L1

算法：在有限时间内，对问题求解的一个清晰指令序列。

算法每个输入确定算法求解的实例。

输入、输出、可行、确定、有穷

$f(n) \leq g(n)$ $O(n)$

$<$ $o(n)$

$=$ $\Theta(n)$

$\geq$ $\Omega(n)$

$>$ $\omega(n)$

$\Theta(n) = O(n) \cap \Omega(n)$

- 传递性
- 对称性
- 互对称性 $O \Leftrightarrow \Omega$ $o \Leftrightarrow \omega$
- 反身性 $f(n) = \Theta(f(n))$

$1 < \log n < n^6 < n\log n < n^2 < n^3 < 2^n < n!$

主定理：$T(n) = aT(n/b) + f(n)$

$f(n) > n^{\log_b a}$ $f(n)$

$=$ $f(n)\log n$

$<$ $n^{\log_b a}$

（不适用非多项式、非标准、复杂递归）

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ $f(n) << g(n)$

输入规模

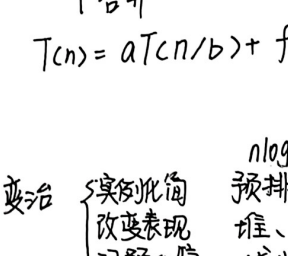
基本操作

平均、最坏、最好

表达式 → 简化

递归方程 + 初始条件 → 边界条件

$T(n) = T(\frac{n}{5}) + T(\frac{n}{10}) + cn$



等比：$\frac{1}{2}$

$T(n) = cn \cdot \frac{1}{1-\frac{1}{2}} = 4cn$

L2 递归

<阶乘> $T(n) = T(n-1) + 1$

<汉诺塔> $T(n) = 2T(n-1) + 1$

<计算二进制位数> $T(n) = T(\lfloor \frac{n}{2} \rfloor) + 1$

<斐> $T(n) = T(n-1) + T(n-2)$

跨1级 1年　2级 2年

选择 n^2

冒泡 n^2 ✓

插入 n^2 ✓

归并 $n\log n$ ✓

快速 $n\log n$ $1/n^2$

堆 $n\log n$

迭代：重复执行

递归：调用自身

- 减一
- 减常数因子
- 减不固定因子

L3 减治

- 自顶向下（递归）
- 自底向上（迭代）

- 减常量
- 减常量因子
- 减可变规模

插、拓

二分、假设(分)、俄氏

<俄氏> n 奇 $\Rightarrow$ $m \times 2 (+m)$

当 $n=1$ 结束，此时 $m + (+m;)$ 为积

L4 分治

- 划分子问题（相互独立，与原问题相同）规模相当，子问题数
- 递归
- 合并

$T(n) = aT(n/b) + f(n)$

- 快排
- 归并排
- 大整数乘法（3）
- strassen 矩阵乘法（7、18）
- 棋盘覆盖 4

L5 变治

- 实例化简 — 预排序、高斯消去
- 改变表现 — 堆、霍纳法则（多项式求值）
- 问题化简 — 线性规划

$n\log n$

L6 回溯法

DFS：前序、中序、后序　链表 O(V+E)　邻接矩阵 $O(V^2)$
BFS

回溯法：DFS
解空间树 { 子集树 2^n 叶子结点　0-1背包
排列树 $n!$　TSP
剪枝函数 { 约束函数
限界 } 减不最优

"解逐步生成，有退回"

分支限界法：BFS
生成所有孩子结点，选一个扩展
{ FIFO 队列
优先队列（代价最小）
背包 ↳ 已装 + 剩余容量 × 剩余单位最大价值
TSP ↳ 已走 + 其余最小出边费用

<装载问题> 特殊0-1　第一艘尽可能装满

回溯：所有解　DFS　成功可能性大
分支：1个　BFS　存储空间大　组合最优化

L7 动态规划

子问题 {阶段 / 状态 / 策略 — 状态转移方程(初始、边界)

{最优子结构 / 重复子问题 / 状态无后效 — {自顶向下(记忆化搜索) dp[] / 自下向上(递推)}

① 找出最优解结构
② 写方程(条件)
③ 计算

<LCS> $x_1 \cdots x_i$ / $y_1 \cdots y_j$

① $C[i,j]=\begin{cases}0 & \\ C[i-1,j-1]+1 & x_i=y_j \\ \max\{C[i,j-1],C[i-1,j]\} & x_i\neq y_j\end{cases}$

② 状态: $B[i,j]$ ↖ ← ↑

③ 输出: "↖" 输出公共

<0-1背包> K种物品,总重y

$dp[i][j]=\max(dp[i-1][j], dp[i-1][j-w]+v_i)$

<动态矩阵>

$A_{i\cdots j}=A_{i\cdots k}\cdot A_{k+1\cdots j}$ ($p_{i-1}\times p_k$ · $p_k\times p_j$)

$M[i,j]=\min\{M[i,k]+M[k+1,j]+p_{i-1}p_kp_j\}$

<多段图>

Floyd $dp[i][j]=\min(dp[i][j], dp[i][k]+dp[k][j])$

不能环 $O(n^3)$

每个K为中间节点,,更新 $d[i][j]$

MDP 多段图 前向/后向 $dp[v]=\min(dp[u]+w(u,v))$

L8 贪心 (当前状态选择)

可行性
局部最优
不可撤销

最优子结构

不保证解是最佳，不用求最大/小
(某种量度下分级处理的最优解)

<迪> 选距离最小

边非负 ①初始∞ ②选最短节点加入 ③更新 ④全部加入

单源最短路径

<分数背包> 利益/容量 单位价值最大

<最小生成树>

Prim 无向图 找权重最小 (从集合节点出发)

Kruskal 并查集 找边.