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

2018-5-18

思考题

- 汉诺塔（Hanoi）　　2^n-1
- 如果有4根柱子怎么办？

- 记 k 根柱子，n个盘子的Hanoi塔问题最优解为 $f(k, n)$

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

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

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

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

$F(4, n)$-$F(4, n$-1$)$ = 2, 2, 4, 4, 4, 8, 8, 8, 8, 16,…

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

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

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

- 有更好的方法吗？
  - 没有，2014年被证明
  - Frame–Stewart algorithm
  - 对更多根柱子，open

算法思维

- 算法例子：排序
  - 冒泡排序、快速排序
- 大O符号
- 分治思想
- P=NP?问题

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：基于比较的排序 **O(*n*log *n*)**
  - 两个数相乘 $O(n^2)$, $O(n^{1.59})$, **O(*n*log *n*)**

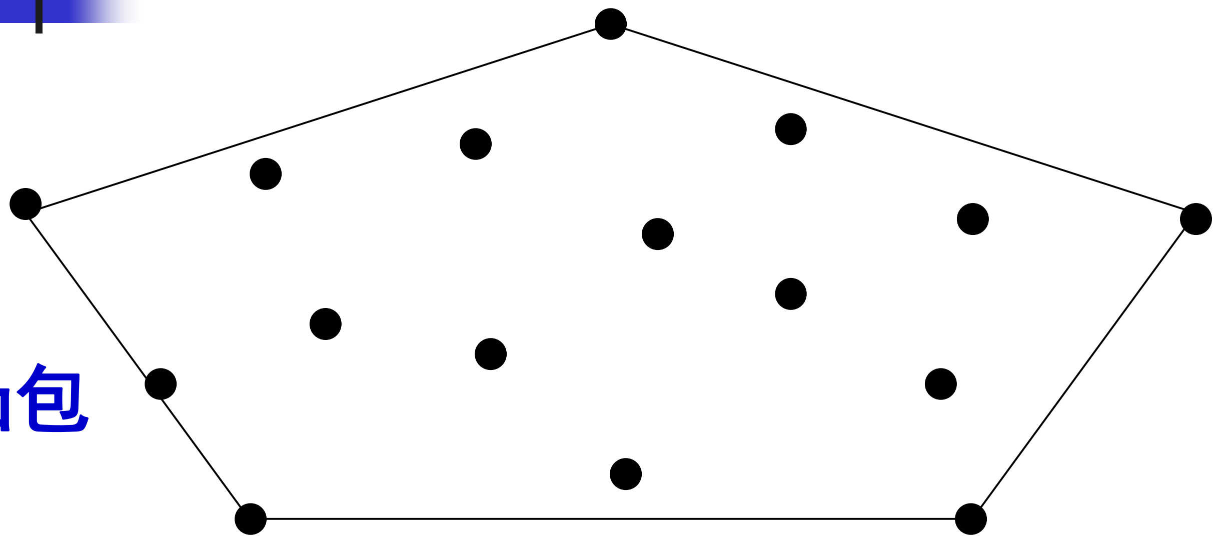
规约

- 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**$A \le_p B$**)：

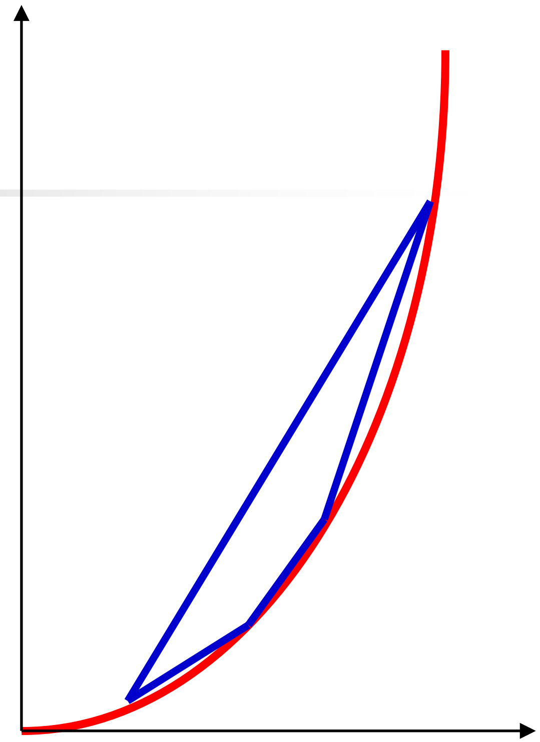
  如果任给一个求解B问题的算法，都可以“使用”此算法求解问题A

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

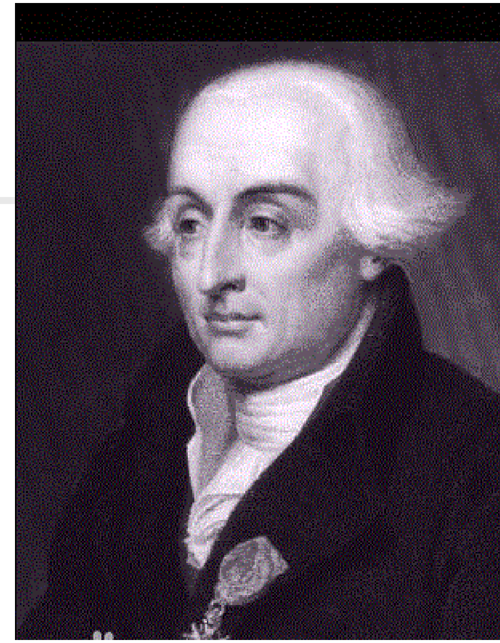
思考题

- 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：$x^3 + y^3 = z^3$，$x^3 + y^3 = z^3 + u^3$

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

- **A $\leq_P$ B:**
  - $f(x, y, z) \rightarrow F(p, q, s, t, u, v)$
    $v)$



Lagrange

1736~1813

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

Lagrange四平方定理

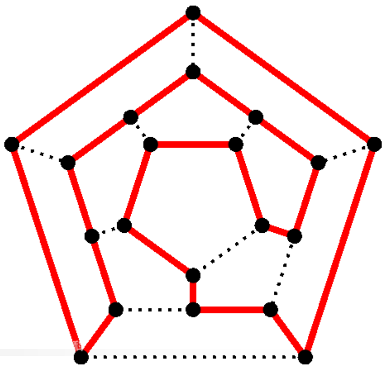
P vs NP 之 P

- 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

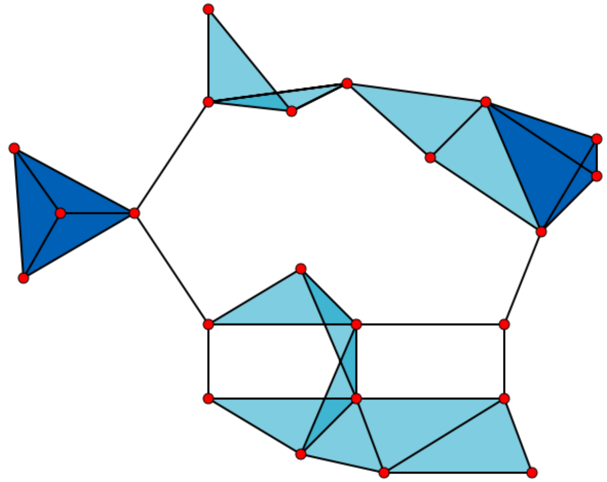
P vs NP 之 NP

- 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
  - 例如：一张地图是否可以进行3染色？
  - P的问题都属于NP, i.e. $P \subseteq NP$

更多NP中的例子



- 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
- 一张图是否存在Hamiltonian回路？
- 给定一张图及参数k，判断图里是否有k个点构成clique？



- 目前为止，不知道这些问题是否属于P！

NP-完全

- SAT, Hamilton, k-clique都是NP-完全的
- NP-完全：NP中最“难”的问题
  - 所有其他的NP问题都可以规约到它
  - 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

素数判定

- 给定正整数n，判定n是否是素数。
- 这个问题属于NP吗？
  - 是！
  - 存在g，$g^{n-1} \equiv 1 \ (mod\ n)$ 且对n-1的任何素因子p，$g^{(n-1)/p} \neq 1 \ (mod\ n)$
  - 证据：g，n-1的所有素因子*
    - 如何判断$g^{n-1} \equiv 1 \ (mod\ n)$？
    - 如何判断给出n-1的素因子都是素数？
  - Agrawal–Kayal–Saxena primality test，2002

有没有不在NP的问题？

- 给定n*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
- 给定一个图灵机M和一个输入字符串x，判定M在x这个输入上是否能停机？
  - 不在NP中
  - 事实上，没有图灵机能判定这个问题。

停机问题：不可判定

	1	2	3	4	5	6	7	8	9	…
1	0	0	0	0	0	0	0	0	0	…
2	1	1	1	1	1	1	1	1	1	…
3	0	1	0	0	0	0	0	0	0	…
4	0	0	1	1	1	1	1	1	0	…
5	1	0	0	0	0	0	0	0	0	…
6	1	1	1	1	1	1	1	1	1	…
7	1	1	0	0	1	1	1	1	1	…
…	.	.	.	.	.	.	.	.	.	.

- 假如存在图灵机H能判定
- 定义图灵机G
  - 对任何输入i,
  - 如果H(i,i) =0, 则停机.
  - 否则无限循环

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

- 基于大整数分解的RSA算法
  - $n = p*q$ is HARD
  - Rivest, Shamir, Adleman (1979)
  - $\Phi(n) = (p-1)(q-1)$, $d \times e \equiv 1 \pmod{\Phi(n)}$
  - $E(M) = M^{e}$, $D(C) = C^{d} \pmod{n}$
- 基于离散对数的公钥加密算法
  - Discrete Logarithm is HARD
- 加密算法能用来做什么
  - 别人发给我的文件只有我能看
  - 别人不能仿造我来发文件
  - ……

思考题

- 分蛋糕问题
  - 2个人分一个蛋糕
  - 怎么分能使每个人都觉得别人手里的蛋糕不比自己手里的蛋糕好？
  - 一个人分，另一个人先选
  - 3个人分一个蛋糕呢？

班级快速排序实验(人体计算机)

- 时间：5月25日上午
- 地点：大礼堂外草地
  - 如下雨：一半在大礼堂，一半在二公寓多功能厅
- 课前准备：
  - 熟悉快速排序算法
  - 思考要统计的量，以及如何在实验中验证正确性（结果正确、算法正确、系统正确）
  - 准备各种材料，如记录材料、拍照工具等

班级快速排序实验(人体计算机)

- 课中阶段：
  - 完成2-3次快速排序实验
  - 有时间的班级可以选做冒泡排序等其他排序算法
  - 各班负责人向全班总结排序实验的情况
- 课后阶段：
  - 各班负责人整理汇报材料，报给年级负责人
  - 年级总负责人在6月1日的课上向所有人汇报

算法思维

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

谢谢！

停机问题（补充）

- 假设有图灵机H能判定停机问题，即
  - H(M,x)=1，如果图灵机M在输入串x下能停机
  - H(M,x)=0，如果图灵机M在输入串x下不能停机
- 图灵机M
  - 可以用七元组表达
  - 可以用有限的二进制串表达
  - 所有的图灵机是可数集合
- 输入串x
  - 有限的二进制串
  - 所有的输入串也是可数集合

停机问题

	1	2	3	4	5	6	7	8	9	…
1	0	0	0	0	0	0	0	0	0	…
2	1	1	1	1	1	1	1	1	1	…
3	0	1	0	0	0	0	0	0	0	…
4	0	0	1	1	1	1	1	1	0	…
5	1	0	0	0	0	0	0	0	0	…
6	1	1	1	1	1	1	1	1	1	…
7	1	1	0	0	1	1	1	1	1	…
…	.	.	.	.	.	.	.	.	.	.

- 第i行第j列
  - 二进制编码为i的图灵机M
  - 二进制编码为j的输入x
  - M在x上会停机，则写1
  - 否则写0
  - 如果i对应的图灵机不合法，默认写1

停机问题

- 假设有图灵机H能判定这个问题，即
  - H(M,x)=1，如果图灵机M在输入串x下能停机
  - H(M,x)=0，如果图灵机M在输入串x下不能停机
- 定义图灵机G
  - 对任何输入i,
  - 如果H(i,i) =0, 则停机.
  - 否则无限循环，即不停机

停机问题

- 定义图灵机G
  - 对任何输入i,
  - 如果H(i,i) =0, 则停机.
  - 否则无限循环，即不停机
- 考察G(G)
  - 假设图灵机G在输入字符串G下停机
  - H(G,G)=1，即第G行第G列填的数是1
  - 由图灵机G的定义，应该无限循环，即图灵机G在输入字符串G下不停机，矛盾

停机问题

- 定义图灵机G
  - 对任何输入i,
  - 如果H(i,i) =0, 则停机.
  - 否则无限循环，即不停机
- 考察G(G)
  - 假设图灵机G在输入字符串G下不停机
  - H(G,G)=0，即第G行第G列填的数是0
  - 由图灵机G的定义，应该停机，即图灵机G在输入字符串G下停机，矛盾
- H不存在，即停机问题不可判定