

- `toc: false`
- `toc-title: "目录"`

title: [13周--第一章]高级算法大模型辅助学习 author:

- name: 杨士康^[2024244062,计算机科学与技术]

`toc: false`

`toc-title: "目录"`

filters:

- `scholarly-metadata.lua`
- `author-info-blocks.lua`

Q01: NPC问题

- `prompt`: NPC问题为什么难以计算?
- `model_name`: kimi
- `model_version`: moonshot-v1-20240416
- `response`: NP完全（NP-Complete，简称NPC）问题难以计算的原因主要涉及以下几个方面：1、多项式时间算法的缺失：NPC问题目前没有已知的多项式时间算法能够解决。这意味着，对于NPC问题，我们无法在合理的时间内找到解决方案，因为随着问题规模的增长，所需的计算时间会呈指数级增长，变得不可行。2、约化概念：NPC问题满足两个条件：首先它是一个NP问题，其次所有的NP问题都可以约化（Reducible）到它。约化意味着如果一个问题A可以转换为另一个问题B，并且问题B可以在多项式时间内解决，那么问题A也可以在多项式时间内解决。因此，如果任何一个NPC问题可以在多项式时间内解决，则所有的NP问题都能在多项式时间内解决。3、指数级或阶乘级复杂度：NPC问题通常需要指数级甚至阶乘级的复杂度的搜索算法来求解，这种复杂度的增长使得问题随着规模的增加而变得难以计算。4、NP完全问题的定义：NPC问题定义为NP问题中最困难的问题，它们是那些在NP问题中，其他所有NP问题都可以在多项式时间内约化到它们的问题。因此，解决任何一个NPC问题都意味着所有NP问题的解决。5、计算复杂性理论：从计算复杂性理论的角度来看，NPC问题的难度与整个NP类的复杂性相关联。这些问题中任何一个如果存在多项式时间的算法，那么所有NP问题都是多项

式时间可解的。6、自然证明障碍：Razborov和Rudich的工作表明，任何自然地证明 $P \neq NP$ 的尝试都会产生一种快速算法，这种算法可以区分包含隐藏信息的伪随机字符串和真正的随机字符串，这与安全密码学的先决条件相矛盾。这意味着，如果安全密码学是可能的，那么自然证明就不能证明 $P \neq NP$ ，从而增加了NPC问题的计算难度。综上所述，NPC问题的难以计算主要源于缺乏有效的多项式时间算法、问题的约化性质、以及与整个NP类问题的复杂性相关联。这些因素共同构成了NPC问题难以计算的原因。

Q02: NP问题和P问题

- prompt: 简单地介绍一下NP问题和P问题。
- model_name: kimi
- model_version: moonshot-v1-20240416
- response: P问题是指可以在多项式时间内解决的问题。多项式时间意味着解决问题所需的时间与输入大小的多项式成正比。具体来说，如果一个问题的输入大小为 n ，那么解决这个问题所需的时间可以用 n^k 来表示，其中 k 是一个常数。多项式时间算法被认为是“有效”的，因为随着输入大小的增加，算法所需的时间增长是可控的。NP问题是指可以在多项式时间内验证一个解的问题。这意味着，如果你给了一个解，那么可以在多项式时间内检查这个解是否正确。NP问题包括所有P问题（因为如果一个问题可以在多项式时间内解决，那么它的解也可以在多项式时间内被验证），但还包括一些目前已知没有多项式时间算法的问题。简而言之，P问题是那些可以快速（多项式时间）解决的问题，NP问题是那些虽然可能难以快速解决，但其解可以快速验证的问题，NP完全问题是NP问题中最“难”的一类，而NP难问题则是一个更广泛的概念，包括了NP问题和一些其他难以解决的问题。

Q03: 程序等价性问题

- prompt: 不可计算问题中的程序等价性问题是什么？
- model_name: kimi
- model_version: moonshot-v1-20240416
- response: 不可计算问题中的程序等价性问题是指在计算理论中，确定两个程序是否在所有可能的输入和状态下都能产生相同输出的问题。这个问题是不可计算的，意味着不存在一个通用算法能够解决所有程序等价性问题。具体来说，这个问题涉及到以下几个方面：
外延封闭性：Rice's theorem 提到，外延封闭的可判定程序集只能是空集或全集。这意味着，如果我们想要确定一个程序集是否包含所有等价的程序，那么这个问题就变得不可判定。
程序等价的定义：程序等价性可以定义为在所有状态下，两个程序的求值结果相同。对于包含变量的表达式（aexp 和 bexp），如果它们在所有状态下的求值结果相同，我们就说它们是行为等价的。
Friedberg Numbering: 存在一种图灵完备的程序语言，使得任何两个程序的功能

都不一样，即任何功能只有唯一的程序实现。这种语言被称为 Friedberg numbering，它通过一个双变量可计算函数（充当解释器）来定义，满足对任意可计算函数 $f(x)$ ，存在唯一的自然数 e_f （即程序）使得 $\text{interp}(e_f, x) = f(x)$ 对任意自然数 x 恒成立。不可计算性：程序等价性问题的不可计算性意味着，尽管理论上存在确定两个程序是否等价的方法，但实际上我们无法构造出一个算法来在有限时间内解决所有情况下的程序等价性问题。综上所述，程序等价性问题是不可计算问题中的一个典型例子，它涉及到计算理论中的一些基本问题，如可判定性、可计算性以及程序行为的等价关系。

Q04: Boolean satisfiability problem

- prompt: Boolean satisfiability problem是什么问题?
- model_name: kimi
- model_version: moonshot-v1-20240416
- response: Boolean satisfiability problem（布尔可满足性问题，简称SAT问题）是计算机科学和逻辑学中的一个经典问题。它属于NP完全问题，这意味着它是NP类问题中最难的问题之一，并且已知没有多项式时间算法能够解决所有的SAT实例。SAT问题的定义如下：给定一个布尔公式（由布尔变量、逻辑与（AND）、逻辑或（OR）和逻辑非（NOT）构成的表达式），问题是判断是否存在一种变量赋值，使得整个公式的值为真（即“可满足”）。例如，考虑以下简单的布尔公式： $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$ ，这个公式可以被满足，因为如果我们赋值 $x_1 = \text{false}$ 和 $x_2 = \text{true}$ ，那么整个表达式的值为真。SAT问题的重要性在于它作为一个理论模型，用于研究计算复杂性理论中的许多问题。由于SAT问题是NP完全的，这意味着如果找到了一个多项式时间算法来解决SAT问题，那么所有的NP问题都可以在多项式时间内解决，这将是一个重大的数学和计算机科学突破。然而，到目前为止，还没有发现这样的算法，而且普遍认为不存在这样的算法。

Q05: P=NP问题的证明进展

- prompt: 关于目前P=NP问题的证明，前沿的进展如何?
- model_name: kimi
- model_version: moonshot-v1-20240416
- response: 1、元复杂性研究：计算机科学家正在探索元复杂性问题，这涉及到评估问题的复杂性。2023年春天，加州大学伯克利分校的西蒙斯研究所举办了关于这个主题的研究项目，展示了复杂性度量之间的联系，并发现了元复杂性与计算机科学其他领域如学习理论和密码学之间的深层联系。2、人工智能的应用：有研究者正在利用人工智能的新方法，从信息论的角度重新构思P vs NP问题，探索机器学习如何与P vs NP相啮合。3、量子计算的局限性：尽管量子计算被认为可能对解决NP完全问题有帮助，但目前的证据表明，除了由格罗弗算法给出的二次加速改

善，量子计算机无法解决NP完备问题。4、复杂性理论的进展：在理解高效计算能力方面取得了几项重大进步，尽管这些结果对于解决P/NP问题没有带来显著的进展，但依旧展现了它们是如何继续启发后续优秀研究的。5、GPT-4与 $P \neq NP$ ：有报道称，通过97轮的“苏格拉底式推理”，GPT-4得出了“ $P \neq NP$ ”的结论，这显示了大型语言模型在科学研究和科学发现中的潜力。6、最小电路门数问题

(MCSP)：近年来，研究人员在证明MCSP是NP完全的这一方面取得了重大进展。特别是，东京国家信息研究所的Shuichi Hirahara（平原秀一）取得了一些进展，而麻省理工学院的Rahul Ilango（拉胡尔·伊兰戈）也致力于证明MCSP的NP完全性，将MCSP与布尔可满足性问题（SAT）联系起来，这是一个已知的NP完全问题。综上所述，尽管 $P=NP$ 问题仍未解决，但研究人员正在从多个角度进行探索，包括元复杂性、人工智能的应用、量子计算的局限性、以及特定问题的NP完全性证明。这些研究正在逐步推进我们对 $P=NP$ 问题的理解。