

1 算法引论

1.1 什么是算法

算法是一系列解决问题的清晰指令

算法是满足下述性质的指令序列：有限性、确定性、输入、输出

1.2 算法的设计

- 问题建模：对输入和解给出形式化描述
- 设计算法：确保正确
- 分析算法：效率

算法复杂性分析：需要的计算机资源量。时间、空间

依赖与：算法本身，数据规模，算法输入

符号： O 上界， Ω 下界， Θ 同时上下界。

2 分治

将一个难以直接解决的规模较大的问题分解为若干个规模较小的子问题，这些子问题互相独立且与原问题相同。递归地解决这些子问题。然后将各个子问题的解合并得到原问题的解。

- 如果规模小，直接解决
- 将 P 划分为 P_1, P_2, \dots, P_k
- for $i \leftarrow 1$ to k
 - $y_i = \text{递归}(P_i)$
- $\text{return Merge}(y_1, y_2, \dots, y_k)$

时间： $T(n) = \begin{cases} \Theta(1) & n \leq c \\ aT(n/b) + D(n) + C(n) & n > c \end{cases}$ 求解：换名法：

令 $n = 2^k$, $g(k) = T(2^k) = T(n) = \begin{cases} 1 & k = 0 \\ 2g(k-1) + 2^k & k > 0 \end{cases}$

主方法：比较 $D(n) + C(n)$ 与 $n^{\log_b a}$ 。

$T(n) = \begin{cases} \Theta(n^{\log_b a}) & n^{\log_b a} > D(n) + C(n) \\ \Theta(n^{\log_b a} \lg n) & n^{\log_b a} = D(n) + C(n) \\ \Theta(D(n) + C(n)) & n^{\log_b a} < D(n) + C(n) \end{cases}$

归并排序、二分检索、芯片测试、多数元素、快速排序、线性时间选择

3 贪心算法

组合优化问题：在某些约束条件下，决定某些**可选择的变量**应该取何值，使所选定的**目标函数达到最优**的问题。

通过对数学方法的研究去寻找离散事件的最优编排、分组、次序或筛选等

组合优化问题的数学模型

$$\begin{aligned} \min f(x) \\ s.t. \quad g(x) \geq 0 \\ x \in \mathcal{D} \end{aligned}$$

$f(x)$ 目标函数， $g(x)$ 约束函数， x 为决策变量， \mathcal{D} 表示有限个点组成的集合

总是做出在当前看来最好的选择，即贪心算法并不是从整体最优考虑，它所做的选择只是在某种意义上的局部最优选择

贪心算法则通常以自顶向下的方式进行，以迭代的方式作出相继的贪心选择，每做一次贪心选择就将所求问题简化为规模更小的子问题

前提条件：**贪心选择性质**和**最优子结构性**质

贪心选择性质：算法可以通过局部最优达到全局最优。

当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构性质。

正确性证明：数学归纳法（步数归纳、问题规模归纳）、交换论证、最优子结构性质 + 狭义的贪心选择性质

- 贪心法适用于组合优化问题。
- 求解过程是多步判断过程，最终的判断序列对应于问题的最优解
- 依据某种“短视的”贪心选择性质判断，性质好坏决定算法的成败
- 贪心法必须进行正确性证明
- 证明贪心法不正确的技巧：举反例

4 动态规划

基本要素：最优子结构、重叠子问题。

基本思想：记忆化搜索

动态规划的基本步骤：

刻画一个最优解的结构特征——子问题划分——最优子结构性质分析证明递归地定义最优值——建立递归关系以自底向上的方式计算出最优值根据计算最优值时得到的信息，构造最优解。