计算机科学导论

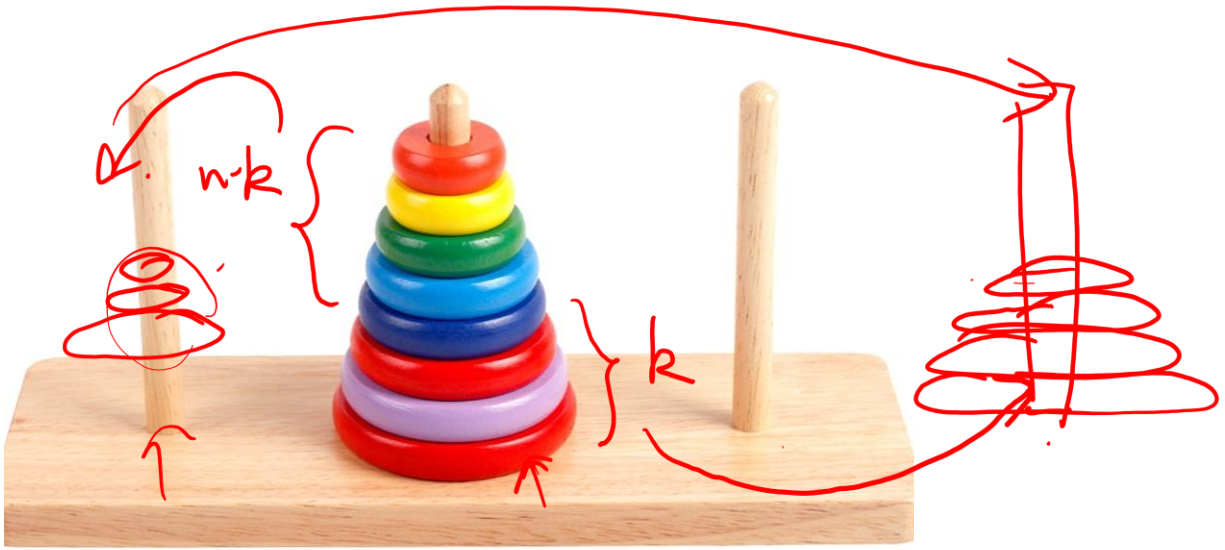
张家琳
中国科学院计算技术研究所

zhangjialin@ict.ac.cn

2025-5-16

思考题

- 汉诺塔（Hanoi）
- 如果有4根柱子怎么办？



$f(3,n)=2^n-1$

$f(4,n)=?$

- 记 l 根柱子，n个盘子的Hanoi塔问题最优解为 $f(l,n)$

$$f(4,n) \le \min_{1\le k\le n-1}\{2f(4,n-k)+f(3,k)\}$$

$$f(4,n) \le \min_{1\le k\le n-1}\{2f(4,n-k)+2^k-1\}$$

定义 $$F(4,n)=\min_{1\le k\le n-1}\{2F(4,n-k)+2^k-1\}$$

n	1	2	3	4	5	6	7	8
$F(4, n)$	1	3	5	9	13	17	25	33

$$F(4, n)\text{-}F(4, n\text{-}1) = 2, 2, 4, 4, 4, 8, 8, 8, 8, 16, \ldots$$

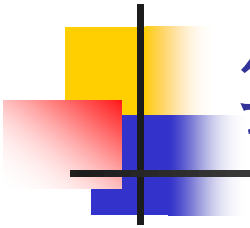
当 $n \in \left[\binom{k+1}{2}, \binom{k+2}{2}\right)$

$$F(4,n) = \sum_{i=1}^{k} i \cdot 2^{i-1} + (n - \binom{k+1}{2})2^k = (n-1-\binom{k}{2})2^k + 1$$

$$f(4,n) \le \min_{1 \le k \le n-1} \{2f(4,n-k) + f(3,k)\}$$

- 有更好的方法吗？
  - 没有，2014年被证明
  - Frame－Stewart algorithm（1941年提出）
  - 对更多根柱子：仍然是开放问题

  - https://en.wikipedia.org/wiki/Tower_of_Hanoi



算法思维

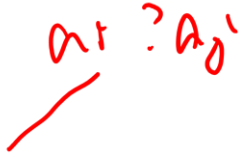
- 算法例子：排序
  - 冒泡排序、快速排序
- 大0符号
- 分治思想
- 其他算法实例
- P=NP?问题

问题的“难”与“易”

- **算法的时间复杂度(complexity)**：算法运行的总“步数”（时间）
  - 通常考虑在最坏的输入情况下
  - 冒泡排序：$O(n^2)$
  - 快速排序的期望运行时间 $O(n \log n)$
    - 即使在最坏输入情况下也如此

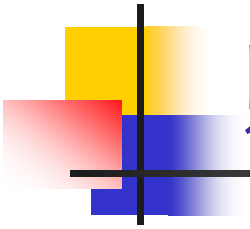
问题的“难”与“易”

- **问题的时间复杂度**：最优算法解决此问题的时间复杂度
  - 基于比较的排序：$\Theta(n \log n)$
    - 有算法可以在$O(n \log n)$内解决排序问题 ✓
    - 最优算法也不能做得更好



问题的“难”与“易”

- **问题的时间复杂度**：最优算法解决此问题的时间复杂度
  - 基于比较的排序：$\Theta(n \log n)$!
  - 两个数相乘: $O(n^2)$, $O(n^{1.59})$, $O(n \log n)$? $\Omega(n)$
- **问题的难易？**
  - 易问题：多项式时间复杂度的计算问题
  - 难问题：时间复杂度超过多项式时间的问题
  - 然而，很多问题的时间复杂度都未知



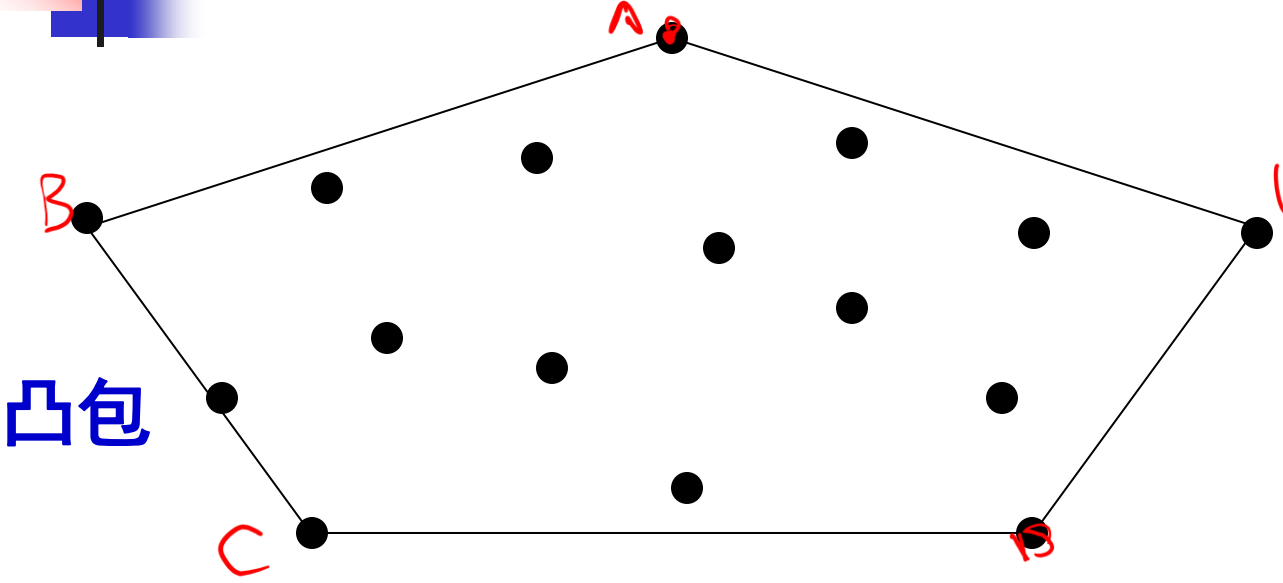
归约

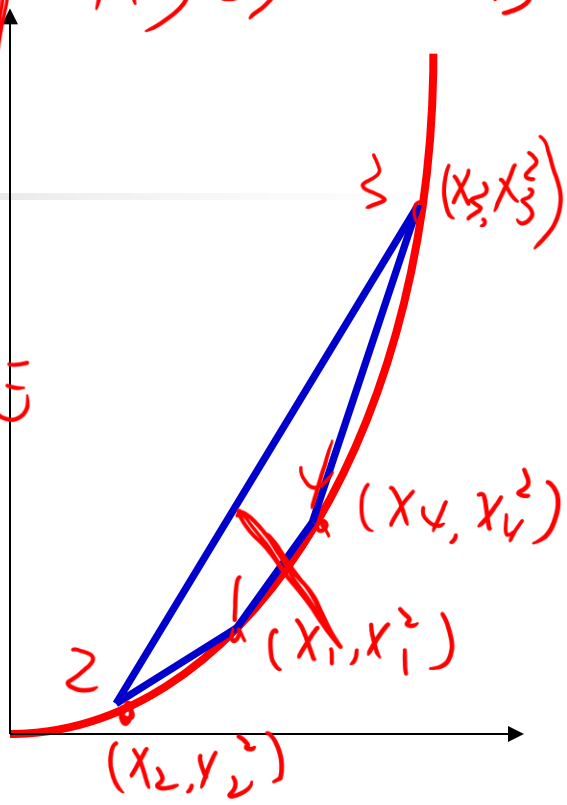
- 假设A和B是两个计算问题，称可以从问题A**归约**到问题B (记做$\mathbf{A \leq_P B}$)：

  如果任给一个求解B问题的算法，都可以“使用”此算法求解问题A

  A is “**easier**” than B

正实数排序 vs. 凸包





- **sorting $\leq_P$ convex-hall**
  - sorting问题的输入$x_1, x_2, \ldots, x_n$ $(x_i > 0)$
  - 构造：$P_1(x_1, x_1^2), P_2(x_2, x_2^2), \ldots, P_n(x_n, x_n^2)$

归约例子

- 问题A：判定一个整系数多项式方程是否有整数解？
- 问题B：判定一个整系数多项式方程是否有非负整数解？

  例如：$x^3 + y^3 = z^3$，$x^3 + 1 = 0$

- **证明：** $\mathbf{A} \leq_P \mathbf{B}$, $\mathbf{B} \leq_P \mathbf{A}$

$x^3+1=0$

整数解 非负

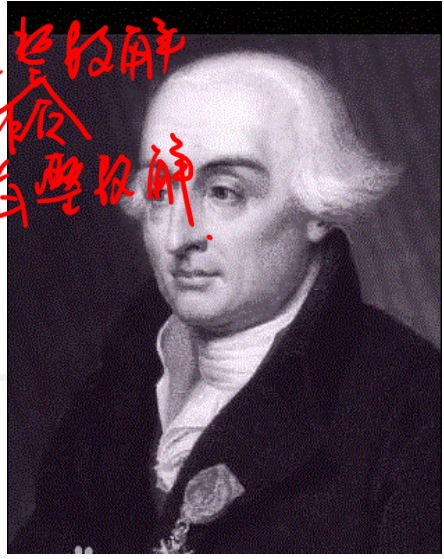
- **$A \leq_P B$:**
  - $f(x, y, z) \rightarrow F(p, q, s, t, u, v) = f(p-q, s-t, u-v)$
  - 例子：$f(x) = x^3 + 1$

转化为$F(p, q) = (p - q)^3 + 1$

非负整数解?

F有非负整数解 $\Rightarrow$ f 有整数解

F没有 $\Rightarrow$ f 没有



- **B** $\leq_p$ **A**:
  - $F(x, y) \rightarrow f(a, b, c, d, p, q, s, t) = F(a^2+b^2+c^2+d^2, p^2+q^2+s^2+t^2)$
  - **Lagrange四平方定理：**每个正整数均可表为四个整数的平方和(其中有些整数可以为零)
  - $23 = 3^2 + 3^2 + 2^2 + 1^2$

Lagrange
1736~1813

归约

- 假设A和B是两个计算问题，称可以从问题A**归约**到问题B (记做**$A \leq_P B$**)：

  如果任给一个求解B问题的算法，都可以“使用”此算法求解问题A
- A is “**easier**” than B
- “使用”：多项式次调用求解B问题的算法，辅助运算也是多项式时间
  - 如果问题B有多项式时间算法，那么A也有

P vs NP 之 P

- 多项式时间（可求解）问题(**P**olynomial time)：存在某个能解决该问题的算法A，它的时间复杂度是O(n^c)，其中c是某常数
  - n是输入的规模
  - O(n), O(n^2), O(n^3), O(n^{10000}), O($n^{2^{100}}$)都是多项式时间
  - 多项式时间问题被认为是计算机能够有效解决的问题
- 等价定义：图灵机可以多项式步求解的问题

P vs NP 之 NP

- P：图灵机（又称确定性图灵机, deterministic Turing machine）能用多项式步判定的问题
- NP：非确定性图灵机（non-deterministic Turing machine）能用多项式步判定的问题
- NP的等价定义：图灵机可以多项式时间验证的问题

P vs NP 之 NP

- 多项式时间可验证问题(**NP,** Non-deterministic Polynomial time)：问题的“答案”可以在多项式时间内验证
  - 存在多项式时间的验证算法，对任何输入：
    - 如果答案是“正确”(接受)，那么存在证据使得验证算法能证明这一点；
    - 反之，一切证据都不能证明
  - P的问题都属于NP, i.e. $P \subseteq NP$

NP的例子

- 一张地图是否可以进行3染色？
  - 暴力穷举算法：$O(3^n \cdot poly(n))$
  - 目前不知道是否有多项式时间可以判定
    - 是否属于P未知
  - 3染色问题属于NP
    - 如果答案是可以3染色，则证据是一种染色方式
    - 验证算法：验证所用的颜色数不超过3种；验证每个区域和相邻区域的染色都不同
    - 如果答案是不可以3染色，任何证据都无法通过验证算法

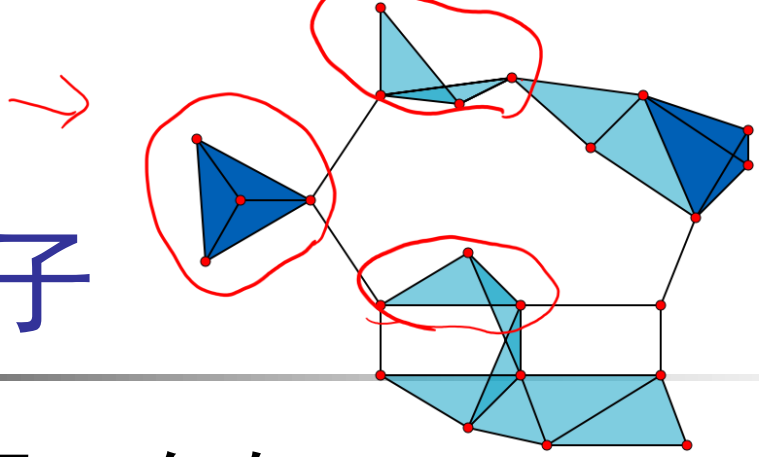
NP的例子

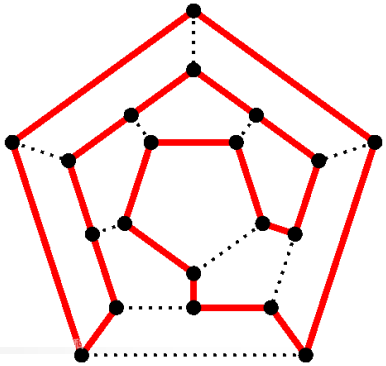
- 给定布尔表达式ϕ，判定是否有一组赋值使得这个布尔表达式的取值为真？SAT
  - 例子：$\phi = x \wedge y$　　$\phi' = x \wedge \bar{x}$
  - 暴力穷举算法：$O(2^n \cdot poly(n))$
  - 不知道是否有多项式时间算法可以判定
  - SAT问题属于NP
    - 证据：使得取值为真的赋值

P vs NP 之 NP

- 多项式时间可验证问题(**NP,** Non-deterministic Polynomial time)：问题的“答案”可以在多项式时间内验证
  - 存在多项式时间的验证算法，对任何输入：
    - 如果答案是“正确”(接受)，那么存在证据使得验证算法能证明这一点；
    - 反之，一切证据都不能证明
  - 证据：长度是输入规模的多项式
  - 验证算法：运行时间是输入规模的多项式

NP的例子





- 一张图是否存在Hamiltonian回路？
  - Hamiltonian回路：经过每个顶点一次且只经过一次的一条回路
  - 证据：一条Hamiltonian回路
- 给定一张图及参数k，判断图里是否有k个点构成团？
  - 团：该集合中任何两个点之间都有边
  - 证据：k个点

P vs NP

- P 是否等于 NP？ P ≟ NP
  - 是否所有多项式时间可验证的问题都可以在多项式时间内求解？
  - 理论计算机领域最重要的开放问题
  - 千禧年七大数学难题之一：https://www.claymath.org/millennium-problems/p-vs-np-problem
  - 对应用领域有深远影响：密码学、信息安全等

NP-完全

- 目前为止，以上NP的例子都不知道是否属于P！
- 这些问题都是NP-完全的
- NP-完全：NP中最“难”的问题
  - 所有其他的NP问题都可以归约到它
  - 如果找到了一个NP-完全问题的多项式时间算法，则所有NP问题都有多项式时间算法，即P=NP

P vs NP

- 尝试证明$P = NP$
  - 找到一个NP-完全的问题，找到这个问题一个多项式时间的求解算法
- 尝试证明$P \neq NP$
  - 找到一个NP问题，证明这个问题不存在多项式时间的求解算法
- 如果$P = NP$：所有NP问题都有多项式时间算法
- 如果$P \neq NP$：所有NP-完全问题都没有多项式时间算法

有没有不在NP的问题？

- 给定$n \times n$的棋盘，两个人下广义的围棋，先手是否必胜？
  - 目前为止，不知道在不在NP中
- 停机问题
  - 不在NP中
  - 事实上，没有图灵机能判定这个问题。

密码学与P vs NP

- 单向函数（one-way function）
  - 密码学基石
  - 给定x，f(x) “易”计算：在P里面
  - 给定f(x)，x “难”计算：不在P里面
- 单向函数是否存在？
  - 不知道
  - 若存在，则$P \neq NP$

密码学-大整数分解

- 大整数分解问题：
  - 最重要的单向函数候选者之一
  - 给定p,q，计算$n = p \times q$是容易的
  - 给定n，分解成$p \times q$目前是难的
- 基于大整数分解的公钥RSA算法
  - Ron Rivest, Adi Shamir, Leonard Adleman (1977)

大整数分解问题

- 目前没有多项式经典算法能解决大整数分解问题
  - 但也没有证明大整数分解问题是NP-完全的
- 量子算法可以在多项式时间内解决
  - Shor算法（Peter Shor，1994）
- BQP：量子计算机可以在多项式时间内求解的问题
  - $\mathrm{P} \subseteq BQP \subseteq PSPACE$
  - BQP vs NP：重要的开放问题！

RSA：非对称加密算法

- 准备两个大素数p和q，$n = p \times q$
- 计算$m = (p-1)(q-1)$
- 找到和m互素的数e以及它的逆元d，即$ed\ (mod\ m) = 1$
- 公布公钥(n, e)，藏好私钥(n, d)
- 加密算法：$x \rightarrow y = x^e\ mod\ n$
- 解密算法：$y \rightarrow x = y^d\ mod\ n$

RSA：非对称加密算法

- 破解RSA加密算法：
  - 根据公钥(n, e)，要求出私钥中的d
  - 如果能把n分解成$p \times q$，则可以计算m以及d
- RSA算法能用来做什么
  - 别人发给我的文件只有我能看
    - 别人用公钥加密文件，我用私钥解密才能看
  - 别人不能仿造我来发文件
    - 我用私钥加密文件，别人都可以用公钥解密看

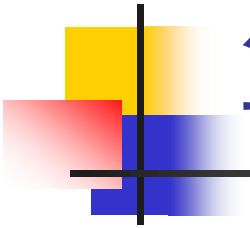
cake-cutting.

思考题：分蛋糕问题

- 问题：2个人分一个蛋糕
  - 每个人对蛋糕不同部分的喜好不同
- 怎么分公平？
  - 公平(fairness)：2人都认为自己的一份不少于 1/2
  - 无怨(envy-free)：2人都不觉得别人拿得比自己多
- 方法：一个人分，另一个人先选
- 思考题：3个人分一个蛋糕呢？
  - 公平？ 1/3.
  - 无怨？ 此事两个人都不喜.

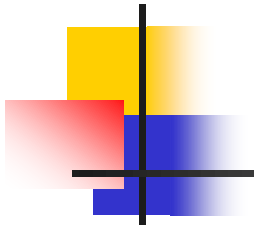
分蛋糕问题

- 更多推广问题
  - 更多个人分
  - 每个人要求分到的比例不同
  - 分房租
  - 小组作业打分
  - ……



算法思维

- 算法例子：排序
  - 冒泡排序、快速排序
- 大0符号
- 分治思想
- P=NP?问题
- 思考题：分蛋糕



谢谢！