**读书笔记**

在网上搜说了下关于算法的书籍，发现很多人推荐了《算法概论》这本书，我抱着试一试的态度看了看这本书，这本书很好，内容很全，涉及数据结构和数学相关知识较多，下面谈谈我的阅读笔记。。

第0章

本章较为简短，没有深入系统地涉及某些内容。主要以fibonacci数列的例子，让我体会了递归和递推思想的差别。针对fibonacci数列例子直接递归解法中涉及的重复计算，优化出递推方式，展示了思考问题中自顶向下与自底向上的不同思考角度可能产生较大的算法效率差别，同时隐约体现记忆化搜索的思想。另外本章较为详细介绍了大o复杂度度量标准。

第1章

本章以rsa算法为例，细致深入讨论了rsa算法涉及的相关数论知识，诸如取模运算、模下的四则运算与逆元概念、取模幂运算、素性检测。

在素性检测部分有经典的欧几里德算法、扩展欧几里德算法，同时引入随机化算法概念，以极高的概率保证素性检测有效性。

通过本章的学习，我对过去不曾深入考虑或者说真正考虑的基础性运算有了更深的理解。之前对乘除运算复杂度总是在以单元操作的概念下以o（1）带过，以后会更加细致地考虑乘除等基本运算的复杂度。另外，本章以rsa为案例，系统地展示了针对某一问题，如何从基础性知识入手，一步一步学习案例所需基础知识，并将其整合从而解决案例。 素性检测与素因子分解，两个看似相去不远的问题，其复杂性天差地别的现实，从一般角度让人们想到的是类似问题的解决难度可能差别很大仅此而已，而rsa算法展示了如何深入的多想一步，利用这种情况设计出优雅的解决方案。这思想很值得我借鉴与利用。

第2章

本章介绍分治算法思想，提及分治，相信每一个学习算法的人都不会陌生，经典的《算法导论》中就已合并排序为例在开篇不久就引入分治概念。本书介绍分治的角度与众不同，不似《导论》中总是介绍比较显而易见的可以分治的案例。本书列举了矩阵相乘、快速傅立叶变换等数学领域分治的应用案例，在这些案例之中，分治的应用很多情况下隐藏的较为深，并非显而易见，加大了分析难度。但是更能让我感受到分治应用之广泛，可能在学习本章之前，许多类型的题目我不会想到去向分治的角度思考，因为不易看出，但是本章给我的备忘录上加了一条：永远不要忽视分治，针对陌生题目，不要轻易就否决掉往分治角度思考的路线。另外，通过本章学习，对于算法复杂度的评估以及根据递推式评估复杂度的能力有了很大的提高。

第3章

学习到本章时，发现本章讲解部分只有15页，算上习题也不过20余页，大致翻看内容，发现讲解的是dfs，便松了一口气，自认为作者真逗，一个dfs也用得着单独分出一章来叙述？岂不知市面上的绝大多数算法书，就是将dfs作为搜索或图、树遍历部分的一小节叙述。可是通过两遍的学习，总算体会到作者的用心良苦及自己过去对dfs认识的肤浅。

另外细节部分，拓扑排序和有向图的强连通分量分解思想的相似性研究，值得好好品味。 做练习题过程中，能体会到如果图模型建立好，我能够反应到dfs针对问题的应用，但是关键难点在于根据题目描述如何联想到图模型，但是这不是说看书能够看会的，看来只有多做题慢慢培养这种关联性思维了。

第4章

本章内容与上一章承接。以bfs为媒介，引出了图论中求解顶点的最短距离相关的一系列算法，诸如dijikstra算法、bellman-ford算法等。由上一章我们知道，dfs的应用一般在于连通分量、结合先、后序操作的算法设计。而bfs的应用一般集中于求解最优化或最短距离方面。

在做本章练习题过程中，我更加体会到为什么自己之前看的算法书不少，而提高却总是很慢的原因。光看书确实是不够的，每一本算法书都配以大量的习题确实是十分必要的。也许对于一本算法书，你看了一遍两遍甚至三遍，对于每一章的内容以及例题都已了然，但是没有经过大量题目的思考解答过程，根本谈不上掌握。如何算作掌握了某一算法？许多人会以掌握其设计思想为由搪塞过去，对于算法的细节往往忽略不谈。自己过去也总是效仿这一种做法，仿佛抠细节是愚蠢之人的做法，其实不然。我当然不赞成一味深入细节，但是我们应当知道算法的某一步骤为何这么设计（这往往是显然的），比如在dijikstra中，当扩展到新的一个节点v，如果有dist[u] dist[v]+l(v,u)时，要更新u的距离，一般人都不会不懂这个操作的原理。但是我们的思考往往也在这一步停止了。在做书中题目时，我发现有一类题目，即到某一点的最短距离路径不唯一时，如何确定？思考了很久，忽然恍然大悟，这不就是dijikstra算法中进行dist[u]和dist[v]+l(v,u)过程中，出现dist[u] = dist[v]+l(v,u)的情况么？单单是对于一个比较符号的深入思考，我们便有了新的收获，同时可以将原算法的应用领域扩展一步。如果没有针对题目的思考，又怎会对算法中一个比较符号的进行分析？又怎会真正体会一个算法的精巧。

bfs作为可获得最优解的一种暴力搜索算法，可以用于状态空间搜索，在这一类应用之中，关键在于状态节点数据结构的设计，以及分析清楚下一步状态节点扩展所依赖的操作，分析清楚这两点之后，便可以以bfs实现求解。

另外，本章算法的应用领域的抽象建模过程较之第3章dfs部分较为简单明了。同时应用的灵活性自然也不如dfs。至此经典的暴力搜索dfs、bfs部分已经结束。

第5章

本章重点介绍贪心算法。贪心算法并非某一特定的算法，而是一类算法或者说是一种算法设计思路。针对某一类满足贪心算法适用的问题背景，我们可以通过每一次都选择当前最优的策略获得最优解。当然，算法的难度并不在于算法实现，而在于对于贪心算法是否适用于某一问题的证明，这也是唯一的难点之一。

本章重点介绍了贪心算法的经典范例最小生成树算法（kruskal与prim），以及huffman编码。另外，引入了数据结构并查集的介绍。内容较为容易理解，习题难度也不大。

第6章

本章内容为动态规划。动态规划作为经典的一类算法设计策略，一直以来都是各算法书籍的重头戏。类似于贪心算法，动态规划并不是某一种特定的算法，而是一种设计策略。在《算法导论》中，作者以多步决策引入了动态规划概念，同时指出动态规划适用的情况是问题同时具有最优子结构和重叠子问题的情况。而在《算法概论》一书中，作者并没有采用这种传统的介绍方式。本书采用了一种结构上的抽象，针对动态规划问题的状态对应于节点，而选择转换对应为边，将动态规划抽象为dag（有向无环图），从而结合求解最短路径思想描述了动态规划。

动态规划的一般实现形式：记忆化搜索（自顶向下）、递推式自底向上。

本章主要范例为lis、lcs、背包（单副本、多副本）、矩阵相乘、最短路及tsp以及独立集。类似之前的章节，在习题中设置了许多范例的变种问题，通过完成习题使我对这些范例的理解更为深刻。总而言之，动态规划题目千变万化，唯有大量练习培养思维敏感性。

第7章

本章介绍线性规划。由于之前已经学习过线性规划相关专著，所以这部分过得比较快。总而言之，这部分内容具有理论上的意义，并且做为数学规划其他内容时必须掌握的。但是，事实上，实际问题中建模后，很难出现这种简单的线性规划模式。所以这一章算是数学规划的一个引言。

第8章

本章介绍np-完全问题。主要要明确以下概念：能够在多项式时间判断某一个解答是否是原问题的正确解，则是np问题；而在np问题中，若还能在多项式时间内求解出解，则是p问题；若在np问题中，若不确定能否在多项式时间内求出原问题的解，则是np-完全问题。换言之，np问题包含p问题与np-完全问题。所以，许多人不求严谨，老是说np问题与p问题求解难度不同，实则是想说np-完全问题与p问题求解难度不同。另外需要明确，所有的np-完全问题都可以规约为同一个问题。

第9章

本章承接上一章，针对np-完全问题的难度，提出了一系列不同的解决策略。主要归结为以下几种：智能化搜索（剪枝、分支定界）、近似算法（退而求其次，不要求一定求得最优解）、局部搜索中的启发式方法（涉及进化算法和模拟退火）。本章算是起到抛砖引玉的作用，如何求解np-完全问题一直是研究的热点，由最初的启发式搜索，包括书中提及的剪枝、分支定界、以及后来的a\*算法，到后来逐步发展的进化算法，虽然一直没有冲破np-完全与p的界限，但是从不同的思考角度都为我们提供了不少在实践中具有实际应用意义的解决方法。正如书中所说，判定一个问题为np-完全问题并不是宣判了该问题的死刑。在np-完全问题的诸多风格的求解方式中，我们更能体会到算法设计领域的博大精深。

第10章

本章讲解量子算法，虽然理解不深，但是本章着实让我大开眼界。