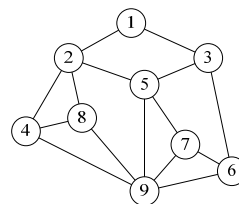
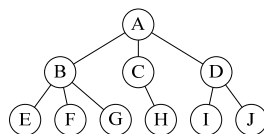
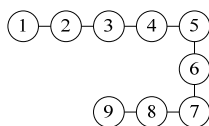
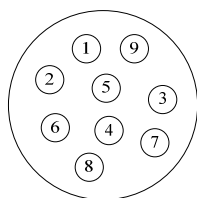


目 录

第 1 章 数据结构绪论 1



1.1 开场白 2

如果你交给某人一个程序，你将折磨他一整天；如果你教某人如何编写程序，你将折磨他一辈子。

1.2 你数据结构怎么学的？ 3

他完成开发并测试通过后，得意地提交了代码。项目经理看完代码后拍着桌子对他说：“你数据结构是怎么学的？”

1.3 数据结构起源 4

1.4 基本概念和术语 5

正所谓“巧妇难为无米之炊”，再强大的计算机，也要有“米”下锅才可以干活，否则就是一堆破铜烂铁。这个“米”就是数据。

1.4.1 数据 5	1.4.4 数据对象 6
1.4.2 数据元素 5	1.4.5 数据结构 6
1.4.3 数据项 6	

1.5 逻辑结构与物理结构 7

1.5.1 逻辑结构 7	1.5.2 物理结构 9
--------------------	--------------------

1.6 抽象数据类型 11

大家都需要房子住，但显然没钱考虑大房子是没有意义的。于是商品房就出现了各种各样的户型，有几百平米的别墅，也有仅两平米的胶囊公寓.....

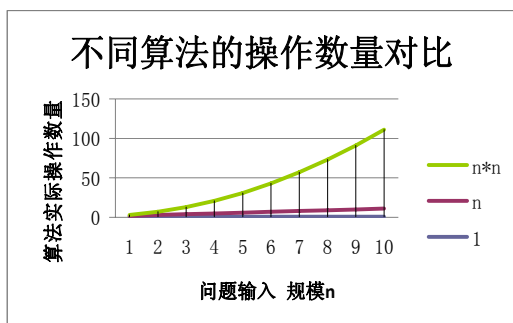
1.6.1 数据类型 11	1.6.2 抽象数据类型 12
---------------------	-----------------------

1.7 总结回顾 14

1.8 结尾语 15

最终的结果一定是，你对着别人很牛的说“数据结构——就那么回事。”

第 2 章 算法 17



2.1 开场白 18

2.2 数据结构与算法关系 18

计算机界的前辈们，是一帮很牛很牛的人，他们使得很多看似没法解决或者很难解决的问题，变得如此美妙和神奇。

2.3 两种算法的比较 19

高斯在上小学的一天，老师要求每个学生都计算 $1+2+\dots+100$ 的结果，谁先算出来谁先回家.....

2.4 算法定义 20

现实世界中的算法千变万化，没有通用算法可以解决所有问题。甚至一个小问题，某个解决此类问题很优秀的算法却未必就适合它。

2.5 算法的特性 21

2.5.1 输入输出 21	2.5.3 确定性 21
2.5.2 有穷性 21	2.5.4 可行性 21

2.6 算法设计的要求 22

求 100 个人的高考成绩平均分与求全省所有考生的成绩平均分在占用时间和内存存储上有非常大的差异，我们自然追求高效率 and 低存储的算法来解决问题。

2.6.1 正确性 22	2.6.3 健壮性 23
2.6.2 可读性 23	2.6.4 时间效率高和存储量低 23

2.7 算法效率的度量方法 24

随着 n 值越来越大，它们在时间效率上的差异也就越来越大。好比有些人每天都在学习，而另一些人，打打游戏、睡睡大觉，毕业后前者名企争着要，后者求职处处无门。

2.7.1 事后统计方法.....24 2.7.2 事前分析估算方法..... 25

2.8 函数的渐近增长27

2.9 算法时间复杂度29

理解大 O 推导不算难，难的其实是对数列的一些相关运算，这考察的更多的是数学知识和能力。

2.9.1 算法时间复杂度定义.....29 2.9.4 线性阶.....31
2.9.2 推导大O阶方法.....30 2.9.5 对数阶.....32
2.9.3 常数阶30 2.9.6 平方阶.....32

2.10 常见的的时间复杂度35

有些时候，告诉你某些东西不可以去尝试，也是一种知识的传递。总不能非要去被毒蛇咬一口才知道蛇不可以去招惹吧。

2.11 最坏情况与平均情况35

2.12 算法空间复杂度36

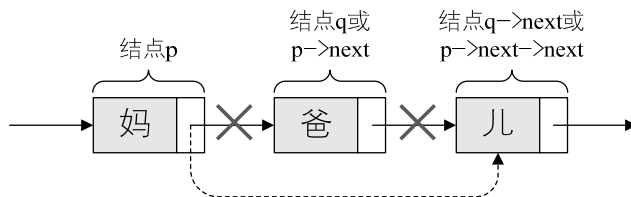
事先建立一个有 2050 大的数组，然后把所有年份按下标数字对应，如果是闰年，此数组项的值就是 1，如果不是就是 0。这样，所谓的判断某一年是否是闰年就变成了查找这个数组的某一项的值是多少的问题。

2.13 总结回顾37

2.14 结尾语38

愚公移山固然可敬，但发明炸药和推土机，可能更加实在和聪明。

第 3 章 线性表41



3.1	开场白.....	42
	<p>门外家长都挤在大门口与门里的小孩子的井然有序，形成了鲜明对比。哎，有时大人的所作所为，其实还不如孩子。</p>	
3.2	线性表的定义.....	42
3.3	线性表的抽象数据类型.....	45
	<p>有时我们想知道某个小朋友（比如麦兜）是否是班级的同学，老师会告诉我说，没有，麦兜是在春田花花幼儿园里。这种查找某个元素是否存在的操作很常用。</p>	
3.4	线性表的顺序存储结构.....	47
	<p>他每次一吃完早饭就冲着去了图书馆，挑一个好地儿，把他书包里的书，一本一本的按座位放好，长长一排，九个座硬是被他占了。</p>	
3.4.1	顺序存储定义.....	47
3.4.2	顺序存储方式.....	47
3.4.3	数据长度与线性表长度区别.....	48
3.4.4	地址计算方法.....	49
3.5	顺序存储结构的插入与删除.....	50
	<p>春运时去买火车票，大家都排队排着好好的，这时来了一个美女：“可否让我排在你前面？”这可不得了，后面的人像蠕虫一样，全部都得退后一步。</p>	
3.5.1	获得元素操作.....	50
3.5.2	插入操作.....	51
3.5.3	删除操作.....	52
3.5.4	线性表顺序存储结构的优缺点.....	54
3.6	线性表的链式存储结构.....	55
	<p>反正也是要让相邻元素间留有足够余地，那干脆所有元素都不要考虑相邻位置了，哪有空位就到哪里。而只是让每个元素知道它下一个元素的位置在哪里。</p>	
3.6.1	顺序存储结构不足的解决办法.....	55
3.6.2	线性表链式存储结构定义.....	56
3.6.3	头指针与头结点的异同.....	58
3.6.4	线性表链式存储结构代码描述.....	58
3.7	单链表的读取.....	60
3.8	单链表的插入与删除.....	61
	<p>本来是爸爸左牵着妈妈的手，右牵着宝宝的手在马路边散步。突然迎面走来一美女，爸爸失神般地望着，此情景被妈妈逮个正着，于是扯开父子俩，拉起宝宝的左手就快步朝前走去。</p>	
3.8.1	单链表的插入.....	61
3.8.2	单链表的删除.....	64
3.9	单链表的整表创建.....	66

3.10 单链表的整表删除69

3.11 单链表结构与顺序存储结构优缺点70

3.12 静态链表71

 对于一些语言，如 Basic、Fortran 等早期的编程高级语言，由于没有指针，这链表结构，按照前面我们的讲法，它就没法实现了。怎么办呢？

 3.12.1 静态链表的插入操作.....73 3.12.3 静态链表优缺点..... 77

 3.12.2 静态链表的删除操作.....75

3.13 循环链表78

 这个轮回的思想很有意思。它强调了不管你今生是穷是富，如果持续行善积德，下辈子就会好过，反之就会遭到报应。

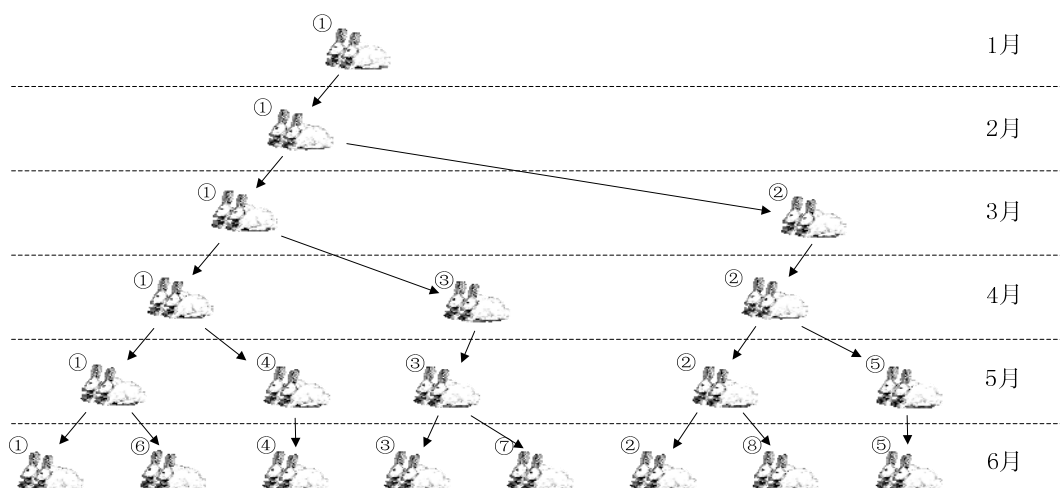
3.14 双向链表81

 就像每个人的人生一样，欲收获就得付代价。双向链表既然是比单链表多了如可以反向遍历查找等的数据结构，那么也就需要付出一些小的代价。

3.15 总结回顾84

3.16 结尾语85

第 4 章 栈与队列.....87



4.1	开场白	88
	想想看，在你准备用枪的时候，突然这手枪明明有子弹却打不出来，这不是要命吗。	
4.2	栈的定义	89
	类似的很多软件，比如 Word、Photoshop 等，都有撤消（undo）的操作，也是用栈这种思想方式来实现的。	
4.2.1	栈的定义	89
4.2.2	进栈出栈变化形式	90
4.3	栈的抽象数据类型	91
4.4	栈的顺序存储结构及实现	92
4.4.1	栈的顺序存储结构	92
4.4.2	栈的顺序存储结构进栈操作	93
4.4.3	栈的顺序存储结构出栈操作	94
4.5	两栈共享空间	94
	两个大学室友毕业同时到北京工作，他们都希望租房时能找到独自住的一室户或一室一厅，可找来找去发现，实在是承受不起。	
4.6	栈的链式存储结构及实现	97
4.6.1	栈的链式存储结构	97
4.6.2	栈的链式存储结构进栈操作	98
4.6.3	栈的链式存储结构出栈操作	99
4.7	栈的作用	100
4.8	栈的应用——递归	100
	当你往镜子前面一站，镜子里面就有一个你的像。但你试过两面镜子一起照吗？如果 A、B 两面镜子相互面对面放着，你往中间一站，嘿，两面镜子里都有你的千百个“化身”。	
4.8.1	斐波那契数列实现	101
4.8.2	递归定义	103
4.9	栈的应用——四则运算表达式求值	104
4.9.1	后缀（逆波兰）表示法定义	104
4.9.2	后缀表达式计算结果	106
4.9.3	中缀表达式转后缀表达式	108
4.10	队列的定义	111
	电脑有时会处于疑似死机的状态。就当你失去耐心，打算了 Reset 时。突然它像酒醒了一样，把你刚才点击的所有操作全部都按顺序执行了一遍。	

4.11 队列的抽象数据类型 112

4.12 循环队列 113

 你上了公交车发现前排有两个空座位，而后排所有座位都已经坐满，你会怎么做？立马下车，并对自己说，后面没座了，我等下一辆？没这么笨的人，前面有座位，当然也是可以坐的。

 4.12.1 队列顺序存储的不足 112 4.12.2 循环队列定义 114

4.13 队列的链式存储结构及实现 117

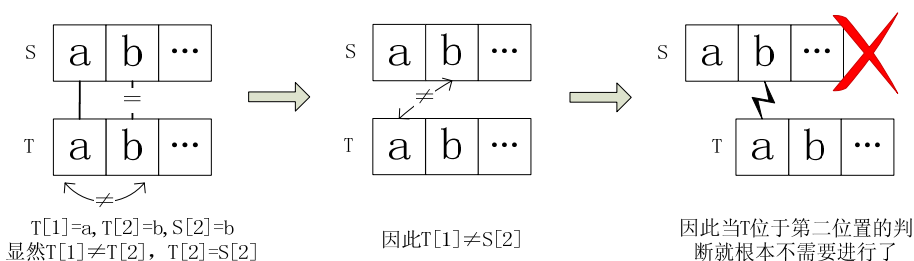
 4.13.1 队列链式存储结构入队操作 118 4.13.2 队列链式存储结构出队操作 119

4.14 总结回顾 120

4.15 结尾语 121

人生，需要有队列精神的体现。南极到北极，不过是南纬 90 度到北纬 90 度的队列，如果你中途犹豫，临时转向，也许你就只能和企鹅相伴永远。可事实上，无论哪个方向，只要你坚持到底，你都可以到达终点。

第 5 章 串 123



5.1 开场白 124

“枯眼望遥山隔水，往来曾见几心知？壶空怕酌一杯酒，笔下难成和韵诗。途路阻人离别久，讯音无雁寄回迟。孤灯夜守长寥寂，夫忆妻兮父忆儿。”……可再仔细一读发现，这首诗竟然可以倒过来读。

5.2 串的定义 124

我所提到的“over”、“end”、“lie”其实就是“lover”、“friend”、“believe”这些单词字符串的子串。

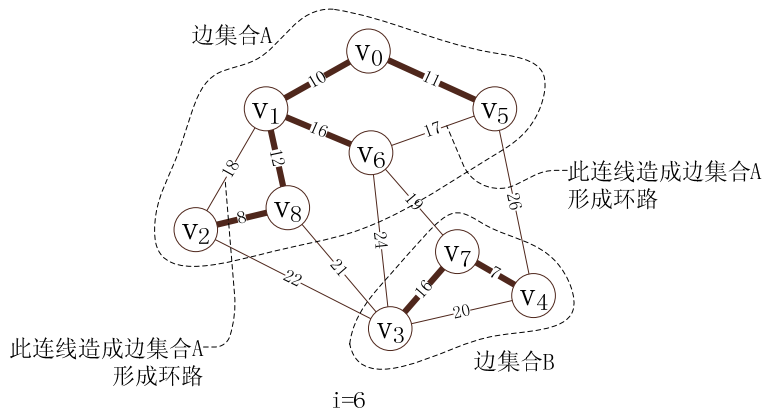
5.3 串的比较 126

5.4 串的抽象数据类型 127

6.1	开场白.....	150
	<p>无论多高多大的树，那也是从小到大的，由根到叶，一点点成长起来的。俗话说十年树木，百年树人，可一棵大树又何止是十年这样容易。</p>	
6.2	树的定义.....	150
	<p>树的定义其实就是我们在讲解栈时提到的递归的方法。也就是在树的定义之中还用到了树的概念，这是比较新的一种定义方法。</p>	
6.2.1	结点分类.....	152
6.2.2	结点间关系.....	152
6.2.3	树的其他相关概念.....	153
6.3	树的抽象数据类型.....	154
6.4	树的存储结构.....	155
6.4.1	双亲表示法.....	155
6.4.2	孩子表示法.....	158
6.4.3	孩子兄弟表示法.....	162
6.5	二叉树的定义.....	163
	<p>苏东坡曾说：“人有悲欢离合，月有阴晴圆缺，此事古难全”。意思就是完美是理想，不完美才是人生。我们通常举的例子也都是左高右低、参差不齐的二叉树。那是否存在完美的二叉树呢？</p>	
6.5.1	二叉树特点.....	164
6.5.2	特殊二叉树.....	166
6.6	二叉树的性质.....	169
6.6.1	二叉树性质 1.....	169
6.6.2	二叉树性质 2.....	169
6.6.3	二叉树性质 3.....	169
6.6.4	二叉树性质 4.....	170
6.6.5	二叉树性质 5.....	171
6.7	二叉树的存储结构.....	172
6.7.1	二叉树顺序存储结构.....	172
6.7.2	二叉链表.....	173
6.8	遍历二叉树.....	174
	<p>你人生的道路上，高考填志愿要面临哪个城市、哪所大学、具体专业等选择，由于选择方式的不同，遍历的次序就完全不同。</p>	
6.8.1	二叉树遍历原理.....	174
6.8.2	二叉树遍历方法.....	175
6.8.3	前序遍历算法.....	178
6.8.4	中序遍历算法.....	181
6.8.5	后序遍历算法.....	184
6.8.6	推导遍历结果.....	184

6.9	二叉树的建立.....	187
6.10	线索二叉树.....	188
<p>我们现在提倡节约型社会，一切都应该节约为本。对待我们的程序当然也不例外，能不浪费的时间或空间，都应该考虑节省。</p>		
6.10.1	线索二叉树原理.....	188
6.10.2	线索二叉树结构实现.....	191
6.11	树、森林与二叉树的转换.....	195
<p>有个乡镇企业也买了同样的生产线，老板发现这个问题后找了个小工来说：你必须搞定，不然炒你鱿鱼。小工很快想出了办法：他在生产线旁边放了台风扇猛吹，空皂盒自然会被吹走。</p>		
6.11.1	树转换为二叉树.....	196
6.11.2	森林转换为二叉树.....	197
6.11.3	二叉树转换为树.....	197
6.11.4	二叉树转换为森林.....	199
6.11.5	树与森林的遍历.....	199
6.12	赫夫曼树及其应用.....	200
<p>压缩而不出错是如何做到的呢？简单的说，就是把我们要压缩的文本进行重新编码，以达到减少不必要的空间的技术。压缩和解压缩技术就是基于赫夫曼的研究之上发展而来，我们应该记住他。</p>		
6.12.1	赫夫曼树.....	200
6.12.2	赫夫曼树定义与原理.....	203
6.12.3	赫夫曼编码.....	205
6.13	总结回顾.....	208
6.14	结尾语.....	209
<p>人受伤时会流下泪水。树受伤时，天将再不会哭。希望我们的未来不要仅仅是钢筋水泥建造的高楼，也要有那郁郁葱葱的森林和草地，我们人类才可能与自然和谐共处。</p>		

第 7 章 图 211



7.1 开场白 212

如果你不善于规划，很有可能就会出现如玩好新疆后到海南，然后再冲向黑龙江这样的荒唐决策。

7.2 图的定义 213

现实中，人与人之间关系就非常复杂，比如我的认识的朋友，可能他们之间也互相认识，这就不是简单的一对一、一对多的关系了，那就是我们今天研究的主题——图。

- 7.2.1 各种图定义 214
- 7.2.2 图的顶点与边间关系 217
- 7.2.3 连通图相关术语 219
- 7.2.4 图的定义与术语总结 222

7.3 图的抽象数据类型 222

7.4 图的存储结构 223

因为美国的黑夜就是中国的白天，利用互联网，他的员工白天上班就可以监控到美国仓库夜间的实际情况，如果发生了像火灾、偷盗这样的突发事件，及时电话到美国当地相关人员处理

- 7.4.1 邻接矩阵 224
- 7.4.2 邻接表 228
- 7.4.3 十字链表 232
- 7.4.4 邻接多重表 234
- 7.4.5 边集数组 236

7.5 图的遍历 237

我有一天早晨准备出门，发现钥匙不见了。一定是我儿子拿着玩，不知道丢到哪个犄角旮旯去了，你们说，我应该如何找？

- 7.5.1 深度优先遍历 238
- 7.5.2 广度优先遍历 242

7.6 最小生成树 245

如果你加班加点，没日没夜设计出的结果是方案一，我想你离被炒鱿鱼应该是不远了（同学微笑）。因为这个方案比后两个方案一半还多的成本会让老板气晕过去的。

7.6.1 普里姆（Prim）算法 247 7.6.2 克鲁斯卡尔（Kruskal）算法 251

7.7 最短路径 257

有人为了省钱，需路程最短，但换乘站间距离长等原因并不省时间；另一些人，他为赶时间，最大的需求是总时间要短；还有一类人，他们都不想多走路，关键是换乘要少，这样可以在车上好好休息一下。

7.7.1 迪杰斯特拉（Dijkstra）算法 259 7.7.3 弗洛伊德（Floyd）算法 265

7.8 拓扑排序 270

电影制作不可能在人员到位进驻场地时，导演还没有找到，也不可能在拍摄过程中，场地都没有。这都会导致荒谬的结果。

7.8.1 拓扑排序介绍 271 7.8.2 拓扑排序算法 272

7.9 关键路径 277

假如造一个轮子要 0.5 天、造一个发动机要 3 天、造一个车底盘要 2 天、造一个外壳要 2 天，其它零部件 2 天，全部零部件集中到一处要 0.5 天，组装成车要 2 天，请问，在汽车厂造一辆车，最短需要多少天呢？

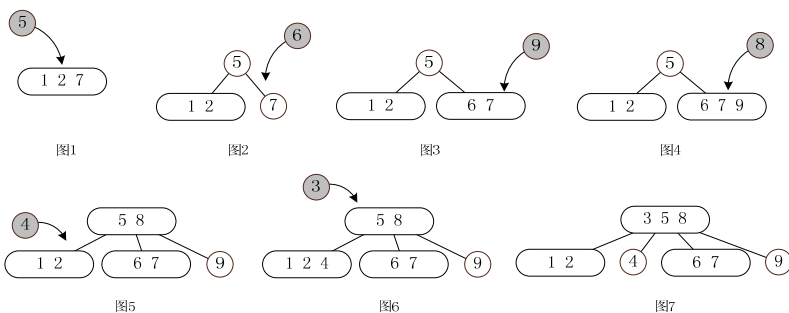
7.9.1 关键路径算法原理 279 7.9.2 关键路径算法 280

7.10 总结回顾 287

7.11 结尾语 289

世界上最遥远的距离，不是牛 A 与牛 C 之间狭小空隙，而是你们当中，有人在通往牛逼的路上一路狂奔，而有人步入大学校园就学会放弃。

第 8 章 查找 291



8.1 开场白 292

当你精心写了一篇博文或者上传一组照片到互联网上，来自世界各地的无数“蜘蛛”便会蜂拥而至。所谓蜘蛛就是搜索引擎公司服务器上软件，它把互联网当成了蜘蛛网，没日没夜的访问上面的各种信息。

8.2 查找概论 293

比如网络时代的新名词，如“蜗居”、“蚁族”等，如果需要将它们收录到汉语词典中，显然收录时就需要查找它们是否存在，以及找到如果不存在时应该收录的位置。

8.3 顺序表查找 295

8.3.1 顺序表查找算法 296 8.3.2 顺序表查找优化 297

8.4 有序表查找 298

我在纸上已经写好了一个 100 以内的正整数请你猜，问几次可以猜出来。当时已经介绍了如何才可以最快的猜出这个数字。我们把这种每次取中间记录查找的方法叫做折半查找。

8.4.1 折半查找 298 8.4.3 斐波那契查找 302
8.4.2 插值查找 301

8.5 线性索引查找 306

我母亲年纪大了，经常在家里找不到东西，于是她用一小本子，记录了家里所有小东西放置的位置，比如户口本放在右手床头柜下面抽屉中，钞票放在衣.....咳，这个就不提了。

8.5.1 稠密索引 307 8.5.3 倒排索引 311
8.5.2 分块索引 308

8.6 二叉排序树 313

后来老虎来了，一人拼命地跑，另一人则急中生智，爬到了树上。而老虎是不会爬树的，结果.....。爬树者改变了跑的思想，这一改变何等重要，捡回了自己的一条命。

8.6.1 二叉排序树查找操作 316 8.6.3 二叉排序树删除操作 320
8.6.2 二叉排序树插入操作 318 8.6.4 二叉排序树总结 327

8.7 平衡二叉树（AVL 树） 328

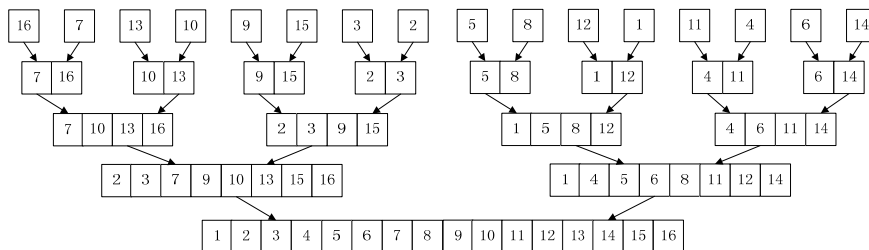
平板就是一个世界，当诱惑降临，人心中的平衡被打破，世界就会混乱，最后留下的只有孤独寂寞失败。这种单调的机械化的社会，禁不住诱惑的侵蚀，最容易被侵蚀的，恰恰是最空虚的心灵。

8.7.1 平衡二叉树实现原理 330 8.7.2 平衡二叉树实现算法 334

8.8 多路查找树 (B 树)	341
<p>要观察一个公司是否严谨，看他们如何开会就知道了。如果开会时每一个人都只是带一张嘴，即兴发言，这肯定是一家不严谨的公司。</p>	
8.8.1 2-3 树	343
8.8.2 2-3-4 树	348
8.8.3 B 树	349
8.8.4 B+树	351
8.9 散列表查找 (哈希表) 概述	353
<p>你很想学太极拳，听说学校有个叫张三丰的人打得特别好，于是到学校学生处找人，工作人员拿出学生名单，最终告诉你，学校没这个人，并说张三丰几百年前就已经在武当山作古了。</p>	
8.9.1 散列表查找定义	354
8.9.2 散列表查找步骤	355
8.10 散列函数的构造方法	356
8.10.1 直接定址法	357
8.10.2 数字分析法	358
8.10.3 平方取中法	359
8.10.4 折叠法	359
8.10.5 除留余数法	359
8.10.6 随机数法	360
8.11 处理散列冲突的方法	360
<p>我们每个人都希望身体健康，虽然疾病可以预防，但不可避免，没有任何人可以说，生下来到现在没有生过一次病。</p>	
8.11.1 开放定址法	361
8.11.2 再散列函数法	363
8.11.3 链地址法	363
8.11.4 公共溢出区法	364
8.12 散列表查找实现	365
8.12.1 散列表查找算法实现	365
8.12.2 散列表查找性能分析	367
8.13 总结回顾	368
8.14 结尾语	369

如果我是个喜欢汽车的人，时常搜汽车信息。那么当我在搜索框中输入“甲壳虫”、“美洲虎”等关键词时，不要让动物和人物成为搜索的头条。

第 9 章 排序373



9.1 开场白 374

假如我想买一台 iphone4 的手机，于是上了某电子商务网站去搜索。可搜索后发现，有 8863 个相关的物品，如此之多，这叫我如何选择。我其实是想买便宜一点的，但是又怕遇到骗子，想找信誉好的商家，如何做？

9.2 排序的基本概念与分类 375

比如我们某些大学为了选拔在主科上更优秀的学生，要求对所有学生的所有科目总分倒序排名，并且在同样总分的情况下将语数外总分做倒序排名。这就是对总分和语数外总分两个次关键字的组合排序。

- 9.2.1 排序的稳定性376
- 9.2.2 内排序与外排序377
- 9.2.3 排序用到的结构与函数378

9.3 冒泡排序 378

无论你学习哪种编程语言，在学到循环和数组时，通常都会介绍一种排序算法，而这个算法一般就是冒泡排序。并不是它的名称很好听，而是说这个算法的思路最简单，最容易理解。

- 9.3.1 最简单排序实现379
- 9.3.2 冒泡排序算法380
- 9.3.3 冒泡排序优化382
- 9.3.4 冒泡排序复杂度分析383

9.4 简单选择排序 384

还有一种做股票的人，他们很少出手，只是在不断观察和判断，等时机一到，果断买进或卖出。他们因为冷静和沉着，以及交易的次数少，而最终收益颇丰。

- 9.4.1 简单选择排序算法384
- 9.4.2 简单选择排序复杂度分析385

9.5 直接插入排序 386

哪怕你是第一次玩扑克牌，只要认识这些数字，理牌的方法都是不用教的。将 3 和 4 移动到 5 的左侧，再将 2 移动到最左侧，顺序就算是理好了。这里，我们的理牌方法，就是直接插入排序法。

- 9.5.1 直接插入排序算法386
- 9.5.2 直接插入排序复杂度分析388

9.6	希尔排序.....	389
<p>不管怎么说，希尔排序算法的发明，使得我们终于突破了慢速排序的时代（超越了时间复杂度为$O(n^3)$），之后，更为高效的排序算法也就相继出现了。</p>		
9.6.1	希尔排序原理.....	391
9.6.2	希尔排序算法.....	391
9.6.3	希尔排序复杂度分析.....	395
9.7	堆排序.....	396
<p>什么叫堆结构呢？回忆一下我们小时候，特别是男同学，基本都玩过叠罗汉的恶作剧。通常都是先把某个要整的人按倒在地，然后大家就一拥而上扑了上去……后果？后果当然就是一笑了之。</p>		
9.7.1	堆排序算法.....	398
9.7.2	堆排序复杂度分析.....	405
9.8	归并排序.....	406
<p>即使你是你们班级第一、甚至年级第一名，如果你没有上分数线，则说明你的成绩排不到全省前1万名，你也基本失去了当年上本科的机会了。</p>		
9.8.1	归并排序算法.....	407
9.8.2	归并排序复杂度分析.....	413
9.8.3	非递归实现归并排序.....	413
9.9	快速排序.....	417
<p>终于我们的高手要登场了，将来你工作后，你的老板让你写个排序算法，而你会的算法中竟然没有快速排序，我想你还是不要声张，偷偷去把快速排序算法找来敲进电脑，这样至少你不至于被大伙儿取笑。</p>		
9.9.1	快速排序算法.....	417
9.9.2	快速排序复杂度分析.....	421
9.9.3	快速排序优化.....	422
9.10	总结回顾.....	428
<p>目前还没有十全十美的排序算法，有优点就会有缺点，即使是快速排序法，也只是在整体性能上优越，它也存在排序不稳定、需要大量辅助空间、对少量数据排序无优势等不足。</p>		
9.11	结尾语.....	430
<p>如果你有梦想的话，就要去捍卫它。当别人做不到的时候，他们就想要告诉你，你也不能。如果你想要些什么，就得去努力争取。就这样！</p>		
附录	参考文献.....	435