

《数据结构 (实践) 》(课程代码 : 13182)

自学考试实践考核大纲

华南理工大学自学考试

I 课程性质与目标

一、课程性质和特点

本课程是一门实践性极强的专业核心实践课程，聚焦数据结构的实际应用与算法实现能力培养。课程紧密结合计算机应用领域的实际开发需求，衔接《数据结构》理论课程中线性表、栈、队列、树、图、排序与查找等核心知识，通过系统化的实操训练，助力考生将数据组织、存储及运算的理论知识转化为解决实际问题的编程能力，适配软件开发、数据处理等相关岗位对数据结构应用能力的要求。

二、课程目标

通过本课程实践学习，考生能够熟练运用至少一种编程语言（如 C++ 语言、Python 等）实现各类数据结构的存储与基本运算，掌握算法设计与分析的实操方法，具备运用数据结构知识解决实际问题的综合能力，为后续职业发展和深入学习筑牢实践基础。具体目标如下：

1. 熟练掌握实践开发环境的搭建与使用，掌握数据结构相关程序的编写、运行与调试流程。
2. 能熟练实现线性表（顺序表、链表）的存储结构，完成初始化、插入、删除、查找等基本运算的编程实现。
3. 掌握栈与队列（顺序栈、链栈、循环队列、链队列）的存储实现与核心运算编程，能运用栈和队列解决简单实际问题。
4. 理解树与二叉树的逻辑结构，能实现二叉树的链式存储、遍历（前序、中序、后序）及哈夫曼树的构造与编码等实操任务。
5. 掌握图的邻接矩阵与邻接表存储实现，能完成深度优先搜索（DFS）、广度优先搜索（BFS）等遍历算法的编程实现。
6. 熟练实现常用内部排序算法（冒泡排序、快速排序、直接插入排序、简单选择排序等）和查找算法（顺序查找、折半查找、哈希查找等），并能分析算法性能。

三、与相关课程的联系与区别

学习本课程前，考生需具备计算机程序设计基础知识（如 C++ 语言、Python 等编程语言基础），已完成《数据结构》理论课程的学习，熟练掌握各类数据结构的逻辑特性、存储原理及算法基本思想等核心知识点。

本实践课程是理论课程的延伸与落地，侧重动手实操能力与问题解决能力的培养，不同于理论课程对概念、原理、思想的讲解，重点解决“如何用编程实现数据结构”“如何用数据结构与算法解决实际问题”的核心诉求，为后续复杂项目开发、毕业设计等课程提供关键的实操支撑。

四、课程的重点和难点

重点：线性表（链表）的存储实现与基本运算编程、栈与循环队列的实操实现、二叉树的遍历算法编程、图的存储与遍历实现、常用排序与查找算法的编程实现。

难点：链表的插入与删除算法的边界条件处理、循环队列的队空与队满判断及实现、二叉树遍历的非递归算法实现、图的复杂应用（如最短路径、最小生成树）算法编程、多数据结构融合解决综合实际问题的能力。

五、考核目标

本课程实践考核目标要求考生熟练运用指定编程语言及开发工具，结合数据结构理论知识，独立完成各类数据结构的实现与算法编程任务。能够灵活运用线性表、栈、队列、树、图等数据结构，以及排序、查找算法，解决实际应用中的数据组织与处理问题。考生需在完成《数据结构》理论课程学习后，参加本实践考核。

II 课程内容与考核要求

第1章 线性表 (包含教材第2章全部内容)

一、课程内容

- 1.1 顺序表的存储实现(结构体定义、数组存储);
- 1.2 顺序表的初始化、插入、删除、查找、遍历等基本运算编程;
- 1.3 单链表、双向链表、循环链表的存储实现(节点定义、指针连接);
- 1.4 链表的建立(头插法、尾插法)、插入、删除、查找、遍历等基本运算编程;
- 1.5 线性表的综合应用(如两个有序线性表的合并、线性表的逆置、数据去重)。

二、学习目的与要求

- 2.1 熟练掌握顺序表的存储结构设计及基本运算编程,理解顺序存储的优缺点。
- 2.2 熟练掌握链表的节点定义与指针操作,能独立完成各类链表的建立与基本运算编程。
- 2.3 能根据实际需求选择合适线性表存储结构,运用线性表知识解决简单综合问题。

三、考核知识点与考核要求

3.1 顺序表实现

综合应用:利用数组与结构体,完成顺序表的存储定义;熟练实现顺序表的初始化、插入、删除及查找运算,能处理插入/删除中的边界条件。

3.2 链表实现

综合应用:利用结构体与指针,完成链表节点定义;熟练实现单链表的头插法、尾插法建立,以及插入、删除、遍历运算;领会双向链表、循环链表的实现逻辑,能完成基本运算编程。

3.3 线性表应用

综合应用:应用线性表知识,完成有序线性表合并、数据去重等综合任务的编程实现。

第2章 栈和队列 (包含教材第3章全部内容)

一、课程内容

- 1.1 栈的存储实现(顺序栈、链栈);
- 1.2 栈的初始化、入栈、出栈、取栈顶元素等基本运算编程;
- 1.3 队列的存储实现(顺序队列、循环队列、链队列);
- 1.4 队列的初始化、入队、出队、取队头元素等基本运算编程;
- 1.5 栈和队列的应用(表达式求值、括号匹配、简单任务调度)。

二、学习目的与要求

- 2.1 熟练掌握栈的两种存储实现方式,能独立完成栈的各类基本运算编程。
- 2.2 理解循环队列的设计原理,熟练掌握循环队列的队空与队满判断方法,能完成队列的各类基本运算编程。
- 2.3 能运用栈和队列的特性解决表达式求值、括号匹配等实际问题。

三、考核知识点与考核要求

3.1 栈的实现

应用:应用数组/结构体实现顺序栈,或用链表实现链栈;熟练实现入栈、出栈等运算;

综合应用:利用栈解决表达式求值、括号匹配问题。

3.2 队列的实现

领会:循环队列的设计逻辑;

应用:利用数组/结构体实现循环队列,或用链表实现链队列;熟练实现入队、出队等运算;

综合应用:应用队列解决简单任务调度问题。

第3章 数组、广义表和字符串

（包含教材第4章全部内容）

一、课程内容

- 1.1 数组的存储实现（行优先、列优先）与元素地址计算编程；
- 1.2 稀疏矩阵的压缩存储（三元组表、十字链表）实现；
- 1.3 字符串的存储实现（定长顺序存储、堆分配存储）；
- 1.4 字符串的赋值、连接、比较、查找（模式匹配）等基本运算编程。

二、学习目的与要求

- 2.1 掌握数组的存储方式，能准确计算数组元素的存储地址。
- 2.2 理解稀疏矩阵压缩存储的意义，能实现三元组表存储及基本运算。
- 2.3 熟练掌握字符串的基本运算编程，能实现简单的模式匹配算法。

三、考核知识点与考核要求

3.1 数组与稀疏矩阵

领会：数组存储原理，计算数组元素地址；领会稀疏矩阵压缩存储思想，实现三元组表的定义与基本运算。

3.2 字符串操作

应用：应用字符数组或指针，实现字符串的赋值、连接、比较运算；

综合应用：利用简单匹配算法实现字符串模式匹配。

第4章 树与二叉树

（包含教材第5章全部内容）

一、课程内容

- 1.1 二叉树的链式存储实现（节点定义、左右指针连接）；
- 1.2 二叉树的初始化、创建（根据遍历序列）、前序/中序/后序遍历（递归与非递归）编程；
- 1.3 树与森林转换为二叉树的编程实现；
- 1.4 哈夫曼树的构造与哈夫曼编码的生成编程。

二、学习目的与要求

- 2.1 熟练掌握二叉树的链式存储结构设计，能根据遍历序列创建二叉树。
- 2.2 能独立完成二叉树三种遍历的递归算法编程，理解并尝试实现非递归遍历算法。
- 2.3 掌握哈夫曼树的构造原理，能编程实现哈夫曼树构建与编码生成。

三、考核知识点与考核要求

3.1 二叉树存储与遍历

应用：应用结构体与指针，实现二叉树链式存储；熟练实现前序、中序、后序递归遍历算法；

领会：非递归遍历算法逻辑，能完成简单实现。

3.2 树与二叉树转换

领会：树与森林向二叉树转换的规则，能完成简单转换的编程实现。

3.3 哈夫曼树

综合应用：应用贪心算法思想，编程实现哈夫曼树的构造，并生成对应的哈夫曼编码。

第5章 图结构

（包含教材第6章全部内容）

一、课程内容

- 1.1 图的存储实现（邻接矩阵、邻接表）；
- 1.2 图的初始化、顶点插入、边插入等基本运算编程；
- 1.3 图的深度优先搜索（DFS）、广度优先搜索（BFS）算法编程；

二、学习目的与要求

- 2.1 熟练掌握图的邻接矩阵与邻接表存储实现，能根据实际图结构完成存储编程。
- 2.2 能独立完成DFS与BFS遍历算法的编程实现，理解两种遍历方式的差异。

三、考核知识点与考核要求

3.1 图的存储实现

综合应用：应用二维数组实现邻接矩阵，或用链表实现邻接表；熟练完成图的初始化、顶点与边的插入运算。

3.2 图的遍历

综合应用：应用递归或栈/队列，实现 DFS 与 BFS 遍历算法，能输出遍历序列。

第 6 章 内部排序 (包含教材第 7 章全部内容)

一、课程内容

- 1.1 插入排序（直接插入排序、希尔排序）算法编程；
- 1.2 交换排序（冒泡排序、快速排序）算法编程；
- 1.3 选择排序（简单选择排序、堆排序）算法编程；
- 1.4 归并排序算法编程；
- 1.5 各类排序算法的性能测试与比较。

二、学习目的与要求

- 2.1 熟练掌握至少 4 种常用内部排序算法的编程实现，理解各算法的核心逻辑。
- 2.2 能分析各类排序算法的时间复杂度、空间复杂度及稳定性。
- 2.3 能根据数据特点选择合适的排序算法，完成数据排序任务。

三、考核知识点与考核要求

3.1 基础排序算法：

综合应用：应用循环、条件判断等语法，熟练实现直接插入排序、冒泡排序、简单选择排序算法。

3.2 高级排序算法

领会：快速排序、堆排序、归并排序的算法思想，能完成编程实现。

3.3 排序算法分析

应用：应用算法复杂度分析方法，比较各类排序算法的性能；

综合应用：能根据数据场景选择合适的排序算法并编程实现。

第 7 章 查找 (包含教材第 8 章全部内容)

一、课程内容

本章节聚焦查找的核心基础算法，哈希表及其查找不纳入实践考核

- 1.1 顺序查找、折半查找算法编程；
- 1.2 二叉排序树的构建与查找算法编程；

二、学习目的与要求

- 2.1 熟练掌握顺序查找与折半查找算法的编程实现，理解折半查找的适用条件。
- 2.2 能独立完成二叉排序树的构建、插入与查找算法编程。

三、考核知识点与考核要求

3.1 基础查找算法

综合应用：应用循环、条件判断，实现顺序查找与折半查找算法；能计算两种算法的平均查找长度。

3.2 二叉排序树查找

综合应用：应用二叉树知识，实现二叉排序树的构建、插入与查找算法。

III 关于实践考核实施的说明

一、关于能力层次的说明

大纲考核要求中，“领会”“应用”“综合应用”三个能力层次的含义如下：

领会：掌握实践中的基本操作步骤、数据结构的存储设计思路及算法的核心逻辑；理解各类知识点的实操应用场景；能读懂给定的数据结构与算法代码，分析其功能。

应用：熟练掌握单一数据结构或算法的实操技能，能独立完成针对性较强的实操任务（如单一排序算法实现、单链表基本运算等）；能处理实操过程中的常见问题（如边界条件、

语法错误等)。

综合应用：融合多个章节的知识点与实操技能，完成综合性较强的实践任务（如结合链表与排序实现有序链表构建、结合图遍历与最短路径实现路径规划等）；具备分析实际问题、选择合适数据结构与算法并编程解决的能力。

二、自学教材

本实践课程使用教材与理论课程一致，为《数据结构》，幸运伟、陈朔鹰主编，机械工业出版社2023年版。教材中第1章，第2章2.4，第6章6.5及第7章7.6和7.7，第8章的8.4不纳入实践考核范围。

三、关于命题考试的要求

1. 本课程命题严格依据本大纲规定的考试内容和考核目标，突出重点内容，侧重综合应用能力的考查，确保考试内容覆盖课程的核心知识点与实操技能。

2. 考核场所为计算机机房，采用现场实操考核方式，考生需在规定时间内完成题目要求的程序编写、编译、运行、调试及结果提交等任务。

3. 考核指定编程语言为C++语言、Java或Python（考生可任选其一），配套开发环境为Visual Studio、Eclipse、Python IDLE（Python）。

4. 考生参加考核可携带指定自学教材，不得携带其他参考资料和电子设备（如U盘、手机等）。

5. 考核时间为120分钟。

6. 实践考核采用五级评分制评定考核成绩：满分100分，0~59分为不合格，60~69分为合格，70~79分为中等，80~89分为良好，90~100分为优秀。

附录

题型举例

一、基础实操题：编写程序实现单链表的建立（尾插法）、插入（指定位置）、删除（指定元素）及遍历功能。要求程序能处理空链表、插入位置无效、删除元素不存在等异常情况，并输出相应提示信息。

二、综合实操题：某校园地图包含多个教学楼、图书馆、食堂等顶点，顶点间的路径为带权边（表示距离）。请完成以下任务：

1. 选择合适的存储结构（邻接矩阵或邻接表）存储该校园地图对应的图结构；
2. 实现图的深度优先搜索（DFS）遍历算法，输出从图书馆出发的遍历序列；