

算法的起源、发展与学习算法的意义

算法分析: 时、空复杂度为指标定量分析算法优劣

- 算法的伪码表示
- 最优、最坏、平均复杂度 (以排序算法为例)
- 函数的渐进复杂度
- 函数阶的计算: 递归方程求解技术 (迭代、递归树、主成分定理)
- 常见的函数阶

## 课程回顾 (2/4): 算法设计技术

### 分治

- 典型问题: 排序选择类、计算几何类、数值计算类 (快速幂、整数乘、矩阵乘、多项式乘)
- 设计: 子问题的划分与子问题解的合并
- 分析: 递归方程求解

### 贪心

- 典型问题 (最优子结构): 最优调度、最优装载、Huffman Coding、单源最短路径、最小生成树
- 设计: 选择贪心策略 (试错: 构造反例说明贪心策略非最优)
- 证明: 数学归纳法证明贪心解是最优解 (对问题的规模或算法的步骤进行归纳)

### 动态规划

- 典型问题 (最优子结构 + 重叠子问题): 字符串类 (最长递增子序列、最大公共子序列、最小编辑距离)、矩阵链、最大子段和、背包类
- 设计: 找出 DAG, 列出递推方程, 设计备忘录和标记函数
- 分析: 根据递推方程或者备忘录分析时空复杂度

搜索技术: 求解优化类问题的通用技术 (将解空间建模为树)

- 暴力搜索 (找到一个解后再利用约束条件判断)
- 回溯 (Domino 性质: 根据优化类问题的约束条件分析部分解, 若不满足则及时停止局部搜索并回溯)
- 分支定界 (引入界和估计函数, 在约束条件的基础上加入更精细的判断)

### 复杂性理论

- 计算模型: Turing Machine (deterministic poly-time and non-deterministic poly-time)
- 重要的复杂性类:  $\mathcal{P}$ ,  $\mathcal{NP}$ ,  $\mathcal{NP}$ -hard,  $\mathcal{NP}$ -complete
- 更多的复杂性类:  $BPP$  (降低错误的方法和为什么能降低、衍生复杂性类、与其他复杂性类的关系)

### 随机算法初步

- 概率论初步: 期望线性性质的应用
- Miller-Rabin 素性检测算法
- Schwartz-Zippel 引理及其应用: 多项式恒等测试、矩阵乘积结果测试

## 考试题型与范围

选择题 (送分题): 约 10 分

- 考察渐进复杂度、经典算法和复杂性以及各类设计范式的基础知识

简答题 (又是送分题): 约 36 分

- 渐进复杂度/递归方程求解
- 复杂性类的基本概念
- 程序理解与分析

算法设计与分析题 (还是送分题): 约 54 分

- 分治、贪心、动态规划

评判标准: 根据算法的正确性和效率评分.

解答需包含以下四部分: (1) 写明算法设计思想; (2) 分治算法需写出递归方程, 贪心算法需给出正确性证明, 动态规划算法需写明递推方程和标记函数; (3) 伪代码 (动态规划需包含求解算法与追踪解算法); (4) 时间、空间复杂度分析

祝万能的青年们  
无所不能，锦绣前程！