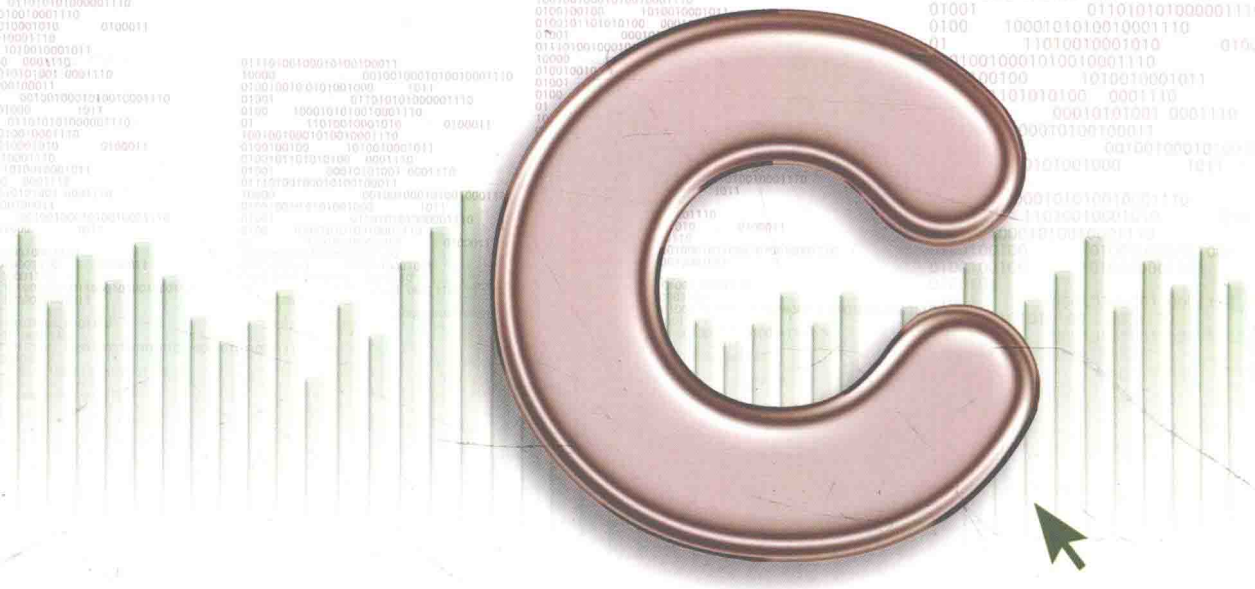
高等教育应用型人才“十三五”规划教材
高等教育应用型人才 计算机类专业 规划教材
本教材学习网站: www.lgsworks.com
江苏高校品牌专业建设工程资助项目



算法与数据结构

（C语言版）

李广水　钱海忠　主编

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等教育应用型人才“十三五”规划教材
高等教育应用型人才计算机类专业规划教材

算法与数据结构
（C语言版）

李广水　钱海忠　主　编
何　静　Fuguo Wei　副主编

電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

内 容 简 介

本书共分 10 章，分别是绪论、线性表、栈和队列、串、多维数组和广义表、树、图、排序、查找和经典算法分析。全书的算法程序基于 C 语言实现。

本书的知识体系符合当前对该课程的主流认知，从文字组织及示例设置上，将教材划分为四个部分，包括基础理论、基础应用、常规应用和经典算法分析。既体现由理论到实践的递进，又保证教材的紧凑完整。

本书虽然是应用型本科教材，但几乎覆盖了该课程的全部知识，并给出了算法实现代码，对比较复杂的问题不仅给出设计思路，还给出具体示例分析，以帮助读者朋友理解掌握，因此本书也适合技术人员参考使用，同时也可以作为相关专业学生的考研辅导用书。

图书在版编目（CIP）数据

算法与数据结构：C 语言版 / 李广水，钱海忠主编. —北京：电子工业出版社，2017.8

ISBN 978-7-121-31513-8

Ⅰ. ①算… Ⅱ. ①李… ②钱… Ⅲ. ①算法分析-高等学校-教材 ②数据结构-高等学校-教材

Ⅳ. ①TP301.6 ②TP311.12

中国版本图书馆 CIP 数据核字（2017）第 105071 号

策划编辑：李 静（lijing@phei.com.cn）
责任编辑：朱怀永
文字编辑：李 静
印 刷：北京京师印务有限公司
装 订：北京京师印务有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：15.25 字数：390.4 千字
版 次：2017 年 8 月第 1 版
印 次：2017 年 11 月第 2 次印刷
定 价：38.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254604，lijing@phei.com.cn

前言

将本门课程多年的教学经验总结归纳成书，是编者的初衷。

作为计算机类相关专业学生的专业基础课程，“算法与数据结构”具有较长远的开设历史，同类教材也非常多，本书知识体系及内容撰写具有以下一些特色。

（1）知识结构符合当前对该课程的主流认知。

（2）针对当前大学生编程普及，但对计算机工作原理却比较薄弱这一特点，内容编写中强调了顺序存储与链式存储，以加强同学们对内存管理及计算机工作原理的了解。

（3）强调函数的重要性，包括示例的函数的讲解、栈在递归函数的应用等，以加强模块化设计的理念。

（4）重视整本书的内在逻辑性，从文字组织及示例设置上，隐性地分为四个部分：基础理论（第1～2章），基础应用（第3～7章），常规应用（第8～9章），经典算法分析（第10章）。既体现教材由理论到实践的递进，也保证72（56+16）课时左右的紧凑完整。

作为应用型本科专业基础教材，“创新”并不是一个主要的追求目标。本书基于多年教学经验，在汇集出本课程的一些重点和难点基础上，针对不同内容设计解析过程，除了文字描述外，还包括图示分析、示例演算、表格总结等，力求将问题讲清讲透。其中，KMP算法，二叉树线索下的前驱后继的查找，图的遍历，图的最短路径、关键路径等内容的讲解，都不同于一般教材直接给出对应的公式或解决方案，而是从思维的习惯性出发，采用朴实的举例或说理，希望更加有助于同学们的理解。诸如此类的尝试，虽然在教学中实践多次，但作为一本正式出版教材中的内容，我们依然谨慎处理，首先由编写老师共同分析，再请学生参照其他教材阅读比较，给出心得体会和改进意见。对每一章结尾的小结，给出具有启发意义的观点，即使未经证实但在没有证伪的情况下，也呈现给广大师生，以提高其深究、探索的兴趣。

全书采用C语言实现函数，方便一般高校培养计划的顺利实施，同时，因为C语言对底层的操作能力要求及非封装特性，可以更为全面地刻画出数据结构与算法的一些关键问题，以帮助学生对计算机编程有更深刻的理解。

由于我们一直采用由唐善策等主编的《算法与数据结构——用C语言描述》，所以在本次编写过程中，参照了该教材部分内容，在此特别申明并表示感谢。其他参考文献，已列于书后，在此对相关作者一并表示感谢。

本书的1～9章内容作为本课程的传统知识体系应该课内讲授，其中一些较难的章节，例

如，约瑟夫问题、迷宫问题等，可以考虑选讲。最后 1 章，可以仅选择一两个算法进行讲解。

本书由金陵科技学院李广水、钱海忠两位老师任主编，广东省科学技术情报研究所何静从应用实践方面、昆士兰科技大学 Fuguo Wei 从软件工程国际合作培养角度提出编写意见，对教材的整体结构及内容组织给予了极大的帮助，任本书副主编，李广水进行全书统稿工作。

真心感谢电子工业出版社李静编辑的悉心审稿，她认真的工作态度、执着的精神，真实地感动了我们，也提高了我们对本书质量保证方面的信心。

即便如此，由于知识及能力的有限，疏漏及不足之处依然难免，恳请广大读者朋友给出批评建议，在此表示衷心感谢！

作者邮箱：yz_lgs@126.com

编　者

2017 年 6 月

目 录

第 1 章　绪论

“算法与数据结构”是计算机及其相关专业的专业基础课，通过该课程的学习，可以比较系统地了解数据在计算机中的存储及运算特点，掌握一些经典的数据结构及算法，为后续编程课程学习和开发应用程序奠定良好的基础。

本章内容主要介绍数据结构的相关概念，利用计算机求解问题的一般过程，算法特点及分析。

本章的其中一个重点是理解真实世界的问题，如何描述出适合计算机解决的过程；另一重点是函数及其特点。

1.1　数据结构的概念

一般而言，可以将**数据结构**理解为：真实世界中的数据经过一定的加工，有效地存储在计算机中并在其上实现相应的运算。其中，在真实世界中的数据依然称为数据，而对应于计算机中的存储称为“**结点**”。

真实世界中同一问题的不同数据称为“**元素**”，如一系列整数、26 个英文字母，当然也包括非简单类型的数据，如学生的自然信息（见表 1.1），其中每一个元素可能包含学号、姓名、性别、籍贯、出生日期、联系电话等多个“**数据项**”，这样的数据在高级语言如 C 语言中须要利用结构体来定义，在这里称为“**结构型数据**”。

表 1.1　学生自然信息表

学号	姓名	性别	籍贯	出生日期	联系电话
1501360001	曾小芹	女	江苏南京	1997-11-8	1891891212
1501360005	王志军	男	浙江宁波	1998-5-11	1891891367
1501360003	朱正红	女	北京市	1997-4-16	1891892286
1501360026	于越	男	上海市	1996-12-22	1891893066

我们将真实世界中数据之间的关系称为“**逻辑关系**”，如一个班级的学生的自然信息依据学号存在且从小到大的线性关系排序，逻辑结构常常可以采用二元组 $S=(D, R)$ 来表示，其中，D 是元素的有限结合，R 是定义在 D 上的关系集合。

为了利用计算机处理数据，需要将其存储在计算机中形成关系，这种关系称为“**物理关**

系”。相同逻辑关系的数据可能采用不同的物理存储结构，其目的一方面是方便地保存数据之间的逻辑关系；另一方面，更加有效地进行相应的运算。数据存储的主要逻辑结构、物理结构及其对应的运算如下：

- 数据的逻辑结构，线性结构、非线性结构；
- 数据的物理结构，顺序存储、链式存储、索引存储、散列储存；
- 数据的运算，插入、删除、查找、修改、排序。

下面首先介绍与数据结构相关的概念。

（1）线性结构

除了第一个数据元素外，每个元素都有一个唯一的前驱；除了最后一个元素外，每个元素都有一个唯一的后继。我们日常排队所形成的人与人之间的关系就是一个线性结构。本书第 2～4 章介绍线性结构。需要注意的是，一些看起来像非线性结构的数据通过适当的转换也可以转化为线性结构来处理，如在第 5 章我们将要学习的矩阵类型的数据集。

（2）非线性结构

一个元素可能有多个前驱和多个后继，两种典型的非线性结构分别是树和图。其中树形结构是一对多的关系，家族中一对父母对应多个孩子，就是典型的树形结构，树及其相关理论将在第 6 章进行学习。图形结构的数据元素之间存在多对多的关系，如教师与学生之间就是图形关系，因为一个老师可以教很多学生，一个学生可以选修几个老师的课程，图及其相关知识将在第 7 章学习。

（3）顺序存储

顺序存储要求逻辑上相邻的元素物理位置也必须相邻，元素间的逻辑关系由存储结点的邻接关系决定。顺序存储在高级语言中一般采用数组实现。

（4）链式存储

该方法不要求逻辑上相邻的元素在内存中必须存在相邻的结点，元素之间的逻辑关系通过结点附加的指针描述，由此得到的存储称为链式存储。高级语言中采用指针来实现。

（5）索引存储

索引存储要求在存储结点信息的同时，建立附加的索引表，索引表中的每一项称为索引项，索引项的形式一般为（关键字，地址），关键字能唯一标志一个结点的哪些数据项。假设学生基本信息包括有照片、声音等大量数据，将每个学生基本信息保存后，基于学号关键字建立成的索引表将包含学号，以及该学生信息存储在内存中的起始地址。

（6）散列存储

散列存储的主要思想是基于所要存储的数据值决定所要存放的内存地址，这种存储方式便于查找，但对内存的要求也非常高，将在第 9 章进行详细介绍。

上面列出的相关运算概念比较易于理解，不再一一解说。需要说明的是，上述 6 种运算是普遍的，对于不同的数据类型和实际情况，可能会强调不同的运算或一些新的运算，如字符串中进行子串的查找、两个串的连接等，这些都可以由这些基本运算组合实现。

1.2 为什么要学习数据结构

早期的计算机主要解决数值计算问题，因此程序设计者关心的是程序的设计技巧、正确

性、运算结果的精度等。而随着计算机的快速发展，非数值计算问题越来越普遍，这类问题所涉及的数据已不是单纯的数值，已无法完全使用数学表达式进行描述，所要解决的问题也不是一个简单的计算结果，可能涉及大量的数据更新、查找、分析等功能。由于数据量的庞大及处理的复杂性，对程序的运行效率也提出了新的要求。这时，针对不同的数据特征及实现功能，设计出对应的存储结构及有效的解决思路就显得非常必要，这正是数据结构研究的内容。这就是所谓的“算法+数据结构=程序”理论。

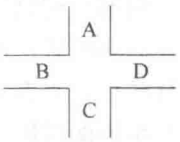


图 1.1　十字路口示意图

下面以信号灯问题为例，举例说明一个真实问题的求解思路。

考虑一个十字路口（如图 1.1 所示）共有 4 条道路，其中 A，B，C 都是双行线，而 D 是单行线，只能往 D 的方向行驶。需要为此设计一个安全有效的信号灯。

1. 分析

依据实际情况，首先确定所有可能的线路。为简单起见，以 AB 表示由从 A 向 B 方向行驶的路线，由此，可以得到以下 9 条路线：AB，AC，AD，BA，BC，BD，CA，CB，CD。如果在某个时间段内只有一条线路可以通行，可将整个路段分成 12 组，显然是可以的，但这种方式的行驶效率太低。我们的目的是希望尽可能少分组，从而保证在一个固定的时间段，每组获得更多的分配时间。

由图 1.1 可以看出，AC，CA，DA，BC 是不冲突的，因此可以分为一组。但实际解决这一问题时，需要首先找出所有可能的冲突线路，为此，分析得到表 1.2。

表 1.2　十字路口各行驶冲突路线

当前线路	冲突线路
AB	无
AC	BD，BA，CB
AD	CA，BA
BA	AC，AD
BC	无
BD	AC，CA，CB
CA	BD，AD
CB	AC，BD，BA
CD	无

2. 建模

依据表 1.2，将每一个线路用一个小圆圈表示，将有冲突的线路的两个圆圈用一条实线相连，由此，得到一个**无向图**，也称为着色图，如图 1.2 所示。

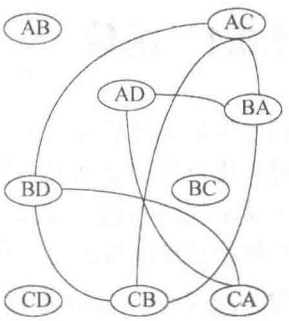


图 1.2　十字路口着色问题

依据图 1.2，可以将没有线相连的线路分为一组，这样，既保证了每组线路的不冲突，也实现了最少分组的原则，见表 1.3。

当然，依据图 1.2，还有多种其他分类方法，由于 CD，AB，BC 三条线路与任何线路没有冲突，因此，放

在任意一组都是可以的。

上述问题的解决思路是一种经典的“着色问题”。其起源是如果将有边相连的所有圆圈涂上同一种颜色，那么有多少种颜色就可以分为多少组。着色问题可以解决很多现实事情，如果将所有小圆圈理解为不同国家，有线相连表示两个国家之间疆域接壤，则可以采用着色问题绘制出世界地图。同样的，如果某高校举办校运动会，由于一个老师可以参加多个比赛项目，为保证比赛的顺利进行和使得比赛日程最短，也可以采用着色问题解决。

表 1.3 依据着色图的十字路口交通分组

组数	线路
第一组	AB，BC，CD，AC，CA
第二组	AD，CB
第三组	BD，BA

3. 实现

利用抽象出来的模型，设计出相应的存储方案并编写应用程序，从而得到问题的求解。

以上内容是解决问题的一般过程，而数据结构知识提供计算机处理数值特别是非数值问题的一些基本原理及其常用方法，例如，上述信号灯问题的核心就是图的相关知识，即使已经抽象出如图 1.2 所示模型，但如何利用计算机程序存储图结构数据并进行有效分组依然是一个比较麻烦的问题。而本书第 7 章将进行图的存储、遍历等知识的讲解，由此也为解决这类问题奠定了坚实的基础。因此，当拥有了数据结构相关知识后，学生在面对实际问题时，将自觉利用所学知识分析问题并进行抽象建模，可以设计并编写出合理的应用程序。

1.3 算法

算法是由有穷规则构成的、为解决某一问题的运算序列。算法包括**输入**、**输出**，输入一般是算法开始时给出的初始条件，而基于这些条件所得到的结果称为输出。

算法必须具有**有穷性**、**确定性**、**可行性** 3 个特性。有穷性是指一个算法在执行了有穷的步骤之后必须结束，确定性是指算法的每一步必须有确切的定义，可行性是指设计的算法必须是可行的，即能够被计算机识别并正确执行。

1.3.1 函数

如果将算法理解成思想，那么函数是程序的一部分，是实现算法的载体。

模块化设计的思路就是将一个大的系统分割成多个小的功能模块，这不仅便于管理维护，并且也易于相同功能的多次复用。而函数可以理解成一个基本功能模块的实现。

类似于其他高级语言，在 C 语言中函数的定义严格规范，其具体格式如下：

```
函数的返回类型  函数名（形式参数）
{
```

```
    说明语句
    执行语句
}
```

1. 算法 1.3.1

以下函数的功能是交换数组中第 i 个元素和第 j 个元素的值，该函树有 4 个参数，分别是整型数组、数组长度、要交换的两个位置，没有返回值。

```
void arrayread（int[   ]a,int n,int i,j）{
  int temp;
  temp=a[i];
  a[i]=a[j];
  a[j]=temp;
}
```

2. 算法 1.3.2

以下函数的功能是求 n 个结点的完全二叉树的深度，该算法有 1 个整型参数，为完全二叉树的结点个数，返回值为整型，为该二叉树的深度。

```
int deeptree（int n）{
  int i=1;
  while （i<=n）
  I=i*2;
  return i;
}
```

3. 算法 1.3.3

以下是选择排序函数，该函树的 2 个参数分别为分布式数组及该数组的有效数据长度，没有返回值。

```
void selectsort（int[] a,int n ）{
  Int  i,j,k,t;
  for（i=0;i<n-1;i++）{
    J=i;
    for（k=i+1;k<n;k++）
    if（ a[k]>a[j] ）  j=k;
    t=a[i];
    a[i]=a[j];
    a[j]=t;
  }
}
```

4. 算法 1.3.4

以下算法是求两个 n 阶方阵的乘积函数，该函树 3 个参数，分别是二维数组描述的矩阵 A，B，C，与 AB 相乘的结果存放于 C 中。

```
void matrixmlt（A,B,C）{
  float A[][n],B[][n],C[][n];
  {
    Int i,j,k;
    for  （i=0;i<n;i++）
```

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

5. 算法 1.3.5

在 C 语言中，函数的定义不可以嵌套，但函数可以嵌套调用自身，这称为递归调用，递归调用在实际编程中得到广泛应用，下列函数是求整数 n 的阶乘，该函数有 1 个整型参数 n，返回值为整型。

```
int  fac（n）{
  If n==1 return 1;
  else return n*fac（n-1）;
}
```

1.3.2 算法分析

一个算法仅仅是正确的还不够，还应该考虑算法的效率问题，这里的效率主要指时间效率和空间效率，就现实情况而言，前者更为重要。

要分析算法的时间或空间效率，首先须要了解问题的规模，问题的规模一般基于问题所处理的数据量的大小而确定。例如，对 n 个自然数累计求和，这时问题的规模就是 n，对 n 个整数进行排序，这时问题的规模也是 n。算法空间效率分析就是在某种算法下，除保存初始数据外还需要额外占用的存储空间大小，而算法的时间效率分析则为执行该算法所有简单语句之和。一般而言，算法的时空效率常常是规模 n 的函数，写作 $O(f(n))$，其中 $f(n)$ 是规模 n 的函数。

算法 1.3.1 中，由于读取数组中某一位置的数据，依据数组初始地址直接计算出所要读取数据的地址（称为“随机读取”），因此与规模 n 无关，此时算法的时间效率记为 $O(C)$，类似的算法仅需要额外占用一个整型空间的内存大小，与规模 n 无关，其空间效率也记为 $O(C)$。

算法 1.3.2 中，该算法不占用额外的存储空间，所以，空间效率为 $O(C)$，该算法的第一条语句与 n 无关，主要分析第二条 while 语句，假设第二条语句须要执行 $f(n)$ 次，则有：

$$2^{f(n)} \leqslant n$$

即 $f(n) \leqslant \log_2 n$，由此可知，该算法的时间效率为 $O(\log_2 n)$。

算法 1.3.3 中，第一层 for 循环须要执行 n 次，当第一层的 i 值由 1 逐渐增大至 n 时，第二层 for 循环可以计算如下：

i=1 执行 1 次
i=2 执行 2 次
i=3 执行 3 次
⋮

故一共需执行 $1+2+3+\cdots+n=n\times(n+1)/2$ 次。考虑到规模 n 很大时，系数 1/2 及 n 与 n^2

相比时很小则可以忽略，因此该算法的时间复杂度记为 $O(n^2)$，同样，该算法不占用额外的存储空间，所以，空间效率为 $O(C)$。

算法 1.3.4 中，该算法也不占用额外的存储空间，所以，空间效率为 $O(C)$。在分析时间效率时，可以发现最里层的 for 循环无论第二层循环的 n 值为多少，总是执行 n 次，因此，第一层循环的 i 值无论为多少，总有 n 次嵌套循环，因此整个循环的次数为

$$n+2n+3n+\cdots+nn=n\times[n\times(n+1)/2]$$

依据前面的分析，该算法的时间效率为 $O(n^3)$。

在算法 1.3.5 中，函数采用递归来实现，递归在实际运行过程中，使用栈来存储中间的过程，即在调用函数 fac（n）时需要首先求出函数 fac（n−1），因此要在内存中保留 fac（n）的状态。以此类推，在该函数的运行过程中，需要开辟与 n 线性相关的内存空间保留中间状态，因此该函数的空间效率为 $O(n)$。另外，虽然程序本身没有与规模 n 相关的循环，但递归调用确是与 n 线性相关的，因此时间效率也是 $O(n)$。

算法分析的目的是针对某一具体问题，设计出时空效率更高的算法。依据数学相关知识，在 n 足够大时有：

$$O(C)<O(\log_2 n)<O(n)<O(n\log_2 n)<O(n^2)<O(n^3)<O(2^n)$$

本章小结

本章简要介绍了数据结构 3 个方面的内容，包括数据的逻辑结构、物理结构及运算，基于一个典型的着色示例探讨了利用计算机求解实际问题的一般思路，分析过程事实找出已知信息和求解的问题；建模过程实际上是针对该问题设计出可行的求解方案，在此借用已经学过的求解方案或将问题演化为某一经典问题，是一种常用方法；求解是针对已经明确的方案进行程序设计及实现。

本章特地开辟一个段落讲解函数的基本特点，一方面是想强调函数在现实编程中的重要性；另一方面，本书所有算法的实现都采用函数的方式，读者在实际上机过程中仅须编写一个主程序来调用该函数即可。

本章最后依据几个函数讲解算法的时空效率问题，并列出常用复杂度排序，虽然是随机给出，但一般而言，给出 $\log_2 n$ 而不是常见的以 10 为底的 $\log_{10} n$ 或自然对数 $\ln n$，其一个重要原因是因为非线性数据结构中一个很重要的逻辑结构——二叉树的很多属性与 $\log_2 n$ 紧密相关。

本章习题

一、填空题

1. 数据结构被形式地定义为（D，R），其中 D 是_____的有限集合，R 是_____的有限集合。

2. 数据结构包括数据的_____、数据的_____ 和数据的_____这 3 个方面的内容。

3. 数据结构按逻辑结构可分为两大类，分别是_____和 _____。

4. 线性结构中元素之间存在_____关系，树形结构中元素之间存在_____关系，图形结构中元素之间存在_____关系。

5. 数据的存储结构可用 4 种基本的存储方法表示，分别是_____、_____、_____、_____。

6. 数据常用的运算有 5 种，分别是_____、_____、_____、_____、_____。

7. 一个算法的效率可分为_____效率和_____效率两种。

二、选择题

1. 非线性结构是数据元素之间存在一种（　　）。

A. 一对多关系　　B. 多对多关系

C. 多对一关系　　D. 一对一关系

2. 数据结构中，与所使用的计算机无关的是数据的什么结构（　　）。

A. 存储　　B. 物理

C. 逻辑　　D. 物理和存储

3. 算法分析的目的是（　　）。

A. 找出数据结构的合理性　　B. 研究算法中的输入和输出的关系

C. 分析算法的效率以求改进　　D. 分析算法的易懂性和文档性

4. 算法分析的两个主要方面是（　　）。

A. 空间复杂性和时间复杂性　　B. 正确性和简明性

C. 可读性和文档性　　D. 数据复杂性和程序复杂性

5. 计算机算法指的是（　　）。

A. 计算方法　　B. 排序方法

C. 解决问题的有限运算序列　　D. 调度方法

三、分析题

分析下面各程序段的时间复杂度。

1.
```
for (i=0;  i<n; i++)
for (j=0; j<m; j++)
A[i][j]=0;
```

2.
```
s=0;
for (i=0; i<n; i++)
for(j=0; j<n; j++)
s+=B[i][j];
sum=s;
```

3.
```
x=0;
for(i=1; i<n; i++)
for (j=1; j<=n-i; j++)
x++;
```

4.
```
i=1;
while(i<=n)
i=i*3;
```

四、简答题

1. 简述将一个现实问题转化为可用计算机解决的问题的过程。

2. 简述构建或使用函数应关注哪些属性。

3. 设有数据逻辑结构 $S=(D, R)$，试按各小题所给条件画出这些逻辑结构的图示，并确定相对应关系 R，哪些结点是开始结点，哪些结点是终端结点。

（1）$D=\{d1, d2, d3, d4\}$　　$R=\{(d1, d2), (d2, d3), (d3, d4)\}$

（2）$D=\{d1, d2, \cdots, d9\}$

$R=\{(d1, d2), (d1, d3), (d3, d4), (d3, d6), (d6, d8), (d4, d5), (d6, d7), (d8, d9)\}$

（3） $D=\{d1, d2, \cdots, d9\}$

$R=\{(d1, d3), (d1, d8), (d2, d3), (d2, d4), (d2, d5), (d3, d9), (d5, d6), (d8, d9), (d9, d7), (d4, d7), (d4, d6)\}$

4. 田径赛的时间安排问题（无向图的着色问题）。设有六个比赛项目，规定每个选手至多可参加3个项目，有5人报名参加比赛（见表1.4），设计比赛日程表，使得在尽可能短的时间内完成比赛。

表1.4 选手报名情况

姓名	项目1	项目2	项目3
王兴元	跳高	跳远	100米
张玉屏	标枪	铅球	
赵刚	标枪	100米	200米
吴建宇	铅球	300米	跳高
程娟	跳远	200米	