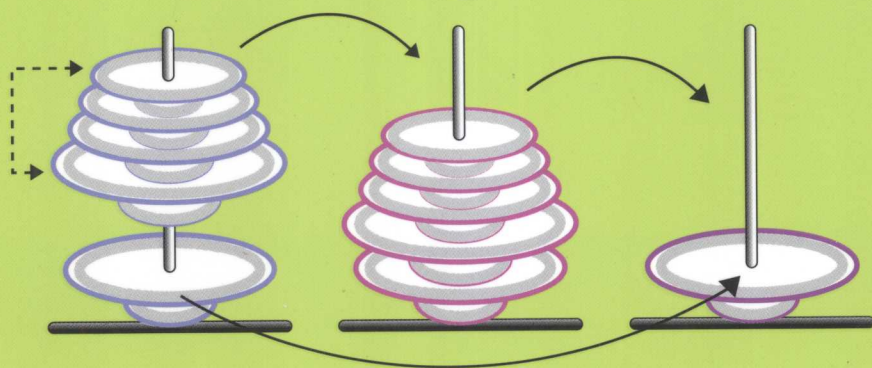


图解 数据结构

第 二 版

胡昭民 著



轻松的图解讲解复杂的
数据结构原理与算法精髓

丰富的范例
演绎数据结构内涵

使用C语言实现经典的算法，
大量的上机操练，实践可用的源代码



提供网络资源下载

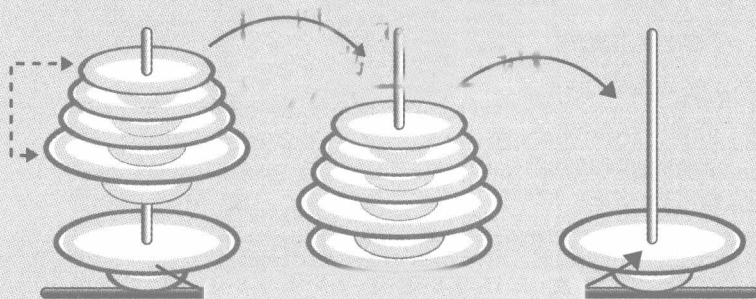


清华大学出版社

图解 数据结构

第二版

胡昭民 著



清华大学出版社
北京

北京市版权局著作权合同登记号：图字 01-2016-4043
本书为荣钦科技股份有限公司授权出版发行的中文简体字版本。

内 容 简 介

本书用最轻松的图解方式来讲解数据结构，全书采用丰富的图例阐述数据结构的基本概念及应用，并将重要理论、演算方法做最详细的诠释与举例，是一本兼具内容及专业的数据结构的教學用书。

由于作者长期从事信息教育及写作，在文字的表达上简洁明了、逻辑清晰，并安排了大量的习题，供读者检验学习成果。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

图解数据结构/胡昭民著. —2 版. —北京：清华大学出版社，2016
ISBN 978-7-302-43935-6

I. ①图… II. ①胡… III. ①数据结构—图解②C 语言—程序设计 IV. ①TP311.12-64②TP312

中国版本图书馆 CIP 数据核字（2016）第 111122 号

责任编辑：夏毓彦
封面设计：王 翔
责任校对：闫秀华
责任印制：李红英

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-62770175 邮 购：010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：北京密云胶印厂

经 销：全国新华书店

开 本：190mm×260mm	印 张：22.75	字 数：582 千字
版 次：2015 年 8 月第 1 版	2016 年 8 月第 2 版	印 次：2016 年 8 月第 1 次印刷
印 数：1~3000		
定 价：49.00 元		

产品编号：068737-01

序

数据结构一直是计算机信息相关学科的必修科目，对于第一次接触数据结构课程的学生来说，内容过多及表达不清楚常常是造成学习障碍最主要的因素，为了让读者能用最轻松的方式快速理解数据结构，本书特别征询多位教师意见，采用丰富的图例来阐述基本概念及应用，并且将重要理论和演算方法做了非常详实的诠释，是一本兼具内容及专业的数据结构教学用书。

由于笔者长期从事咨询教育及写作工作，在文句的表达上尽量以简洁有力、逻辑清楚阐述为主。为了验收各章的学习成果，特别搜集了大量的习题，并参阅重要考试，提供给读者更多的实战演练经验。

另外，为了减轻读者的学习压力及经济负担，在整本书所涉及的主题分量不减的情况下，书中仅收录精华的演算方法及程式范例的执行画面，本书所有的程序代码在提供的下载地址中可以下载，读者可直接下载并运行验证。最后希望本书能带给各位对数据结构更完整的认识。

胡昭民 敬笔

改编说明

数据结构毫无疑问是计算机科学既经典又核心的课程之一，不管是从事计算机软件还是硬件的开发工作，如果没有系统地学习数据结构或者没有专心自学过，很容易被人打上“非专业”的标签。对于任何在信息技术行业工作的专业人员或者想进入此行业的人来说，什么时候开始学数据结构都不会晚，更不会过时。

从“数据结构”的名字看，它不仅仅只是讲授数据的结构以及在计算机内如何存储和组织数据的方式，这些只是它的表面现象。数据结构背后真正蕴含的是与之息息相关的算法，精心选择的数据结构配合恰如其分的算法就意味着数据或者信息在计算机内被高效率地存储和高效率地处理。算法其实就是数据结构的灵魂，它既神秘又神奇“好玩”，当然对初学者也比较难，算法可以说是“聪明人在计算机上的游戏”。

本书是一本综合而且全面讲述数据结构及其算法分析的教科书，为了便于高校的教学或者读者自学，作者在描述数据结构原理和算法时文字清晰并且严谨，为每个算法及其数据结构提供了演算的详细图解。另外，为了适合在教学中让学生上机实践或者自学者上机“操练”，本书为每个经典的算法都提供了 C 语言编写的完整范例程序的源代码，每个范例程序都不需要经过修改，直接通过编译就可以运行，目的就是让本书的学习者以这些范例程序作为参照迅速掌握数据结构和算法的要点。

全书的所有范例程序都可以在标准的 C 语言编程环境中编译通过并且成功运行，我们在改编本书的过程中选用了免费的 Dev C++ 5.11 集成开发环境，对原书的所有范例程序进行编译、修改、调试和测试，并确保它们都可以准确无误地运行。附录 A 包含了“C/C++编译程序的介绍与安装”，其中重点就介绍了 Dev C++。附录 B 则包含了“C 语言快速入门”。

本书范例程序源代码下载地址为：<http://pan.baidu.com/s/1eStzSC6>。

如果下载有问题，请电子邮件联系 booksaga@126.com，邮件主题为“求图解数据结构的范例程序代码”。

赵军

2016 年 1 月

目 录

第 1 章 数据结构导论	1
1-1 数据结构简介	2
1-1-1 数据与信息	2
1-1-2 算法	3
1-1-3 算法的条件	3
1-1-4 数据结构的应用	6
1-2 数据抽象化	7
1-2-1 基本数据类型	7
1-2-2 抽象数据类型	7
1-3 算法与程序设计	8
1-3-1 认识程序设计	8
1-3-2 程序开发流程	9
1-3-3 程序设计的风格	9
1-4 面向对象程序设计	11
1-4-1 封装 (Encapsulation)	12
1-4-2 继承 (Inheritance)	13
1-4-3 多态 (Polymorphism)	13
1-5 模块化设计与 C 语言	13
1-5-1 函数的基本概念	13
1-5-2 参数类型的介绍	14
1-5-3 参数的传递方式	15
1-6 递归算法	15
1-6-1 递归的定义	15
1-6-2 斐波拉契数列	17
1-6-3 汉诺塔问题	18
1-7 程序效率的分析	23
1-7-1 Big-oh	25
1-7-2 $\Omega(\omega)$	26
1-7-3 $\theta(\theta)$	27
本章习题	27
第 2 章 线性表	32
2-1 线性表的定义	33



2-1-1 线性表的用途	33
2-2 数组	34
2-2-1 一维数组	34
2-2-2 二维数组	36
2-2-3 多维数组	40
2-2-4 结构数组	44
2-2-5 字符数组	46
2-2-6 字符串数组	48
2-2-7 指针数组	49
2-3 矩阵	50
2-3-1 矩阵的运算	51
2-3-2 稀疏矩阵	53
2-3-3 上三角形矩阵	55
2-3-4 下三角形矩阵	59
2-3-5 带状矩阵	64
本章习题	65
第3章 链表	69
3-1 动态分配内存	70
3-1-1 C 的动态分配变量	70
3-1-2 C++ 的动态分配变量	72
3-2 单向链表	73
3-2-1 建立单向链表	74
3-2-2 遍历单向链表	75
3-2-3 释放单向链表节点的空间	76
3-2-4 单向链表插入新节点	77
3-2-5 单向链表删除节点	79
3-2-6 单向链表的反转	81
3-3 环形链表	83
3-3-1 环形链表的建立与遍历	83
3-3-2 环形链表中插入新节点	85
3-3-3 环形链表节点的删除	86
3-3-4 环形链表的连接功能	88
3-4 双向链表	89
3-4-1 双向链表的建立与遍历	90
3-4-2 双向链表中加入新节点	92
3-4-3 双向链表节点的删除	94
3-5 链表相关应用简介	96
3-5-1 多项式表式法	96
3-5-2 稀疏矩阵表示法	100

本章习题	102
第 4 章 堆栈与队列	109
4-1 堆栈简介	110
4-1-1 堆栈的基本操作	111
4-1-2 用数组实现堆栈	111
4-1-3 用链表实现堆栈	112
4-1-4 堆栈类样板的实现	114
4-1-5 老鼠走迷宫	116
4-1-6 八皇后问题	119
4-2 算术表达式的表示法	120
4-2-1 中序转为前序与后序	121
4-2-2 前序与后序转为中序	126
4-2-3 中序表示法求值	129
4-2-4 前序法的求值运算	130
4-2-5 后序法的求值运算	131
4-3 队列	132
4-3-1 队列的基本操作	133
4-3-2 用数组实现队列	133
4-3-3 环形队列	135
4-3-4 双向队列	139
4-3-5 双向队列	141
4-3-6 优先队列	143
本章习题	144
第 5 章 树状结构	156
5-1 树的基本概念	157
5-1-1 专有名词介绍	158
5-2 二叉树	159
5-2-1 二叉树的特性	159
5-2-2 特殊二叉树简介	160
5-3 二叉树的存储方式	161
5-3-1 一维数组表示法	161
5-3-2 链表表示法	164
5-4 二叉树的遍历	166
5-4-1 中序遍历	166
5-4-2 后序遍历	167
5-4-3 前序遍历	167
5-4-4 二叉树节点的插入与删除	170
5-4-5 二叉运算树	174

5-5	线索二叉树.....	176
5-5-1	二叉树转为线索二叉树.....	176
5-6	树的二叉树表示法.....	180
5-6-1	树转化为二叉树.....	180
5-6-2	二叉树转换成树.....	182
5-6-3	森林化为二叉树.....	183
5-6-4	二叉树转换成森林.....	184
5-6-5	树与森林的遍历.....	185
5-6-6	确定唯一二叉树.....	189
5-7	优化二叉查找树.....	191
5-7-1	扩充二叉树.....	191
5-7-2	霍夫曼树.....	192
5-8	平衡树.....	194
5-8-1	平衡树的定义.....	194
5-9	高级树状结构的研究.....	196
5-9-1	决策树.....	196
5-9-2	B 树.....	198
5-9-3	二叉空间分割树.....	198
5-9-4	四叉树与八叉树.....	199
	本章习题.....	200
第 6 章	图形结构.....	210
6-1	图形简介.....	211
6-1-1	图的定义.....	212
6-1-2	无向图.....	212
6-1-3	有向图.....	214
6-2	图的数据表示法.....	215
6-2-1	邻接矩阵法.....	215
6-2-2	邻接表法.....	218
6-2-3	邻接复合链表法.....	220
6-2-4	索引表格法.....	222
6-3	图的遍历.....	225
6-3-1	深度优先遍历法.....	225
6-3-2	广度优先遍历法.....	227
6-4	生成树.....	229
6-4-1	DFS 生成树和 BFS 生成树.....	229
6-4-2	最小生成树.....	231
6-4-3	Kruskal 算法.....	231
6-4-4	Prim 算法.....	235
6-5	图的最短路径.....	236



6-5-1 单点对全部顶点	237
6-5-2 两两顶点间的最短路径	240
6-6 AOV 网络与拓扑排序	244
6-6-1 拓扑排列简介	244
6-7 AOE 网络	246
6-7-1 关键路径	246
本章习题	248
第 7 章 排序	257
7-1 排序简介	258
7-1-1 排序的分类	259
7-2 内部排序法	260
7-2-1 冒泡排序法	260
7-2-2 选择排序法	262
7-2-3 插入排序法	264
7-2-4 希尔排序法	266
7-2-5 合并排序法	268
7-2-6 快速排序法	269
7-2-7 堆积排序法	271
7-2-8 基数排序法	278
7-3 外部排序法	280
7-3-1 直接合并排序法	280
7-3-2 k 路合并法	284
7-3-3 多相合并法	284
本章习题	285
第 8 章 查找	295
8-1 常见的查找方法	296
8-1-1 顺序查找法	296
8-1-2 二分查找法	297
8-1-3 插值查找法	299
8-1-4 斐波那契查找法	301
8-2 哈希查找法	305
8-2-1 哈希法简介	305
8-3 常见的哈希函数	306
8-3-1 除留余数法	306
8-3-2 平方取中法	307
8-3-3 折叠法	308
8-3-4 数字分析法	308
8-4 碰撞与溢出问题的处理	309



8-4-1 线性探测法	309
8-4-2 平方探测	310
8-4-3 再哈希	310
8-4-4 链表	311
本章习题	312
附录 A C/C++编译程序的介绍与安装	318
A-1 C/C++编译程序简介	319
A-1-1 Visual C++ 2010 Express	319
A-1-2 C++ Builder	320
A-1-3 Visual C++	320
A-1-4 Dev C++	321
A-1-5 GCC	322
A-2 Dev C++的安装与介绍	322
A-2-1 下载 Dev-C++	323
A-2-2 安装 Dev C++	323
附录 B C语言快速入门介绍与安装	329
B-1 轻松学C程序	330
B-1-1 编译与执行	331
B-1-2 编译程序	332
B-1-3 开始执行程序	333
B-2 C的基本数据处理	333
B-2-1 变量	333
B-2-2 常数	334
B-2-3 数据类型简介	334
B-3 C语言的输出与输入	335
B-3-1 printf()函数	336
B-3-2 scanf()函数	337
B-4 流程控制	338
B-4-1 顺序结构	338
B-4-2 选择结构	339
B-4-3 重复结构	343
B-5 数组简介	346
B-5-1 字符串简介	347
B-5-2 字符串数组	347
B-6 函数介绍	349
B-6-1 传递参数的方式	350
B-6-2 标准函数库	352

第 1 章

数据结构导论

- 1-1 数据结构简介
- 1-2 数据抽象化
- 1-3 算法与程序设计
- 1-4 面向对象程序设计
- 1-5 模块化设计与 C 语言
- 1-6 递归算法
- 1-7 程序效率的分析

计算机 (computer) 或者称为电脑, 是一种具备数据处理与计算功能的智能电子设备。它可以接受人类的指令或程序设计语言, 经过运算处理后, 输出所期待的结果。

对于一个有志于从事信息技术 (IT) 专业领域的人员来说, 数据结构 (data structure) 是一门和计算机硬件与软件都息息相关的学科, 数据结构这门学科中包含了算法 (algorithm)、数据存储结构、排序、搜索、程序设计概念与哈希函数。

1-1 数据结构简介

大家可以把数据结构看成是在数据处理过程中, 一种分析、组织数据的方法与逻辑, 它考虑到了数据之间的特性与相互关系。在现代的社会中, 计算机与信息是密切相关的, 计算机具有处理速度快和存储容量大的两大特点, 如图 1-1 所示, 所以在数据处理的角色上更为举足轻重。

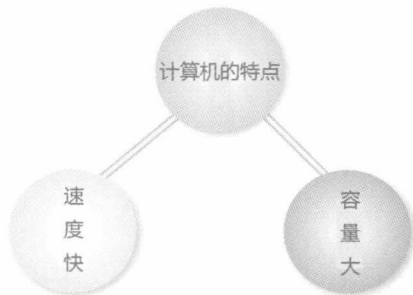


图 1-1 计算机的两大特点

数据结构就是数据进入计算机处理的一套完整逻辑, 程序设计人员必须选择一种数据结构来进行数据的添加、修改、删除、存储等操作, 如果在选择数据结构时做了错误的决定, 那么程序执行起来的速度将可能变得非常低效。如果选错了数据类型, 后果则更加不堪设想。

因此, 当我们要求计算机为我们解决问题时, 必须以计算机所能接受的模式来确认问题, 并且要选用适当的算法来处理数据, 这就是数据结构讨论的重点。简单地说, 数据结构就是数据与算法的研究。

1-1-1 数据与信息

谈到数据结构, 首先就必须了解什么是数据 (data) 与信息 (information)。从字义上来看, 所谓数据 (data), 指的就是一种未经处理的原始文字 (word)、数字 (number)、符号 (symbol) 或图形 (graph) 等, 它所表达出来的只是一种没有评估价值的基本元素或项目。例如姓名或我们常看到的课程表、通讯录等等都可泛称为一种“数据” (data)。

当数据经过处理 (process), 例如以特定的方式系统地整理、归纳甚至进行分析后, 就成为“信息” (information)。这样处理的过程就称为“数据处理” (data processing)。如图 1-2 所示。

从严谨的角度来形容“数据处理”, 就是用人力或机器设备, 对数据进行系统的整理, 如记录、排序、合并、整合、计算、统计等, 以使原始的数据符合需求, 成为有用的信息。



图 1-2 数据处理过程的示意图

不过大家可能会有疑问：“那么数据和信息的角色是否绝对一成不变呢？”。这倒也不一定，同一份文件可能在某种情况下为数据，而在另一种情况下则为信息。例如美伊战争的某场战役死伤人数报告，对平民百姓而言，当然只是一份不痛不痒的“数据”，不过对于英美联军指挥官而言，这份报告则是弥足珍贵的“信息”。

1-1-2 算法

数据结构与算法是程序设计实践中最基本的内涵。程序能否快速而高效地完成预定的任务，取决于是否选对了数据结构，而程序是否能清楚而正确地把问题解决，则取决于算法。所以大家可以这么认为：“数据结构加上算法等于可执行的程序。”

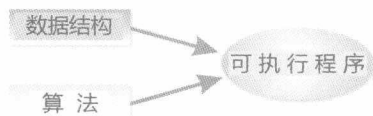


图 1-3 数据结构加上算法等于可执行的程序

在韦氏辞典中算法定义为：“在有限步骤内解决数学问题的程序。”如果运用在计算机领域中，我们也可以把算法定义成：“为了解决某项工作或某个问题，所需要有限数量的机械性或重复性指令与计算步骤。”其实日常生活中有许多工作都可以用算法来描述，例如员工的工作报告、宠物的饲养过程、学生的课程表等。

1-1-3 算法的条件

认识了算法的定义之后，下面再来说明一下算法所必须符合的 5 个条件，如图 1-4 和表 1-1 所示。

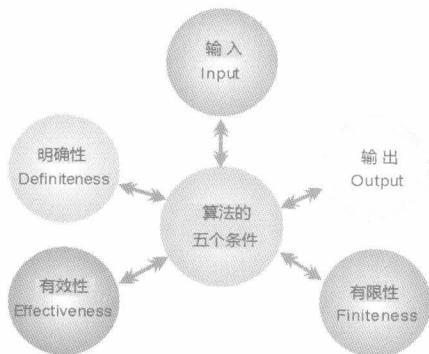


图 1-4 算法的 5 个条件

表 1-1 算法必须符合的 5 个条件

算法的条件(特性)	内容与说明
输入 (Input)	0 个或多个输入数据，这些输入必须有清楚 的描述或定义
输出 (Output)	至少会有一个输出结果，不可以没有输出结果
明确性 (Definiteness)	每一个指令或步骤必须是简洁明确的
有限性 (Finiteness)	在有限步骤后一定会结束，不会产生无限循环
有效性 (Effectiveness)	步骤清楚且可行，能让用户用纸笔计算而求出答案

大家认识了算法的定义与条件后，接着要来思考：该用什么方法来表达算法最为适当呢？其实算法的主要目的在于让人们了解所执行的工作流程与步骤，只要能清楚地体现算法的 5 个条件即可，常用的算法如下。

- 一般文字叙述：中文、英文、数字等。文字叙述法的特色在于使用文字或语言叙述来说明算法的演算步骤。例如，下面就是一个学生小华早上去上学并且买早餐的简单文字算法，如图 1-5 所示。



图 1-5 文字叙述的算法例子

- 伪语言 (pseudo-language)：接近高级程序设计语言的写法，也是一种不能直接放进计算机中执行的语言。一般都需要一种特定的预处理器 (preprocessor)，或者要用人工编写转换成真正的计算机语言，经常使用的有 SPARKS、PASCAL-LIKE 等语言。以下是用 SPARKS 写成的链表反转的算法：

```
Procedure Invert(x)
  p←x; q←Nil;
  WHILE p≠NIL do
    r←q; ←p;
    p←LINK(p);
    LINK(q)←r;
```

```
END
x←q;
END
```

- 表格或图形：如数组、树形图、矩阵图等，以下是井字游戏的某个决策区域，下一步是 X 方下棋，很明显的，X 方绝对不能选择第二层的第二种下法，因为这样下的话 X 方必败无疑，下面使用博弈树来表示其算法，如图 1-6 所示。

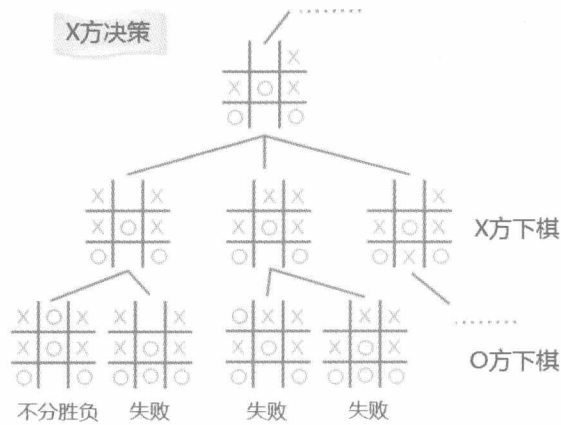


图 1-6 图形描述算法的例子

- 流程图：流程图（Flow Diagram）是一种通用的图形符号表示法，例如请用户输入一个数值，然后判断这个数值是奇数还是偶数，如图 1-7 所示。

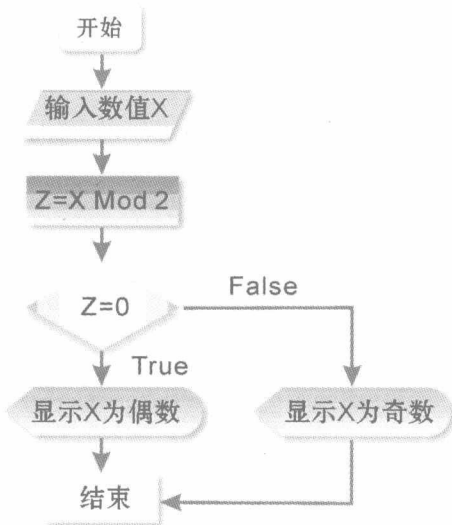


图 1-7 流程图描述算法的示意图

- 程序设计语言：目前算法也能够直接以可读性高级程序设计语言来表示，例如 Visual Basic 语言、C 语言、C++语言、Java 语言。



提示

算法和过程 (procedure) 有何不同? 与流程图又有什么关系?

算法和过程是有区别的, 因为过程不一定要满足有限性的要求, 如操作系统或机器上运行的过程。除非宕机, 否则永远在等待循环中 (waiting loop), 这就违反了算法 5 大条件中的“有限性”。

另外, 只要是算法都能够使用过程流程图来表示, 但反过来, 过程无法用算法来描述, 因为过程流程图可以包含无限循环。

1-1-4 数据结构的应用

在现实生活中, 计算机的主要工作就是把所称的数据 (data), 通过某种运算处理过程, 转换为实用的信息 (information)。数据是一种如文字、数字、图形等基本符号, 例如一个学生的语文成绩是 90 分, 可以说这是一科成绩的数据, 但无法判断它具备什么意义。

如果经过某些如排序 (sorting) 的处理, 就可以知道这位学生的语文成绩在班上同学中的名次, 也就是清楚这个班学生的成绩如何, 这就是一种信息, 而排序就是数据结构的一种应用。以下将介绍一些数据结构的常见应用:

■ 树状结构

树状结构是一种相当重要的非线性数据结构, 广泛运用于如人类社会的族谱或是公司、机关的组织结构、计算机上的 MS-DOS 和 Unix 操作系统、平面绘图应用、游戏设计等。

例如在年轻人喜爱的大型网络游戏中, 需要获取某些物体所在的地形信息, 如图 1-8 所示, 如果程序是依次从构成地形的模型三角面去寻找, 往往会耗费计算机很多运行时间, 非常低效。因此, 程序设计人员一般就会使用树状结构中的二叉树 (BSP tree)、四叉树 (Quadtree)、八叉树 (Octree) 等来分割场景数据。

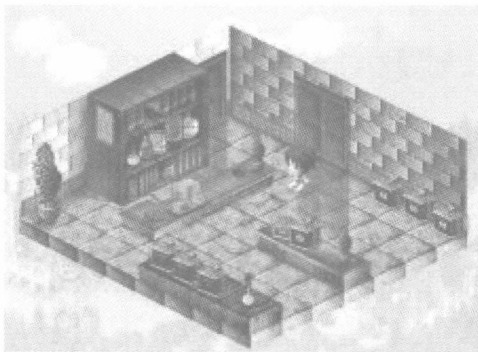


图 1-8 网络游戏场景

■ 最短路径

最短路径的作用是在众多不同的路径中, 寻找距离最短或者所花费成本最少的路径。最传统的应用是在公共交通运输或网络架设上的最短路径问题, 如都市运输系统、铁道运输系统、通信网络系统等。