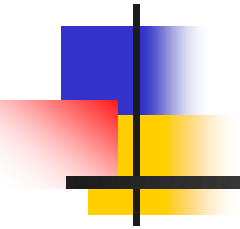


计算机科学导论

孙晓明

中国科学院计算技术研究所

2022-4-22

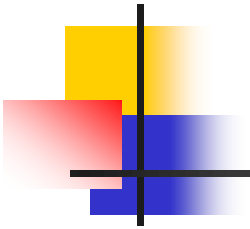
问题

- 下列复杂度中不是$o(1.01^n)$有___?

- 1) $n^{\sqrt{n}}$
- 2) $n!$
- 3) $2^{\log^{20} n}$
- 4) $2^{\sqrt{n}}$

乘法与傅里叶变换

- $f(z) = x_0 + x_1 z + \ldots + x_n z^n$
- $g(z) = y_0 + y_1 z + \ldots + y_n z^n$
- $h(z) = f(z)\, g(z)$, $h(10)$ or $h(2)$
- 令 $N=2n+1$，取$\alpha_1, \alpha_2, \ldots, \alpha_N$
  - 计算$f(\alpha_1), \ldots, f(\alpha_N), g(\alpha_1), \ldots, g(\alpha_N)$
  - 计算 $h(\alpha_i) = f(\alpha_i) g(\alpha_i)$ $(i = 1, 2, \ldots, N)$
  - 计算$h(z)$



- 基于快速傅里叶变换
- $O(n \log n \log\log n)$ (Schönhage and Strassen, 1971)
- $O(n \log n \, 2^{O(\log^* n)})$ (Fürer, 2007)
- $O(n \log n \, 8^{\log^* n})$ (Harvey and Hoeven, 2016)
- $O(n \log n \, 4^{\log^* n})$ (Harvey and Hoeven, 2019)
- $O(n \log n)$ (Harvey and Hoeven, 2021)
  - $\log^* n \triangleq \min\{k : \overbrace{\log\log\cdots\log n}^{} \leq 1\}$

矩阵乘法(1)

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \\ \vdots & & \ddots & \\ a_{n1} & & & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & \ddots & & \\ \vdots & & \ddots & \\ b_{n1} & & & b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & \ddots & & \\ \vdots & & \ddots & \\ c_{n1} & & & c_{nn} \end{bmatrix}$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

$O(n^3)$次**乘法**，　$O(n^3)$次加法

矩阵乘法(2)

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

$$m_1 = (a_{12} - a_{22})(b_{21} + b_{22})$$
$$m_2 = (a_{11} + a_{22})(b_{11} + b_{22})$$
$$m_3 = (a_{11} - a_{21})(b_{11} + b_{12})$$
$$m_4 = (a_{11} + a_{12})b_{22}$$
$$m_5 = a_{11}(b_{12} - b_{22})$$
$$m_6 = a_{22}(b_{21} - b_{11})$$
$$m_7 = (a_{21} + a_{22})b_{11}$$

$$c_{11} = m_1 + m_2 - m_4 + m_6$$
$$c_{12} = m_4 + m_5$$
$$c_{21} = m_6 + m_7$$
$$c_{22} = m_2 - m_3 + m_5 - m_7$$

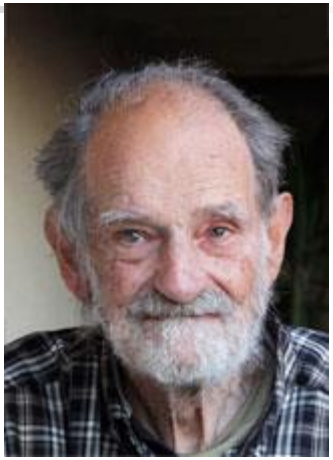
$O(n^{\log 7 \approx 2.81})$次**乘法**

矩阵乘法(3)

- Strassen algorithm'69 $O(n^{2.81})$
- $O(n^{2.79}), O(n^{2.55}), O(n^{2.48})$ …
- Coppersmith–Winograd algorithm'89 $O(n^{2.376})$
- Stothers'10 $O(n^{2.374})$
- Williams'11 $O(n^{2.373})$
- Le Gall'14 $O(n^{2.3729})$
- Alman, Williams'21 $O(n^{2.3728596})$

2012年经济学诺贝尔奖

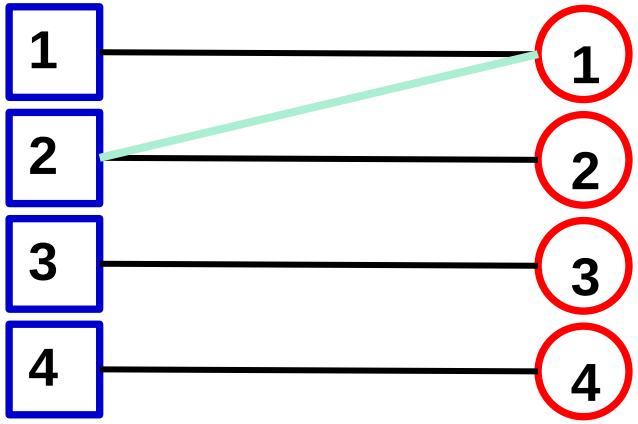
Alvin E. Roth



Lloyd S. Shapley

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2012 was awarded jointly to Alvin E. Roth and Lloyd S. Shapley "for the theory of stable allocations and the practice of market design"

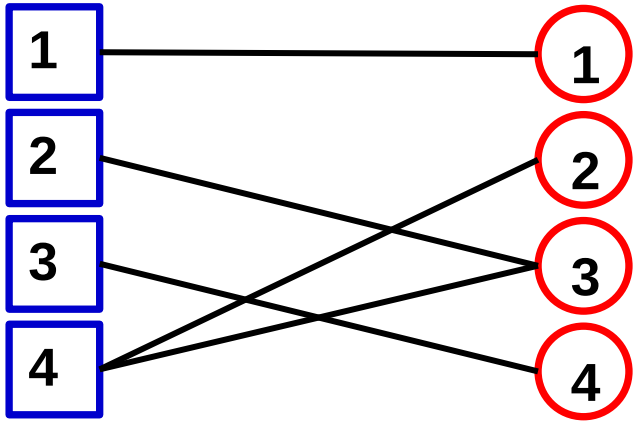
Stable matching



1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

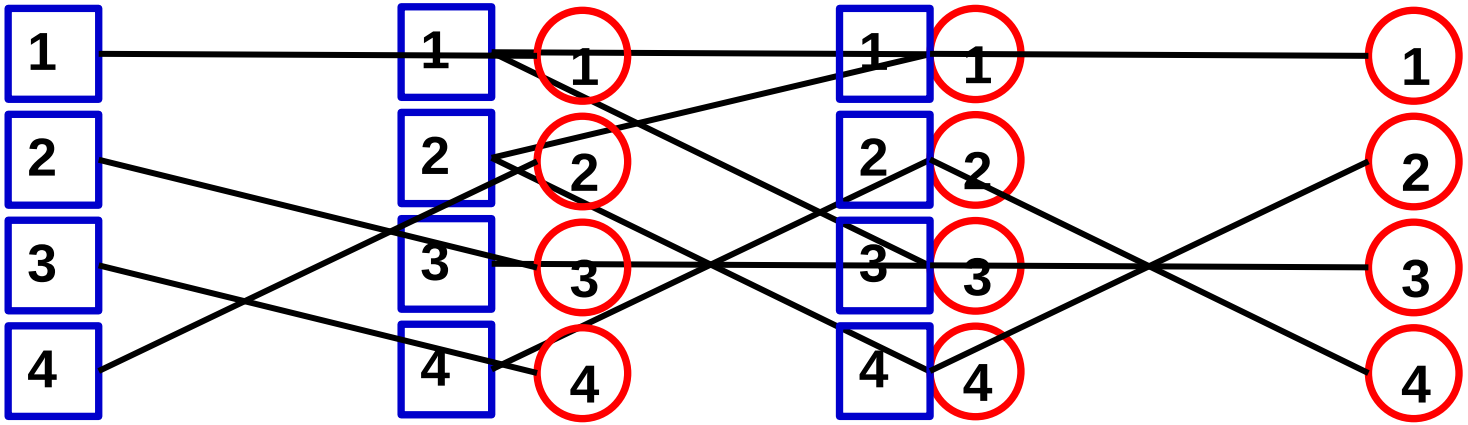
Gale-Shapley Algorithm (1962)



1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

Gale-Shapley Algorithm (1962)



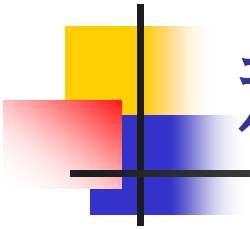
1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

问题的“难”与“易”

- **算法复杂度(complexity)**：算法运行的总“步数”（时间）
  - 通常考虑在最坏的输入情况下
  - 例如：冒泡排序

- **问题的复杂度**：最优算法解决此问题的算法复杂度
  - 例如：排序问题 $O(n^2)$, **$O(n\log n)$**



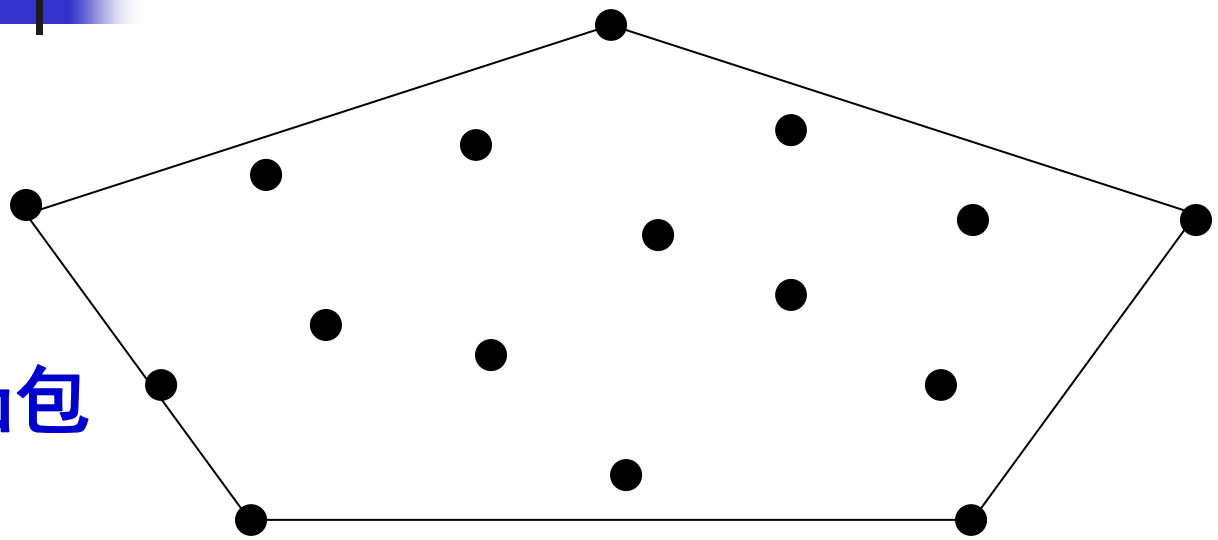
规约

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

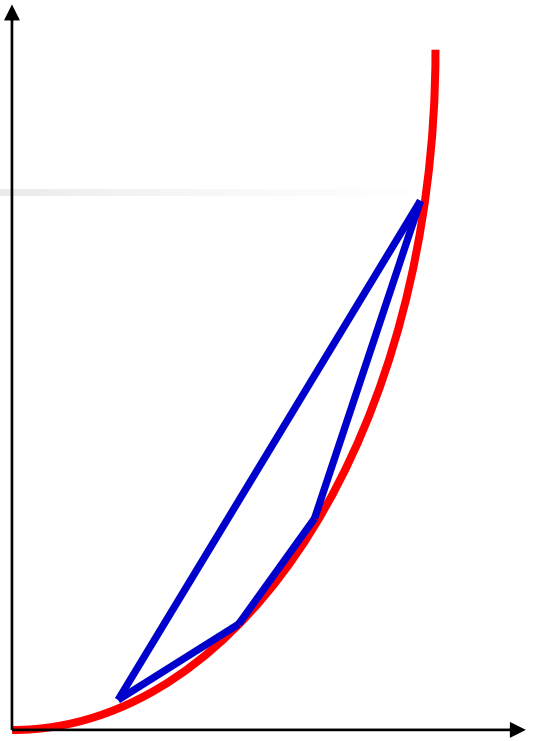
  如果任给一个求解B问题的算法，都可以“使用”此算法求解问题A

  A is “**easier**” than B

排序 vs. 凸包



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：$P_1(x_1, x_1^2), P_2(x_2, x_2^2), \ldots, P_n(x_n, x_n^2)$

思考题

- 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：$x^3 + y^3 = z^3 + 2022,$

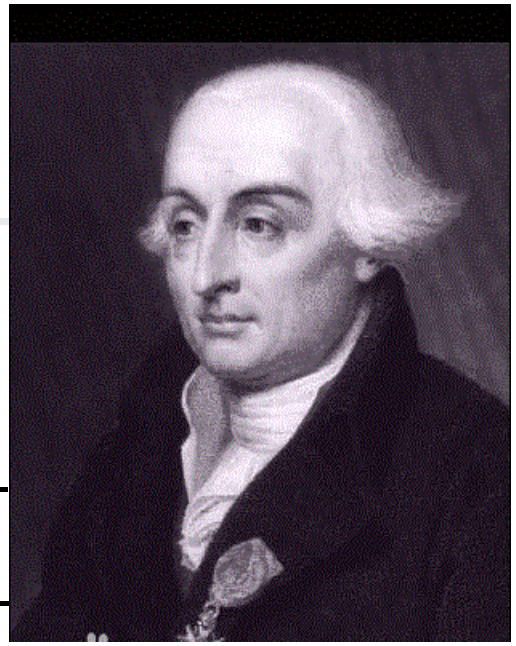
  $$x^{2021} + 3y^{2022} = 5z^{2023} - 7$$

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

- **A $\leq_P$ B:**
  - $f(x,y,z) \rightarrow F(x_+, x_-, y_+, y_-, z_+, z_-)$
    $= f(x_+ - x_-, y_+ - y_-, z_+ - z_-)$

- **B $\leq_P$ A:**
  - $F(x,y) \rightarrow f(a,b,c,d,p,q,r,s)$
    $= F(a^2+b^2+c^2+d^2, p^2+q^2+r^2+s^2)$
  - $23 = 3^2 + 3^2 + 2^2 + 1^2$ **Lagrange四平方定理**

- **A $\leq_P$ B:**
  - $f(x,y,z) \rightarrow F(x_+, x_-, y_+, y_-$
    $= f(x_+ - x_-, y_+ - y_-, z_+ - z_-$



Lagrange
1736~1813

- **B $\leq_P$ A:**
  - $F(x,y) \rightarrow f(a,b,c,d,p,q,r,s)$
    $= F(a^2+b^2+c^2+d^2, p^2+q^2+r^2+s^2)$
  - $23 = 3^2 + 3^2 + 2^2 + 1^2$

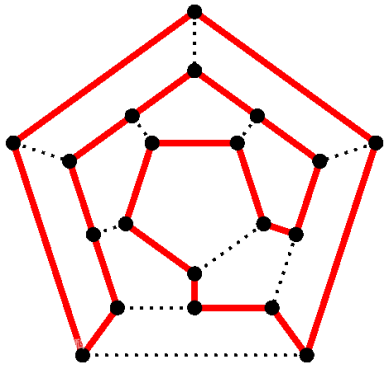
Lagrange四平方定理

多项式时间复杂性类

- 多项式时间（可求解）问题(**P**olynomial time)：存在某个常数c, 问题的算法复杂度是O(n^c)
  - 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

非确定性多项式时间复杂性类

- 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
  - 例如：一张地图是否可以进行3染色？一张图是否存在Hamiltonian回路？CNF的可满足性？
  - P的问题都属于NP, i.e. $P \subseteq NP$

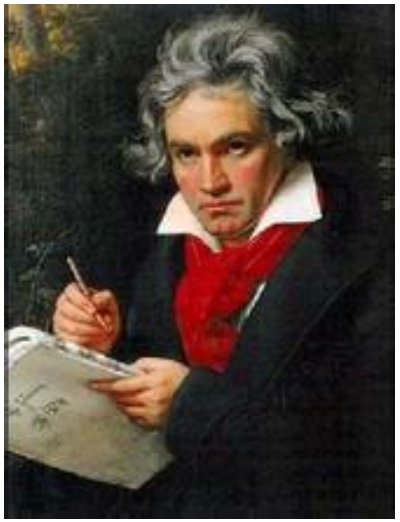


Millennium Prize

- Birch and Swinnerton-Dyer Conjecture
- Hodge Conjecture
- Navier-Stokes Equations
- **P vs NP**
- Poincaré Conjecture (solved)
- Riemann Hypothesis
- Yang-Mills Theory

First Clay Mathematics Institute Millennium Prize Announced: Prize for Resolution of the Poincaré Conjecture Awarded to Dr. Grigoriy Perelman

- 如果 P = NP



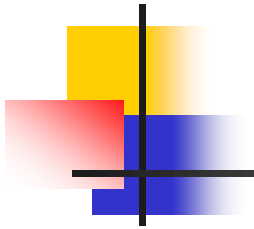
- 如果 P ≠ NP
  - **密码学！**

密码学

- Rivest, Shamir, Adleman (1979)
  - Factoring $n = p \times q$ is HARD
  - $\Phi(n) = (p-1) \times (q-1)$, pick $d \times e \equiv 1 (mod\ \Phi(n))$
  - Public key e, private key d
  - $E(M) = M^e$, $D(C) = C^d\ (mod\ n)$
- Play poker via Wechat

思考题

- 300名同学参加乒乓球赛，要想决出冠军至少需要赛多少场？
- 如果想决出前二名需要赛多少场？前三名呢？
- 如果改成比赛每次可以多人参加的棋牌运动(例如4人)，又需要多少次？
- 请就一般的n, k回答上述问题。
- 注：这里假定每位同学的实力非常稳定，需要决出实力最强的前k名同学。



谢谢！