《算法设计与分析》复习

一、章节要求

第 1 章 1.1（重点）、1.2（了解）

第 2 章 2.1（熟悉循环不变式，其它了解）、2.2（了解）、2.3（了解）

第 3 章 3.1（重点）、3.2（了解）

第 4 章 4.1（重点）、4.2（熟悉）、4.3（重点）、4.4（重点）、4.5（重点）、4.6（不要求）

第 5 章 掌握概率分析、随机算法的基本概念，指示器随机变量的定义和使用。其他内容只要求一般了解

第6、7章 排序算法本身只要求一般了解，但要能够熟练使用这些排序算法，要了解不同排序算法的性质，在不同的场景下选择合适的算法解决问题

第 8 章 8.1（熟悉）、8.2~8.4（了解）

第 9 章 9.1（掌握顺序统计量、中位数的基本概念，其它内容一般了解），9.2~9.3（熟悉选择算法的设计思想，要会用）

第15章 15.1（熟悉）、15.2（重点）、15.3（熟悉）、15.4（重点）、15.5（重点）

第16章 16.1（重点）、16.2（重点）、16.3（重点，包括最优归并模式问题）、16.4~16.5（不要求）

第22章 22.1（熟悉）、22.2（熟悉）、22.3（熟悉）、22.4（不要求）、22.5（不要求）

回溯（重点）、分支限界（重点）

第23章 23.1（熟悉）、23.2（熟悉）

第24章 24.1（重点）、24.2（不要求）、24.3（重点）、24.4（重点）、24.5（熟悉）

第25章 25.1（了解）、25.2（重点）、25.3（熟悉）

第26章 26.1（重点）、26.2（重点）、26.3（熟悉）、26.4（不要求）、26.5（不要求）

第34章（了解）

“重点”：占比大，会出独立题目或综合题目；“熟悉”：综合性题目里会涉及到相关内容；

“了解”：考察涉及的内容很少； “不要求”：不在本次考察范围内

二、理论部分

1. 算法的基本概念和性质。

算法就是任何良定义的计算过程，该过程取某个值或值的集合作为输入，并产生某个值或值的集合作为输出；算法是一组有穷的规则，它规定了解决某一特定类型问题的一系列计算。

五个性质：确定性、能行性、输入、输出、有穷性

1. 渐近表示：O、Ω、Θ记号的定义和相关性质及其证明。
2. 什么是循环不变关系？用循环不变关系证明循环的正确性。

在第一次进入循环之前成立、每次循环之后还成立的关系。

分为三步：初始化（证明在初始状态时循环不变式成立，即在循环开始之前为真），保持（在每次循环之后、下一次循环开始之前循环不变式仍为真），终止（证明循环可以在有限次循环之后终止）

1. 分治法的基本思想。

在问题规模比较大而无法直接求解时，将原始问题分解为几个规模较小、性质与原始问题一样的子问题，然后递归地求解这些子问题，最后合并子问题的解以得到原始问题的解。

分为：分解、解决、合并三步骤。

1. 以比较为基础的检索和分类算法的时间下界及其证明，熟练掌握其中几个典型算法。
2. 为什么我们通常更关心算法的最坏情况执行时间？

一个算法的最坏情况执行时间给出了任何输入的运行时间的一个上界，知道了这个界，就能确保算法绝不需要更长的时间，我们就不需要再对算法做更坏的打算，并且最坏情况经常出现。

1. 用代换法、递归树法解递推式的基本思想

代换法：先猜测解的形式，然后用数学归纳法验证猜测的正确性。

递归树法：根据递归式的定义，画一棵递归树，来帮助我们猜测递归式的解。

1. 主方法及其使用。
2. 一个算法的平均情况运行时间和期望运行时间分别指什么？
3. 了解期望时间和最坏情况时间是线性时间的选择算法的基本思想。
4. 最优化问题是一类什么问题？

这一类问题的可行解可能有很多个，每个解都有一个评价优劣的值，我们希望寻找具有最优值的解，这种找最优解的问题通常就叫作最优化问题。

1. 什么是最优子结构性、无后效性？用剪切-粘贴法证明一个问题满足最优子结构性。

最优子结构性：无论过程的初始状态和初始决策是什么，其余的决策都必须相对于初始决策所产生的状态构成一个最优决策序列（一个最优策略的子策略应该是最优的）

无后效性：如果在某个阶段上过程的状态已知，则从此阶段以后过程的发展变化仅与此阶段的状态有关，而与过程在此阶段以前的阶段所经历的状态无关。

剪切-粘贴法：反证法，假定原问题最优解中对应于某个子问题的最优解不是该子问题的最优解，而存在“更优的子解”，那么我们可以从原问题的解中剪切掉一部分，而将更优的子解粘贴进去，从而得到比最优解“更优”的解，这就出现了矛盾。

1. 什么是状态转移方程？

状态转移方程，是动态规划过程中本阶段的状态往往是上一阶段状态和上一阶段决策的结果，如果给定了第k阶段的状态Sk和决策uk，那么第k+1阶段的状态Sk+1也就完全确定。

1. 子问题图的画法。
2. 简述对动态规划所能带来计算性能改进的理解

一方面，动态规划仅仅保存了所有子问题的最优解，而舍去了所有不能导致问题最优解的次优决策序列，因此可能有多项式的计算复杂度；另一方面，动态规划通过仔细安排求解顺序，避免对子问题的解的重复计算，提高了效率。有两种等价的实现方法，分别是：带备忘的自顶向下法和自底向上法。

1. 贪心算法的基本思想和一般步骤。什么是贪心选择性和贪心选择？

贪心算法的思想与步骤：分步骤实施，在每一步做出局部最优的选择，并希望通过这样的选择最终能找到全局最优解。

贪心选择性：可以通过做出局部最优选择来构造全局最优解的性质。

贪心选择：每一步所做的局部最优选择。

1. 比较动态规划和贪心方法的异同。

动态规划每一步要进行一次选择，但这种选择通常依赖于子问题的解，而贪心不依赖任何将来的选择或子问题的解。动态规划通常自底向上完成计算，而贪心算法通常是自顶向下的。

1. 什么叫切割、横跨切割、轻量级边、安全边？

切割：无向图G=(V,E)的一个切割(S,V-S)是结点集V的一个划分

横跨切割：一条边(u,v)∈E的一个端点在集合S中，另一个端点在集合V-S中

轻量级边：在横跨一个切割的所有边中，权重最小的边

安全边：一条不违反循环不变式的边(u,v)加入集合A，{A∪{(u,v)}仍是某棵最小生成树

的子集

1. 什么是松弛操作？

测试是否可以对u和v两点之间的距离进行改善，如果可以改善，就将v.d更新为新的最短路径估计值，且将v的前驱v.pi更新为新的前驱结点。

1. **最短路和松弛操作的相关性质：三角不等式性质、上界性质、路径松弛性质及其它。**
2. 举例说明在带有负权重边的图上Dijkstra算法工作异常。
3. Bellman-Ford算法是如何检查图中可能存在的负权重回路的？

如果不存在回路，算法返回TRUE，且对于所有节点，前驱子图是一个根节点为s的最短路径树；

存在返回False

1. 生成树：宽度优先生成树、深度优先生成树、最小成本生成树、单源点最短路径生成树，了解这些生成树的生成过程和相关性质。
2. 什么是回溯法、分支-限界法？限界函数的作用是什么？
3. 有关状态空间、解空间、状态空间树及检索和周游的基本概念。
4. 简述LC-检索的基本思想。
5. 什么是结点成本函数和结点成本估计函数？结点成本估计函数(X)中h函数和函数会分别对算法带来什么影响？
6. 了解函数的性质，了解C(X)上界的作用。
7. 什么是流网络、最大流？
8. Ford-Fulkerson方法：残存网络、增广路径、最大流最小切割定理。

增广路径：残存网络中一条从源节点s到汇点t的简单路径

1. Edmonds-Karp算法的思路

使用广度优先搜索寻找源节点到汇点的最短路径作为增广路径

三、算法部分

1.算法设计的基本策略

1）增量式算法设计策略

2）分治策略

3）贪心策略

4）动态规划

5）回溯

6）分支限界

2.问题和算法

1）排序算法：插入排序、归并排序、快速排序、堆排序、计数排序

（只要求了解算法性质后会用）

2）分治相关：最大子数组问题、Strassen矩阵乘法、最近点对问题、期望为线性时间的选择算法、最坏是Ο(n)的选择算法

3）概率分析和随机算法：雇佣问题

4）查找问题：二分查找、找最大最小元素、找最大次大元素

5）中位数相关：石油管的最优位置问题、一维邮局问题、二维邮局问题

6）动态规划相关：钢条切割问题、矩阵链乘法问题、LCS问题、最优二叉搜索树问题、0-1背包问题、Bellman-ford算法、Floyd-Warshall算法。

7）贪心策略相关：活动选择问题、分数背包问题、Huffman编码问题、最优归并模式问题、最小生成树问题、Dijkstra算法。

8）最小生成树：MST性质、构造最小生成树的贪心策略、Prim算法、Kruskal算法。

9）最短路相关：单源最短路径问题（Bellman-ford算法、Dijkstra算法）、每对结点间最短路径问题（Floyd-Warshall算法、Johnson算法），应用扩展：差分约束系统

10）深度优先搜索：DFS、回溯算法，n-皇后问题、子集和数问题。

11) 宽度优先搜索：BFS、分支限界算法、A\*算法，15-谜问题、带有限期的作业排序问题

12）网络流：流网络、最大流，Ford-Fulkerson方法、Edmonds-Karp算法

以上红色标识的问题和算法为一般性了解，不做要求。

四、复习范围

1.**要求为“重点”和“熟悉”的章节要重点复习，**“了解”的章节有时间可以看一下；“不要求”的章节不在本次考察范围内。

对要求为“重点”和“熟悉”的章节，可以**参考上面二、三中的列出的问题和要点有针对性地复习**。

2.**书上或课件中的例题和作业**

“重点”和“熟悉”章节涉及的例题和作业题目应重点复习。做到：

计算题：能够熟练计算，换个算例也会算

证明题：掌握证明策略和证明过程，类似问题可以自行证明

设计题：能够用伪代码描述算法，算法描述要完整、正确、准确

五、作业清单

第一次作业：

抄写算法（熟悉伪代码的写法，不要求）

1.2-2

1.2-3

3.1-5（Ω、Ο、Θ的性质证明）

第二次作业：

2-4（逆序对）

4.1-5（最大子序列和）

4.3-2（证明lgn）

4.3-9（代换法）

4.4-6（递归树法）

4.5-1（主方法）

4.5-4（不能用主方法）

第三次作业：

9.1-1（第一、第二小元素）

9.3-1（讨论select的线性特点，不要求）

9.3-5（Select算法的应用）

9.3-9（Olay教授，不要求）

9.2（中位数，不要求）

分金币（不要求）

第四次作业：

DP： 计算：15.2-1（矩阵链乘法）

15.4-1（LCS）

15.5-2（最优二分搜索树）

算法：15.1-3（钢条切割）

15.9（切割字符串）

15.2-5

15.3-6

贪心：16.1-4（最少教室）

16.2-6

16.2-7（指数乘积）、

16.3-3（Huffman）

16.1（找零问题）

补充：分数背包求解

第五次作业：

计算：

24.1-1（Bellman-Ford）

24.4-1（差分约束）

25.2-1（Floyd-Warshell）

算法：24.1-3（Bellman-Ford）

24.3

25.2-6（Floyd-Warshell）

25.2-7（Floyd-Warshell）

第七次作业：

（1）算法：作业分配问题

（2）计算：子集和数问题

第八次作业（网络流）：

26.1-1

26.2-3

26.3-1