

AI 生成代码背景下算法课程的“理解导向”教学重构

王晓川 王晨 刘彩霞 李海生**

北京工商大学计算机与人工智能学院, 北京 100141

摘要 针对生成式人工智能编程助手普及背景下算法课程教学面临的适应性危机, 探索从“代码实现”向“正确理解代码”转型的教学重构路径。分析传统“三会、三题、三考”教学模式导致的课程目标错位、能力培养局限与评价方式固化问题, 提出“理解导向”的算法课程教学重构方案, 构建以“能溯源、能比较、能审计”为目标、“问题解构、算法溯源、AI 对抗”为活动、“课堂研讨、过程档案、多维量规”为评价的三维改革框架。教学实践覆盖计算机科学与技术专业两个班共 78 名学生。结果表明, 学生课堂发言率与研讨活动参与率较上届提升约 3 倍, 算法分析题平均得分提升 14%, 课程满意度由 95.6 提升至 96.1。实践证实, 理解导向的教学重构在 AI 辅助编程背景下具有可行性与有效性, 能够针对性提升学生的算法理解能力。

关键字 理解导向, 教学目标, 教学活动, 算法分析与设计

Understanding-Oriented Reconstruction of Algorithm Courses in the Era of AI-Generated Code

Xiaochuan Wang Chen Wang Caixia Liu Haisheng Li*

School of Computing and Artificial Intelligence
Beijing Technology and Business University
Beijing 100141, China

Abstract—In response to the adaptive crisis in algorithm course instruction amid the widespread adoption of generative AI programming assistants, this study explores a teaching reconstruction pathway shifting from "code implementation" to "correct understanding of code." It analyzes how the traditional "three abilities, three topics, three assessments" teaching model leads to misaligned course objectives, limited skill development, and rigid evaluation methods. A "comprehension-oriented" reform framework for algorithm courses is proposed, structured around three dimensions: goals focused on "tracing origins, comparing approaches, and auditing logic"; activities centered on "problem deconstruction, algorithm tracing, and AI adversarial learning"; and assessment based on "classroom discussions, process portfolios, and multidimensional rubrics." The teaching practice involved 78 students across two classes in the Computer Science and Technology program. Results show that student participation rates in classroom discussions and collaborative activities increased approximately threefold compared to the previous cohort, average scores on algorithm analysis questions improved by 14%, and course satisfaction rose from 95.6 to 96.1. The practice confirms that comprehension-oriented teaching reform is both feasible and effective in the context of AI-assisted programming, significantly enhancing students' algorithmic understanding.

Keywords—understanding-oriented, instructional objectives, teaching activities, algorithm design and analysis

1 引言

算法设计与分析课程是计算机类本科专业的核心课程, 是培养学生计算思维、算法设计与分析能力的重要载体, 也是深化新工科建设、推动本科教育适应人工智能时代发展的关键基础。近年来, 以GitHub Copilot、ChatGPT、Claude等为代表的生成式人工智能编程助手迅速普及, 其代码生成能力已达到甚至超越普通本科生的编程水平, 这对传统以“代码实现”为核心目标的算法课程教学提出了严峻挑战。

现有算法课程教学实践主要集中于教学内容优化、教学模式改革、课程思政融合与评价体系完善等方面^[1]。在教学内容方面, 研究者强调计算思维的系统性培养, 注重算法理论与问题求解能力的平衡^[2]; 在教学模式方面, 翻转课堂、研讨式教学、OJ (Online Judge) 驱动等模式被广泛采用, 以提升学生的课堂参与度与实践能力^[3-6]; 在课程思政方面, 陆玲等^[7]构建“三协同四融合”的思政教学体系, 将算法思想与哲学智慧、算法实例与民族自信、算法实现与职业素养、算法应用与社会责任进行深度耦合, 实现了价值塑造与知识传授的有机统一。在评价方式方面, OJ平台与过程性考核相结合, 构成了主流的课程质量评价体系^[8]。

**通讯作者: 李海生 lihsh@btbu.edu.cn。

上述探索显著提升了课程教学质量,然而,在生成式人工智能快速迭代的背景下,传统教学范式面临新的适应性危机。卜东波^[9]基于教学实验指出,大语言模型已能自主设计并通过算法考核,算法课程必须转向核心思维与验证能力的培养;赵明华^[10]等探索了生成式AI+项目驱动的程序设计课程实验教学方式;王一凡等^[11]证实ChatGPT可自动生成算法习题、解决方案并完成智能评阅;陈晶玉^[12]在数据结构与算法课程中验证了AI对备课、答疑与编程辅助的全环节渗透;尹良泽等^[13]分析了AI编程助手对程序设计课程的深刻冲击,指出学生借助AI即可完成知识问答、代码生成与自动纠错。王帅等^[14]进一步揭示,大模型的“即问即答”特性实质上构建了认知捷径,导致学习者跳过问题解析的深度思考,陷入“代码生成即终结”的认知断裂,难以建立从问题约束到算法策略的推导能力。更为深层的问题在于,若课程目标仍停留在培养编码熟练度,算法课程将与程序设计入门课程同质化,丧失其作为高阶专业核心课程的独特价值。

基于此,本文提出“理解导向”的算法课程教学重构思路,从教学目标、课堂活动与考核评价三个维度,系统探讨人工智能生成代码背景下算法课程从“写出正确代码”向“正确理解代码”的转型路径。通过构建能溯源、能比较、能审计的新教学目标,设计问题解构、算法溯源、AI对抗的新型课堂活动,建立课堂研讨、过程档案、多维量规的多维评价体系,为算法课程在人工智能时代的教学重塑提供实践参考。

2 现状与问题分析

2.1 课程目标的错位:从算法思维滑向编码熟练度

传统算法课程的目标体系长期因循编码实现的逻辑,将“会写、会调、会AC”作为核心能力标准。所谓会写,即要求学生能够独立手写算法代码;会调,即通过调试处理边界数据与异常情况;会AC,即最终通过OJ系统的全部测试点。上述目标在程序设计能力培养中具有基础性价值,然而,当生成式人工智能已能秒级生成高质量代码并自动处理常见边界条件时,这一目标体系的区分度与培养价值急剧下降。

更为深层的问题在于,算法课程与程序设计入门课程在目标层面出现同质化倾向。算法设计与分析课程的核心价值本应在于培养计算复杂性直觉、问题结构化能力与算法创新思维,而非单纯的编码熟练度。当课程目标过度聚焦于写出正确代码,算法理论中的主方法证明、贪心最优性论证、动态规划状态转移方程推导等高阶内容往往被压缩或弱化,导致课程降维为高级编程训练。在人工智能辅助编程普及的背景下,这种目标错位使得算法课程面临价值危机,课程难以

对其真实理解水平进行有效度量。

2.2 能力培养的局限:从算法设计退化为题型复现

受限于上述目标体系,算法课堂的活动设计长期被困于“例题、习题、考题”的闭环。在例题环节,教师通常演示标准解法,学生以观摩记忆为主,往往只见解法而不见问题结构,难以建立从问题约束到算法策略的推导能力。在习题环节,学生课后通过大量刷题巩固模板,陷入见题套算法的机械复现模式,对算法适用的边界条件、时空权衡与变种问题的迁移能力培养不足。在考题环节,期末考核侧重于实现速度与准确率,无法真实反映学生对算法本质的理解深度。

这一“三题”闭环在人工智能环境下被进一步放大。由于AI工具能够直接生成通过测试的代码,学生甚至跳过复现环节,将刷题异化为刷提示词。其后果是,学生虽能提交正确代码,却无法针对具体数据特征选择并论证最优算法,更难以在问题结构发生变化时进行算法调整。长此以往,算法课程所应培养的算法设计者角色被矮化为“题型复现者”,甚至“AI指令操作员”,课程在能力培养层面的局限性愈发凸显。

2.3 产出评价的固化:从过程理解坍缩为结果判定

与上述目标与活动相对应,现有算法课程的评价体系主要依赖“考勤、考查、考试”的三维结构,且三者均呈现结果化、单一化的特征。考勤环节侧重于出勤行为管理,难以反映学生的课堂思维参与度;考查环节以平时作业与OJ通过率为依据,但在人工智能辅助下,OJ的“Accepted”结果已无法区分独立实现与AI代写;考试环节无论是限时手写代码还是上机测试,均只能测量学生的瞬时记忆与编码熟练度,对算法论证、复杂度推导、多方案比较等理解性能力缺乏有效测评手段。

这种“三考”体系本质上将评价维度坍缩为单一的结果判定,对算法理解的过程性证据完全失明。由于评价是指棒,学生自然倾向于选择成本最低的路径达成考核要求,借助AI工具完成代码任务便成为理性策略。评价的固化不仅掩盖了教学过程中的真实问题,更在客观上激励了重结果、轻理解的行为取向,使得算法课程陷入“目标错位—活动局限—评价固化”的恶性循环。因此,从“理解导向”出发重构教学目标、课堂活动与评价体系,已成为算法课程适应人工智能时代的必然选择。

3 理解导向的算法教学重构

针对上述“三会、三题、三考”的固化倾向,算法课程的教学重构不应止步于局部修补或简单的AI工

具禁用，而需进行系统性的范式转换。理解导向的教学重构以“正确理解代码”替代“写出正确代码”作为课程核心价值，从教学目标、课堂活动与考核评价三个层面同步发力，形成相互支撑的改革闭环，如图1所示。

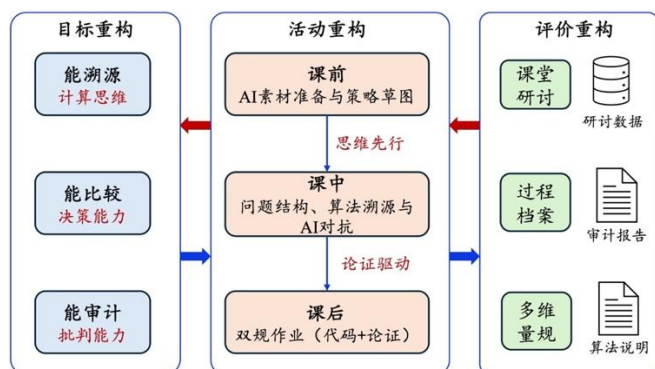


图1 理解导向的算法教学方案设计示意

3.1 超越“三会”的教学目标重构

理解导向的教学重构首先需要在目标层面实现从“编码熟练度”到“算法理解力”的跃迁。新的目标体系以“能溯源、能比较、能审计”为支柱，将课程的核心价值从培养独立编程能力转向培养计算思维与算法审美。其中，“能溯源”强调学生应具备从问题约束推导算法结构的能力。面对具体应用场景，学生能够识别数据规模、分布特征、实时性要求等关键约束，进而判断应采用何种算法策略，而非机械套用已知模板。“能比较”要求学生掌握多方案的严格论证方法。传统教学中，算法比较往往停留在时间复杂度分析层次；理解导向则要求学生从时间复杂度、空间复杂度、稳定性、实现难度、适用边界等多维度进行系统权衡，在约束条件下寻求最优折中。“能审计”是人工智能时代的新增维度，要求学生具备批判性审视AI生成代码的能力。学生应能识别AI代码中的隐藏漏洞（如边界处理缺失、稳定性破坏、未考虑的特殊输入分布），判断其复杂度是否真正最优，并能在必要时进行结构性修正。

3.2 超越“三题”的课堂活动重构

传统课堂以例题讲解、习题训练、考题模拟为活动主轴，学生处于被动接受与机械复现的位置。理解导向的课堂活动通过“问题解构、算法溯源、AI对抗”三种新型范式，将课堂重心从算法复现转向算法设计。具体表现为：

问题解构替代例题讲解。教师不再直接给出标准解法，而是呈现问题场景（如“十亿级几乎有序数据流中的Top-K查询”），要求学生先绘制问题结构图，识别约束条件与优化目标，再自主推导可能的算法路径。这一活动迫使学生在接触具体代码之前，先建立对问

题本质的结构性理解，避免见题套模板的思维惰性。

算法溯源替代习题训练。针对经典算法，课堂活动聚焦于“消融”与“溯源”——若去掉某个前提条件（如非负权、独立性、稠密图假设），算法结构应如何调整？以最短路径为例，学生需论证Dijkstra算法在负权图下的失效机理，并推导Bellman-Ford算法的补偿策略。这种活动将习题从代码熟练度训练转化为算法原理的迁移与验证。

AI对抗替代考题模拟。教师利用人工智能编程助手生成针对特定问题的代码，要求学生在课堂上进行分组审计。AI生成的代码往往表面正确但暗藏漏洞——例如，用标准Dijkstra处理含负权边的图、或用朴素递归求解重叠子问题严重的动态规划实例。学生需找出理论漏洞、分析复杂度缺陷，并给出修正方案。该活动将AI的威胁性存在转化为教学资源，使批判性审视成为可训练、可考核的课堂行为。

3.3 超越“三考”的考核评价重构

在传统考核评价方式的基础上，理解导向的评价重构建立“课堂研讨、过程档案、多维量规”的三维体系，具体包括：

课堂研讨。通过观察量表记录学生在问题解构与AI审计中的参与度，评价指标包括问题分析的准确性、对同伴方案的批判性反馈、以及算法论证的逻辑严密性。该维度将课堂出勤行为转化为可观测的思维参与证据。

过程档案。要求学生持续提交算法论证日志、AI审计报告与错题反思录。例如，每次课后需针对AI生成的代码撰写审计文档，说明其时间复杂度、空间复杂度、适用假设，并指出至少一处潜在风险。这些档案构成理解能力的纵向追踪数据。

多维量规。期末考核侧重复杂度推导、正确性证明与多方案比较；OJ部分不仅判定结果，还强制要求提交算法说明文档，标注关键不变式与复杂度依据。代码实现环节引入注释质量分与算法说明文档，未通过文档审核的提交即使AC也不获满分。这一设计从机制上迫使学生必须“理解”而非“复制粘贴”。

4 理解导向的算法教学实践路径

本节以动态规划单元为例，介绍课程团队实践理解导向算法教学的实践路径，并对实施效果进行初步验证与反思。

4.1 教学设计

动态规划教学过程中，最优子结构与重叠子问题的特征恰好构成检验学生算法理解力的理想场景。表1展示了传统教学与理解导向教学在教学设计上的具

体差异。

表 1 传统教学与理解导向教学设计对比

教学设计	传统教学	理解导向教学
教学目标	编码熟练度（会写、会调、会AC）	算法理解力（能溯源、能比较、能审计）
课前环节	自主预习课本内容	完成预研任务单，绘制问题结构图
课中环节	例题讲解→习题训练→考题模拟	问题解构→算法溯源→AI对抗
课后作业	手写代码提交OJ系统	双轨作业（可执行代码+算法论证文档）
评价方式	OJ通过率+期末笔试	课堂研讨+过程档案+多维量规

在上述设计中，理解导向并非取消代码实现，而是重构了“理解”与“编码”的先后顺序。学生必须在课前完成对AI代码的批判性阅读，在课中完成对算法结构的溯源性推导，方能在课后进行有意识的编码实践。

4.2 实施流程

理解导向教学的实施在CDIO的框架下，遵循“课前预研—课中对抗—课后论证”的三阶段流程，每一阶段均有明确的角色分工与交付物要求，如表2所示。

表 2 理解导向教学三阶段实施流程

阶段	教师任务	学生任务	交付物
课 前 预研	生成AI瑕疵代码素材库，发布问题场景与预研任务单	提交算法思路草图（含约束识别、策略比较、复杂度分析）	预研任务单
课 中 对抗	引导识别约束条件与策略因果关系，记录课堂参与度	展示课前草图，推导状态转移方程，审计AI代码漏洞	课堂研讨记录
课 后 论证	批改代码与论证文档	提交双轨作业，回答四个论证问题	可执行代码+算法论证文档

课前预研阶段。教师首先利用人工智能编程助手针对教学重点生成多份表面正确但暗藏瑕疵的参考代码，作为课堂对抗的素材库。随后，发布问题场景与预研任务单，要求学生提交算法思路草图。该草图须包含三个要素：问题约束识别、候选策略比较、预期复杂度分析。此阶段明确禁止学生编写可执行代码，以防止AI代写对预研环节的干扰。

课中对抗阶段。课堂时间按“问题解构—算法溯源—AI对抗”三模块分配，每模块约20分钟。在问题解构模块，学生展示课前草图，教师引导识别约束条件与策略选择的因果关系；在算法溯源模块，学生从子问题结构推导状态转移方程的必然形式，而非背诵标准模板；在AI对抗模块，学生分组审计教师提供的

AI代码，找出理论漏洞，并现场提出修正方案。课堂研讨的参与度通过观察量表实时记录，纳入过程性评价。

课后论证阶段。学生提交双轨作业：可执行代码与算法论证文档。论证文档须回答四个问题：（1）状态定义为何选择此种而非彼种形式；（2）时间复杂度下界如何推导；（3）AI代码与最终代码的差异及修正理由；（4）若问题约束发生变化（如从0/1背包变为完全背包或多维约束），当前结构应如何调整。

5 效果验证与反思

课程团队于2024—2025学年秋季学期在算法设计与分析课程中试点上述方案，覆盖计算机科学与技术专业两个班共78人。通过对比试点教学与上一届传统教学在考核中的表现，得到以下初步结论。

课堂参与度提升。与上一届相比，本届学生课堂发言率与研讨活动参与率显著提升。基于飞书协同平台的课堂行为数据统计，学生在算法溯源与AI对抗环节的评论与标注数量相较上一届提升了约3倍。

考核成绩结构改善。与上一届相比，本届学生在算法分析题（论证题、计算题、应用题）上的平均得分较去年提升14%。印证了理解导向教学对算法理解能力的针对性提升。然而，编程实现题的得分较去年有所下降，说明AI辅助生成代码确实冲击了学生对于代码本身的熟悉程度。

课程满意度向好。学评教结果显示，学生整体满意度较上一届有所提升（95.6到96.1）。课程满意度问卷显示，63%的学生认可新的教学活动；47.4%的学生认为考评方式加深了对算法的理解。同时，46%的学生表现出一定的抵触清晰，认为新的教学活动和文档考评增加了认知负担。

综上，理解导向的算法教学重构在人工智能辅助编程的背景下具有一定的可行性与有效性，但需关注两方面平衡：一是理解深度与编码训练的比例平衡，避免从一个极端滑向另一个极端；二是教师投入与教学产出的成本平衡，需建立可复用的案例库以降低持续实施门槛。

6 结束语

算法设计与分析课程是计算机类本科专业体系中承上启下的核心课程，其教学效果直接影响学生计算思维与问题求解能力的养成。在生成式人工智能快速迭代的背景下，传统以“代码实现”为核心的教学目标、课堂活动与评价体系面临适应性危机，课程价值有被技术替代而稀释的风险。本文针对上述问题，提出“理解导向”的算法课程教学重构思路，构建了以“能溯源、能比较、能审计”为目标引领，以“问题

解构、算法溯源、AI对抗”为活动载体，以“课堂研讨、过程档案、多维量规”为评价支撑的教学范式。未来，课程团队将进一步扩大实践范围，深化人工智能技术与算法教学的融合机制，推动算法课程从“培养编码熟练度”向“培养计算思维与算法审美”的高阶目标持续转型。

参 考 文 献

- [1] 王忠, 侯博, 刘延飞. “内容为基, 方法为要, 思政为纲, 考评为导”的算法设计与分析课程教学改革[J]. 计算机教育, 2026(1): 170-174.
- [2] 乔丽娟, 周子力, 高鹏, 等. 新工科背景下算法设计与分析课程的教学改革与实践[J]. 曲阜师范大学学报(自然科学版), 2026, 52(1): 124-128.
- [3] 柳欣, 胡晓鹏, 李保田, 等. 探究式实验驱动的算法设计与分析课程改革与实践——以创新能力培养为导向[J]. 大学教育, 2025(24): 71-75.
- [4] 陈艳. 案例驱动式教学法: 传统与 AI 算法融合的教学创新探索[J]. 信息与电脑, 2026(7): 213-216.
- [5] 陈致超, 董雯. 知识图谱与智能体融合的“算法设计与分析”课程智慧化教学实践[J]. 工业和信息化教育, 2026(2): 61-65.
- [6] 柳欣, 陈龙溪, 李保田, 等. 面向创新能力培养的探究式实验教学改革——以“算法设计与分析”课程为例[J]. 工业和信息化教育, 2026(3): 80-86.
- [7] 陆玲, 李新建, 李亚珍, 等. 基于“三协同四融合”的算法设计与分析课程思政教学探索[J]. 计算机教育, 2026(4): 88-92.
- [8] 冯艳娜, 张峰, 吴敏宁. 应用型本科高校人才培养成效数智化评价方法研究[J]. 计算机技术与教育学报, 2026(1): 12-17.
- [9] 卜东波. AI 时代的算法课——从一个教学小实验谈起[J]. 计算机教育, 2025(6): 1-5.
- [10] 王一凡, 邹文龙, 刘梦君. 生成式 AI 支持的人机协同智能教学及应用探索——以 ChatGPT 协同算法教学与智能评阅为例[J]. 中国高校科技, 2025(6): 70-78.
- [11] 赵明华, 张朝晖, 王秀青. 生成式 AI+项目驱动的程序设计类课程实验教学改革与实践[J]. 计算机技术与教育学报, 2026(1): 39-44.
- [12] 陈晶玉. 生成式 AI 工具在数据结构与算法课程中的教学应用探究[J]. 信息与电脑, 2025(13): 176-178.
- [13] 尹良泽, 徐建军, 李姗姗, 等. 人工智能时代下的计算机程序设计课程教学探索[J]. 计算机教育, 2025(2): 123-127.
- [14] 王帅, 杨大智, 盛浩, 等. AI 辅助编程教学中思维链式启发策略探索[J]. 计算机教育, 2026(2): 143-147.