《现代密码学》第三讲

复杂性理论

上讲内容回顾

- Shannon 的通信保密系统
- 熵和无条件保密
- 分组密码设计思想

本章主要内容

- 问题的定义及分类
- 算法复杂度定义及分类
- P 问题和 NP 问题
- 规约思想与 NPC 类
- 密码算法的计算安全性

本章主要内容

- 问题的定义及分类
- 算法复杂度定义及分类
- P 问题和 NP 问题
- 规约思想与 NPC 类
- 密码算法的计算安全性

问题的定义及分类

1 设 A=($a_1, a_2, \cdots, a_n$) 是由 n 个不同的正整数构成的 n 元组， S 是另一已知的正整数．A 称为背包向量，S 称为背包容积． 求 A 中元素集合 $\sum_{a_i \in A} a_i = S$ 使 .

2 设背包向量 A=(1, 2, 5, 10, 20, 50, 100)，$X = \{0,1\}^7$，$\sum_{i=1}^{7} x_i a_i = S$ 背包容积为 177，求向量 ，使得

问题的定义及分类

3　已知整数 N，问 N 是否是一个素数？

4　试问 77 是否是素数？

5　试问 79 是否是素数？

6　已知整数 N，求 N 的素分解式.

7　已知整数 177，求其素分解式.

问题的定义及分类

- **问题**：描述参量陈述解答应当满足的性质（称为**询问**）.

  参量为具体数值时，称为问题的一个实例.

- **判定问题**：回答只有 Yes 或 No.
- **计算问题**：从其可行解集合中搜索出最优解.

本章主要内容

- 问题的定义及分类
- 算法复杂度定义及分类
- P 问题和 NP 问题
- 规约思想与 NPC 类
- 密码算法的计算安全性

算法复杂度的定义

例 设 x 是小于 100 的某个整数，问 x 是否是素数？

解答一：

取 2 ~ $\lfloor\sqrt{x}\rfloor$ 的所有整数，依次试除 ***x***，若存在某个整数可以整除 ***x***，则程序停止，输出 ***x*** 为合数，否则输出 ***x*** 为素数．

最坏试除次数：$\lfloor\sqrt{x}\rfloor$ 存储空间：**0**

解答二： 有小于 **100** 的素数存储在寄存器中；然后将 ***X*** 与存储器中的元素比较，若存在某个素数等于 ***x***，则程序停止，输出 ***x*** 为素数，否则输出 **x** 为合数．

最坏比较次数：***100/ln100***, 存储空间：***100/ln100***

算法复杂度的定义

- **时间（计算）复杂性：** 考虑算法的主要操作步骤，计算执行中所需的总操作次数.
- **空间复杂性：** 执行过程中所需存储器的单元数目.
- **数据复杂性：** 信息资源 .

计算模型 ---- 确定性图灵机（有限带符号集合，有限状态集，转换函数）（读写头，读写带）.

算法复杂度的定义

不同的编程语言，不同的编译器导致执行一次操作的时间各不相同，为了方便不同算法比较，通常假定所有计算机执行相同的一次基本操作所需时间相同，而把算法中基本操作执行的最大次数作为执行时间 .

$$\text{运行时间} = \frac{\text{基本操作数量}}{\text{机器速度}}$$

算法复杂度的定义

北京郵電大學 BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

定义 假设一个算法的计算复杂度为 $O(n^t)$，其中 t 为常数，n 为输入问题的长度，则称这算法的复杂度是多项式的。具有多项式时间复杂度的算法为多项式时间算法.

函数 $g(n)=O(n^t)$ 表示存在常数 $c>0$ 和 $n_0>=0$，对一切 $n> n_0$ 均有 $|g(n)|<=c|n^t|$ 成立，也就是说，当 n 足够大时，$g(n)$ 存在上界.

定义 非多项式时间算法： 算法的计算复杂性

算法复杂度的定义

例 设 x 是小于 100 的某个整数，问 x 是否是素数？

解法1是否是多项式时间算法？

解法2是否是多项式时间算法？

本章主要内容

- 问题的定义及分类
- 算法复杂度定义及分类
- **P 问题和 NP 问题**
- 规约思想与 NPC 类
- 密码算法的计算安全性

P 问题和 NP 问题

- 定义（P 问题）如果一个判定问题存在解它的多项式时间的算法，则称该问题属于 P 类 .
- 定义（NP 问题） 如果一个判定问题不存在解它的多项式时间的算法，且对于一个解答可以在多项式时间验证其是否正确，则称该问题属于 NP 类 .
- **公开问题： P≠ NP ？**

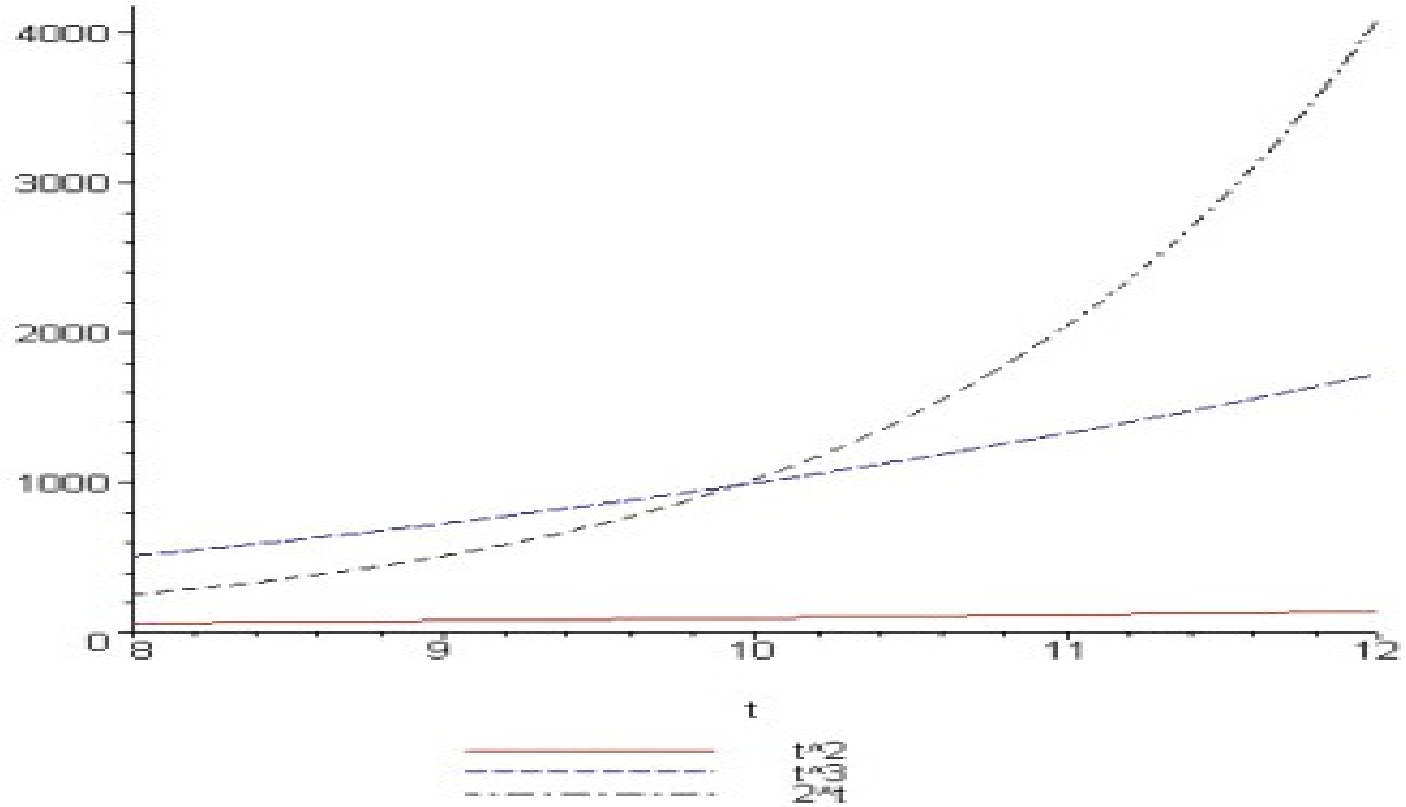
  它是 Clay 研究所的七个百万美元大奖问题之一

本章主要内容

- 问题的定义及分类
- 算法复杂度定义及分类
- P 问题和 NP 问题
- 规约思想与 NPC 类
- 密码算法的计算安全性

密码算法的计算安全性

二次函数、三次函数、2^x 函数的示意图



密码算法的计算安全性

例：

设问题输入长度为 n，在一个每秒钟运行百万次的计

算机上的运行时间如下：

	10	30	50	60
$T(n)=n^2$	0.0001s	0.0009s	0.0025s	0.0036s
$T(n)=2^n$	0.001s	17.9 月	35.7 年	366 世纪

密码算法的计算安全性

当问题输入长度足够大，分析密码体制的算法的复杂度较大，可能的计算能力下，在保密的期间内可以保证算法不被攻破，这就是密码体制的计算安全性思想。

密码算法的计算安全性

密码系统设计：

合法用户——易 （多项式）

攻 击 者——难 （非多项式）

密码算法的实际安全性

- 实际安全是指密码系统满足以下准则之一：

  - 破解该密码系统的成本超过被加密信息本身的价值；
  - 破译该密码系统的时间超过被加密信息的有效生命周期。

主要知识点小结

- 算法的复杂度定义及分类
- 密码算法的计算复杂度

THE END !