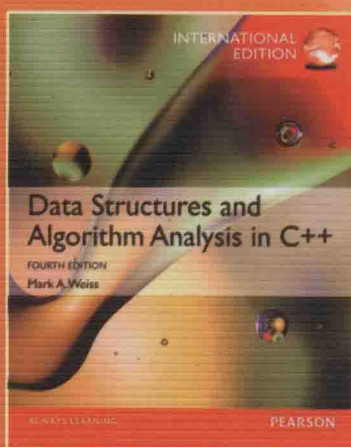


数据结构与算法分析

—— C++语言描述 (第四版)

Data Structures and Algorithm Analysis in C++
Fourth Edition



英文版

[美] Mark Allen Weiss 著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

国外计算机科学教材系列

数据结构与算法分析

——C++ 语言描述

(第四版) (英文版)

Data Structures and Algorithm Analysis in C++
Fourth Edition

[美] Mark Allen Weiss 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是数据结构与算法分析的经典教材, 书中使用主流的程序设计语言C++的新标准C++11作为具体的实现语言。书中内容包括表、栈、队列、树、散列表、优先队列、排序、不相交集算法、图论算法、算法分析、算法设计、摊还分析、查找树算法、后缀数组、后缀树、 k - d 树和配对堆等。本书把算法分析与C++程序的开发有机地结合起来, 深入分析每种算法, 内容全面、缜密严格, 并细致讲解精心构造程序的方法。

本书概念清楚, 逻辑性强, 内容新颖, 适合作为高等学校计算机软件与计算机应用等相关专业的双语教材或参考书, 也适合计算机工程技术人员参考。

Authorized Reprint from English language edition, entitled Data Structures and Algorithm Analysis in C++, Fourth Edition, by Mark Allen Weiss, published by Pearson Education, Ltd., Copyright © Pearson Education Inc. 2014.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

ENGLISH language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2017.

This edition is manufactured in the People's Republic of China, and is authorized for sale and distribution only in the mainland of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版专有出版权由 Pearson Education (培生教育出版集团) 授予电子工业出版社。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书在中国大陆地区出版, 仅限在中国大陆发行。

本书贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

版权贸易合同登记号 图字: 01-2017-0855

图书在版编目(CIP)数据

数据结构与算法分析: C++语言描述: 第四版: 英文版=Data Structures and Algorithm Analysis in C++, Fourth Edition/ (美) 马克·艾伦·韦斯 (Mark Allen Weiss) 著. —北京: 电子工业出版社, 2017.8

国外计算机科学教材系列

ISBN 978-7-121-32316-4

I. ①数… II. ①马… III. ①C++语言—数据结构—算法分析—高等学校—教材—英文 IV. ①TP312.8
②TP311.12

中国版本图书馆CIP数据核字(2017)第181834号

策划编辑: 冯小贝

责任编辑: 冯小贝

印刷: 三河市双峰印刷装订有限公司

装订: 三河市双峰印刷装订有限公司

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: 787×980 1/16 印张: 41 字数: 1194千字

版次: 2017年8月第1版(原著第4版)

印次: 2017年8月第1次印刷

定价: 109.00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, 88258888。

质量投诉请发邮件至zltts@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: fengxiaobei@phei.com.cn。

To my kind, brilliant, and inspiring Sara.

前 言

目的/目标

本书是《数据结构与算法分析——C++语言描述》的第四版，论述组织大量数据的方法——数据结构，以及算法运行时间的估计——算法分析。随着计算机的速度越来越快，对于能够处理大量输入数据的程序需求变得日益急迫。可是，由于在输入量很大时程序的低效率显得非常突出，因此又要求对效率问题给予更加严密的关注。通过在实际编程之前对算法进行分析，学生们可以确定一个特定的解决方案是否可行。例如，在本书中学生可查阅一些特定的问题并看到精心的实现怎样能够把处理大量数据的时间限制从若干个世纪减至不到一秒。因此，若无运行时间的阐释，就不会有算法和数据结构的提出。在某些情况下，对于影响实现的运行时间的一些微小细节都需要认真的探究。

一旦解法被确定，接着就要编写程序。随着计算机功能的日益强大，它们必须解决的问题就变得更加庞大和复杂，这就要求开发更加复杂的程序。本书的目标是同时教授学生良好的程序设计技巧和算法分析能力，使他们开发的程序能够具有最高的效率。

本书适合于高等数据结构课程或是算法分析第一年的研究生课程。学生应该具有中等程度的程序设计知识，包括指针、递归以及面向对象程序设计这样一些内容，此外还要具有一些离散数学的基础。

处理方法

虽然本书的内容大部分都与语言无关，但是，程序设计还是需要使用某种特定的语言。正如书名指出的，我们为本书选择了C++。

C++已经成为主流系统编程语言。除修复C语言的许多语法漏洞之外，C++还提供一些直接结构（类和模板）来实现作为抽象数据类型的泛型数据结构。

编写本书最困难的部分是决定C++所占的篇幅。使用太多C++的特性将使本书难以理解，使用太少又会使读者得不到比支持类的C语言教材更多的收获。

我们所采取的做法是以一种基于对象的处理方式展示书中的内容。这样，本书几乎不存在对继承的使用。书中以类模板描述泛型数据结构。一般我们避免深奥的C++特性，但是使用了vector类和string类，如今它们已是C++标准的一部分。本书以前各版通过把类模板接口从其实现中分离出来已经做到了对类模板的实现。虽然这是有争议的首选方式，但它揭示出编译器使读者难以具体使用代码的一些问题。因此，这一版中联机代码把类模板作为一个独立的单元来介绍，无须接口与实现的分离。第1章提供了用于全书的C++特性的综述，并描述了对类模板的处理方式。附录A 阐述如何能够重写类模板以使用分离式编译。

以C++和Java二者描述的数据结构的完全版可在互联网上获得。我们使用类似的编码约定以使得这两种语言间平行的结构表现得更加明显。

第四版最重要的变化概述

本书第四版吸收了大量对错误的修正，书中许多部分经过修订以增加表述的清晰度。此外：

- 第4章包含AVL树删除算法的实现，它是一个常常被读者提出的论题。
- 第5章已被广泛修订和扩展，现已包含两个更新的算法：杜鹃散列和跳房子散列。此外，还添加了论述通用散列的新的一节。对C++中引入的unordered_set和unordered_map两个类模板的简要讨论也是新加的内容。
- 第6章基本没有变化，不过，二叉堆的实现用到了C++11版引入的移动操作（move operation）。
- 第7章增加了基数排序的内容，并添加了新的一节，论述下界的证明。排序的程序用到C++11版中引入的移动操作。
- 第8章使用由Seidel和Sharir所做的新的union/find分析，并证明了 $O(M\alpha(M, N))$ 界以代替前面各版较弱的 $O(M \log^* N)$ 界。
- 第12章添加了论述后缀树和后缀数组的材料，包括Karkkainen和Sanders的后缀数组线性时间构建算法（及其实现）。删除了讨论确定性跳跃表和AA树的两节。
- 全书所出现的代码均用C++11做了更新。值得注意的是，这意味着C++11新特性的使用，包括auto关键字、范围for循环、移动构造和赋值，以及统一初始化（uniform initialization）。

内容提要

第1章包含离散数学和递归的一些复习材料。我相信熟练处置递归唯一的办法是反复不断地研读一些好的用法。因此，除第5章外，递归遍及本书每一章的例子之中。另外，第1章还介绍了一些材料，作为对基本C++的回顾，包括在C++类设计中对模板和一些重要结构的讨论。

第2章处理算法分析。这一章阐述渐近分析和它的主要弱点。这里提供了许多例子，包括对对数运行时间的深入解释。通过直观地把一些简单递归程序转变成迭代程序而对它们进行分析。介绍了更复杂的分治程序，不过有些分析（求解递推关系）将推迟到第7章再详细阐述。

第3章包括表、栈和队列。这一章包括对STL中vector类和list类的讨论，其中涉及迭代器的一些材料，并提供对STL中vector类和list类的重要子集的实现。

第4章讨论树，重点在查找树，包括外部查找树（B树）。UNIX文件系统和表达式树是作为例子来使用的。本章还介绍了AVL树和伸展树。关于查找树实现细节的更详细的处理放在第12章介绍。树的另外一些内容，如文件压缩和博弈树，推迟至第10章讨论。外部媒体上的数据结构作为几章中的最后论题来考虑。STL中set类和map类的讨论也在本章进行，其中包括一个重要的例子，该例使用3个分离的映射高效地解决一个问题。

第5章讨论散列表，包括诸如分离链接法以及线性探测法和平方探测法这样一些经典算法，此外还有几个新算法，即杜鹃散列和跳房子散列。通用散列也在这里讨论，而对可扩散列的讨论则放在本章末尾进行。

第6章是关于优先队列的。二叉堆也安排在这里，还有些额外的材料论述优先队列某些理论上有趣的实现。斐波那契堆在第11章讨论，配对堆在第12章讨论。

第7章是排序。它是关于编程细节和分析的非常特殊的一章。所有重要的通用排序算法均被讨论并进行了比较。详细分析了4种算法：插入排序，希尔排序，堆排序，以及快速排序。本版新增加了基数排序和一些与选择相关问题的下界证明。外部排序的讨论安排在本章的末尾进行。

第8章讨论不相交集算法并证明其运行时间。这是短小且特殊的一章，如果不讨论Kruskal算法则该章可以跳过。

第9章讲授图论算法。图论算法的趣味性不仅因为它们在实践中经常发生，而且还因为它们的运行时间强烈地依赖于数据结构的恰当使用。实际上，

所有标准算法都是和相应的数据结构、伪代码以及运行时间的分析一起介绍的。为把这些问题放在一个适当的上下文环境下，本章对复杂性理论（包括NP完全性和不可判定性）进行了简要的讨论。

第10章通过考查一些常见问题的求解技巧来讨论算法设计。这一章通过大量实例而得到强化。这里及后面各章使用的伪代码使得学生对一个算法实例的理解不至于被实现的细节所困扰。

第11章处理摊还分析。对来自第4章和第6章的3种数据结构以及本章介绍的斐波那契堆进行了分析。

第12章讨论查找树算法、后缀树和后缀数组、 k -d树、配对堆。不同于其他各章，本章为查找树和配对堆提供了完全和审慎的实现。材料的安排使得教师可以把一些内容整合到其他各章的讨论中。例如，第12章中的自顶向下红黑树可以和AVL树（第4章）一起讨论。

第1章~第9章为大多数一学期的数据结构课程提供了足够的材料。如果时间允许，那么第10章也可以包括进来。研究生的算法分析课程可以使用第7章~第11章的内容。在第11章所分析的高级数据结构可以容易地在前面各章中查到。第9章中对NP完全性的讨论太过简单，以至于不足以用于这样的一门算法分析课程。读者将会发现，参阅一些论述NP完全性的著述对深化本书内容大有裨益。

练习

在每章末尾提供的练习与正文中讲授内容的顺序相匹配。最后的一些练习是把一章作为一个整体来处理而不是针对特定的某一节来考虑的。难做的练习标以一个星号，更难的练习标注两个星号。

参考文献

参考文献列于每章的最后。一般来说，这些参考文献或者是历史性质的，代表着书中材料的原始来源，或者阐述对正文中给出结果的扩展和改进。有些文献提供一些练习的解法。

补充材料

所有读者均可从网站<http://cssupport.pearsoncmg.com/>上获取下列补充资料：

- 示例程序的源代码^①
- 勘误表

此外，下列材料只提供给Pearson Instructor Resource Center (www.pearsonhighered.com/irc) 上有资格的教师。如欲获取这些资料，可访问IRC或咨询Pearson Education 销售代表。^②

- 书中挑选的一些练习的解答
- 本书中的一些图示
- 勘误表

致谢

在该丛书几部著作的准备过程中作者得到了许许多多朋友的帮助，有些人在本书的其他版本中列出。谢谢所有诸位。

如同往常一样，Pearson专家们的努力使得本书的写作过程更加轻松。我愿意借此机会感谢我的编辑Tracy Johnson，以及制作编辑Marilyn Lloyd。贤妻Jill因其所做的每一件工作应该得到我特别的感谢。

最后，我还要感谢广大的读者，他们发送E-mail并指出较早各版的错误和矛盾之处。我的网站www.cis.fiu.edu/~weiss也将包含更新后（C++和Java）的源代码，一个勘误表，以及提交问题报告的一个链接。

M. A. W.
Miami, Florida

① 也可登录华信教育资源网 (www.hxedu.com.cn) 免费注册下载。

② 教辅申请方式请参见书末的“教学支持说明”。

Chapter 1 Programming: A General Overview 19

- 1.1 What's This Book About? 19
- 1.2 Mathematics Review 20
 - 1.2.1 Exponents 21
 - 1.2.2 Logarithms 21
 - 1.2.3 Series 22
 - 1.2.4 Modular Arithmetic 23
 - 1.2.5 The P Word 24
- 1.3 A Brief Introduction to Recursion 26
- 1.4 C++ Classes 30
 - 1.4.1 Basic class Syntax 30
 - 1.4.2 Extra Constructor Syntax and Accessors 31
 - 1.4.3 Separation of Interface and Implementation 34
 - 1.4.4 vector and string 37
- 1.5 C++ Details 39
 - 1.5.1 Pointers 39
 - 1.5.2 Lvalues, Rvalues, and References 41
 - 1.5.3 Parameter Passing 43
 - 1.5.4 Return Passing 45
 - 1.5.5 std::swap and std::move 47
 - 1.5.6 The Big-Five: Destructor, Copy Constructor, Move Constructor, Copy Assignment operator=, Move Assignment operator= 48
 - 1.5.7 C-style Arrays and Strings 53
- 1.6 Templates 54
 - 1.6.1 Function Templates 55
 - 1.6.2 Class Templates 56
 - 1.6.3 Object, Comparable, and an Example 57
 - 1.6.4 Function Objects 59
 - 1.6.5 Separate Compilation of Class Templates 62
- 1.7 Using Matrices 62
 - 1.7.1 The Data Members, Constructor, and Basic Accessors 62
 - 1.7.2 operator[] 63

1.7.3 Big-Five	64
Summary	64
Exercises	64
References	66

Chapter 2 Algorithm Analysis

69

2.1 Mathematical Background	69
2.2 Model	72
2.3 What to Analyze	72
2.4 Running-Time Calculations	75
2.4.1 A Simple Example	76
2.4.2 General Rules	76
2.4.3 Solutions for the Maximum Subsequence Sum Problem	78
2.4.4 Logarithms in the Running Time	84
2.4.5 Limitations of Worst-Case Analysis	88
Summary	88
Exercises	89
References	94

Chapter 3 Lists, Stacks, and Queues

95

3.1 Abstract Data Types (ADTs)	95
3.2 The List ADT	96
3.2.1 Simple Array Implementation of Lists	96
3.2.2 Simple Linked Lists	97
3.3 vector and list in the STL	98
3.3.1 Iterators	100
3.3.2 Example: Using erase on a List	101
3.3.3 const_iterators	102
3.4 Implementation of vector	104
3.5 Implementation of list	109
3.6 The Stack ADT	121
3.6.1 Stack Model	121
3.6.2 Implementation of Stacks	122
3.6.3 Applications	122
3.7 The Queue ADT	130
3.7.1 Queue Model	131
3.7.2 Array Implementation of Queues	131
3.7.3 Applications of Queues	133
Summary	134
Exercises	134

Chapter 4 Trees

139

- 4.1 Preliminaries 139
 - 4.1.1 Implementation of Trees 140
 - 4.1.2 Tree Traversals with an Application 141
- 4.2 Binary Trees 144
 - 4.2.1 Implementation 146
 - 4.2.2 An Example: Expression Trees 146
- 4.3 The Search Tree ADT—Binary Search Trees 150
 - 4.3.1 contains 152
 - 4.3.2 findMin and findMax 153
 - 4.3.3 insert 154
 - 4.3.4 remove 157
 - 4.3.5 Destructor and Copy Constructor 159
 - 4.3.6 Average-Case Analysis 159
- 4.4 AVL Trees 162
 - 4.4.1 Single Rotation 165
 - 4.4.2 Double Rotation 167
- 4.5 Splay Trees 176
 - 4.5.1 A Simple Idea (That Does Not Work) 176
 - 4.5.2 Splaying 178
- 4.6 Tree Traversals (Revisited) 184
- 4.7 B-Trees 186
- 4.8 Sets and Maps in the Standard Library 191
 - 4.8.1 Sets 191
 - 4.8.2 Maps 192
 - 4.8.3 Implementation of set and map 193
 - 4.8.4 An Example That Uses Several Maps 194
- Summary 199
- Exercises 200
- References 207

Chapter 5 Hashing

211

- 5.1 General Idea 211
- 5.2 Hash Function 212
- 5.3 Separate Chaining 214
- 5.4 Hash Tables without Linked Lists 219
 - 5.4.1 Linear Probing 219
 - 5.4.2 Quadratic Probing 220
 - 5.4.3 Double Hashing 225
- 5.5 Rehashing 226
- 5.6 Hash Tables in the Standard Library 228

5.7	Hash Tables with Worst-Case $O(1)$ Access	230
5.7.1	Perfect Hashing	231
5.7.2	Cuckoo Hashing	233
5.7.3	Hopscotch Hashing	245
5.8	Universal Hashing	248
5.9	Extendible Hashing	251
	Summary	254
	Exercises	255
	References	259

Chapter 6 Priority Queues (Heaps)

263

6.1	Model	263
6.2	Simple Implementations	264
6.3	Binary Heap	265
6.3.1	Structure Property	265
6.3.2	Heap-Order Property	266
6.3.3	Basic Heap Operations	267
6.3.4	Other Heap Operations	270
6.4	Applications of Priority Queues	275
6.4.1	The Selection Problem	276
6.4.2	Event Simulation	277
6.5	d -Heaps	278
6.6	Leftist Heaps	279
6.6.1	Leftist Heap Property	279
6.6.2	Leftist Heap Operations	280
6.7	Skew Heaps	287
6.8	Binomial Queues	289
6.8.1	Binomial Queue Structure	289
6.8.2	Binomial Queue Operations	289
6.8.3	Implementation of Binomial Queues	294
6.9	Priority Queues in the Standard Library	300
	Summary	301
	Exercises	301
	References	306

Chapter 7 Sorting

309

7.1	Preliminaries	309
7.2	Insertion Sort	310
7.2.1	The Algorithm	310
7.2.2	STL Implementation of Insertion Sort	311
7.2.3	Analysis of Insertion Sort	312
7.3	A Lower Bound for Simple Sorting Algorithms	313

7.4	Shellsort	314
7.4.1	Worst-Case Analysis of Shellsort	315
7.5	Heapsort	318
7.5.1	Analysis of Heapsort	319
7.6	Mergesort	322
7.6.1	Analysis of Mergesort	324
7.7	Quicksort	327
7.7.1	Picking the Pivot	329
7.7.2	Partitioning Strategy	331
7.7.3	Small Arrays	333
7.7.4	Actual Quicksort Routines	333
7.7.5	Analysis of Quicksort	336
7.7.6	A Linear-Expected-Time Algorithm for Selection	339
7.8	A General Lower Bound for Sorting	341
7.8.1	Decision Trees	341
7.9	Decision-Tree Lower Bounds for Selection Problems	343
7.10	Adversary Lower Bounds	346
7.11	Linear-Time Sorts: Bucket Sort and Radix Sort	349
7.12	External Sorting	354
7.12.1	Why We Need New Algorithms	354
7.12.2	Model for External Sorting	354
7.12.3	The Simple Algorithm	355
7.12.4	Multiway Merge	356
7.12.5	Polyphase Merge	357
7.12.6	Replacement Selection	358
	Summary	359
	Exercises	359
	References	365

Chapter 8 The Disjoint Sets Class

369

8.1	Equivalence Relations	369
8.2	The Dynamic Equivalence Problem	370
8.3	Basic Data Structure	371
8.4	Smart Union Algorithms	375
8.5	Path Compression	378
8.6	Worst Case for Union-by-Rank and Path Compression	379
8.6.1	Slowly Growing Functions	380
8.6.2	An Analysis by Recursive Decomposition	380
8.6.3	An $O(M \log^* N)$ Bound	387
8.6.4	An $O(M \alpha(M, N))$ Bound	388
8.7	An Application	390

Summary	392
Exercises	393
References	394

Chapter 9 Graph Algorithms

397

9.1	Definitions	397
9.1.1	Representation of Graphs	398
9.2	Topological Sort	400
9.3	Shortest-Path Algorithms	404
9.3.1	Unweighted Shortest Paths	405
9.3.2	Dijkstra's Algorithm	409
9.3.3	Graphs with Negative Edge Costs	418
9.3.4	Acyclic Graphs	418
9.3.5	All-Pairs Shortest Path	422
9.3.6	Shortest Path Example	422
9.4	Network Flow Problems	424
9.4.1	A Simple Maximum-Flow Algorithm	426
9.5	Minimum Spanning Tree	431
9.5.1	Prim's Algorithm	432
9.5.2	Kruskal's Algorithm	435
9.6	Applications of Depth-First Search	437
9.6.1	Undirected Graphs	438
9.6.2	Biconnectivity	439
9.6.3	Euler Circuits	443
9.6.4	Directed Graphs	447
9.6.5	Finding Strong Components	449
9.7	Introduction to NP-Completeness	450
9.7.1	Easy vs. Hard	451
9.7.2	The Class NP	452
9.7.3	NP-Complete Problems	452
	Summary	455
	Exercises	455
	References	463

Chapter 10 Algorithm Design Techniques

467

10.1	Greedy Algorithms	467
10.1.1	A Simple Scheduling Problem	468
10.1.2	Huffman Codes	471
10.1.3	Approximate Bin Packing	477
10.2	Divide and Conquer	485
10.2.1	Running Time of Divide-and-Conquer Algorithms	486
10.2.2	Closest-Points Problem	488

10.2.3	The Selection Problem	493
10.2.4	Theoretical Improvements for Arithmetic Problems	496
10.3	Dynamic Programming	500
10.3.1	Using a Table Instead of Recursion	501
10.3.2	Ordering Matrix Multiplications	503
10.3.3	Optimal Binary Search Tree	505
10.3.4	All-Pairs Shortest Path	509
10.4	Randomized Algorithms	512
10.4.1	Random-Number Generators	513
10.4.2	Skip Lists	518
10.4.3	Primality Testing	521
10.5	Backtracking Algorithms	524
10.5.1	The Turnpike Reconstruction Problem	524
10.5.2	Games	529
	Summary	536
	Exercises	536
	References	545

Chapter 11 Amortized Analysis

551

11.1	An Unrelated Puzzle	552
11.2	Binomial Queues	552
11.3	Skew Heaps	557
11.4	Fibonacci Heaps	559
11.4.1	Cutting Nodes in Leftist Heaps	560
11.4.2	Lazy Merging for Binomial Queues	562
11.4.3	The Fibonacci Heap Operations	566
11.4.4	Proof of the Time Bound	567
11.5	Splay Trees	569
	Summary	573
	Exercises	574
	References	575

Chapter 12 Advanced Data Structures and Implementation

577

12.1	Top-Down Splay Trees	577
12.2	Red-Black Trees	584
12.2.1	Bottom-Up Insertion	585
12.2.2	Top-Down Red-Black Trees	586
12.2.3	Top-Down Deletion	588
12.3	Treaps	594

12.4 Suffix Arrays and Suffix Trees	597
12.4.1 Suffix Arrays	598
12.4.2 Suffix Trees	601
12.4.3 Linear-Time Construction of Suffix Arrays and Suffix Trees	604
12.5 k -d Trees	614
12.6 Pairing Heaps	620
Summary	624
Exercises	626
References	630

Appendix A Separate Compilation of Class Templates

633

A.1 Everything in the Header	634
A.2 Explicit Instantiation	634

Index	637
-------	-----