



算法复习课

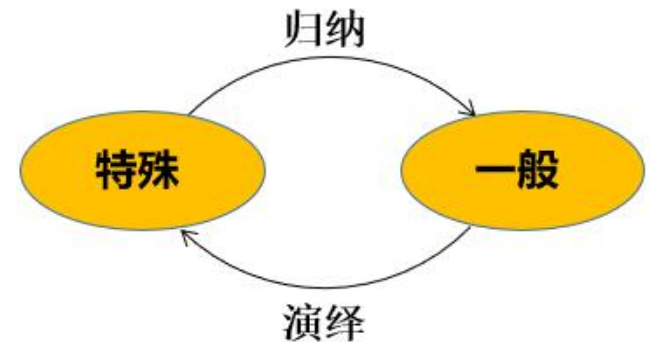
周可, zhke@hust.edu.cn

武汉光电国家研究中心

2024.10

overview

Algorithm: a set of rules to solve a certain problem



Search problem
Sort problem
Path problem
Optimization
Traversal problem
...

Rule design



Algorithm

discover

guide

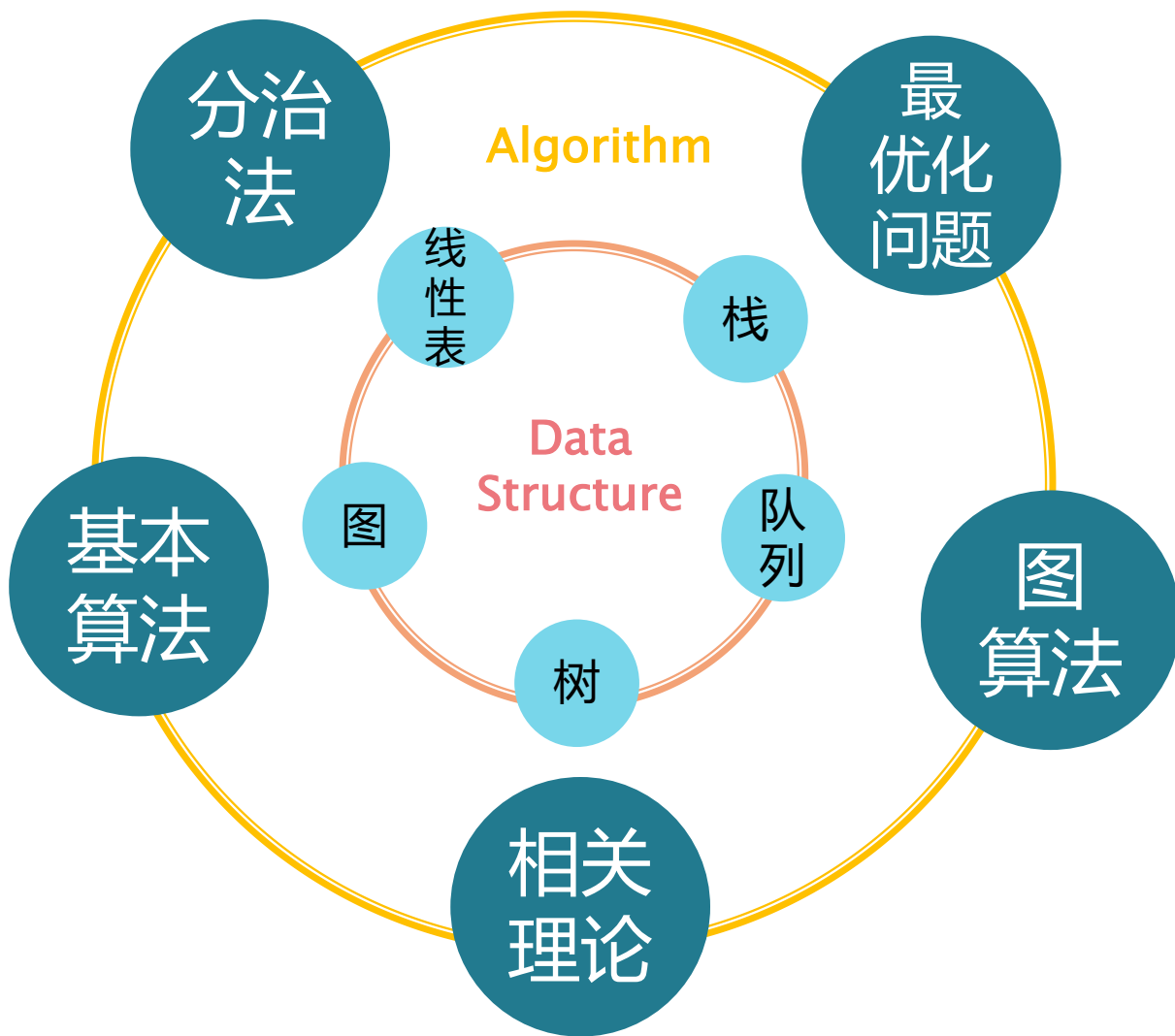
Design strategy



Guiding ideology
and basic law for
algorithm design


Divide and conquer
Greedy strategy
Dynamic programming
Search
....

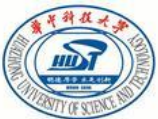
Main content





Main content (in detail)

1. 基本算法
排序
 - (比较排序) (Ω)  决策树
 - 插入、归并、堆、快速
 - 计数排序 (线性)
2. 理论相关
 - 算法复杂度 (Θ 、 O 、 Ω 、 o 、 ω)
 - 递归式求解 (代入法、递归树法、主方法)
 - NP完全性 (P、NP、NP-hard、NPC)
 - 近似算法 (VERTEX-COVER、TSP)
3. 图算法
 - 图的表示和搜索 (BFS、DFS)
 - Dijkstra算法 (路径)、MST(Prim/Kruskal)



Main content (in detail)

4. 分治法

应用

联系与区别?

归并排序、快速排序

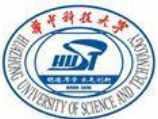
最大子数组、矩阵乘的strassen算法

5. 最优化问题

动态规划 (矩阵链乘法、最长公共子序列)



贪心算法 (硬币找零、背包问题、哈夫曼编码、Dijkstra/Prim/Kruska算法)



数据结构

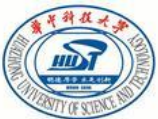
▶ 分类

- 线性结构：线性表、栈 (LIFO)、队列 (FIFO)
- 非线性结构：树和二叉树、图

▶ 主要内容

- 存储表示（顺序、链表等）
- 主要操作（插入、删除、查找、遍历等）

（详见L1、L2、L5、L6、L13）



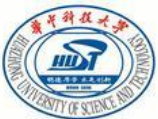
L6 二叉搜索树与红黑树

1. 树 → 二叉树 → 搜索二叉树 → 平衡二叉树
演化逻辑？

2. 红黑树

性质、左/右旋操作

红黑树的插入、删除（不要求）



L3 插入排序 & 归并排序

▶ 插入排序

- 通常 $T(n) = O(n^2)$
- 最好 $T(n) = O(n)$ (说明: 算法时间不仅与输入规模相关, 而且与输入数据的分布相关)

▶ 分治法

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq c, \\ aT(n/b) + D(n) + C(n) & \text{otherwise.} \end{cases}$$

▶ 归并排序

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1; \\ 2T(n/2) + \Theta(n) & \text{if } n > 1. \end{cases}$$



L4 最大子数组 & Strassen算法

▶ 最大子数组

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1. \end{cases}$$

▶ 矩阵乘的一般分治法

$$T(n) = 8T(n/2) + \Theta(n^2)$$

主方法



$$T(n) = \Theta(n^3)$$

▶ 矩阵乘的Strassen算法

$$T(n) = 7T(n/2) + \Theta(n^2)$$

主方法

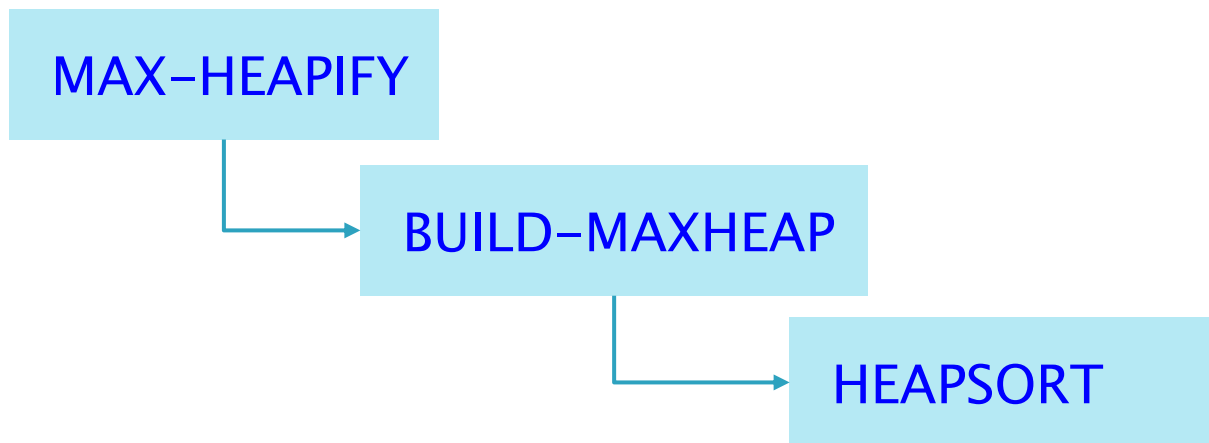


$$T(n) = \Theta(n^{\lg 7})$$



L7 堆排序

- ▶ 堆：完全二叉树
- ▶ 堆排序几个核心算法的逻辑关系



- ▶ Priority queue

应用：哈夫曼编码（Extract-min）

Dijkstra算法（Extract-min, Decrease-key）



L8 快速排序、决策树、计数排序

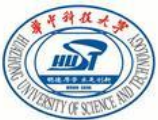
▶ 快速排序

- 分治法（划分，partitioning） $T(n)=\Theta(n\lg n)$
- Worst-case：极不平衡的划分 $T(n)=\Theta(n^2)$
- 快速排序的随机化版本

▶ 决策树

- 比较排序的下界： $\Omega(n\lg n)$

▶ 计数排序（线性时间）



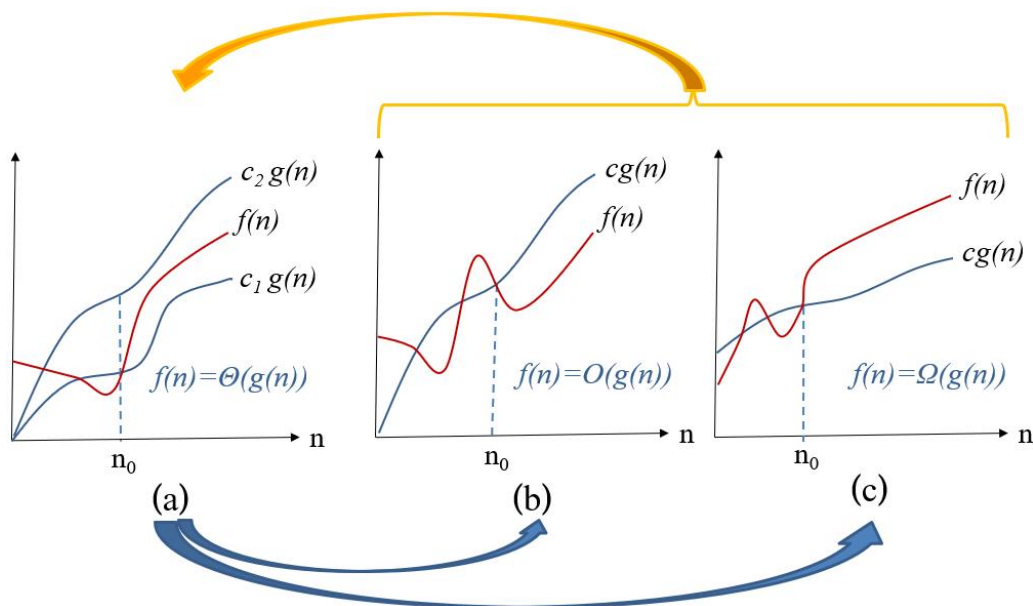
L9 算法复杂度

▶ Running time

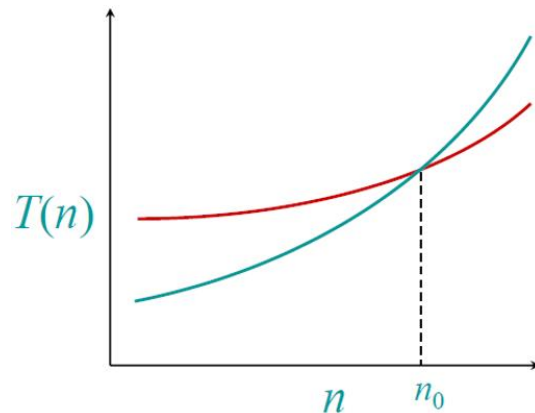
- Size、Distribution、*Upper-bound*

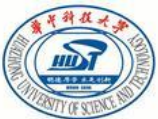
▶ 渐进表示

- Θ , O , Ω , o , ω



When n gets large enough, a $\Theta(n^2)$ algorithm *always* beats a $\Theta(n^3)$ algorithm.





L10 递归式求解

▶ 代换法

- Guess \rightarrow Verify (定义!)

▶ 递归树法

- $T(n) = aT(n/b) + \boxed{C(n) + D(n)}$ $\nearrow f(n)$

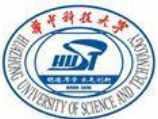
子树大小一样 (/不一样), 递归树平衡 (/不平衡)

- 节点含义 (分解、处理、合并)

▶ 主方法

注意定义, 尤其是第一、三种情况

$$1. f(n) = O(n^{\log_b a - \varepsilon}) \quad 2. f(n) = \Theta(n^{\log_b a}) \quad 3. f(n) = \Omega(n^{\log_b a + \varepsilon})$$



L1 1 动态规划

▶ 矩阵乘

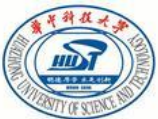
◦ $\underline{A_1 A_2 A_3 A_4}$ $\underline{A_1 A_2 A_3} A_4$ $A_1 \underline{A_2 A_3 A_4}$ $A_1 A_2 \underline{A_3 A_4}$

▶ DP性质

- 最优子结构（自下而上、局部最优→全局最优）
- 重叠子问题（重用思想、空间换时间）

▶ 最长公共子序列（LCS）

$$c[i, j] = \begin{cases} c[i-1, j-1] + 1 & \text{if } x[i] = y[j], \\ \max \{c[i-1, j], c[i, j-1]\} & \text{otherwise.} \end{cases}$$



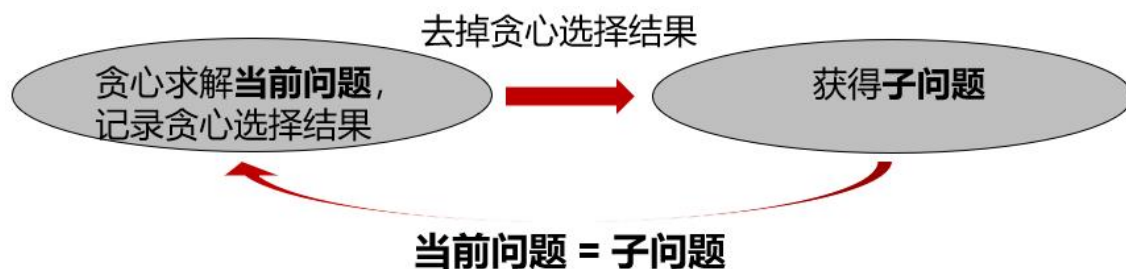
L12 贪心策略

▶ 整数背包 & 分数背包

- 度量标准

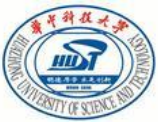
▶ 相关性质

- 贪心选择性质：大问题 \rightarrow 小问题，迭代求解
- 最优子结构：全局最优 & 局部最优



▶ 哈夫曼编码

- 变长码：频率 \rightarrow 码长



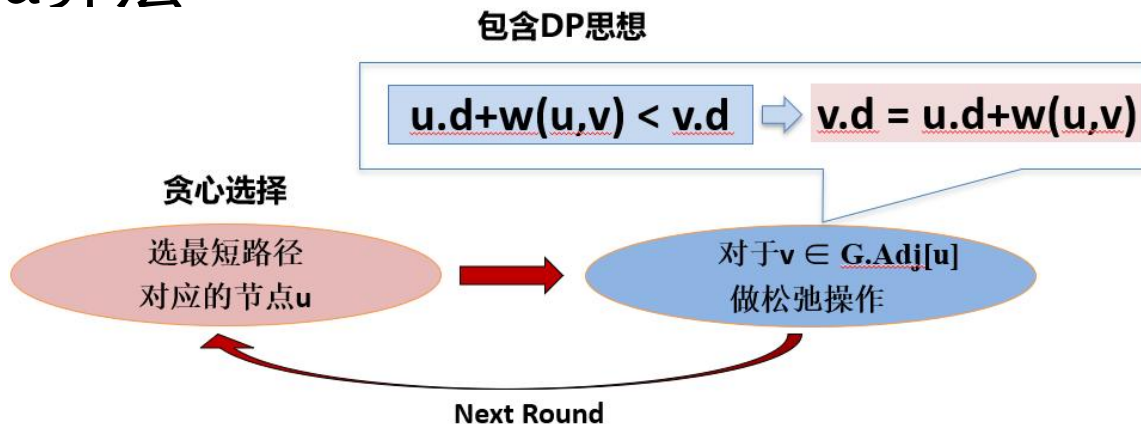
L13 图的表示与搜索

- ▶ 图的表示
 - 邻接矩阵
 - 邻接链表
- ▶ 图的搜索
 - BFS
 - DFS（递归，堆栈操作）



L14 Dijkstra算法 & MST

► Dijkstra算法



► MST

- 最优子结构性性质

贪心思想

- Prim算法：加点，树的扩展
- Cruskal算法：加边，森林中不同树的合并



L15-16 NP完全性

▶ 几个关键概念

- P, NP, NP-hard, NPC (理解概念, 不要求证明)

If L is a language such that $L' \leq_p L$ for **some** $L' \in \text{NPC}$, then L is NP-hard. In addition, $L \in \text{NP}$, then $L \in \text{NPC}$.

every $L' \in \text{NP}$

▶ 两种问题

- 最优化问题 $I \rightarrow S$
- 判定问题 $I, S \rightarrow \text{判定 (Y/N)}$

▶ 几个关键词

- 验证、证书、规约



L17 近似算法

- ▶ 概念
- ▶ 顶点覆盖
- ▶ TSP
 - 满足三角不等式的TSP
 - 通用TSP

考试题型

- 填空、判断、简答、分析/证明、综合