
《数据结构》硕士研究生招生初试考试大纲

考试科目：834 数据结构

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、考试形式

考试形式为闭卷、笔试。

三、学习内容

(一) 数据结构基本概念

主要考核数据结构的基本概念和内涵，包括逻辑结构和存储结构的分类、逻辑结构和存储结构之间的关系；算法的含义及其特性、算法的时间复杂度分析方法。

学习要求：

1. 掌握数据结构、逻辑结构和存储结构的定义，以及逻辑结构和存储结构之间的关系。
2. 掌握逻辑结构和存储结构的分类，深刻理解顺序存储和链式存储结构。
3. 理解渐进时间复杂度和大 O 表示法。
4. 了解算法的含义及其基本特性。

(二) 线性表

主要考核线性结构的特点、线性结构的顺序存储和链式存储的定义、基本操作和简单应用。

学习要求：

1. 掌握顺序表的定义及基本操作，包括增加元素、删除元素、查找元素、求表长等。
2. 掌握带头结点的和不带头节点的单向链表的定义及基本操作，包括增加元素、删除元素、查找元素、求表长、判断表空等。
3. 掌握单向循环链表和双向链表的基本操作，包括增加元素、删除元素、查找元素、求表长、判断表空等。
4. 掌握基于线性表解决简单应用问题的方法。
5. 理解线性表的不同存储结构对线性表基本操作效率的影响。
6. 了解线性结构的特点。

(三) 栈和队列

主要考核栈和队列的特性、栈和队列的顺序存储和链式存储的定义、基本操作和简单应用。

学习要求：

1. 掌握顺序栈和链栈的定义及其基本操作，包括入栈、出栈、判断栈空、判断栈满等。
2. 掌握循环队列和链队列的定义及其基本操作，包括入队、出队、判断队空、判断队满等。
3. 掌握基于栈或者队列解决简单应用问题的方法。
4. 理解栈和队列的不同实现对栈和队列的基本操作效率的影响。
5. 理解栈和队列的特性。

(四) 数组和串

主要考核数组的存储方式、矩阵的压缩存储、字符串的简单模式匹配。

学习要求：

1. 理解数组的行主序和列主序存储方式。
2. 理解对称矩阵和三角矩阵这两种特殊矩阵的压缩存储方式。
3. 了解稀疏矩阵的三元组表压缩存储方式。
4. 了解字符串的简单模式匹配算法。

(五) 树和二叉树

主要考核二叉树的性质、二叉树链式存储的定义、二叉树的遍历方法及其简单应用、线索二叉树、赫夫曼树和赫夫曼编码。

学习要求：

1. 掌握二叉树的基本性质。
2. 掌握二叉树的先序、中序和后序遍历，以及二叉树遍历方法的应用。
3. 掌握二叉链表的定义。
4. 掌握赫夫曼树的构造方法、求赫夫曼编码的方法和带权路径长度的计算方法。
5. 理解树和二叉树的相关概念，如子树、叶子结点、结点的层次和树的深度等。
6. 理解线索二叉树的定义。
7. 了解二叉树的顺序存储。

8. 了解树的定义以及树与二叉树之间的转换方法。

(六) 图

主要考核图的基本概念、图的顺序存储和链式存储的定义、图的遍历方法及其简单应用、最小生成树、拓扑排序、关键路径、最短路径。

学习要求：

1. 掌握图的定义和相关概念，包括顶点的入度和出度、有向图、无向图、子图、连通图、连通分量、完全图等。
2. 掌握图的邻接矩阵和邻接表定义，深刻理解其含义。
3. 掌握图的深度优先和广度优先遍历方法及其实现，能用这两种遍历方法解决简单应用问题。
4. 掌握最小生成树的构造方法。
5. 掌握拓扑排序的方法。
6. 理解关键路径的计算方法。
7. 了解从源点到其余各点最短路径的计算方法。

(七) 查找

主要考核在静态查找表和动态查找表上执行的有代表性的查找算法。

学习要求：

1. 掌握折半查找的过程、算法实现和平均查找长度的计算方法。
2. 掌握二叉排序树的构造、基于二叉排序树的查找过程和平均查找长度的计算方法。
3. 掌握哈希表的构造方法和哈希查找的过程。
4. 理解静态查找表和动态查找表的区别。
5. 理解平衡二叉树的概念。
6. 理解影响哈希查找效率的因素。
7. 了解 B 树和 B+树的概念。

(八) 内部排序

主要考核插入排序、交换排序、选择排序和归并排序中有代表性的排序算法。

学习要求：

1. 掌握直接插入排序、快速排序、简单选择排序、堆排序和 2 路归并排序的操作过程和算法实现。

-
2. 理解直接插入排序、快速排序、简单选择排序、堆排序和 2 路归并排序的时间复杂度。
 3. 理解插入排序、交换排序、选择排序和归并排序这种分类方法的含义。
 4. 了解希尔排序、冒泡排序和基数排序的操作过程。
 5. 了解排序的相关概念，包括内部排序、外部排序、排序的稳定性等。

四、考核主要形式

- 1、选择、填空、判断题(涵盖较广，包括基本概念、简单计算、基本方法的简单运用等)；
- 2、解答题(基本原理和基本方法在具体问题上的运用，包括分析、构造和求解等)；
- 3、算法设计题(灵活运用数据结构知识，通过设计算法和实现程序，解决规模较小的具体问题)。