

# 文华学院 2024 年普通专升本考试

## 《数据结构》考试大纲

### 第一部分 考试说明

#### 一、 考试概况

本课程考试是为在计算机及相关专业中选拔优秀专科毕业生进入本科阶段学习而组织的考试。《数据结构》是计算机学科的一门专业核心基础课，是所有计算机应用程序都要用到的基础知识，是计算机类后续专业课程的基础。通过本课程的学习，使学生掌握数据常用的逻辑结构、存储结构与基本操作以及一些经典的算法实现，为后续课程的学习及今后的实际工作打下基础。考试的指导思想是考查学生对本课程概念、理论与主要知识点的掌握程度，以及对具体问题进行分析解决实际问题的能力。

#### 二、 考试方式与考试时间

- (1) 答卷方式：闭卷，笔试
- (2) 记分方式：满分为 150 分
- (3) 考试时间：90 分钟

#### 三、 参考书目

- (1) 数据结构 刘畅等主编 上海交通大学出版社.
  - (2) C 语言程序设计 陈维等主编，人民邮电出版社
- ### 第二部分 考试范围、考试内容及试卷结构

#### 一、 考试范围及考试内容

##### 1. 绪论

###### 1) 内容与要求

- (1) 理解数据结构的基本概念和基本术语；
- (2) 掌握算法的时间复杂度分析方法；
- (3) 掌握 C 语言的基本语法规则和 C 语言程序结构；。

###### 2) 考核要点

- (1) 基本知识点：数据结构的一些基本概念；数据常用的逻辑结构和物理结构；C 语言的基本语法规则和 C 语言基本程序结构；
- (2) 拔高知识点：时间复杂度的分析和求解；

##### 2. 线性表、栈和队列、数组

###### 1) 内容与要求

- (1) 理解并掌握线性表的基本特点；
- (2) 掌握线性表的顺序存储和链式存储的实现；

(3) 理解栈和队列的特点及存储实现;

(4) 掌握数组的定义及特点;

## 2) 考核要点

(1) 基本知识点: 顺序存储和链式存储的特点; 用 C 语言实现顺序存储和链式存储插入和删除操作; 栈和队列的特点以及插入和删除实现; 数组元素地址的求解;

(2) 拔高知识点: 循环链表和双向链表的插入和删除; 栈和队列的应用;

## 3. 树和二叉树

### 1) 内容与要求

(1) 理解树的概念及基本术语;

(2) 掌握二叉树的定义和性质;

(3) 掌握二叉树三种遍历及递归算法;

(4) 掌握树与二叉树的转换;

(4) 掌握哈夫曼树

### 2) 考核要点

(1) 基本知识点: 树与二叉树的一些基本概念; 二叉树的存储方法; 二叉树的三种遍历方法; 树与二叉树的转换;

(2) 拔高知识点: 构建二叉树; 二叉树的递归算法实现; 哈夫曼树;

## 4. 图

### 1) 内容与要求

(1) 掌握图的基本概念以及图的存储结构 (邻接矩阵、邻接表);

(2) 掌握图的深度优先和广度优先遍历算法;

(3) 掌握图的最小生成树算法;

(4) 掌握拓扑排序;

### 2) 考核要点

(1) 基本知识点: 图的基本概念; 图的存储结构; 图的遍历;

(2) 拔高知识点: prim 算法及 kruskal 算法; 拓扑序列;

## 5. 查找

### 1) 内容与要求

(1) 理解静态查找表和动态查找表的特征;

(2) 掌握常见几种查找算法;

### 2) 考核要点

(1) 基本知识点: 顺序查找、折半查找的特点以及实现;

(2) 拔高知识点: 二叉排序树; 哈希表的概念和查找方法和哈希函数的构造方法, 解决冲突的基本方法;

## 6. 排序

### 1) 内容与要求

- (1) 理解排序的概念；
- (2) 掌握几种常见的排序算法；

### 2) 考核要点

(1) 基本知识点：直接插入排序、冒泡排序、简单选择排序的特点；排序方法的稳定性；

- (2) 拔高知识点：快速排序和堆排序特点；

## 二、试卷结构

### 1. 命题范围

命题范围涵盖所列章节，会涉及 C 语言的一些基本知识，本大纲所提到的知识点是重点。

### 2. 难易程度

本试题难易程度可分为四档：易、较易、较难、难，这四档在试卷中所占的比例约为 1:4:3:2。

### 3. 试卷题型

无选择题，无判断题，其它题型不限。

## 附录 题型举例

### 1、填空题

1. 数据的最小单位是\_\_\_\_\_。

### 2、问答题

1. 栈和队列的特点是什么。

### 3、求解题

1. 根据给定的二叉树写出前序，中序和后序序列。

### 4、补充程序题

1. 下面程序段的功能是利用从尾部插入的方法建立单链表的算法，请在下划线处填上正确的内容。

```
typedef struct node
{ int data;
  struct node *next;
} llist;
void llistcreate(_____ *&head )
{
    for (i=1;i<=n;i++)
    {
        p=(llist *)malloc(sizeof(llist));
        scanf("%d",&(p->data));p->next=null;
        if(i==1)
            head=q=p;
        else
            {q->next=p;
              _____;}
    }
}
```

### 5、算法设计

给出一个高效算法，求出 1, 3, 6………n 这串数中大于 M 小于 N 的数。(M 和 N 是给定的数)