算法设计与分析 课程总结

算法课程的知识框架

分 治 策 歌 秘 规 規

贪心法

回溯 与分限 界 算法的数学基础: 序列求和、递推方程求解

算法的基本概念: 算法的伪码表示 算法的两种时间复杂度 复杂度函数阶的表示

函数的阶

- 阶的符号: $O, \Omega, \Theta, o, \omega$
- 阶的高低

至少指数级: 2^n , 3^n , n!, ...

多项式级: $n, n^2, n \log n, n^{1/2}, ...$

logn的多项式级: logn, log^2n ,...

注意

阶反映的是大的 $n(n>n_0)$ 的情况可以忽略前面的有限项

序列求和

- 基本求和公式等比数列等 差数列调和数级数
- 估计和式的阶 放大,然后估计上界 用积分估计上下界

递推方程求解

- 主要的求解方法迭代+进行序列求和递归树+求和主定理:注意条件验证
- 一些常见的递推方程的解

$$f(n) = af(\frac{n}{b}) + d(n)$$

算法设计技术

- 设计技术 分治策略 动态规划 贪心法 回溯和分支限界
- 关注问题 使用条件 主要设计步骤 时间复杂度分析方法 改进途径 典型例子

分治策略

- 适用条件: 归约为独立求解子问题
- 设计步骤: 归约方法, 初始子问题的计算, 子问题解的综合方法. 注意子问题划分均衡, 类型相同
- 递归算法分析: 求解递推方程
- 改进途径:减少子问题数,预处理
- 典型问题: 二分检索,归并排序,芯片 测试,幂乘,矩阵乘法,最临近点对,多 项式求值

动态规划

- 适用条件: 优化问题, 多步判断求解, 满足优化原则, 子问题重叠
- 设计步骤:确定子问题边界,列关于目标函数的递推方程及初值;自底向上,备忘录存储;标记函数及解的追踪方法
- 复杂度分析: 备忘录, 递推方程
- 典型问题:矩阵链相乘,投资,背包,最 长公共子序列,图像压缩,最大子段和, 最优二分检索树,生物信息学应用

贪心法

- 适用条件:组合优化问题,多步判断求解,有贪心选择性质
- 设计步骤:局部优化策略的确定及 算法正确性证明(直接证明,数学归 纳法,交换论证)
- 复杂度分析
- 典型问题:活动选择,装载问题,最小 延迟调度,最优前缀码,最小生成树, 单源最短路

回溯和分支限界

- 适用条件: 搜索或优化问题, 多步 判断求解, 满足多米诺性质
- 设计步骤:确定解向量,搜索树结构, 搜索顺序,结点分支搜索的约束条件 与代价函数,路径存储
- 搜索树结点数估计
- 复杂度分析
- 典型问题: n后问题, 背包问题, 货郎问题, 装载问题, 最大团问题, 圆排列问题, 连续邮资问题

算法设计

- 设计思想:尽量选复杂度低的算法
- 算法实现依赖于数据结构,选择合适的数据结构
- 实际问题中的综合考虑: 时空权衡, 实现成本的权衡,...