

计算机专业教学辅导丛书

数据结构（C 语言篇）

——习题与解析

（修订版）

李 春 葆 编著

清 华 大 学 出 版 社

<http://www.tup.tsinghua.edu.cn>

(京)新登字 158 号

内 容 提 要

本书根据数据结构课程的教学大纲的要求,提供了作者多年教学中积累、收集与验证的有关数据结构的基本内容及相关题解。全书共分 13 章,每章先给出内容概述,然后给出该章的题解,题解分为基本题和习题解析两部分,前者由选择题和填空题两种题型组成,直接给出答案;后者对每个习题的解答给出了完整的过程。

全书概念清晰,习题覆盖面广,既收集了较容易的题目,也收集了难度适中和较高难度的题目,如一些高校计算机专业招收硕士研究生的数据结构试题。

本次修订主要对第 1 版中部分程序进行了补充和完善;对其中少量错误和遗漏进行了订正,并对第 1 版中部分只给出解题思路的习题增补了算法分析。

本书可作为计算机专业本、专科学生的学习参考书,也是报考计算机专业硕士研究生的必读参考书,同时也可供自学考试的读者和计算机等级(三级或四级)考试者研习。

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:数据结构(C 语言篇)——习题与解析(修订版)

作 者:李春葆

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:北京市耀华印刷有限公司

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:22.625 字数:550 千字

版 次:2002 年 4 月第 1 版 2002 年 4 月第 1 次印刷

印 数:0001~8000

书 号:ISBN 7-302-05411-8/TP·3187

定 价:29.00 元

前 言

计算机编程中加工处理的对象是数据，而数据具有一定的组织结构，所以学习编写计算机程序仅仅了解计算机语言是不够的，还必须掌握数据组织、存储和运算的一般方法，这便是数据结构课程中所学习和研究的内容，也是我们编写计算机程序的重要基础。由于它对计算机学科起到承前启后的作用，因此本课程列为计算机等相关专业最重要的专业基础课程。

由于数据结构的原理和算法较抽象，而该课程一般在本科低年级开设，对于只具有计算机程序设计基础知识的初学者，理解和掌握其中的原理就更困难了。在解答数据结构的习题时，往往感到无从下手，作者在多年的教学中感受颇深。本人通过长期的实践、收集与整理，编写了本书，其目的是：通过对习题的解答，使学生充分掌握数据结构的原理以及求解数据结构问题的思路与方法，深化对基本概念的理解，提高分析与解决问题的能力。

本书遵循数据结构课程的教学大纲的要求，从内容上分为 13 章：第 1 章是概述，讨论数据结构的基本概念及相关题解；第 2 章是顺序表，讨论基本顺序表即向量、栈和队列的基本内容及相关题解；第 3 章是链表，讨论各种链表的基本内容及相关题解；第 4 章是串，讨论串的基本内容及相关题解；第 5 章是数组和稀疏矩阵，讨论数组和稀疏矩阵的基本内容及相关题解；第 6 章是递归，讨论基本递归设计方法及相关题解；第 7 章是广义表，讨论广义表的基本内容及相关题解；第 8 章是树形结构，讨论树和二叉树的基本内容及相关题解；第 9 章是图，讨论图的基本内容及相关题解；第 10 章是查找，讨论基本查找方法及相关题解；第 11 章是内排序，讨论基本内排序方法及相关题解；第 12 章是文件，讨论基本文件组织结构及相关题解；第 13 章是外排序，讨论基本外排序方法及相关题解。

每章的内容介绍与习题有关，精选了该章所讨论的数据结构的概念、存储方式和基本运算。每章的题解分为基本题和习题解析两部分，前者由选择题和填空题两种题型组成，由于这部分习题是一些基本概念方面的题目，书中只给出答案；习题解析是对每个习题的解答并给出求解思路和解答的完整的过程，这部分内容中包含一些难度较大的习题，也包含一些高校计算机专业招收硕士研究生的数据结构试题，这部分习题前面加有“*”号。本书的大部分程序在 Turbo C 系统中调试通过。

本书习题覆盖面广，既收集了较容易的题目，也收集了难度适中和较高难度的题目。因此，本书不仅可以作为计算机专业本、专科生数据结构课程的学习参考书，也是报考计算机专业硕士研究生的考生必读复习书，同时适合于数据结构课程自学者和计算机等级（三级或四级）考试者研习。

在编写本书时，作者力求从方法上提高解题的能力，例如，递归问题是学生较难理解的知识点，但在计算机专业知识中又是经常遇到的问题。为此，作者专门编写了递归一章，较深入地分析了递归的执行过程，提出了从递归模型到递归设计的步骤。在其他几章中，也采用了类似的解题方法。

本次修订主要对第 1 版中部分程序进行了补充和完善；对其中少量错误和遗漏之处进行了订正，并对第 1 版中部分只给出解题思路的习题增补算法分析。

在本书的编写和修订过程中，得到武汉大学计算机学院和万力先生的大力支持，在此表示衷心感谢。

由于习题较多，解答上可能存在不准确或不完美之处，内容编排上也可能存在不合理的地方，敬请广大读者批评指正。

作 者

2002.3

目 录

第 1 章 概述	1
1.1 基本概念	1
1.1.1 数据结构	1
1.1.2 存储方式	2
1.1.3 算法及评价	3
1.2 基本题	5
1.2.1 单项选择题	5
1.2.2 填空题（将正确的答案填在相应的空中）	7
1.3 习题解析	8
第 2 章 线性表	14
2.1 基本概念和运算	14
2.1.1 向量	14
2.1.2 栈	16
2.1.3 队列	18
2.2 基本题	20
2.2.1 单项选择题	20
2.2.2 填空题（将正确的答案填在相应的空中）	22
2.3 习题解析	23
2.3.1 向量	23
2.3.2 栈	28
2.3.3 队列	35
第 3 章 链表	47
3.1 基本概念和运算	47
3.1.1 单链表	47
3.1.2 双链表	52
3.1.3 链栈和链队	56
3.2 基本题	60
3.2.1 单项选择题	60
3.2.2 填空题（将正确的答案填在相应的空中）	62
3.3 习题解析	64
3.3.1 单链表	64
3.3.2 双链表	88

第 4 章 串	95
4.1 串的存储及其运算	95
4.1.1 顺序存储及其基本运算	95
4.1.2 链接存储及其基本运算	98
4.2 基本题	102
4.2.1 单项选择题	102
4.2.2 填空题（将正确的答案填在相应的空中）	103
4.3 习题解析	103
第 5 章 数组和稀疏矩阵	120
5.1 基本概念和运算	120
5.1.1 多维数组	120
5.1.2 稀疏矩阵	122
5.2 基本题	128
5.2.1 单项选择题	128
5.2.2 填空题（将正确的答案填在相应的空中。其中 $A[i, j]$ 表示下标从 i 到 j ）	130
5.3 习题解析	130
第 6 章 递归	149
6.1 递归设计方法	149
6.1.1 递归模型	149
6.1.2 递归的执行过程	149
6.1.3 递归设计	150
6.1.4 递归到非递归的转换	151
6.2 基本题	153
6.2.1 单项选择题	153
6.2.2 填空题（将正确的答案填在相应的空中）	153
6.3 习题解析	156
第 7 章 广义表	177
7.1 广义表的表示及其运算	177
7.1.1 广义表的表示	177
7.1.2 广义表的基本运算	178
7.2 基本题	182
7.2.1 单项选择题	182
7.2.2 填空题（将正确的答案填在相应的空中）	183
7.3 习题解析	183
第 8 章 树形结构	196
8.1 基本概念和运算	196
8.1.1 树	196

8.1.2	二叉树	198
8.1.3	二叉排序树	203
8.1.4	树和森林	206
8.1.5	哈夫曼树	207
8.2	基本题	208
8.2.1	单项选择题	208
8.2.2	填空题（将正确的答案填在相应的空中）	213
8.3	习题解析	218
第 9 章	图	261
9.1	图的存储及其运算	261
9.1.1	图的基本术语	261
9.1.2	图的存储方式	262
9.1.3	图的基本运算	266
9.2	基本题	274
9.2.1	单项选择题	274
9.2.2	填空题（将正确的答案填在相应的空中）	276
9.3	习题解析	277
第 10 章	查找	287
10.1	基本查找方法	287
10.1.1	顺序查找	287
10.1.2	二分查找	288
10.1.3	分块查找	289
10.1.4	哈希表查找	290
10.1.5	背包问题及其求解函数	292
10.2	基本题	295
10.2.1	单项选择题	295
10.2.2	填空题（将正确的答案填在相应的空中）	297
10.3	习题解析	298
第 11 章	内排序	309
11.1	基本排序方法	309
11.1.1	直接插入排序	309
11.1.2	希尔（Shell）排序	310
11.1.3	起泡排序	311
11.1.4	快速排序	311
11.1.5	直接选择排序	312
11.1.6	堆排序	313
11.1.7	归并排序	314

11.1.8 基数排序	315
11.2 基本题	316
11.2.1 单项选择题	316
11.2.2 填空题（将正确的答案填在相应的空中）	318
11.3 习题解析	319
第 12 章 文件	332
12.1 基本文件组织方式	332
12.1.1 顺序文件	332
12.1.2 索引文件	332
12.1.3 直接存取文件	334
12.1.4 多关键字文件	334
12.1 基本题	335
12.1.1 单项选择题	335
12.1.2 填空题（将正确的答案填在相应的空中）	335
12.3 习题解析	335
第 13 章 外排序	342
13.1 基本归并排序法	342
13.1.1 磁盘文件归并排序	342
13.1.2 磁带文件归并排序	344
13.2 基本题	346
13.2.1 单项选择题	346
13.2.2 填空题（将正确的答案填在相应的空中）	346
13.3 习题解析	347

第 1 章 概 述

自从 1946 年第一台计算机问世以来,计算机技术的发展日新月异。其应用已不再局限于科学计算,而是更多地用于控制、管理及数据处理等非数值计算的处理工作。与此相应,计算机加工处理的对象由纯粹的数值发展到字符、表格和图像等各种具有一定结构的数据,数据结构就是研究数据组织、存储和运算的一般方法的学科。本章讨论数据结构的基本概念及相关题解。

1.1 基 本 概 念

数据是信息的载体,在计算机科学中是指所有能输入到计算机中并由计算机程序处理的符号的总称。

1.1.1 数据结构

数据结构是指同一数据元素类中各数据元素之间存在的关系。数据结构又可以分为下述三个组成部分,它们分别是数据的逻辑结构、数据的存储结构和数据的运算。

数据的逻辑结构是对数据之间关系的描述,所以有时就把数据的逻辑结构简称为数据结构。逻辑结构形式上用二元组

$$B=(K, R)$$

来表示,其中 K 是结点即数据元素的有穷集合,即 K 是由有限个结点所构成的集合, R 是 K 上的关系的有限集合,即 R 是由有限个关系所构成的集合,而每个关系都是从 K 到 K 的关系。设 r 是一个 K 到 K 的关系, $r \in R$,若 $k, k' \in K$,且 $\langle k, k' \rangle \in r$,则称 k' 是 k 的后续, k 是 k' 的前驱,这时 k 和 k' 是相邻的结点(相对 r 而言);如果不存在一个 k' 使 $\langle k, k' \rangle \in r$,则称 k 为 r 的终端结点;如果不存在一个 k' 使 $\langle k', k \rangle \in r$,则称 k 为 r 的开始结点;如果 k 既不是终端结点也不是开始结点,则称 k 是内部结点。

数据的存储结构是数据的逻辑结构在计算机存储器中的实现,逻辑结构是从逻辑关系上观察数据,它与数据的存储无关,即独立于计算机,而存储结构是依赖于计算机的。计算机存储器是由有限多个存储单元组成的,每个存储单元有惟一的地址,各存储单元的地址是连续编码的,每个存储单元 Z 都有惟一的后续单元 $Z' = \text{succ}(Z)$, Z 和 Z' 称为相邻单元。一片相邻的存储单元的整体叫做存储区域,记做 M 。把 B 存储在计算机中,首先必须建立一个从 K 的结点到 M 单元的映象 $S: K \rightarrow M$,即对于每一个 $k \in K$,都有惟一的 $Z \in M$ 使得 $S(k)=Z$, Z 为 K 结点所占存储空间中的起始单元。通常有四种基本的存储映象方法,即顺序方法、链接方法、索引方法和散列方法。



数据的运算是在数据的逻辑结构上定义的操作算法,如检索、插入、删除、更新和排序等。

从逻辑上可以把数据结构分为线性结构和非线性结构。在线性结构中有且仅有一个终端结点和一个开始结点,并且所有结点都最多只有一个前驱和后续,顺序表就是典型的线性结构。非线性结构中可能有多个终端结点和多个开始结点,每个结点可能有多个前驱和多个后续。非线性结构中最重要的是树形结构和图形结构,树形结构中每个结点只有一个前驱和多个后续;图形结构中每个结点有多个前驱和多个后续。

1.1.2 存储方式

1. 线性结构的存储方式

线性结构的数据有顺序、链接、索引和散列等四种存储方式。

顺序方式是把逻辑上相邻的结点存储在物理上相邻的存储单元里,结点之间的关系由存储单元的邻接关系来体现。其优点是占用最少的存储空间,其缺点是由于每个结点只能使用一整块存储区域,因此可能产生较多的碎片现象。另外,在作插入或删除操作时需移动大量元素。

链接存储方式是将结点所占的存储单元分为两部分,一部分存放结点本身的信息,即为数据项。另一部分存放该结点的后续结点所对应的存储单元的地址,即为指针项。其优点是充分利用所有的存储单元,不会出现碎片现象,其缺点是每个结点占用较多的存储空间。

索引方式是用结点的索引号来确定结点存储地址,其优点是检索速度快,其缺点是增加了额外的索引表,会占用较多的存储空间。另外,在增加和删除数据时还要修改索引表,因而会花费较多时间。

散列的方式是根据结点的值确定它的存储地址,其优点是检索、增加和删除结点的操作都很快,其缺点是采用不好的散列函数时可能出现结点存储单元的冲突,为解决冲突需要额外的时间和空间开销。

2. 树形结构的存储方式

在树形结构的数据中,每个结点可能有多个后续结点,因此一般只能采用链接的方式进行存储,链接的方式正好能表达树形结构中的父子和兄弟两种层次关系。由于链接的方式不能表达任意多个儿子结点,所以常常使用较规则的树,如二叉树,限制后续结点的最多个数。而其他几种存储方式难以达到这种要求。

可以在特定规则的情况下,采用顺序结构(如一维数组)来存储树形结构。如图 1.1 的一颗二叉树,用图 1.1 中的数组来存储。

3. 图形结构的存储方式

在图形结构的数据中,每个结点可能有多个前驱结点和多个后续结点,因此一般只能采用链接的方式进行存储,同样,由于链接的方式不能表达任意多个后续结点,因此,将链接方式改进成邻接表,即图形结构中的每个结点对应有一个链表,该链表存储这个结点的所有相邻结点。

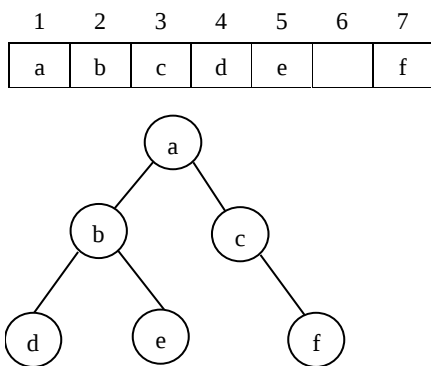


图 1.1 一颗二叉树

另处还可以采用矩阵存储图形结构，即用矩阵表示图形结构中任意两个结点之间的关系，这种矩阵称为邻接矩阵。

1.1.3 算法及评价

算法是解决某一特定类型问题的有限运算序列。描述一个算法可以采用某一种计算机语言，也可以采用流程图等。本书的算法是采用 C 语言描述的。

算法具有 5 个基本特性：有穷性、确定性、可行性、输入和输出。

评价一个算法一般从 4 个方面进行：正确性、运行时间、占用空间和简单性。其中最主要的是算法的运行时间和占用空间。

正确性是指算法是否正确。运行时间是指一个算法在计算机上运行所花费的时间，采用时间复杂度来度量，所谓时间复杂度是指算法中包含简单操作的次数，一般不必要精确计算出算法的时间复杂度，只要大致计算出相应的数量级，如 $O(1)$ 、 $O(\log_2 n)$ 、 $O(n)$ 或 $O(n^2)$ 等。 O 的形式定义为：若 $f(n)$ 是正整数 n 的一个函数，则 $x_n = O(f(n))$ 表示存在一个正的常数 M ，使得当 $n \geq n_0$ 时满足 $|x_n| \leq M|f(n)|$ 。一般地，常用的时间复杂度有如下关系：

$$O(1) \leq O(\log_2 n) \leq O(n) \leq O(n \log_2 n) \leq O(n^2) \leq O(n^3) \leq \dots \leq O(n_k) \leq O(2^n)$$

占用空间指在计算机存储器上所占用的存储空间，主要考虑在算法运行过程中临时占用的存储空间的大小，称之为空间复杂度，一般以数量级形式给出。简单性是指算法的易读性等。

【例 1】 分析以下程序段的时间复杂度。

```

for (i=1; i<n; i++)
{
    y=y+1;
    for (j=0; j<=(2*n); j++)
        x++;
}
  
```

语句的频度指的是该语句重复执行的次数。一个算法中所有语句的频度之和构成了该算法的运行时间。在本例算法中，其中语句 `y=y+1;` 的频度是 $n-1$ ，语句 `for (j=0; j<=(2*n); j++)` 的频度是



$(n-1)(2n+1)=2n^2 - n - 1$ 。则该程序段的时间复杂度 $T(n)=n-1+2n^2-n-1=O(n^2)$ 。

【例 2】 分析以下程序段的时间复杂度。

```
i=1;
while (i<=n)
    i=i*2;
```

其中语句 的频度是 1, 设语句 的频度 $f(n)$, 则有:

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

即 $f(n) \leq \log_2 n$, 取最大值 $f(n)=\log_2 n$ 。

则该程序段的时间复杂度 $T(n)=1 + f(n)=1 + \log_2 n=O(\log_2 n)$ 。

【例 3】 分析以下程序段的时间复杂度。

```
a=0; b=1;
for (i=2; i<=n; i++)
{
    s=a+b;
    b=a;
    a=s;
}
```

其中语句 的频度是 2 ;语句 的频度是 n ;语句 的频度是 $n-1$;语句 的频度是 $n-1$;
语句 的频度是 $n-1$ 。则该程序段的时间复杂度 $T(n)=2 + n + 3*(n-1)=4n-1=O(n)$ 。

【例 4】 有以下程序, 分析其中 `order()` 函数的时间复杂度。

```
int a[]={2, 5, 1, 7, 9, 3, 6, 8};
order(int j, int m)
{
    int i, temp;
    if (j<m)
    {
        for (i=j; i<=m; i++)
            if (a[i]<a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        j++;
        order(j, m);
    }
}
main()
{
```



```
int i;
order(0, 7);
for (i=0; i<=7; i++)
    printf("%d ", a[i]);
}
```

order()函数是一个递归排序过程, 设 $T(n)$ (这里 $n=m+1$) 是排序 n 个元素所需要的时间。在排序 n 个元素时, 算法的计算时间主要花费在递归调用 order() 上。第一次调用时, 处理的元素序列个数为 $n-1$, 也就是对余下的 $n-1$ 个元素进行排序, 这部分所需要的计算时间应为 $T(n-1)$ 。

又因为在其中的循环中, 需要 $n-1$ 次比较, 所以排序 n 个元素所需要的时间为:

$$T(n)=T(n-1)+n-1 \quad n>1$$

这样得到如下方程:

$$T(n)=\begin{cases} 0 & n=1 \\ T(n-1)+n-1 & n>1 \end{cases}$$

求解过程为:

$$\begin{aligned} T(n) &= [T(n-2)+(n-2)]+(n-1) \\ &= [T(n-3)+(n-3)]+(n-2)+(n-1) \\ &= \dots \\ &= (T(1)+1)+2+\dots+n-1 \\ &= 0+1+2+\dots+n-1 \\ &= n(n-1)/2 \\ &= O(n^2) \end{aligned}$$

故 order()函数的时间复杂度为 $O(n^2)$ 。

1.2 基 本 题

1.2.1 单项选择题

1. 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_____以及它们之间的_____和运算等的学科。

A. 数据元素 B. 计算方法 C. 逻辑存储 D. 数据映像

A. 结构 B. 关系 C. 运算 D. 算法

答: A B

2. 数据结构被形式地定义为 (K, R) , 其中 K 是_____的有限集, R 是 K 上的_____有限集。



- A. 算法 B. 数据元素 C. 数据操作 D. 逻辑结构
A. 操作 B. 映象 C. 存储 D. 关系

答: B D

3. 在数据结构中,从逻辑上可以把数据结构分成_____。

- A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
C. 线性结构和非线性结构 D. 内部结构和外部结构

答: C

4. 线性结构的顺序存储结构是一种_____的存储结构,线性表的链式存储结构是一种_____的存储结构。

- A. 随机存取 B. 顺序存取
C. 索引存取 D. 散列存取

答: A B

5. 算法分析的目的是_____,算法分析的两个主要方面是_____。

- 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
A. 空间复杂度和时间复杂度 B. 正确性和简单性
C. 可读性和文档性 D. 数据复杂性和程序复杂性

答: C A+B

6. 计算机算法指的是_____,它必须具备输入、输出和_____等五个特性。

- 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

A. 可执行性、可移植性和可扩充性。

B. 可行性、确定性和有穷性

C. 确定性、有穷性和稳定性

D. 易读性、稳定性和安全性

答: C B

7. 线性表的逻辑顺序与存储顺序总是一致的,这种说法_____。

- A. 正确 B. 不正确

答: B

8. 线性表若采用链式存储结构时,要求内存中可用存储单元的地址_____。

- A. 必须是连续的 B. 部分地址必须是连续的
C. 一定是不连续的 D. 连续不连续都可以

答: D

9. 在以下的叙述中,正确的是_____。

- A. 线性表的线性存储结构优于链表存储结构
B. 二维数组是其数据元素为线性表的线性表
C. 栈的操作方式是先进先出
D. 队列的操作方式是先进后出

答: B



10. 每种数据结构都具备三个基本运算：插入、删除和查找，这种说法_____。

A. 正确

B. 不正确

答：B

1.2.2 填空题（将正确的答案填在相应的空中）

1. 数据逻辑结构包括_____、_____和_____三种类型，树形结构和图形结构合称为_____。

答： 线性结构 树形结构 图形结构 非线性结构

2. 在线性结构中，第一个结点_____前驱结点，其余每个结点有且只有_____个前驱结点；最后一个结点_____后续结点，其余每个结点有且只有_____个后续结点。

答： 没有 1 没有 1

3. 在树形结构中，树根结点没有_____结点，其余每个结点有且只有_____个前驱结点；叶子结点没有_____结点，其余每个结点的后续结点可以_____。

答： 前驱 1 后续 任意多个

4. 在图形结构中，每个结点的前驱结点数和后续结点数可以_____。

答：任意多个

5. 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。

答： 一对一 一对多 多对多

6. 算法的五个重要特性是_____、_____、_____、_____、_____。

答： 有穷性 确定性 可行性 输入 输出

7. 下面程序段的时间复杂度是_____。

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

答： $O(m*n)$

8. 下面程序段的时间复杂度是_____。

```
i=s=0;
while (s<n)
{
    i++; /*i=i+1*/
    s+=i; /*s=s+i*/
}
```

答： $O(n)$

9. 下面程序段的时间复杂度是_____。

```
s=0;
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
```



```
s+=B[i][j];  
sum=s;
```

答： $O(n^2)$

10. 下面程序段的时间复杂度是_____。

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

答： $\log_3 n$

1.3 习题解析

1. 设有数据逻辑结构为：

$B=(K, R)$

$K=\{k_1, k_2, \dots, k_9\}$

$R=\{<k_1, k_3>, <k_1, k_8>, <k_2, k_3>, <k_2, k_4>, <k_2, k_5>, <k_3, k_9>, <k_5, k_6>, <k_8, k_9>, <k_9, k_7>, <k_4, k_7>, <k_4, k_6>\}$

画出这个逻辑结构的图示，并确定相对于关系 R ，哪些结点是开始结点，哪些结点是终端结点？

解：该题的逻辑结构图示如图 1.2 所示。

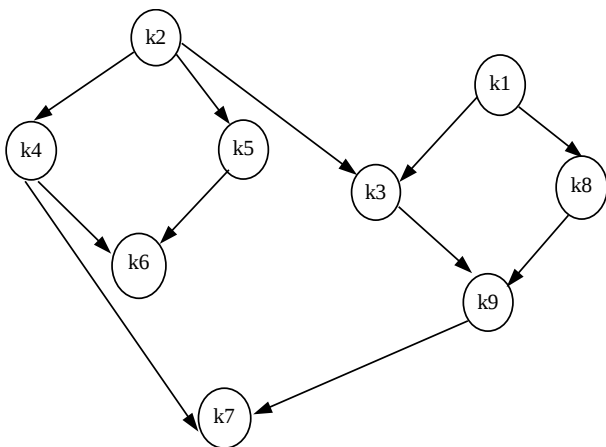


图 1.2 一个逻辑结构图示

开始结点是指无前驱的结点，这里满足该定义的开始结点为 k_1, k_2 。终端结点是指无后续结点，这里满足该定义的终端结点为 k_6, k_7 。该逻辑结构是非线性结构中的图形结构。



2. 设有如图 1.3 所示的逻辑图, 给出它的逻辑结构。

解: 本题的逻辑结构如下:

$B=(K, R)$

$K=\{k_1, k_2, \dots, k_9\}$

$R=\{<k_1, k_2>, <k_1, k_3>, <k_3, k_4>, <k_3, k_6>, <k_6, k_8>, <k_4, k_5>, <k_6, k_7>, <k_8, k_9>\}$

该逻辑结构是一个树形结构, 其树根为 k_1 , 叶子结点为 k_2 、 k_5 、 k_7 和 k_9 。

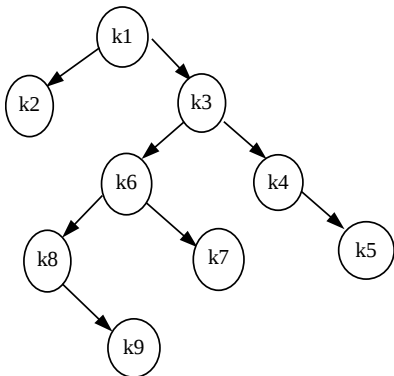


图 1.3 一个逻辑结构图示

3. 有下列几种用二元组表示的数据结构, 画出它们分别对应的逻辑图形表示, 并指出它们分别属于何种结构。

(1) $A=(K, R)$, 其中:

$K=\{a, b, c, d, e, f, g, h\}$

$R=\{r\}$

$r=\{<a, b>, <b, c>, <c, d>, <d, e>, <e, f>, <f, g>, <g, h>\}$

(2) $B=(K, R)$, 其中:

$K=\{a, b, c, d, e, f, g, h\}$

$R=\{r\}$

$r=\{<d, b>, <d, g>, <d, a>, <b, c>, <g, e>, <g, h>, <e, f>\}$

(3) $C=(K, R)$, 其中:

$K=\{1, 2, 3, 4, 5, 6\}$

$R=\{r\}$

$r=\{(1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6)\}$

这里的圆括号对表示两结点是双向的。

(4) $D=(K, R)$, 其中:

$K=\{48, 25, 64, 57, 82, 36, 75\}$

$R=\{r_1, r_2\}$

$r_1=\{<25, 36>, <36, 48>, <48, 57>, <57, 64>, <64, 75>, <75, 82>\}$

$r_2=\{<48, 25>, <48, 64>, <64, 57>, <64, 82>, <25, 36>, <82, 75>\}$