算法复习笔记

算法基本概念

递归与分治

动态规划

贪心算法

随机算法

回溯法

分枝限界法

NP完全性

近似算法

1. 算法的基本概念：

1、定义：一般认为，算法是由若干条指令组成的又穷序列，具有以下五个特性：确定性、可行性、又穷性、输入、输出。

2、如何衡量算法的好坏：最初，用所需要的计算时间来衡量一个算法的好坏但不同的机器相互之间无法比较需要用独立于具体计算机的客观衡量标准：问题规模、基本运算、算法的计算量函数。

问题规模：一个或多个整数，作为输入数据量的测度

基本运算：解决给定问题时占支配地位的运算

算法的计算量函数：用问题规模的某个函数来表示算法的基本运算量,这个表示基本运算量的函数称为算法的时间复杂性（度）

渐进时间复杂度：当问题的规模趋于极限情形时（相当大）的时间复杂度

3、描述算法的时间复杂度：用问题规模的某个函数来表示算法的基本运算量,这个表示基本运算量的函数称为算法的时间复杂性（度）：

算法的时间复杂度概念：O ΩΘ描述；

分类：最坏时间复杂度：规模为n的所有输入中，基本运算执行次数为最多的时间复杂度

平均时间复杂度：规模为n的所有输入的算法时间复杂度的平均值

4、如何评价算法：①正确性：评价算法的首要因素：程序正确性证明、程序测试

②健壮性 算法/程序不仅对正确的输入要能计算出正确的结果，对不正确的输入也要能够应对处理

③简单性 算法/程序的可读性好，易调试、改进

④高效性 时间、空间复杂度较小，特别是时间复杂度

⑤最优性 证明所给算法是解决同一类问题中最好的

二、分治与递归。

什么是分治法：

分解、解决、合并

分治法与平衡的概念

平衡：使子问题规模尽量接近的做法，就是平衡

在使用分治法和递归时，要尽量把问题分成规模相等，或至少是规模相近的子问题以提高算法的效率

分治与递归

递归式的解法

当一个算法包含对自身的递归调用时，其运行时间通常可以用递归式来表示

1. 代换法：利用经验进行猜测，主要解决一些容易看出来的问题
2. 递归树方法：每一个节点代表递归函数调用集合中一个子问题的代价，将所有层的代价相加得到总代价
3. 主方法：主定理

分治法的适用条件

1. 问题的规模缩小到一定程度就可以容易地解决
2. 问题可以分解为若干个规模较小的相同问题，即该问题具有最优子结构性质
3. 基于子问题的解可以合并为原问题的解
4. 问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子问题
   1. 动态规划

与分治法类似，也是将问题分解为规模逐渐减小的同类型的子问题

与分治法不同，分解所得的子问题很多都是重复的

动态规划方法的适用范围

适合用动态规划方法求解的问题

若一个问题可以分解为若干个高度重复的子问题，且问题也具有最优子结构性质，就可以用动态规划法

求解具体方式：可以递推的方式逐层计算最优值并记录必要的信息，最后根据记录的信息构造最优解

动态规划方法总体思想

保存已解决的子问题的答案，在需要时使用，从而避免大量重复计算

动态规划方法求解实例

矩阵连乘问题

LCS

最大子段和

……

四、贪心算法

贪心算法的基本思想

贪心算法总是作出在当前看来是最好的选择

贪心算法并不从整体最优上加以考虑，它所作出的选择只是在某种意义上的局部最优选择

贪心算法不能对所有问题都得到整体最优解，但对许多问题它能产生整体最优解

在一些情况下，即使贪心算法不能得到整体最优解，其最终结果却是最优解的很好近似

贪心算法的基本要素

可以根据下列步骤设计贪心算法

将最优化问题转化为这样的一个问题，即先做出选择，再解决剩下的一个子问题

证明原问题总有一个最优解是做贪心选择得到的，从而说明贪心选择的安全

说明在做出贪心选择之后，子问题的最优解与所作出的贪心选择联合起来，可以得出原问题的一个最

优解

贪心算法求解实例

活动安排问题

单源最短路径

最小生成树

……

* 1. 随机算法

随机算法的分类

Las Vegas和Monte Carlo

Las Vegas和Monte Carlo算法的区别

Monte Carlo 算法一定可以获得解，但是这个解不一定是正确的最优解；

Las Vegas 算法不一定可以获得解，但是如果获得解了，那么一定是正确的最优解。

Sherwood算法

随机算法求解实例

快速排序随机化版本

求第k小元素

Testing String Equality

Pattern Matching

主元素问题

六、回溯法与分枝限界法

回溯法（DFS）与分枝限界法（BFS）的基本概念

回溯法的基本做法是搜索，它是一种可以避免不必要搜索的穷举式搜索法

回溯法适用于求解一些组合数较大的问题

如果肯定不包含，则跳过对该结点为根的子树的搜索，逐层向其祖先结点回溯

否则，进入该子树，继续按深度优先策略搜索

具有限界函数的深度优先生成法称为回溯法

不同点

求解目标：回溯法的求解目标是找出解空间树中满足约束条件的所有解，而分枝限界法的求解目标则是找出满足约束条件的一个解，或是在满足约束条

件的解中找出在某种意义下的最优解

搜索方式的不同：回溯法以深度优先的方式搜索解空间树，而分枝限界法以广度优先或以最小耗费优先的方式搜索解空间树

分枝限界法常以广度优先或以最小耗费（最大效益）优先的方式搜索问题的解空间树

回溯法求解时常见的两类解空间树

子集树：当所给问题是从n个元素的集合S中找出S满足某种性质的子集时，相应的解空间树称为子集树

排列树：当所给问题是确定n个元素满足某种性质的排列时，相应的解空间树称为排列树

求解实例

0-1背包

TSP

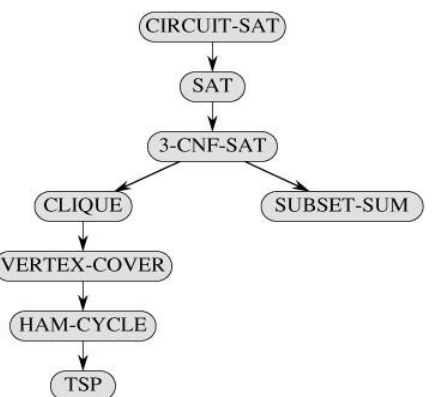
……

七、NP完全性

判定问题

P、NP、NPC、NP Hard的定义及相互关系

NPC问题实例



八、近似算法

近似算法的分类

常数近似比的近似算法

多项式时间近似方案

完全多项式时间近似方案

近似算法的性能

近似算法一般都比较简单，但设计近似算法时必须关注所设计的算法所得到的近似解与最优解之间的差距到底有多大

若一个最优化问题的最优值为c\*，求解该问题的一个近似算法求得的近似最优解相应的目标函数值为c，则将该近似算法的近似比定义为max{c\*/c, c/ c\*}