

全国重点名校系列

新版

# 全国硕士研究生招生考试 考研专业课精品资料

【电子书】2024年中原工学院

819计算机专业基础（数据结构、C++程序设计）  
考研精品资料

策划：辅导资料编写组

真题汇编 直击考点  
考研笔记 突破难点  
核心题库 强化训练  
模拟试题 查漏补缺

高分学长学姐推荐



**【初试】2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研精品资料**

**说明：本套考研资料由本机构多位高分研究生潜心整理编写，2024 年考研初试首选资料。**

**一、重点名校考研真题汇编**

**1. 附赠重点名校：数据结构 2016-2022 年考研真题汇编（暂无答案）**

说明：本科目没有收集到历年考研真题，赠送重点名校考研真题汇编，因不同院校真题相似性极高，甚至部分考题完全相同，建议考生备考过程中认真研究其他院校的考研真题。

**二、2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研资料**

**2. 《数据结构》考研相关资料**

**（1）《数据结构》[笔记+课件+提纲]**

**①2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之《数据结构》考研复习笔记。**

说明：本书重点复习笔记，条理清晰，重难点突出，提高复习效率，基础强化阶段必备资料。

**②2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之《数据结构》本科生课件。**

说明：参考书配套授课 PPT 课件，条理清晰，内容详尽，非本校课件，版权归属制作教师，本项免费赠送。

**③2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之《数据结构》复习提纲。**

说明：该科目复习重难点提纲，提炼出重难点，有的放矢，提高复习针对性。

**（2）《数据结构》考研核心题库（含答案）**

**①2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研核心题库之《数据结构》算法设计题精编。**

**②2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研核心题库之《数据结构》应用题精编。**

说明：本题库涵盖了该考研科目常考题型及重点题型，根据历年考研大纲要求，结合考研真题进行的分类汇编并给出了详细答案，针对性强，是考研复习首选资料。

**（3）《数据结构》考研题库[仿真+强化+冲刺]**

**①2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研专业课五套仿真模拟题。**

说明：严格按照本科目最新专业课真题题型和难度出题，共五套全仿真模拟试题含答案解析。

**②2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研强化五套模拟题及详细答案解析。**

说明：专业课强化检测使用。共五套强化模拟题，均含有详细答案解析，考研强化复习必备。

**③2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研冲刺五套模拟题及详细答案解析。**

说明：专业课冲刺检测使用。共五套冲刺预测试题，均有详细答案解析，最后冲刺必备资料。

**3. 《C++程序设计》考研相关资料**

**（1）《C++程序设计》[笔记+提纲]**

①2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之《C++程序设计》考研复习笔记。

说明：本书重点复习笔记，条理清晰，重难点突出，提高复习效率，基础强化阶段必备资料。

②2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）之《C++程序设计》复习提纲。

说明：该科目复习重难点提纲，提炼出重难点，有的放矢，提高复习针对性。

### 三、资料全国统一零售价

4. 本套考研资料包含以上一、二部分（不含教材），全国统一零售价：[¥]

特别说明：

①本套资料由本机构编写组按照考试大纲、真题、指定参考书等公开信息整理收集编写，仅供考研复习参考，与目标学校及研究生院官方无关，如有侵权、请联系我们将立即处理。

②资料中若有真题及课件为免费赠送，仅供参考，版权归属学校及制作老师，在此对版权所有者表示感谢，如有异议及不妥，请联系我们，我们将无条件立即处理！

### 四、2024 年研究生入学考试指定/推荐参考书目（资料不包括教材）

5. 中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研初试参考书

《数据结构》（c 语言版）严蔚敏，吴伟民主编清华大学出版社

《C++程序设计》郑莉主编清华大学出版社第三版

### 五、本套考研资料适用学院和专业

电子信息学院

### 版权声明

编写组依法对本书享有专有著作权，同时我们尊重知识产权，对本电子书部分内容参考和引用的市面上已出版或发行图书及来自互联网等资料的文字、图片、表格数据等资料，均要求注明作者和来源。但由于各种原因，如资料引用时未能联系上作者或者无法确认内容来源等，因而有部分未注明作者或来源，在此对原作者或权利人表示感谢。若使用过程中对本书有任何异议请直接联系我们，我们会在第一时间与您沟通处理。

因编撰此电子书属于首次，加之作者水平和时间所限，书中错漏之处在所难免，恳切希望广大考生读者批评指正。

目录

封面.....	1
目录.....	4
2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）备考信息 .....	9
中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研初试参考书目 .....	9
中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研招生适用院系 .....	9
2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研核心笔记 .....	10
<b>《数据结构》考研核心笔记 .....</b>	<b>10</b>
第 1 章 绪论 .....	10
考研提纲及考试要求 .....	10
考研核心笔记 .....	10
第 2 章 线性表 .....	22
考研提纲及考试要求 .....	22
考研核心笔记 .....	22
第 3 章 栈和队列 .....	37
考研提纲及考试要求 .....	37
考研核心笔记 .....	37
第 4 章 串 .....	41
考研提纲及考试要求 .....	41
考研核心笔记 .....	41
第 5 章 数组和广义表 .....	48
考研提纲及考试要求 .....	48
考研核心笔记 .....	48
第 6 章 树和二叉树 .....	54
考研提纲及考试要求 .....	54
考研核心笔记 .....	54
第 7 章 图 .....	78
考研提纲及考试要求 .....	78
考研核心笔记 .....	78
第 8 章 动态存储管理 .....	96
考研提纲及考试要求 .....	96
考研核心笔记 .....	96
第 9 章 查找 .....	104
考研提纲及考试要求 .....	104
考研核心笔记 .....	104
第 10 章 内部排序 .....	120
考研提纲及考试要求 .....	120

考研核心笔记.....	120
第 11 章 外部排序.....	130
考研提纲及考试要求.....	130
考研核心笔记.....	130
第 12 章 文件.....	136
考研提纲及考试要求.....	136
考研核心笔记.....	136
<b>《C++程序设计》考研核心笔记.....</b>	<b>141</b>
<b>2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研辅导课件.....</b>	<b>355</b>
《数据结构》考研辅导课件.....	355
<b>2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研复习提纲.....</b>	<b>466</b>
《数据结构》考研复习提纲.....	466
《C++程序设计》考研复习提纲.....	469
<b>2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研核心题库.....</b>	<b>475</b>
《数据结构》考研核心题库之算法设计题精编.....	475
《数据结构》考研核心题库之应用题精编.....	509
<b>2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研题库[仿真+强化+冲刺].....</b>	<b>532</b>
中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研仿真五套模拟题.....	532
2024 年数据结构五套仿真模拟题及详细答案解析（一）.....	532
2024 年数据结构五套仿真模拟题及详细答案解析（二）.....	538
2024 年数据结构五套仿真模拟题及详细答案解析（三）.....	543
2024 年数据结构五套仿真模拟题及详细答案解析（四）.....	548
2024 年数据结构五套仿真模拟题及详细答案解析（五）.....	555
中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研强化五套模拟题.....	561
2024 年数据结构五套强化模拟题及详细答案解析（一）.....	561
2024 年数据结构五套强化模拟题及详细答案解析（二）.....	566
2024 年数据结构五套强化模拟题及详细答案解析（三）.....	574
2024 年数据结构五套强化模拟题及详细答案解析（四）.....	580
2024 年数据结构五套强化模拟题及详细答案解析（五）.....	585
中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研冲刺五套模拟题.....	590
2024 年数据结构五套冲刺模拟题及详细答案解析（一）.....	590
2024 年数据结构五套冲刺模拟题及详细答案解析（二）.....	596
2024 年数据结构五套冲刺模拟题及详细答案解析（三）.....	603
2024 年数据结构五套冲刺模拟题及详细答案解析（四）.....	609
2024 年数据结构五套冲刺模拟题及详细答案解析（五）.....	617
<b>附赠重点名校：数据结构 2016-2022 年考研真题汇编（暂无答案）.....</b>	<b>624</b>

第一篇、2022 年数据结构考研真题汇编 .....	624
2022 年沈阳工业大学数据结构考研专业课真题 .....	624
2022 年广东财经大学 809 数据结构考研专业课真题 .....	626
2022 年中国人民解放军陆军工程大学 811 数据结构考研专业课真题 .....	631
2022 年南京审计大学 814 数据结构考研专业课真题 .....	635
2022 年常州大学 858 数据结构考研专业课真题 .....	637
2022 年湖南师范大学 865 数据结构考研专业课真题 .....	639
2022 年南京审计大学 911 数据结构考研专业课真题 .....	643
2022 年北京邮电大学 809 数据结构考研专业课真题 .....	645
2022 年暨南大学 830 数据结构考研专业课真题 .....	653
2022 年河北科技大学 847 数据结构考研专业课真题 .....	657
第二篇、2021 年数据结构考研真题汇编 .....	661
2021 年杭州电子科技大学数据结构考研专业课真题 .....	661
2021 年河北科技大学 847 数据结构考研专业课真题 .....	670
2021 年青岛理工大学 813 数据结构考研专业课真题 .....	673
2021 年北京化工大学 842 数据结构考研专业课真题 .....	676
2021 年常州大学 858 数据结构考研专业课真题 .....	684
2021 年广东财经大学 809 数据结构考研专业课真题 .....	686
2021 年广东财经大学 809 数据结构考研专业课真题 .....	690
2021 年湖南师范大学 865 数据结构考研专业课真题 .....	694
2021 年暨南大学 830 数据结构考研专业课真题 .....	698
2021 年南京审计大学 814 数据结构考研专业课真题 .....	702
2021 年南京审计大学 911 数据结构考研专业课真题 .....	704
2021 年宁波大学 916 数据结构与算法考研专业课真题 .....	706
2021 年沈阳工业大学 808 数据结构考研专业课真题 .....	711
2021 年四川化工大学 816 数据结构与算法考研专业课真题 .....	713
2021 年扬州大学 858 程序设计与数据结构考研专业课真题 .....	721
2021 年浙江工业大学 850 数据结构与计算机网络考研专业课真题 .....	723
第三篇、2020 年数据结构考研真题汇编 .....	727
2020 年青岛理工大学 813 数据结构考研专业课真题 .....	727
2020 年南京审计大学 814 数据结构考研专业课真题 .....	730
2020 年河北师范大学 823 计算机专业基础（数据结构）考研专业课真题 .....	732
2020 年广东工业大学 829 数据结构考研专业课真题 .....	735
2020 年暨南大学 830 数据结构考研专业课真题 .....	741
2020 年河北师范大学 833 数据结构考研专业课真题 .....	749
2020 年长沙理工大学 850 数据结构考研专业课真题 .....	757
2020 年浙江工业大学 850 数据结构与计算机网络考研专业课真题 .....	761
2020 年浙江工业大学 851 数据结构与软件工程考研专业课真题 .....	764
2020 年扬州大学 858 程序设计与数据结构考研专业课真题 .....	768
2020 年常州大学 858 数据结构考研专业课真题 .....	771



2020 年河北师范学 911 计算机专业基础（数据结构）考研专业课真题 .....	773
2020 年南京审计大学 911 数据结构考研专业课真题 .....	777
2020 年宁波大学 916 数据结构与算法考研专业课真题 .....	779
第四篇、2019 年数据结构考研真题汇编 .....	786
2019 年青岛理工大学 813 数据结构考研专业课真题 .....	786
2019 年重庆邮电大学 802 数据结构考研专业课真题 .....	789
第五篇、2018 年数据结构考研真题汇编 .....	797
2018 年贵州财经大学 808 数据结构考研专业课真题 .....	797
2018 年贵州财经大学 808 数据结构考研专业课真题 .....	799
2018 年湖南师范大学 865 数据结构考研专业课真题 .....	801
2018 年青岛理工大学 817 数据结构考研专业课真题 .....	805
2018 年山东大学 909 数据结构考研专业课真题 .....	808
2018 年沈阳工业大学 808 数据结构考研专业课真题 .....	810
2018 年天津城建大学 815 数据结构考研专业课真题 .....	812
2018 年长沙理工大学 850 数据结构考研专业课真题 .....	816
2018 年浙江理工大学 991 数据结构考研专业课真题 .....	820
2018 年重庆邮电大学 802 数据结构考研专业课真题 .....	825
第六篇、2017 年数据结构考研真题汇编 .....	831
2017 年广东工业大学 829 数据结构考研专业课真题 .....	831
2017 年桂林电子科技大学 910 数据结构考研专业课真题 .....	837
2017 年暨南大学 830 数据结构考研专业课真题 .....	841
2017 年南京师范大学 821 数据结构考研专业课真题 .....	846
2017 年青岛大学 910 数据结构考研专业课真题 .....	848
2017 年沈阳工业大学 808 数据结构考研专业课真题 .....	852
2017 年温州大学 831 数据结构考研专业课真题 .....	854
2017 年武汉纺织大学 848 数据结构考研专业课真题 .....	860
2017 年重庆邮电大学 802 数据结构 A 考研专业课真题 .....	864
第七篇、2016 年数据结构考研真题汇编 .....	870
2016 年安徽工业大学 861 数据结构考研专业课真题 .....	870
2016 年北京化工大学数据结构考研专业课真题 .....	876
2016 年广东工业大学 829 数据结构考研专业课真题 .....	885
2016 年暨南大学 830 数据结构考研专业课真题 .....	891
2016 年江苏大学 851 数据结构考研专业课真题 .....	896
2016 年解放军信息工程大学 820 数据结构考研专业课真题 .....	902
2016 年南京师范大学 829 数据结构考研专业课真题 .....	906
2016 年青岛大学 910 数据结构考研专业课真题 .....	909
2016 年沈阳工业大学 808 数据结构考研专业课真题 .....	914
2016 年武汉纺织大学 848 数据结构考研专业课真题 .....	915
2016 年湘潭大学 848 数据结构考研专业课真题 .....	918
2016 年湘潭大学 870 数据结构考研专业课真题 .....	921

## 2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）备考信息

### 中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研初试参考书目

《数据结构》（c 语言版）严蔚敏，吴伟民主编清华大学出版社

《C++程序设计》郑莉主编清华大学出版社第三版

### 中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研招生适用院系

电子信息学院



2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研核心笔记

《数据结构》考研核心笔记

第 1 章 绪论

考研提纲及考试要求

- 考点：数据（Data）
- 考点：数据对象（DataObject）
- 考点：数据抽象与抽象数据类型
- 考点：逻辑结构
- 考点：数据结构课程地位

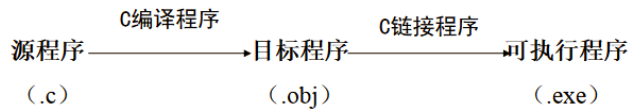
考研核心笔记

【核心笔记】数据结构的基本概念（定义）

1.数据（Data）

定义

数据是描述客观事物的数值、字符以及能输入机器且能被处理的各种符号集合。  
数据包含整型、实型、布尔型、图象、字符、声音等一切可以输入到计算机中的符号集合。  
例如对 C 源程序



2.数据元素（DataElement）

定义

数据元素是组成数据的基本单位，是数据集合的个体，在计算机中通常作为一个整体进行考虑和处理。

例如：

学号	姓名	性别	籍贯	出生年月	住址
101	赵虹玲	女	河北	1983.11	北京
...	...	...	...	...	...

数据项 ↓

← 数据元素

3.数据对象（DataObject）

定义

数据对象是性质相同的数据元素的集合，是数据的一个子集。

例如：整数集合：N={0, ±1, ±2, ...}无限集

字符集合：C={'A', 'B', ..., 'Z'}有限集

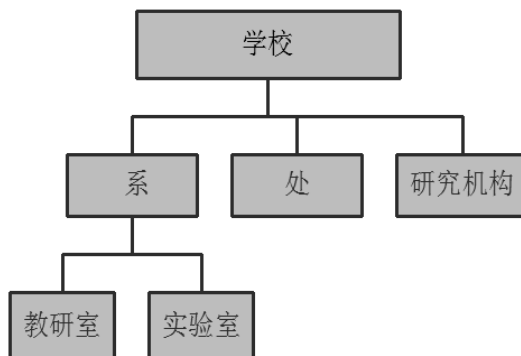
4.数据结构（Data Structure）

(1) 定义

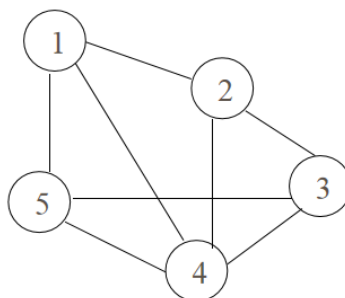
数据结构是指相互之间存在一种或多种特定关系的数据元素集合，是带有结构的数据元素的集合，它指的是数据元素之间的相互关系，即数据的组织形式。例如表结构：

学号	姓名	性别	籍贯	出生年月	住址
101	赵虹玲	女	河北	1983.11	北京
...	...	...	...	...	...

(2) 树型结构



(3) 图结构



### 5.数据类型(DataType)

定义

数据类型是一组性质相同的值集合以及定义在这个值集合上的一组操作的总称。

如在高级语言中，整型类型的取值范围为：

-32768~+32767，运算符集合为加、减、乘、除、取模，即+、-、\*、/、%

(1) 高级语言中的数据类型分为两大类：

①原子类型，其值不可分解。如 C 语言中的标准类型（整型、实型、字符型、）。

②结构类型，其值是由若干成分按某种结构组成的，因此是可以分解的，并且它的成分可以是非结构的，也可以是结构的。

### 6.数据抽象与抽象数据类型

- (1) 数据的抽象
- (2) 抽象数据类型(AbstractDataType)
- (3) 抽象数据类型实现
- (4) ADT 的表示与实现
- (5) 面向对象的概念
- (6) 结构化的开发方法与面向对象开发方法不同点

#### 【核心笔记】数据结构的内容

#### 1.逻辑结构

(1) 定义

数据的逻辑结构是指数据元素之间逻辑关系描述。

(2) 形式化描述:  $Data\_Structure = (D,R)$  其中  $D$  是数据元素的有限集,  $R$  是  $D$  上关系的有限集。

(3) 四类基本的结构

集合结构、线性结构、树型结构、图状结构。

2.集合结构

(1) 定义: 结构中的数据元素之间除了同属于一个集合的关系外, 无任何其它关系。

例如



3.线性结构

定义: 结构中的数据元素之间存在着一对一的线性关系。

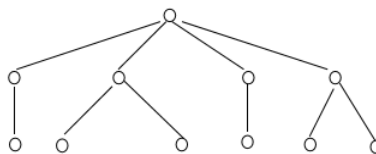
例如: 线性表



4.树型结构

定义: 结构中的数据元素之间存在着一对多的层次关系。

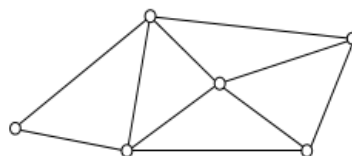
例如: 树



5.图状结构或网状结构

定义: 结构中的数据元素之间存在着多对多的任意关系。

例如



6.逻辑结构

综上所述, 数据的逻辑结构可概括为

逻辑结构

线性结构——线性表、栈、队、字符串数组、广义表

非线性结构——树、图

7.存储结构

(1) 定义: 存储结构 (又称物理结构) 是逻辑结构在计算机中存储映象, 是逻辑结构在计算机中的

实现，它包括数据元素的表示和关系的表示。

(2) 形式化描述:

D 要存入机器中，建立一从 D 的数据元素到存储空间 M 单元映象 S,  $D \rightarrow M$ , 即对于每一个  $d, d \in D$ , 都有唯一的  $z \in M$  使  $S(d) = z$ , 同时这个映象必须明显或隐含地体现关系 R。

(3) 逻辑结构与存储结构的关系为:

存储结构是逻辑关系的映象与元素本身映象，是数据结构的实现；逻辑结构是数据结构的抽象。

(4) 数据元素之间关系在计算机中的表示方法:

顺序映象（顺序存储结构）

非顺序映象（非顺序存储结构）

## 8. 运算集合

例如工资表

编号	姓名	性别	基本工资	工龄工资	应扣工资	实发工资
100001	张爱芬	女	345.67	145.45	30.00	451.12
100002	李林	男	445.90	185.60	45.00	586.50
100003	刘晓峰	男	345.00	130.00	25.00	450.00
100004	赵俊	女	560.90	225.90	65.00	721.80
100005	孙涛	男	450.60	190.80	50.00	591.80
...	...	...	...	...	...	...
100121	张兴强	男	1025.98	365.53	100.00	1291.51

## 9. 数据结构的内容

综上所述，数据结构的内容可归纳为三个部分

逻辑结构、存储结构和运算集合：按某种逻辑关系组织起来的一批数据，按一定的映象方式把它存放在计算机存储器中，并在这些数据上定义了一个运算的集合，就叫做数据结构。

### 【核心笔记】算法

#### 1. 算法 (Algorithm) 定义

定义：算法是规则的有限集合，是为解决特定问题而规定的一系列操作。

#### 2. 算法的特性

- (1) 有限性：有限步骤之内正常结束，不能形成无穷循环
- (2) 确定性：算法中的每一个步骤必须有确定含义，无二义性得以实现。
- (3) 输入：有多个或 0 个输入
- (4) 输出：至少有一个或多个输出
- (5) 可行性：原则上能精确进行，操作可通过已实现基本运算执行有限次而完成

#### 3. 算法设计的要求

- (1) 算法特征
  - ① 算法的正确性
  - ② 可读性
  - ③ 健壮性
  - ④ 效率和低存储量

### 【核心笔记】算法描述的工具

《C++程序设计》考研核心笔记

# 第1章 概述

## 1.1 计算机的工作模型

自 1946 年第一台电子计算机 (ENIAC) 问世以来, 计算机在理论、技术以及应用等方面有了很大的发展, 特别是计算机的应用, 它已从早期的数值计算拓广到现在的大量的非数值计算, 如: 管理信息系统、文字处理系统等都属于计算机在非数值计算方面的应用。现在, 计算机已经渗透到人类社会活动的各个领域并发挥着巨大的作用。

一台计算机由硬件和软件二部分构成。硬件是指计算机的物理构成, 即构成计算机的元器件和设备。软件是指计算机程序以及相关的文档资料。硬件是计算机的物质基础, 软件是计算机的灵魂。没有硬件就没有计算机, 但是, 如果只有硬件没有软件, 可以说计算机什么事情也做不了。要想用计算机来解决各种问题, 必须要有相应的软件。从某种意义上讲, 一台计算机的性能主要由硬件决定, 而它的功能主要是由软件来提供的。

计算机的应用领域在不断扩展, 应用的规模、层次和类型也在不断扩大, 社会对计算机软件的需求急剧增长。如何设计出大量的满足用户需求的高质量软件是软件工作者所面临的严峻挑战。计算机程序不同于其它程序 (如: 音乐会程序), 它是由计算机来执行的, 计算机程序的编制 (程序设计) 通常要按照计算机解决问题的方式来进行。因此, 要进行程序设计, 就必须对计算机的工作模型有一定的了解。

### 1.1.1 硬件结构

虽然计算机有了很大的发展, 但目前大部分计算机基本上采用的还是传统的冯·诺依曼 (Von Neumann) 体系结构, 即存储程序式结构。图 1.1 给出了冯·诺依曼计算机的硬件构成。

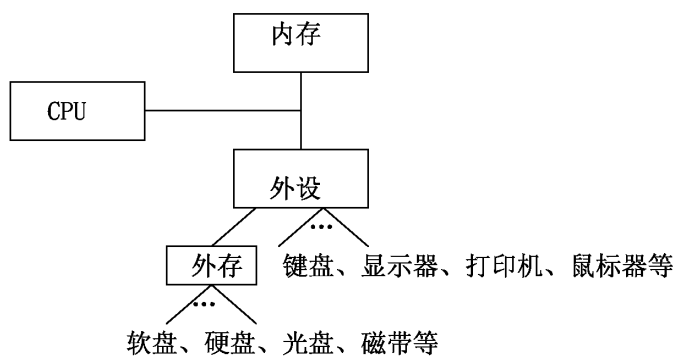


图 1.1 典型的计算机硬件组成

#### 1、中央处理器, 简称 CPU (Central Processing Unit)

CPU 是计算机的核心，用于执行计算机指令以实现对整个计算机的控制。它由控制器、运算器以及寄存器等构成。控制器负责取指令和根据指令发出控制信号以引起其它部件的动作。运算器执行运算指令所规定的运算。寄存器用于暂时保存指令的计算结果供下一（几）条指令使用，其作用是减少访问内存的次数，提高指令执行效率。

## 2、内部存储器或主存储器，简称内存（Memory）

内存用于存储计算机程序（指令和数据）。内存由许多存储单元构成，存储单元的大小视计算机而定，一般为一个字节（Byte）。每个存储单元都有一个地址，对存储单元的访问是通过其地址来进行的。

## 3、外部设备，简称外设（Device）

外设用于计算机的输入/输出和提供大容量的信息存储。它包括输入/输出设备和外部存储器或辅助存储器（简称外存）。

输入/输出设备提供了计算机与用户的接口，用于用户数据的输入和程序执行结果的输出。键盘和鼠标器等属于输入设备；显示器和打印机等属于输出设备。

外存是大容量的低速存储部件（与内存相比），用于永久性地存储程序、数据以及各种文档等信息，包括软盘、硬盘、光盘、磁带等。存储在外存中的信息通常以文件形式进行组织和访问。内存与外存除了容量和速度不同外，它们的另一个区别在于：内存中存储的是正在运行的程序和数据。

冯·诺依曼计算机的工作模型是：待执行的程序从外存装入到内存中，CPU 从内存中逐条地取程序中的指令执行。程序执行中所需要的数据从内存或从外设中获得，程序执行中产生的临时数据保存在内存中，程序的执行结果通过外设输出，程序执行结束后从内存退出。

CPU 所能执行的指令通常有：

- 算术指令。实现加、减、乘、除等运算。
- 比较指令。比较两个操作数的大小。
- 数据传输指令。实现 CPU 的寄存器、内存以及外设之间的数据传输。
- 执行流程控制。确定下一条指令的内存地址，包括转移、循环以及子程序调用/返回。通常情况下，CPU 从某个内存地址开始依次取指令来执行。

在冯·诺依曼计算机中存在着几个影响程序执行效率的瓶颈，其中包括 CPU 与内存之间以及内存与外设之间的数据传输。现在的计算机中往往在 CPU 中为内存提供了高速缓存（memory cache）；在内存中为外存提供了高速缓存（disk cache），以解决 CPU 与内存以及内存与外存之间速度不匹配问题。

### 1.1.2 软件

计算机硬件为计算机提供了物质基础，但它必须通过计算机软件来发挥作用。计算机软件是计算机系统上的程序以及有关的文档。程序是计算任务的处理对象与处理规则的描述；文档是为了便于人理解程序所需的资料说明，供程序开发与维护使用。



软件可以分为：系统软件、支撑软件和应用软件。系统软件居于计算机系统中最靠近硬件的一级，它与具体的应用领域无关，其它软件一般要通过系统软件发挥作用，如操作系统就属于系统软件。支撑软件是指支持软件开发与维护的软件，一般由软件开发人员使用，如软件开发环境就是典型的支撑软件。应用软件是指用于特定领域的专用软件，如：人口普查软件、财务软件等。图 1.2 给出了计算机软件的分类。

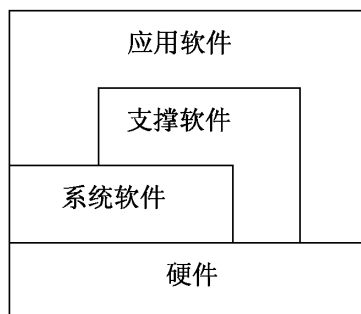


图 1.2 计算机软件分类

一个软件从无到有，一直到最后的消亡（报废），通常要经历一个过程，这个过程称为软件的生命周期，它包括若干阶段：软件需求分析、软件设计、编程实现、测试以及运行与维护。软件需求分析的主要任务是明确待实现的软件要解决什么问题，即做什么。软件设计是根据软件的需求说明给出抽象的解决方案，它包括概要设计和详细设计。概要设计是指软件的整体结构设计；详细设计是指抽象的数据/算法描述；编程实现是指根据软件设计说明，采用某种程序设计语言书写程序；测试是对书写好的程序进行测试，确认其是否满足所规定的需求；运行与维护是指使用软件并对软件进行维护，其中的维护包括：正确性维护、完善性维护和适应性维护。

早期的软件开发工作主要花费在编程实现阶段，并且采用的是个体的小作坊开发模式。随着计算机应用领域的不断扩大和应用层次的不断加深，使得软件的规模不断扩大、软件的复杂度不断提高，早期的软件开发模式难以驾驭软件开发过程，程序的正确性难以保证，软件生产率急剧下降，出现了“软件危机”。为了解决软件危机，软件工程概念应运而生，其主要思想是采用工程方法来开发软件。在软件工程中，软件开发工作的中心从实现阶段转移到软件需求分析、设计和维护阶段，并且强调对软件开发过程的管理和加强各个阶段的文档制作。方法和工具构成了软件工程的两大支柱，它们贯穿于软件开发过程，对软件工程思想提供具体的支持。

## 1.2 程序设计

简单地说，程序设计就是为计算机编制程序的过程，它涉及程序设计方法和程序设计语言等方面的内容。从现代软件工程的视角讲，程序设计是指编程实现阶段的工作，而实际上，我们现在所说的程序设计是指传统意义上的程序设计，它包含软件工程中其

它阶段的一些工作，只是更多地考虑实现技术而已。因此，不能把程序设计仅仅理解成用某种语言来实现设计好的软件，其中还必须考虑需求分析、软件设计、测试以及维护等一些问题。

### 1.2.1 程序设计风范 (Programming Paradigm)

以不同的方式来给出计算的描述就形成了不同的程序设计风范 (Programming Paradigms)。具体地讲，程序包括数据以及对数据的加工两部分，程序设计风范就是指以何种观点和方式来看待、组织和描述它们。不同的程序设计风范支持不同的程序设计方法。

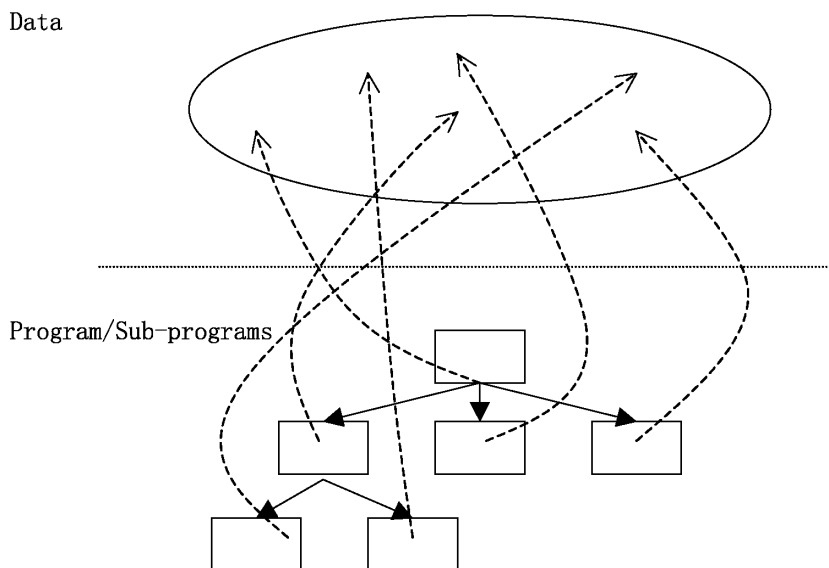
目前存在若干种程序设计风范，典型的程序设计风范有：过程式、对象式、函数式以及逻辑式等。

#### 1、过程式

过程式程序设计是一种以功能为中心、基于功能分解的程序设计风范。在这种程序设计中，首先从程序的功能出发对系统的功能进行分解，然后再对分解出的各个模块进行算法和数据结构的设计。下面给出的公式刻划了过程式程序设计的本质特征：

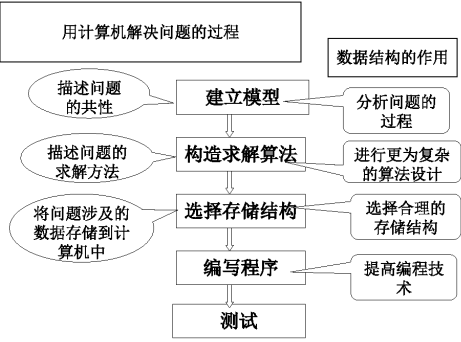
$$\text{程序} = \text{算法} + \text{数据结构}$$

过程式程序设计的优点在于：它对程序功能的描述比较清晰。不足之处是：数据与操作分离，缺乏数据保护 (图 1.3)；程序难以复用；不能适应功能的变化。



2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研辅导课件

《数据结构》考研辅导课件

<h3 style="text-align: center;">第一章 绪论</h3> <p>1.1 什么是数据结构</p> <p>1.2 基本概念</p> <p>1.3 抽象数据类型</p> <p>1.4 算法及其分析</p> <p style="text-align: right;">1</p>	<h3 style="text-align: center;">学习提要</h3> <ul style="list-style-type: none"> <li>掌握本课程所涉及到的基本名词、术语和概念，特别是数据的逻辑结构和存储结构之间的关系及性质。</li> <li>了解抽象数据类型的定义、表示和实现方法。</li> <li>理解算法设计的五个要素和基本要求；掌握算法效率的度量方法，着重学习算法的时间复杂度分析。</li> </ul> <p style="text-align: right;">2</p>																																										
<h3 style="text-align: center;">教学重点</h3> <p>(1)数据、数据元素、数据项；</p> <p>(2)逻辑结构和存储结构在概念上的联系与区别；</p> <p>(3)数据结构及其三个组成部分；</p> <p>(4)抽象数据类型和数据抽象；</p> <p>(5)评价算法优劣的标准及方法。</p> <p style="text-align: right;">3</p>	<h3 style="text-align: center;">1.1什么是数据结构</h3> <p>一、为什么要学习数据结构？</p> <p>1、电子计算机的主要用途：</p> <ul style="list-style-type: none"> <li>早期：                     <ul style="list-style-type: none"> <li>主要用于数值计算。</li> </ul> </li> <li>后来：                     <ul style="list-style-type: none"> <li>处理逐渐扩大到非数值计算领域（能处理多种复杂的具有一定结构关系的数据）。</li> </ul> </li> </ul> <p style="text-align: right;">4</p>																																										
 <p>用计算机解决问题的过程</p> <p>数据结构的的作用</p> <p>描述问题的共性 → 建立模型 → 分析问题的过程</p> <p>描述问题的求解方法 → 构造求解算法 → 进行更为复杂的算法设计</p> <p>将问题涉及的数据存储到计算机中 → 选择存储结构 → 选择合理的存储结构</p> <p>编写程序 → 提高编程技术</p> <p>测试</p>	<h3 style="text-align: center;">书目卡片</h3> <p>登录号： 书名： 作者名： 分类号： 出版单位： 出版时间： 价格：</p> <p style="text-align: right;">6</p>																																										
<h3 style="text-align: center;">引例</h3> <p>书目自动检索系统的数学模型</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>001</td><td>高等数学</td><td>樊映川</td><td>S01</td></tr> <tr><td>002</td><td>理论力学</td><td>罗远祥</td><td>L01</td></tr> <tr><td>003</td><td>高等数学</td><td>华罗庚</td><td>S01</td></tr> <tr><td>004</td><td>线性代数</td><td>秦汝书</td><td>S02</td></tr> <tr><td>.....</td><td>.....</td><td>.....</td><td>.....</td></tr> </table> <p>按书名      按作者名      按分类号</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>高等数学</td><td>001, 003, .....</td></tr> <tr><td>理论力学</td><td>002, .....</td></tr> <tr><td>线性代数</td><td>004, .....</td></tr> <tr><td>.....</td><td>.....</td></tr> </table> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>樊映川</td><td>001, .....</td></tr> <tr><td>华罗庚</td><td>002, .....</td></tr> <tr><td>秦汝书</td><td>004, .....</td></tr> <tr><td>.....</td><td>.....</td></tr> </table> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>L</td><td>002, .....</td></tr> <tr><td>S</td><td>001, 003, .....</td></tr> <tr><td>.....</td><td>.....</td></tr> </table> <p style="text-align: right;">7</p>	001	高等数学	樊映川	S01	002	理论力学	罗远祥	L01	003	高等数学	华罗庚	S01	004	线性代数	秦汝书	S02	.....	.....	.....	.....	高等数学	001, 003, .....	理论力学	002, .....	线性代数	004, .....	.....	.....	樊映川	001, .....	华罗庚	002, .....	秦汝书	004, .....	.....	.....	L	002, .....	S	001, 003, .....	.....	.....	<h3 style="text-align: center;">求解非数值计算的问题：</h3> <p>主要考虑的是设计出合适的数据结构及相应的算法。</p> <p>即：首先要考虑对相关的各种信息如何表示、组织和存储？</p> <p>因此，简单说来，数据结构是一门研究非数值计算的程序设计问题中计算机的操作对象以及它们之间的关系和操作的学科。</p> <p style="text-align: right;">8</p>
001	高等数学	樊映川	S01																																								
002	理论力学	罗远祥	L01																																								
003	高等数学	华罗庚	S01																																								
004	线性代数	秦汝书	S02																																								
.....	.....	.....	.....																																								
高等数学	001, 003, .....																																										
理论力学	002, .....																																										
线性代数	004, .....																																										
.....	.....																																										
樊映川	001, .....																																										
华罗庚	002, .....																																										
秦汝书	004, .....																																										
.....	.....																																										
L	002, .....																																										
S	001, 003, .....																																										
.....	.....																																										

**问题：学习数据结构有什么用？**

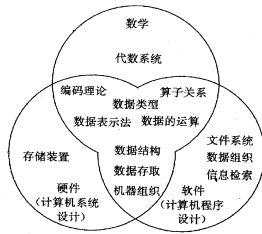
- 答：计算机内的数值运算依靠数学方程，而非数值运算（如表、树、图等）则要依靠数据结构。
- 同样的数据对象，用不同的数据结构来表示，运算效率可能有明显的差异。
  - 程序设计的实质是对实际问题选择一个好的数据结构，加之设计一个好的算法。而好的算法在很大程度上取决于描述实际问题的数据结构。
- 《算法+数据结构=程序》
- ①提高复杂程序设计的能力
  - ②培养算法设计能力
  - ③为后继课程（如操作系统、编译原理等）打基础。

**数据结构课程的形成和发展**

- 形成阶段
  - 60年代初期，“数据结构”有关的内容散见于操作系统、编译原理和表处理语言等课程。
  - 1968年，美唐·欧·克努特教授开创了数据结构的最初体系，《计算机程序设计技巧》第一卷《基本算法》，“数据结构”被列入美国一些大学计算机科学系的教学计划。
- 发展阶段：
  - 数据结构的概念不断扩充，包括了网络、集合代数学、关系等“离散数学结构”的内容。
  - 70年代后期，80年代初，我国高校陆续开设该课程。

**《数据结构课程》所处的地位：**

介于数学、计算机硬件和计算机软件三者之间的一门核心课程。



**1.2 基本概念和术语**

**1.2.1 基本概念**

**1. 数据 (data) :**

数据是信息的载体，是描述客观事物的数、字符、以及所有能输入到计算机中并被计算机程序识别和处理的符号的集合，是计算机程序加工的“原料”。

分类： $\begin{cases} \text{数值性数据} \\ \text{非数值性数据} \end{cases}$

**2、数据元素 (data element)**

- 数据的基本单位。在计算机程序中常作为一个整体进行考虑和处理。有时一个数据元素可以由若干数据项(Data Item)组成。
- 数据项是数据不可分割的最小标识单位。
- 数据元素又称为元素、结点、记录。

学号	姓名	成绩
001	LiLy	89
002	Yang	98
003	Zhao	78

**3、数据对象 (data object)**

- 数据对象是具有相同性质的数据元素的集合。
- 整数数据对象
 
$$N = \{0, \pm 1, \pm 2, \dots\}$$
- 字母字符数据对象
 
$$C = \{ 'A', 'B', \dots, 'Z' \}$$
- 学生成绩数据对象
 
$$C_j = \{ ( '101', 'jane', 80 ), ( '102', 'jack', 90 ), ( '103', 'jerry', 75 ) \}$$

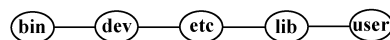
学号	姓名	成绩
001	LiLy	89
002	Yang	98
003	Zhao	78

**4、数据结构(data structure)**

- 数据结构是相互之间存在一种或多种特定关系的数据元素的集合。
- 在任何问题中，数据元素都不是孤立存在的，而是在它们之间存在着某种关系，这种数据元素相互之间的关系称为结构(Structure)。
- 数据结构是一堆数据元素和这些数据元素之间的关系的总和。

按数据元素之间关系的不同特性，通常有4类基本结构

- (1) 集合 结构中的数据元素除了“同属于一个集合”外，别无其它关系。
- (2) 线性结构 结构中的数据元素之间存在一对一的关系。
- (3) 树型结构 结构中的数据元素之间存在一对多的关系。
- (4) 图状结构或网状结构 结构中的数据元素之间存在多对多的关系。



**集合**  
数据元素之间无特殊关系

**图形（网状）结构**  
数据元素之间存在着多对多的关系。

17

**树形结构**  
数据元素之间存在着一个对多个的关系

树                      二叉树                      二叉搜索树

**线性结构**  
数据元素之间存在着一个对一个的关系

### 数据结构的定义

用一个二元组表示，记为：  
 $DS = (D, S)$

其中，D是数据元素的有限集（即一个数据对象），S是该对象中所有数据成员之间的关系有限集。

在计算机科学中，对复数的定义：复数是一种数据结构  
 $Complex = (C, R)$

其中：C是包含两个实数的集合  $\{C_1, C_2\}$ ，  
 $R = \{P\}$ , P是定义在C上的一种关系  $\{<C_1, C_2>\}$ 。

**例、用数据结构如何描述2行3列矩阵：**

- 它是一个含6个数据元素  $\{a_1, a_2, a_3, a_4, a_5, a_6\}$  的集合，且集合上存在“行关系”和“列关系”两个次序关系，其中  
 行关系为  $\{<a_1, a_2>, <a_2, a_3>, <a_4, a_5>, <a_5, a_6>\}$   
 列关系为  $\{<a_1, a_4>, <a_2, a_5>, <a_3, a_6>\}$ 。
- $<x, y>$  意为 x 和 y 之间存在“x领先于y”的次序关系。

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix}$$

**思考：**  
如何描述一个一行六列的矩阵？

20

**例1-5** 假设我们需要编制一个事务管理的程序，管理学校科学研究课题小组的各项事务，则首先要为程序的操作对象——课题小组设计一个数据结构。假设每个小组由一位教师、一至三名研究生及一至六名本科生组成，小组成员之间的关系是：教师指导研究生，而由每位研究生指导一至两名本科生。

则可以如下定义数据结构：  
 $Group = (P, R)$

其中：P =  $\{T, G_1, \dots, G_n, S_{11}, \dots, S_{nm} \mid 1 \leq n \leq 3, 1 \leq m \leq 2\}$

$R = \{R_1, R_2\}$

$R_1 = \{<T, G_i> \mid 1 \leq i \leq n, 1 \leq n \leq 3\}$

$R_2 = \{<G_i, S_{ij}> \mid 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq n \leq 3, 1 \leq m \leq 2\}$

### 5、数据的逻辑结构

- 数据的逻辑结构从逻辑关系上描述数据，可以看作是从具体问题抽象出来的数据模型，与数据的存储无关，也与数据元素本身的形式、内容、相对位置无关；
- 数据的逻辑结构分类：
  - 线性结构  
线性表、栈、队列、串
  - 非线性结构  
树、图（或网络）

22

### 6、数据的存储结构

- 数据结构在计算机中的表示（或称映像）称为数据的存储结构，又称为物理结构。它包括数据元素的表示和关系的表示。
  - 数据元素的表示：位、字长、元素、结点、数据域
  - 关系的表示两种基本的存储方法：
    - 顺序映像（顺序存储结构）
    - 非顺序映像（链式存储结构）

23

- 顺序存储结构：  
借助数据元素在存储器中的相对位置来表示数据元素之间的逻辑关系。  
所有元素存放在一片连续的存储单元中，逻辑上相邻的元素存放在计算机内存中仍然相邻。
- 链式存储结构：  
在每一个数据元素中增加一个存放地址的指针，借助该指针来表示数据元素之间的逻辑关系。  
所有元素存放在可以不连续的存储单元中，但元素之间的关系可以通过地址（指针）确定，逻辑上相邻的元素存放在计算机内存中不一定是相邻的。

24

### 存储结构的描述

- 存储结构的描述方法随编程环境的不同而不同，通常可用高级编程语言中提供的数据类型描述之。
- 例如，用一维数组类型描述顺序存储结构，用指针描述链式存储结构。
- 例如，定义“日期”为：  

```
typedef struct {
    int y; // 年号 Year
    int m; // 月号 Month
    int d; // 日号 Day
} DateType; // 日期类型
```

 同样，此时对数据元素也要借用高级编程语言中的数据类型描述之。

25

### 7、数据类型 (data type)

- 数据类型是一个值的集合和定义在这个值集上的一组操作的总称。
- 数据类型可分两类：原子类型和结构类型。
- 例如：在C语言中  
 原子类型：整型、实型、字符型等  
 结构类型：数组、结构体、联合等

26

### 9、抽象数据类型 (Abstract Data Type简称ADT)

- 由用户定义，用以表示应用问题的数据模型
- 指一个数学模型以及定义在此数学模型上的一组操作。
- 例如：计算机拥有的整数类型。
- 它与数据类型实质上是一个概念，但其特征是使用与实现分离，实行封装和信息隐蔽（独立于计算机）。

27

### 抽象数据类型的描述方法

- 抽象数据类型可用 (D, S, P) 三元组表示，其中，D是数据对象，S是D上的关系集，P是对D的基本操作集。

```
ADT 抽象数据类型名 {
    数据对象: (数据对象的定义)
    数据关系: (数据关系的定义)
    基本操作: (基本操作的定义)
} ADT 抽象数据类型名
```

数据对象和数据关系的定义用伪码（不真正执行的符号）描述。

基本操作的定义格式为：  
 基本操作名 (参数表)  
 初始条件: (初始条件描述)  
 操作结果: (操作结果描述)

28

- 基本操作有两种参数：  
 赋值参数只为操作提供输入值；  
 引用参数以&打头，除了可以提供输入值外，还将返回操作结果。
- “初始条件”描述了操作执行之前数据结构和参数应满足的条件，若不满足，则操作失败，并返回相应出错信息。若初始条件为空，则省略之。
- “操作结果”说明了操作正常完成之后，数据结构的变化状况和应返回的结果。

29

### 举例——抽象数据类型复数的定义：

```
ADT Complex {
    数据对象: D={e1, e2 | e1, e2 ∈ RealSet }
    数据关系: R1={<e1, e2| e1是复数的实数部分, e2是复数的虚数部分}
    基本操作:
    InitComplex( &Z, v1, v2 )
        操作结果: 构造复数Z, 其实部和虚部分别被赋以参数v1和v2的值。
    DestroyComplex( &Z )
        操作结果: 复数Z被销毁。
```

30

```
GetReal( Z, &realPart )
    初始条件: 复数已存在。
    操作结果: 用realPart返回复数Z的实部值。
GetImag( Z, &ImagPart )
    初始条件: 复数已存在。
    操作结果: 用ImagPart返回复数Z的虚部值。
Add( z1, z2, &sum )
    初始条件: z1, z2是复数。
    操作结果: 用sum返回两个复数z1, z2的和值。
} ADT Complex
```

假设: z1和z2是上述定义的复数, 则Add(z1, z2, z3)操作的结果将得到z3=z1+z2

31

### 1.3抽象数据类型的表示与实现

- 抽象数据类型可通过固有数据类型来表示和实现，即利用处理器中已存在的数据类型来说明新的结构，用已经实现的操作来组合新的操作。

一、类C语言简要说明

(1) 预定义常量和类型：  
 //函数结果状态代码  

```
#define TRUE 1
#define FALSE 0
#define OK 1
#define ERROR 0
#define INFEASIBLE -1
#define OVERFLOW -2
//Status是函数的类型，其值是函数结果状态代码
typedef int Status;
```

32



2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研复习提纲

《数据结构》考研复习提纲

《数据结构》复习提纲

第一章 绪论

本章主要内容：

- 1.1 什么是数据结构
- 1.2 基本概念和术语
- 1.3 抽象数据类型的表示与实现
- 1.4 算法和算法分析

复习要求：理解数据结构的相关概念，对本章内容及知识体系结构有初步的认识，并能够对算法的时间效率进行分析。

重点及难点：本章复习重点是对数据结构知识体系的理解，难点是算法分析。

第二章 线性表

本章主要内容：

- 2.1 线性表的类型定义
- 2.2 线性表的顺序表示和实现
- 2.3 线性表的链式表示和实现
- 2.4 一元多项式的表示及相加

本章复习要求：理解线性表的逻辑结构和两种存储结构，并能够应用线性表解决实际问题。

本章重点及难点：本章复习重点是对线性表的理解，难点是线性链表的各种操作。

第三章 栈和队列

本章主要内容：

- 3.1 栈
- 3.2 栈的应用举例
- 3.3 栈与递归的实现(\*)
- 3.4 队列
- 3.5 离散事件模拟(\*)

本章复习要求：理解操作受限的线性结构栈和队列的特点，从栈和队列的应用体会栈和队列在解决实际问题中的重要性。

本章重点及难点：本章复习重点是对栈和队列的理解，难点是栈与递归的关系。

第四章 串

本章主要内容：

- 4.1 串类型的定义
- 4.2 串的实现和实现
- 4.3 串的模式匹配算法(\*)
- 4.4 串操作应用举例

本章复习要求：理解串结构的特点，能够综合应用串的常见操作方法解决问题。

本章重点及难点：本章复习重点是对串的理解的串操作的应用，难点是模式匹配算法。

第五章 数组和广义表

本章主要内容：

- 5.1 数组的定义
- 5.2 数组的顺序表示和实现
- 5.3 矩阵的压缩存储
- 5.4 广义表的定义
- 5.5 广义表的存储结构
- 5.6 M 元多项式的表示(\*)
- 5.7 广义表的递归算法(\*)

本章复习要求：从数组和广义表的表示和实现进一步加深对线性结构的理解，掌握矩阵的压缩存储和广义表的存储。

本章重点及难点：本章复习重点是对矩阵的压缩存储，难点是广义表的存储和操作。

第六章 树和二叉树

本章主要内容：

- 6.1 树的定义和基本术语

- 6.2 二叉树
- 6.3 遍历二叉树和线索二叉树
- 6.4 树和森林
- 6.5 树与等价问题(\*)
- 6.6 赫夫曼树及其应用
- 6.7 回溯法与树的遍历
- 6.8 树的计数

本章复习要求：理解非线性结构树和二叉树的特点。

本章重点及难点：本章复习重点是是二叉树的遍历和赫夫曼树，难点是线索二叉树的操作以及回溯法与树的遍历。

### 第七章 图

本章主要内容：

- 7.1 图的定义和术语
- 7.2 图的存储结构
- 7.3 图的遍历
- 7.4 图的连通性问题
- 7.5 有向无环图及其应用
- 7.6 最短路径

本章复习要求：理解图的结构特点以及图和树间的关系, 掌握图的遍历和常见算法。

本章重点及难点：本章复习重点是是图的遍历和图的操作，难点是图的生成树算法和最短路径算法。

### 第八章 动态存储管理

本章主要内容：

- 8.1 概术
- 8.2 可以得用空间表及分配方法
- 8.3 边界标识法
- 8.4 伙伴系统

本章复习要求：体会数据结构在内存管理中的作用, 掌握内存管理中常用结构和方法。

本章重点及难点：本章复习重点是是可利用空间表，难点是伙伴系统

### 第九章 查找

本章主要内容：

- 9.1 静态查找表
- 9.2 动态查找表
- 9.3 哈希表

本章复习要求：掌握各种查找方法和它们的优缺点, 能够针对实际问题选择合适的查找方法。

本章重点及难点：本章复习重点是是二分查找，树表查找和哈希表，难点是 B-树查找。

### 第十章 内部排序

本章主要内容：

- 10.1 概述
- 10.2 插入排序
- 10.3 快速排序
- 10.4 选择排序
- 10.5 归并排序
- 10.6 基数排序
- 10.7 各种内部排序方法的比较讨论

本章复习要求：掌握各种排序方法和它们的优缺点, 能够针对实际问题选择合适的排序方法。

本章重点及难点：本章复习重点是是各种排序方法的实现，难点是针对实际问题如何选择合适的排序方法。

### 第十一章 外部排序

本章主要内容：

- 11.1 外存信息的存取
- 11.2 外部排序的方法
- 11.3 多路平衡归并的实现
- 11.4 置换-选择排序
- 11.5 最佳归并树

本章复习要求：体会外部排序和内部排序的不同之处,掌握各种外部排序方法。

本章重点及难点：本章复习重点是是各种外部排序方法，难点是败者树的实现。

## 第十二章 文件

本章主要内容：

- 12.1 有关文件的基本概念
- 12.2 顺序文件
- 12.3 索引文件
- 12.4 ISAM 文件和 VSAM 文件
- 12.5 直接存取文件(散列文件)
- 12.6 多关键字文件

本章复习要求：理解文件的概念和各种文件的结构特点。

本章重点及难点：本章复习重点是是各种文件的结构特点，难点是 ISAM 文件和 VSAM 文件。

## 《C++程序设计》考研复习提纲

### 一、课程的性质

C 语言由 Dennis Ritchie 设计, 并首先在 UNIX 下运行。它是在 B 语言的基础上发展起来的一种快速、高效的编程语言。c 语言具有允许直接对位、字节、字、串和指针操作的特点, 因此, 它适用于作系统程序的基本语言。c 语言支持强类型转换, 仅有 32 个关键字构成全部指令, 编写的程序具有简洁性, 因此又被誉为编程者的语言。C 语言支持结构化的程序设计方法, 它的主要结构成份是函数, 函数可以被定义完成独立的任务, 并独立的编译成目标代码。C 语言提供的复合语句实现了程序的模块化和分隔, 使得运算过程清晰、简洁和高效, 所以, C 语言往往被选做为学习程序设计的主要工具。本课程对先修课程没有要求, 它是程序设计的基础和工具。通过该课程的学习, 可为今后的软件工程实践和毕业设计打下坚实基础, 是本专业中一门重要的专业基础课。

### 二、教学目标和教学基本要求

本课程通过教学活动使学生掌握 C 语言的语法、C 语言程序设计的基本技能, 并能编写具有良好风格的程序。本课程基本要求是: 掌握 C 语言的语法、C 语言的程序结构、组织与管理、C 语言丰富的数据类型以及对数据的各种操作方法。掌握面向过程的程序设计方法, 了解它系统设计中的应用, 通过实验掌握 C 语言的使用方法和程序设计方法, 具有较强的计算机编程能力。

### 三、教学方法及建议

#### 1、课堂讲授

在教学过程中, 教师应注重加强基础。对 c 语言的语法要素和程序设计的基本概念进行详细的讲解, 并指出每章的重点和难点部分。讲授中应尽量纳入程序设计方法的最新发展成果, 注重理论联系实际, 通过电子课件、机器演示等多种形式展示、讨论, 启迪学生的思维, 加深学生对有关概念、内容和方法的理解, 使学生理解并掌握 c 语言的基本内容和 c 语言的分析方法与设计方法。讲授时间为 36 学时。

#### 2、教学方法

采取课堂讲授、上机操作、课后自学、课堂讨论等形式。教学中适当采用 CAI 课件加强基本概念的阐述, 改善教学效果。鉴于学生开始接受一些基本概念和分析方法有一定难度, 在教学中要采用启发式教学方法, 使学生逐步了解并掌握本课程的主要内容。在每章教学中均安排习题课结合实例说明分析方法和设计方法, 使学生提高分析和设计能力。

### 四、实验内容

通过实验是学好计算机语言最有效的办法之一。通过实验一方面可加深对 C 语言的理解, 另一方面也可使实验者得到编程的锻炼, 为今后开发工作打下坚实的基础。根据教学内容拟做如 18 个实验, 合计 36 学时。

五、习题数量及要求: 知识点均安排一定数量的习题。

六、考核方式及要求

2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研核心题库

《数据结构》考研核心题库之算法设计题精编

1. 已知顺序表中有  $n$  个记录，表中记录不依关键字有序排序，编写一算法，为该顺序表建立一个有序的索引表（依关键字递增排列），索引表中的每一项应含有记录的关键字和该记录在顺序表中的序号。要求算法的时间复杂度在最好的情况下能达到  $O(n)$ 。

```

【答案】 typedef struct index{           //索引类型
    Keytype key;                          //关键字
    int no;                                //在主表中的序号
}indextype;

typedef struct{                            //表结点类型
    ElemType data;                        //表中元素
    KeyType key;                          //表中元素关键字
}RectType;

void createindex (RectType r[], indextype &idx[ ], int n){ //为主表 r 建立一个索引表 idx
    int i,j;
    for(i=1; i<=n; i++){
        j = i-1;
        while(j>0&&idx[j].key){          //把大于 r[i].key 的所有 idx[j].key 后移
            idx[j+1] = idx[j];
            j--;
        }
        idx[j+1].key = r[i].key;         //把 r[i].key 插入到 idx 中，并记录下其序号 i
        idx[j+1].no = i;
    }
}
    
```

2. 已知中序线索二叉树  $T$  右子树不空。设计算法，将  $S$  所指的结点作为  $T$  的右子树中的一个叶子结点插入进去，并使之成为  $T$  的右子树的（中序序列）第一个结点（同时要修改相应的线索关系）。

**【答案】** 若使新插入的叶子结点  $S$  成  $T$  右子树中序序列的第一个结点，则应在  $T$  的右子树中最左面的结点（设为  $p$ ）处插入，使  $S$  成为结点  $p$  的左子女，则  $S$  的前驱是  $T$ ，后继是  $p$ 。

```

p=T->rchild;                               //p 指向 T 的右子女
while(p->ltag==0) p=p->lchild;              //p 最终指向 T 的右子树中最左面的结点
S->ltag=1; S->rtag=1;                       //S 是叶子，其左右标记均为 1
S->lchild=T; S->rchild=p;                   //S 的前驱是根结点 T，后继是结点 p
p->lchild=S; p->ltag=0;                     //将 p 的左子女指向 S，并修改左标志为 0
    
```

3. 有 15 个人围成一圈，顺序从 1 到 15 编号。从第一个人开始报数，凡报到  $n$  的人退出圈子。用 C 语言写出程序，输入  $n$  ( $n \geq 1$ ) 的值，输出最后留在圈子中的人的编号。

**【答案】** 算法描述如下：

```

#include <stdio.h>
#define people 15
    
```

```

struct person{
    int number;
    int nextp;
    link[peplenum+1];
}
main()
{
    int n;
    int i,count,h;
    scanf('%d",&n);
    IF (n<1) {
        printf('n 的值必须大于 1');
        return;
    }
    FOR(i=1;i<=peplenum;i++){
        IF(i==peplenum) link[i].nextp=1;
        ELSE link[i].nextp=i+1;
        link[i].number=1;
    }
    printf('\n');
    count=0;
    h=peplenum;
    WHILE (count<peplenum-1){
        WHILE(i!=n){
            h=link[h].nextp;
            IF (link[h].number) i++;
        }
        link[h].number=0;
        count++;
    }
    printf('\n 最后留在圈里的人的序号是: ');
    FOR(i=1;i<=peplenum;i++)
        IF(link[i].number)
            printf('%6d\n',link[i].number);
}

```

4. 结点类型和存储方式如下:

```

TYPE node=RECORD
    key:integer;
    info:datatype;
    count:integer
END;
VAR R:ARRAY[1..n] OF node;

```

请给出一个排序方法,不移动结点存储位置,只在结点的 count 字段记录结点在排序中的序号(设排序码最大的结点序号为 1)。

**【答案】**算法描述如下:



```

PROCEDURE CountSort(VAR R:ARRAY[1..n] OF node);
VAR i,j:integer;
BEGIN
FOR i:=1 TO n DO
    R[i].count:=0;
FOR i:=1 TO n DO
    FOR j:=1 TO n DO
        IF R[i].key<R[j].key
            THEN R[i].count:=R[i].count+1
            ELSE R[i].count:=1
    END;
END;

```

5. 已知一个递增有序表 $R[1..4n]$ ，并且表中没有关键字相同的记录。按如下方法查找一个关键字为  $k$  的记录：先在编号为 4, 8, 12, ...,  $4n$  的记录中进行顺序查找，或者查找成功，或者由此确定一个继续进行折半查找的范围。

- (1) 设计满足上述过程的查找算法。
- (2) 分析成功查找情况下的平均查找长度，和对整个表进行折半查找相比哪个算法较好？
- (3) 为了提高效率，对本算法可以做何改进。

**【答案】** (1) 先在编号为 4, 8, 12, ...,  $4n$  的记录中进行顺序查找，若没有找到，找到大于  $k$  的位置  $i$ ，然后在  $R[i-3..i-1]$  的范围内进行折半查找。对应的算法如下。

```

int FindElem(SeqList R,int n,KeyType k)
{
    int i=4,low,high,mid;
    if (k<R[1].key || k>R[4*n].key) return -1;
    while(i<=4*n)
    {
        if (R[i].key==k)
            return i; //查找成功返回
        else if (R[i].key<k)
            i+=4;
        else //找到大于 k 的位置 i
            break;
    }
    low=i-3;high=i-1;
    while (low<=high)
    {
        mid=(low+high)/2;
        if (R[mid].key==k) //查找成功返回
            return mid;
        if (R[mid].key>k) //继续在 R[low..mid-1]中查找
            high=mid-1;
        else //继续在 R[mid+1..high]中查找
            low=mid+1;
    }
    return -1;
}

```

(2) 在成功查找情况下，顺序查找中平均关键字比较次数为  $(n+1)/2$ ，然后在 3 个元素的范围内进行折半查找，其平均关键字比较次数为  $\log_2 4 - 1 = 1$ ，所以总的平均查找长度为  $(n+1)/2 + 1 = (n+3)/2$ 。若对整个表进行折半查找，平均查找长度为  $\log_2(4n+1) - 1$ ，显然折半查找更好些。

(3) 对本算法可做这样的改进：由于编号为 4, 8, 12, ..., 4n 的记录是递增有序的，可以将顺序查找改为折半查找，然后在确定范围内再进行折半查找。

6. 设计一个算法，从一给定的顺序表 L 中删除元素值在 x 到 y ( $x \leq y$ ) 之间的所有元素，要求以较高的效率来实现，空间复杂度为  $O(1)$ 。

【答案】可以采用上例的解法，只是将  $L.data[i] == x$  的条件改为  $L.data[i] \geq x \ \&\& \ L.data[i] \leq y$ 。

对应算法如下。

```
void delallxyl(SqList &L, ElemType x, ElemType y)
{
    int i, k=0;
    for (i=0; i<L.length; i++)
        if (!(L.data[i] >= x && L.data[i] <= y))
            {
                L.data[k] = L.data[i];
                k++;
            }
    L.length = k; //表长置为 k
}
```

7. 设 A 和 B 均为下三角矩阵，每一个都有 n 行 n 列。因此在下三角区域中各有  $n(n+1)/2$  个元素。另设有一个二维数组 C，它有 n 行 n+1 列。试设计一个方案，将两个矩阵 A 和 B 中的下三角区域元素存放于同一个 C 中。要求将的下三角区域中的元素存放于 C 的下三角区域中，B 的下三角区域中的元素转置后存放于 C 的上三角区域中。并给出计算 A 的矩阵元素  $a_{ij}$  和 B 的矩阵元素  $b_{ij}$  在 C 中的存放位置下标的公式。

【答案】将 n 阶方阵的下三角 A 复制到 C，B 逆置后复制到 C。核心语句段如下：

```
for(i=0; i<n; i++)
    {
        for(j=0; j<=i; j++) C[i][j] = A[i][j];
        for(j=i+1; j<=n; j++) C[i][j] = B[j-1][i];
    }
```

A 的矩阵元素  $a_{ij}$  和 B 的矩阵元素  $b_{ij}$  在 C 中的存放位置下标的公式：

$$A[i][j] = C[i][j]; \quad \{ 0 \leq i < n, 0 \leq j \leq i \}$$

$$B[i][j] = C[j][i+1]; \quad \{ 0 \leq i < n, i \leq j \leq n \}$$

8. 完成下述用模板类表示文件记录的折半查找算法，算法实现在线性表表示的文件 A 中查找检索码等于 key 值的记录，若找到，则返回其下标值，否则返回 -1。

```
template < class T >
int BinarySearch(T A[], int n, int key)
{
    int low, high, mid;
    low = 0;
    high = n - 1;
    ...
}
```

【答案】算法如下：

```
template < class T >
int BinarySearch(T A[], int n, int key)
{
    int low, high, mid;
    low = 0;
```

2024 年中原工学院 819 计算机专业基础（数据结构、C++程序设计）考研题库[仿真+强化+冲刺]

中原工学院 819 计算机专业基础（数据结构、C++程序设计）之数据结构考研仿真五套模拟题

2024 年数据结构五套仿真模拟题及详细答案解析（一）

一、算法设计题

1. 有一种简单的排序算法，叫做计数排序(Count Sorting)。这种排序算法对一个待排序的表(用数组表示)进行排序，并将排序结构存放到另一个新的表中。必须注意的是，表中所有待排序的关键字互不相同。计数排序算法针对表中的每个记录，扫描待排序的表一趟，统计表中有多少个记录的关键字比该记录的关键字小。假设对某一个记录，统计出计数值为  $c$ ，那么，这个记录在新的有序表中的合适的存放位置即为  $c$ 。

- (1) 给出适用于计数排序的数据表定义。
- (2) 使用 Pascal 或 C 语言编写实现计数排序的算法。
- (3) 对于有  $n$  个记录的表，关键字比较次数是多少？
- (4) 与简单选择排序相比，这种方法是否更好？为什么？

【答案】(1) 数据表定义为：

```
TYPE sortlist=ARRAY[0..n-1] OF key
```

(2) 用 Pascal 语言实现的计数排序算法为：

```
PROCEDURE countsort(VAR oldlist,newlist:sortlist;n:integer);
VAR i,j,count:integer;
BEGIN
  FOR i:=0 TO n DO
    BEGIN
      count:=0;
      FOR j:=0 TO n DO
        IF (oldlist[j]<oldlist[i]) THEN count:=count+1;
        newlist[count]:=oldlist[i]
      END;
    END;
```

(3) 对于有  $n$  个记录的表，关键字比较次数是  $n^2$ 。

(4) 简单选择排序比这种排序好，因为对具有  $n$  个记录的数据表进行简单排序时只需进行  $1+2+\dots+(n-1)=\frac{n(n-1)}{2}$  次比较，且可在原地进行排序。

2. 我们用一个  $m$  行  $n$  列的网格表示一个彩色图像，每个格点表示一个单位区域，且每个格点可以着一种颜色。为简单起见，我们用数字  $0\sim 9$  分别表示 10 种不同的颜色，具有相同颜色的格点可以通过上下左右四个方向连成一片而构成一个区域。例如，在下图 1 中就存在着四个区域。假设我们已对网格中的每个格点都着了色，并指定一个格点  $(x,y)$  和一种颜色  $color$ 。请编写一个将格点  $(x,y)$  所在区域改着  $color$  颜色的程序。例如，对于下图 2，若指定的格点为  $(2, 2)$ ，指定的颜色为 9，则改色后的网格如下图 3 所示；若指定的颜色不是 9，而是 2，则改色后的网格如下图 4 所示。

```
1 0 0
0 0 1
1 0 0
```

图1

```
5 0 6 4 0
0 0 0 6 0
3 0 0 8 0
2 0 0 8 0
0 2 0 0 0
```

图2

```
5 9 6 4 9
9 9 9 6 9
3 9 9 8 9
2 9 9 8 9
0 2 9 9 9
```

图3

```
5 2 6 4 2
2 2 2 6 2
3 2 2 8 2
2 2 2 8 2
0 2 2 2 2
```

图4

【答案】算法描述如下：

**TYPE** SelemType=**RECORD**

ord:integer; {某一网格在路径中的位置}

seat:PosType; {某一网格的坐标位置}

di:integer; {从当前位置走向下一位置的方法: 1:向上 2:向下 3:向左 4:向右}

**END;**

**PROCEDURE** ChangeColor(VAR A:ARRAY[1..m,1..n] OF color;

VAR start:PosType; c:color);

VAR curpos:PosType;

curstep:integer;

**BEGIN**

InitStack(S);

curpos:=start;

curstep:=1; {试探第一步}

**REPEAT**

**IF** (Pass(curpos)) {当前网格可通过}

**THEN BEGIN**

e:=(curstep,curpos,1); {e是入栈信息结构}

push(S,e);

**IF** EmptyStack(S) **THEN** break;

curpos:=NextPos(curpos,1); {下一位置是当前位置的上方}

curstep:=curstep+1 {试探下一位置}

**END**

**ELSE BEGIN**

**IF** (not StackEmpty(S))

```

THEN BEGIN
    pop(S,e);
    WHILE ((e.di=4) and (not StackEmpty(S))) DO
        BEGIN
            MarkPrint(e.seat); {留下不能通过的标记
            pop(S,e)           并退回一步}
        END;
    IF (e.di<4) THEN
        BEGIN
            e.di:=e.di+1; {换下一方向试探}
            push(S,e);
            curpos:=NextPos(e.seat,e.di); {设置当前位置
            END           是该新的向上的相邻位置}
        END
    END
END
UNTIL (not StackEmpty(S));
END;
```

3. 二项式 $(a + b)^n$ 展开式的系数为

$$\begin{cases} C(n,0) = 1, C(n,n) = 1, & \text{对于 } n \geq 0 \\ C(n,k) = C(n-1,k) + C(n-1,k-1), & \text{对于 } 0 < k < n \end{cases}$$

形成著名的杨辉三角形，如下所示：

n = 0										1							
n = 1									1	1							
n = 2									1	2	1						
n = 3									1	3	3	1					
n = 4									1	4	6	4	1				
n = 5									1	5	10	10	5	1			
n = 6									1	6	15	20	15	6	1		
n = 7									1	7	21	35	35	21	7	1	
n = 8									1	8	28	56	70	56	28	8	1

(1) 试写一个递归算法，根据以上公式生成 $C(n,k)$ ;

(2) 试画出计算 $C(6,4)$ 的递归树;

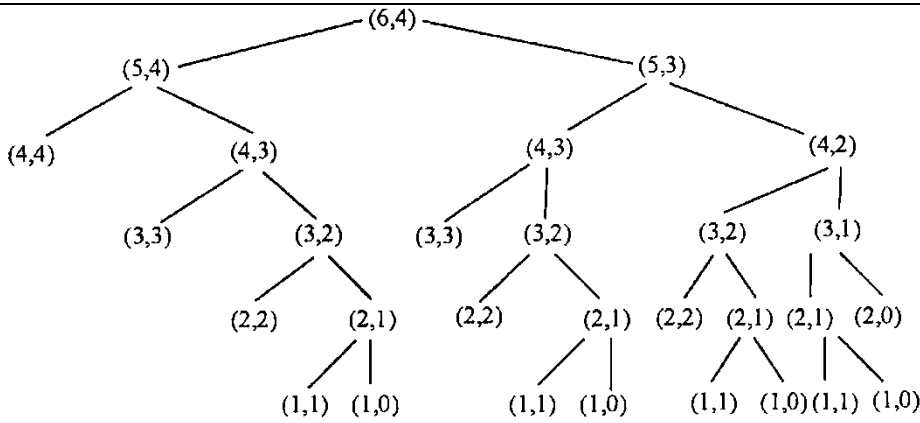
(3) 试写一个递归算法，既不用数组也不用栈，对于任意的 $0 \leq k \leq n$ 计算 $C(n,k)$ 。

**【答案】** (1) 求 $C(n,k)$ 的递归算法如下：

```

int c(int n, int k)
{
    if (k == 0 || k == n) return(1);
    return(c(n-1,k) + c(n-1,k-1));
}
```

(2)  $C(6,4)$ 的递归树如下图所示：



(3) 求  $C(n,k)$  的非递归算法如下 (根据二项式定理  $C(n,k) = \frac{n(n-1)\cdots(n-k+1)}{k!}$ ):

```

int c(int n, int k)
{
    long x = 1, y = 1;
    int i;
    for (i = 1; i <= k; i++) x * = i;
    for (i = n; i >= n - k + 1; i--) y * = i;
    return(y/x);
}

```

4. 请给出遍历广义表的算法, 按照广义表的逻辑结构顺序, 打印广义表所有元素结点上的数据域。

**【答案】** 与二叉树前序遍历的递归算法相似。

```

preorder(gnode * p)
{
    if (p! = NULL)
    {
        if (p->tag == 0) printf('%c', p->val.data);
        else preorder(p->val.sublist);
        if (p->link! = NULL) preorder(p->link);
    }
}

```

5. 请编写直接插入排序算法。

**【答案】** 假定第 1 个元素有序, 从第 2 个元素起, 依次插入前面有序子文件中。核心语句如下:

```

for(i=2; i<=n; i++) //假定第一个记录有序
{
    R[0]=R[i]; j=i-1; //将待排序记录放进监视哨
    while(R[0].key<R[j].key) //从后向前查找插入位置,同时向后移动记录
        {R[j+1]=R[j]; j=j-1;}
    R[j+1]=R[0] //将待排序记录放到合适位置
}

```

## 二、应用题

6. 简述直接插入排序、简单选择排序、2 路归并排序的基本思想以及在时间复杂度和排序稳定性上的差别。

**【答案】** 直接插入排序的基本思想是基于插入, 开始假定第一个记录有序, 然后从第二个记录开始,

附赠重点名校：数据结构 2016-2022 年考研真题汇编（暂无答案）

第一篇、2022 年数据结构考研真题汇编

2022 年沈阳工业大学数据结构考研专业课真题

沈阳工业大学

2022 年硕士研究生招生考试题签

（请考生将题答在答题册上，答在题签上无效）

科目名称：数据结构

第 1 页 共 2 页

一、解释下列名词(共 20 分，每小题 4 分)

1. 队列      2. 二叉树      3. 平均查找长度      4. 时间复杂度      5. 有向完全图

二、填空(共 30 分，每空 3 分)

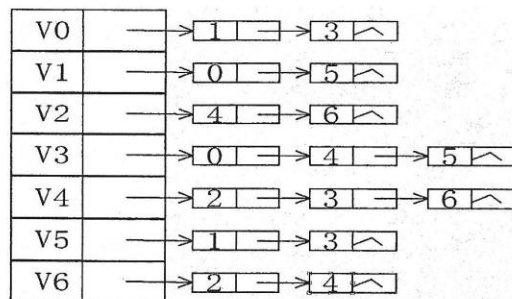
- 数据的存储结构包括：顺序存储、\_\_\_\_\_、索引存储和散列存储。树形结构中的数据元素之间存在\_\_\_\_\_的关系。
- 在栈中存取数据应遵循的原则是\_\_\_\_\_。解决顺序队列“假溢出”的方法是采用\_\_\_\_\_。
- 二维数组  $A[0..3][0..4]$  的首地址是 1000，每个数据元素长度是 6，则  $A[2][3]$  的地址是\_\_\_\_\_。广义表  $((c, d), c, d)$  的表尾是\_\_\_\_\_。
- 具有  $n$  个结点的完全二叉树的深度是\_\_\_\_\_。如果一棵二叉树度为 2 的结点个数为 5，度为 1 的结点个数为 3，那么叶子结点的个数为\_\_\_\_\_。
- 有向图  $G$  是由  $(V, VR)$  组成，其中顶点的集合  $V=\{A, B, C, D, E\}$ ，弧的集合  $VR=\{\langle A, B \rangle, \langle A, C \rangle, \langle C, B \rangle, \langle C, D \rangle, \langle B, D \rangle, \langle E, A \rangle, \langle E, B \rangle, \langle E, C \rangle\}$ ，则该图的拓扑排序序列为\_\_\_\_\_，顶点  $D$  的入度为\_\_\_\_\_。

三、解答下列问题(共 50 分，每题 10 分)

- 设一棵二叉树结点的先序遍历序列是：ABFCGDEH，中序遍历序列是：FBGCADHE，要求：
  - 画出这棵二叉树，并写出这棵二叉树的后序遍历序列的结果。
  - 画出该二叉树对应的森林。
  - 画出该二叉树的后序线索二叉树。
- 设一个关键字序列为 {25, 32, 48, 16, 21, 37, 9, 12}，按要求完成以下操作：
  - 画出其对应的二叉排序树，并求在等概率情况下查找成功的平均查找长度 (ASL)。
  - 如果哈希表的长度为 12，哈希函数为  $H(\text{key})=\text{key} \bmod 9$ ，用线性探测再散列方法创建哈希表，并求在等概率情况下查找成功的平均查找长度 (ASL)。
- 设一组权值序列为 (8, 11, 5, 17, 7, 29)，按要求完成以下操作：
  - 根据权值构建哈夫曼树。
  - 写出每个权值对应的哈夫曼编码。
  - 求带权路径的长度 (WPL)。



4. 已知一个无向图的存储结构如下图所示：



按要求完成以下操作：

- (1) 画出对应的无向图。
  - (2) 写出无向图对应的邻接矩阵。
  - (3) 写出从顶点 V0 出发深度优先和广度优先遍历的序列。
5. 已知一组关键字的序列为 {23, 48, 17, 9, 27, 3, 20, 69, 55}，请写出采用归并排序法对该序列作升序排序时每一趟的结果。

#### 四、编程题，程序设计语言不限(除 C、C++ 外请标明是哪种语言) (共 50 分，每题 10 分)

1. 从键盘输入 10 个整数存储在一个顺序表中，再输入一个整数赋值给变量 x，然后将这个顺序表划分成两部分，其中左半部分的每个整数均小于 x，右半部分的每个整数均大于等于 x。
2. 设有两个集合 A 和 B 分别存储在两个单链表中，设计求 A 和 B 的交集 C 的程序，集合 C 也存储在一个单链表中。(集合元素类型自行定义)
3. 从键盘输入二叉树数据，采用二叉链表建立二叉树，并写出对该二叉树前序遍历的递归程序。(注意：程序前面要用文字说明键盘输入二叉树数据的格式)
4. 从键盘输入一个有向图，建立一个邻接矩阵。再输入一个顶点编号 num，输出编号为 num 的顶点的度(入度和出度之和)。
5. 从键盘输入 10 个整数到一个一维整型数组，然后采用简单选择排序法进行排序并输出。

2022 年广东财经大学 809 数据结构考研专业课真题

## 广东财经大学硕士研究生入学考试试卷

考试年度：2022 年 考试科目代码及名称：809-数据结构（自命题）

适用专业：085400 电子信息

[友情提醒：请在考点提供的专用答题纸上答题，答在本卷或草稿纸上无效!]

### 一、单项选择题（10 题，每题 2 分，共 20 分）

- 算法的时间复杂度取决于（ ）。  
A. 问题的规模      B. 待处理数据的初态      C. 计算机的配置      D. A 和 B
- 线性表的顺序存储结构中，数据元素的逻辑位置和物理位置的关系是（ ）。  
A. 不一致的      B. 一致的      C. 大致相同      D. 个别元素相同
- 在一个有  $n$  个元素的顺序表中，插入一个元素平均要移动的元素个数为（ ）。  
A.  $(n-1)/2$       B.  $n/2$       C.  $(n+1)/2$       D.  $n$
- 若顺序栈  $S$  存储在数组  $stack[MAXSIZE]$  中，栈顶位置  $top$  初值为  $-1$ ，则元素  $e$  进栈的操作是（ ）。  
A.  $S.stack[S.top++] = e;$       B.  $S.stack[++S.top] = e;$   
C.  $S.stack[S.top--] = e;$       D.  $S.stack[--S.top] = e;$
- 链队列  $Q$  的结点结构为： $(data, link)$ ，指针  $front$  指向队首元素， $rear$  指向队尾元素，则出队元素到变量  $x$  中的操作（ ）。  
A.  $x=Q.front->data; Q.front=Q.front->link;$   
B.  $Q.front=Q.front->link; x=Q.front->link;$   
C.  $x=Q.rear->data; Q.rear=Q.rear->link;$   
D.  $x=Q.rear->data; Q.rear=Q.front;$
- 一个递归算法必须包括（ ）。  
A. 递归部分      B. 终止条件和递归部分      C. 迭代部分      D. 终止条件和迭代部分
- 一棵非空二叉树的先序遍历序列和中序遍历序列相同，则该二叉树一定满足（ ）。  
A. 所有的结点均无左孩子      B. 所有的结点均无右孩子  
C. 只有一个叶子结点      D. 不存在这样的二叉树
- 按照满二叉树的编号顺序对深度为  $k$  的完全二叉树编号，则编号最小的叶结点的编号是（ ）。  
A.  $2^{k-1}-1$       B.  $2^{k-1}$       C.  $2^{k-2}+1$       D.  $2^k-1$
- 一棵完全二叉树的第 7 层有 24 个叶子结点，则整个二叉树的结点数至多为（ ）个  
A. 87      B. 206      C. 207      D. 231
- $G$  是一个非连通无向图，共有 36 条边，则该图至少有（ ）个顶点  
A. 7      B. 8      C. 9      D. 10

### 二、名词解释（10 题，每题 3 分，共 30 分）

- 队列
- 算法的空间复杂度
- 深度优先搜索
- 最小生成树
- 有向无环图
- 关键路径

以上为本书摘选部分页面仅供预览，如需购买全文请联系卖家。

全国统一零售价： **¥ 368.00元**

卖家联系方式：

微信扫码加卖家好友：

