答与应用题

已知一棵二叉树的中序遍历和后序遍历的序列如下，画出这棵树，并写出该二叉树的前序遍历序列。

中序遍历序列：c, b, d, e, a, g, i, h, j, f

后序遍历序列：c, e, d, b, i, j, h, g, f, a

答案：

前序遍历序列:abcdefghij

五、算法分析与程序设计题

已知一个带有表头结点的单链表，结点结构为data |link,,假设该链表只给出了头指针list。在不改变链表的前提下，设计一个尽可能高效的算法，查找链表中倒数第K个位置上的结点（K为正整数）。若查找成功，算法输出该结点的data域的值，并返回1；否则，只返回0。要求：

（1）写出算法的基本设计思想；

（2）写出算法详细实现步骤；

（3）根据设计思想和实现步骤，采用相关程序设计语言描述算法，关键之处给出注释。

答案：

（1）算法的基本设计思想：

问题的关键是设计一个尽可能高效的算法，通过链表的一趟遍历，找到倒数第k个结点的位置。算法的基本设计思想是：定义两个指针变量p和q，初始时均指向头结点的下一个结点（链表的第一个结点)。p指针沿链表移动；当p指针移动到第k个结点时，q指针开始与p指针同步移动；当p指针移动到最后一个结点时，q指针所指示结点为倒数第k个结点。以上过程对链表仅进行一遍扫描。

（2）算法的详细实现步骤：

①count=0，p和q指向链表表头结点的下一个结点；

②若p为空，转⑤；

③若count等于k，则q指向下一个结点；否则，count=count+1；

④p指向下一个结点，转②；

⑤若count等于k，则查找成功，输出该结点的data域的值，返回1；否则，说明k值超过了线性表的长度，查找失败，返回0；

⑥算法结束。

（3）算法实现：

typedef int ElemType； //链表数据的类型定义

typedef struct LNode { //链表结点的结构定义

ElemType data； //结点数据

struct Lnode \*link； //结点链接指针

} \*LinkList；

int Search\_ k(LinkList list, int k){

//查找链表list倒数第k个结点，并输出该结点data域的值

Linklist p=list->link，q=list->link；

//指针p、q指示第一个结点

int count=0；

while(p != NULL) { //遍历链表直到最后一个结点

if(count < k) count++； //计数，若count < k只移动p

else q= q->link；p = p->link； // 之后让p、q同步移动

} //while

if(count < k)

return 0； // 查找失败返回0

else { //否则打印并返回1

printf("%d" ,q->data)；

return 1；

}

} //search\_k