



西南大学

内部资料，
请勿外传。

算法引论-补充

张里博

lbzhang@swu.edu.cn



- 算法设计一般遵循的原则：自顶向下、逐步求精；
- 依赖于抽象数据类型等工具，算法的顶层设计与底层实现分离
- 进行顶层设计时不用考虑数据类型、运算表示和具体实现。算法的复杂性降低，条理性和模块化也增强了；
- 顶层设计与底层实现局部化，容易查找和纠正错误。



算法的描述

- 算法的描述方法有很多，常见的有：**自然语言、流程图和伪码**；
- 高级程序设计语言十分接近算法语言，但并不是算法的描述方式；
- 算法描述主要关注算法的主干部分，即算法的顶层设计，是求解问题的主要步骤；
- 此部分描述的首要目的是方便阅读、理解和交流



PART 05

NP问题





五、NP问题

- 计算复杂度理论中研究的主要内容之一，就是问题的内在复杂性，即问题是“易计算”，还是“难计算”？
- 如果知道了一个问题的时间复杂度的下界，就可以正确评价解决该问题算法的效率，以及现有最优算法上可以改进的空间；
- 常见的总共有四类问题：
- **P问题、NP问题、NPC问题和NPH问题。**



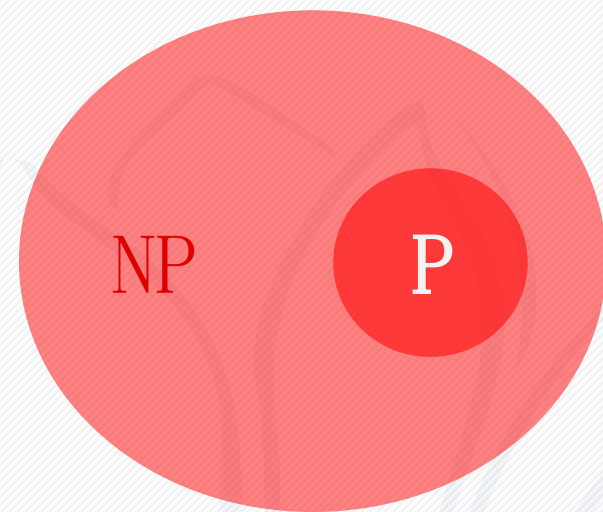
五、NP问题

- P (Polynomial-time) 问题
 - 能够在**多项式时间**内用算法**求解**的问题。
 - 存在多项式时间算法的问题被认为是“**简单问题**”；
- 一个问题的有效算法
 - 渐进时间复杂度是问题规模的**多项式或者更低阶函数**。



五、NP问题

- P(Polynomial-time) 问题
 - 能够在**多项式时间**内用算法求解的问题。
- NP(Nondeterministic polynomial-time) 问题
- 非确定性图灵计算模型
- 非确定性算法的
- 两个步骤：猜测（非确定性）和验证（确定性）
- **不确定是否存在**多项式时间的求解算法，但可以在**多项式时间**内**验证**一个猜测解的**正确性**的问题。



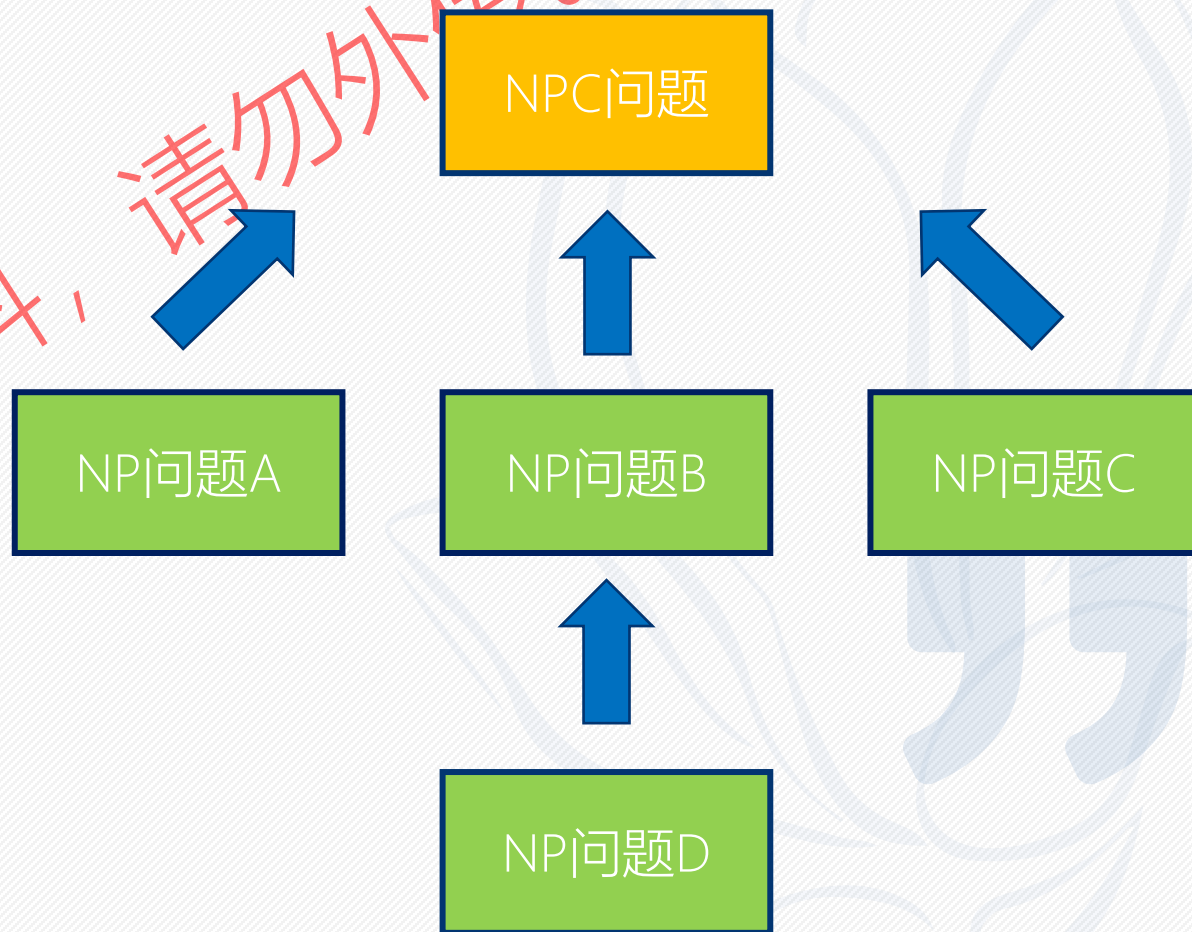
$$P \subseteq NP$$



五、NP问题

- NPC (Non-deterministic Polynomial complete problem) 问题
- 如果**所有NP问题**可在多项式时间内**归约成某个NP问题**，则该NP问题称为NP完全问题。

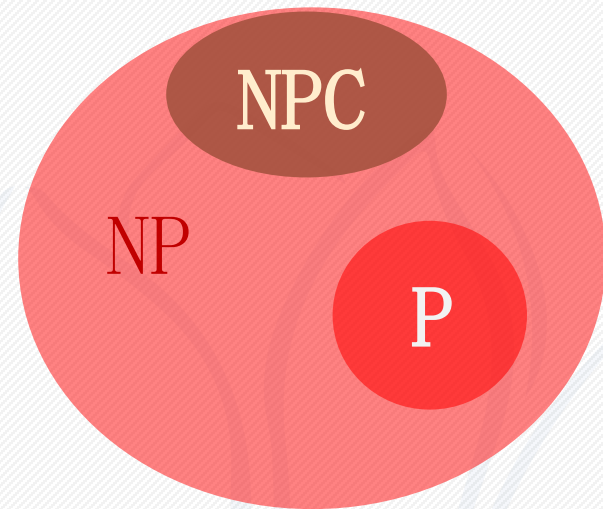
- NPC问题 (2个条件)
- ①自身是NP问题；
- ②其他NP问题能约化到它；





五、NP问题

- NPC (Non-deterministic Polynomial complete problem) 问题
- 如果**所有NP问题**可在多项式时间内**归约成某个NP问题**，则该NP问题称为NP完全问题。
- NPC的存在是 $P \neq NP$ 的关键
- 但是，NPC问题是否存在多项式时间的求解算法，**尚未得到证明**。若能证明存在，则 $P \neq NP$ ；若能证明不存在，则 $P = NP$ ；
- 一般认为 $P \neq NP$ ，NPC问题**是没有多项式时间**复杂度的求解算法，需要**指数级甚至更高阶时间**复杂度才能求解。



$$\begin{aligned} P &\subseteq NP \\ NPC &\subseteq NP \end{aligned}$$



五、NP问题

- NPH (Non-deterministic Polynomial hard problem) 问题
- 如果所有NP问题可在**多项式时间内转化**（归约，意思是解决了后者也就相应的解决了前者）成某个问题，则该问题称为NP难问题。
- 满足NPC问题的条件②但不满足条件①。
- NPH问题不一定是NP问题，有可能是**不可判定问题**，即无法在多项式时间内验证一个解的正确性。

