

<<算法精解>>

图书基本信息

书名：<<算法精解>>

13位ISBN编号：9787111394266

10位ISBN编号：7111394267

出版时间：2012-8

出版时间：机械工业出版社

作者：Kyle Loudon

页数：401

译者：肖翔,陈舸

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

本书封面上的动物是海马，属于海龙科。

海马这个词来源于希腊语中的“弯曲的马”。

海马那不同寻常的身体由大约50块左右包围着身体的骨板构成，宛如一圈盔甲的形状。

海马依靠它狭窄的鼻口作为进食的管道，主要吸食浮游生物和小鱼的幼虫。

公海马的肚子上有一个袋子，母海马每次将100枚或更多的海马蛋放在公海马的袋子里。

公海马使袋子内的海马蛋受精，并一直照料这些蛋直到小海马孵化出来。

根据海马的种类，这个过程大约需要10天到6个星期。

尽管也有一些种类的海马居住在海洋中，但是海马通常都出现在热带和亚热带的浅海水域。

所有海马都使用骨盆和胸鳍来完成转向的动作。

它们采用直立的姿势游动，但速度很慢且常常停下来休息。

在休息的时候，它们用自己的尾巴缠绕住海藻或珊瑚使自己停住。

除了能提供一个休息的地方外，海藻和珊瑚还能为海马提供良好的伪装效果。

世界上体型最大的海马是太平洋海马，大约有12英寸长。

最小的海马是矮海马，大约只有1.5英寸长。

## &lt;&lt;算法精解&gt;&gt;

## 内容概要

本书是数据结构和算法领域的经典之作，十余年来，畅销不衰！

全书共分为三部分：第一部分首先介绍了数据结构和算法的概念，以及使用它们的原因和意义，然后讲解了数据结构和算法中最常用的技术——指针和递归，最后还介绍了算法的分析方法，旨在为读者学习这本书打下坚实的基础；第二部分对链表、栈、队列、集合、哈希表、堆、图等常用数据结构进行了深入阐述；第三部分对排序、搜索数值计算、数据压缩、数据加密、图算法、几何算法等经典算法进行了精辟的分析和讲解。

本书的众多特色使得它在同类书中独树一帜：具体实现都采用正式的C语言代码而不是伪代码，在很多数据结构和算法的实现过程中，有大量细节问题是伪代码不能解决的；每一章都有精心组织的主题和应用；全部示例来自真实的应用，不只是一般的练习；对每种数据结构、算法和示例都进行了详细分析；每一章的末尾都会有一系列问题和对应的回答，旨在强调这一章的重要思想……

本书中的代码尤为值得强调：所有实现都采用C语言编写，所有代码都优先用于教学目的，所有代码都在4种平台上经过完整测试，头文件记录了所有公共的接口，命名规则适用于全书所有的代码，所有的代码都包含大量注释……

本书内容包括：

- 数据结构和算法的概念，以及使用它们的原因和意义
- 指针和递归
- 算法分析
- 常用数据结构：链表、栈、队列、集合、哈希表、树、堆、优先级队列以及图
- 排序和搜索
- 数值计算
- 数据压缩
- 数据加密
- 图算法
- 几何算法

## &lt;&lt;算法精解&gt;&gt;

## 作者简介

Kyle Loudon，是美国加州洛斯加托斯Jeppesen Dataplan公司的一名软件工程师，主管图形接口开发小组，主攻航迹规划软件的研发，这些软件主要用于商业航空公司、私营航空部门和其他一些航空制造业。在来到Jeppesen之前，Kyle在IBM公司是一名系统程序员。在技术上，Kyle主要对操作系统、网络、人机交互等领域感兴趣。1992年，Kyle在普渡大学拿到了计算机科学学士学位，并取得了法语的第二学位，同时他还被选入斐陶斐荣誉学会（美国大学优等生之荣誉学会）。他在普渡大学计算机系教了三年的计算机课程。在这期间，他完成了他个人的第一本书《Understanding Computers》，这本书用理论结合实践的方式介绍计算机的方方面面。如今，尽管他继续工作在硅谷的软件业，但他仍然坚韧不拔地在追求一个更高的学位。

除了计算机，Kyle多年来喜欢打网球、教网球。他还喜欢山地骑行、滑冰，偶尔也和朋友们一起参加高尔夫课程。另外，Kyle还喜欢各种形式的戏剧、美食，以及某些风格的音乐和艺术；他期望成为钢琴家和艺术家，但希望渺茫。他现在在Jeppesen的工作是从他1992年开始驾驶飞机之后找到的。现在，他是一个拥有美国联邦航空局颁发的商业飞行员执照的飞行员。

## &lt;&lt;算法精解&gt;&gt;

## 书籍目录

1. 前言
2. 第1部分 预备知识
3. 第1章 概述
4. 数据结构简介
5. 算法简介
6. 小酌软件工程
7. 如何使用本书
8. 第2章 指针操作
9. 指针基础
10. 存储空间分配
11. 数据集合与指针的算术运算
12. 作为函数参数的指针
13. 泛型指针与类型转换
14. 函数指针
15. 问与答
16. 相关主题
17. 第3章 递归
18. 基本递归
19. 尾递归
20. 问与答
21. 相关主题
22. 第4章 算法分析
23. 最坏情况分析
24. O表示法
25. 计算的复杂度
26. 实例分析：插入排序
27. 问与答
28. 相关主题
29. 第2部分 数据结构
30. 第5章 链表
31. 单链表介绍
32. 单链表接口的定义
33. 单链表的实现与分析
34. 使用链表的例子：页帧管理
35. 双向链表介绍
36. 双向链表接口的定义
37. 双向链表的实现与分析
38. 循环链表介绍
39. 循环链表接口的定义
40. 循环链表的实现与分析
41. 使用循环链表的例子：第二次机会页面置换法
42. 问与答
43. 相关主题
44. 第6章 栈和队列
45. 栈的描述

## &lt;&lt;算法精解&gt;&gt;

46. 栈的接口定义
47. 栈的实现与分析
48. 队列的描述
49. 队列的接口定义
50. 队列的实现与分析
51. 队列示例：事件处理
52. 问与答
53. 相关主题
54. 第7章 集合
55. 集合介绍
56. 集合的性质
57. 集合接口的定义
58. 集合抽象数据类型的实现和分析
59. Set示例：集合覆盖
60. 问与答
61. 相关主题
62. 第8章 哈希表
63. 链式哈希表的描述
64. 链式哈希表的接口定义
65. 链式哈希表的实现与分析
66. 链式哈希表的例子：符号表
67. 开地址哈希表的描述
68. 开地址哈希函数的接口定义
69. 开地址哈希表的实现与分析
70. 问与答
71. 相关主题
72. 第9章 树
73. 二叉树介绍
74. 二叉树的接口定义
75. 二叉树的实现与分析
76. 二叉树示例：表达式处理
77. 二叉搜索树介绍
78. 二叉搜索树的接口定义
79. 二叉搜索树的实现与分析
80. 问与答
81. 相关主题
82. 第10章 堆和优先队列
83. 堆的描述
84. 堆的接口定义
85. 堆的实现与分析
86. 优先队列的描述
87. 优先队列的接口定义
88. 优先队列的实现与分析
89. 优先队列的示例：包裹分拣
90. 问与答
91. 相关主题
92. 第11章 图

## &lt;&lt;算法精解&gt;&gt;

93. 图的描述
94. 图的接口定义
95. 图的实现与分析
96. 关于图的应用举例：计算网络跳数
97. 关于图的应用举例：拓扑排序
98. 问与答
99. 相关主题
100. 第3部分 算法
101. 第12章 排序和搜索
102. 插入排序的描述
103. 插入排序的接口定义
104. 插入排序的实现与分析
105. 快速排序的描述
106. 快速排序的接口定义
107. 快速排序的实现与分析
108. 快速排序的例子：目录列表
109. 归并排序的描述
110. 归并排序的接口定义
111. 归并排序的实现与分析
112. 计数排序的描述
113. 计数排序的接口定义
114. 计数排序的实现与分析
115. 基数排序的描述
116. 基数排序的接口定义
117. 基数排序的实现与分析
118. 二分查找的描述
119. 二分查找的接口定义
120. 二分查找的实现与分析
121. 二分查找的例子：拼写检查器
122. 问与答
123. 相关主题
124. 第13章 数值计算
125. 多项式插值法
126. 多项式插值的接口定义
127. 多项式插值的实现与分析
128. 最小二乘估计法
129. 最小二乘估计的接口定义
130. 最小二乘估计的实现和分析
131. 方程求解介绍
132. 方程求解的接口定义
133. 方程求解的实现与分析
134. 问与答
135. 相关主题
136. 第14章 数据压缩
137. 位操作的描述
138. 位操作的接口定义
139. 位操作的实现与分析

## &lt;&lt;算法精解&gt;&gt;

140. 霍夫曼编码的描述
141. 霍夫曼编码的接口定义
142. 霍夫曼编码的分析与实现
143. 霍夫曼编码的例子：网络优化
144. LZ77的描述
145. LZ77的接口定义
146. LZ77的实现与分析
147. 问与答
148. 相关主题
149. 第15章 数据加密
150. DES算法介绍
151. DES的接口定义
152. DES算法的实现和分析
153. DES应用举例：分组加密模式
154. RSA算法介绍
155. RSA的接口定义
156. RSA算法的实现与分析
157. 问与答
158. 相关主题
159. 第16章 图算法
160. 最小生成树的描述
161. 最小生成树的接口定义
162. 最小生成树的实现与分析
163. 最短路径的描述
164. 最短路径的接口定义
165. 最短路径的实现与分析
166. 最短路径的例子：路由表
167. 旅行商问题的描述
168. 旅行商问题的接口定义
169. 旅行商问题的实现与分析
170. 问与答
171. 相关主题
172. 第17章 几何算法
173. 测试线段是否相交
174. 测试线段是否相交的标准方法
175. 检测线段是否相交的接口定义
176. 检测线段是否相交的实现与分析
177. 凸包简介
178. Jarvis ' s March
179. 凸包的接口定义
180. 凸包的实现与分析
181. 球面弧长
182. 求解球面弧长的接口定义
183. 求解球面弧长的实现和分析
184. 球面弧长的应用举例：地球上两点之间的近似距离
185. 问与答
186. 相关主题





## &lt;&lt;算法精解&gt;&gt;

## 章节摘录

版权页：插图：问与答问：链表比数组优越的地方前面已经介绍过了。但是，数组同样也有比链表优越的地方，那么什么情况下适合使用数组呢？

答：当我们期望进行频繁的插入和删除操作时，链表比数组更有优势。

然而，当我们期望进行随机访问的次数高于插入和删除操作的次数时，数组就显得更有优势了。随机访问是数组的强项，因为它们的元素在内存中是连续排列的。

这种连续的排列方式使得数组中的任何元素能够在 $O(1)$ 的时间内通过其索引访问。

回顾一下访问链表中元素的方法，我们必须得有一个指向元素的指针。

如果我们对访问元素的方式不甚了解，那么要获取某个指向特定元素的指针的代价将非常高。

在实践中，对于许多应用来说，我们至少需要遍历链表的一部分。

如果存储数据的总量是恒定的，则数组也有更大的优势，因为它们不需要增加额外的指针来使得它们所有的元素“链接”起来。

问：关于链表的插入、删除以及访问元素的操作和数组相比有何差异？

答：回顾一下本章中各种形式的链表，除了销毁链表操作外，其他的操作都具有 $O(1)$ 的运行时复杂度。

确实，这种表现似乎很难控制。

然而，在分析过程中有一点并没有说明，那就是对于许多链表的操作来说，想得到指向链表中某个特定元素的指针其代价是很高的。

例如，在最坏的情况下，可能需要遍历整个链表，此时的开销就是 $O(n)$ ，这里 $n$ 代表链表中的元素个数。

另一方面，在一个设计得当的应用中，比如本章的页帧管理，则对此就不会有任何性能上的额外开销。

因此，观察应用的特点是非常重要的。

对于数组，插入和删除都是 $O(n)$ 级别的操作，因为在最坏的情况下，插入或删除索引为0的元素需要将其他所有的元素都移动一个位置来调整整个数组的布局。

如果我们知道索引值，则访问数组中的元素就是 $o(1)$ 的操作。

问：假设我们想在本章给出的单链表基础上写一个名为`list_ins_pos`的函数，它的作用是在给定的位置之后插入一个新元素。

假设我们希望在第9个元素之后插入新元素，但并不直接提供指向第9个元素的指针。

那么这个函数的运行时复杂度是什么？

<<算法精解>>

编辑推荐

《算法精解:C语言描述》编辑推荐：数据结构和算法领域最具特色著作之一，公认权威经典，畅销不衰！

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>