

嵌入式思政教学在《数据结构与算法》课程中的设计与实践*

王燕** 孙涛

内蒙古大学计算机学院, 呼和浩特 010021

摘要 数据结构与算法在计算机类专业的课程体系中处于承上启下的位置, 它既要训练学生的计算思维和算法设计分析能力, 也担负着价值引领的责任。但实际教学中往往算法效率讲得多、伦理层面考量少, 结果对不对看得重、过程严不严看得轻。本文选取“排序算法的稳定性”和“递归算法的栈空间管理”这两个经典知识点作为切入点, 依托情境讨论、角色扮演、代码走查及反思写作等实践环节, 把公平正义意识、科学精神与责任担当融进具体的教学环节。实践过程数据表明, 该嵌入式教学有效提升了学生的算法伦理、严谨态度与系统责任感, 为同类课程思政融入提供了可借鉴的样板。

关键词 数据结构与算法; 课程思政; 排序稳定性; 递归; 教学成效

Design and Practice of Embedded Ideological and Political Teaching in the Data Structures and Algorithms Course*

Yan Wang** Tao Sun

College of Computer Science, Inner Mongolia University
Hohhot 010021, China
cswy@imu.edu.cn

Abstract—"Data Structures and Algorithms" occupies a pivotal, transitional position in the computer science curriculum. While it is tasked with cultivating students' computational thinking and algorithmic design and analysis skills, it also bears the responsibility of value guidance. However, in actual teaching practice, there is often an overemphasis on algorithmic efficiency at the expense of ethical considerations, and a disproportionate focus on the correctness of results over the rigor of the process. Taking two classic concepts—"the stability of sorting algorithms" and "stack space management in recursive algorithms"—as entry points, this paper integrates values such as fairness and justice, scientific spirit, and a sense of responsibility into specific teaching modules. This is achieved through practical activities including scenario discussions, role-playing, code reviews, and reflective writing. Practical process data indicates that this embedded teaching approach effectively enhances students' algorithmic ethics, rigorousness, and systemic responsibility, providing a replicable model for integrating ideological and political education into similar courses.

Keywords—Data Structures and Algorithms, Curriculum Ideological and Political Education, Sorting Stability, Recursion, Teaching Effectiveness

1 引言

数据结构与算法一般安排在大二开设, 是计算机科学与技术、软件工程、人工智能等专业的核心基础课。学生走到这一步, 基本的程序设计能力已经具备, 接下来要系统地学怎么组织数据、怎么设计算法、怎么分析效率。本课程旨在引导学生掌握线性表、树、

图、哈希表等经典数据结构, 以及排序、查找、递归、动态规划、图算法等核心算法; 但更为关键的是, 要重点锤炼抽象建模能力、复杂度分析能力, 并同步培养工程判断力与伦理反思意识。

换个角度, 从思政育人上看这门课, 它其实有别的课未必具备的优势。算法早已渗进生活的各个角落——搜索引擎给出的结果排序、推荐系统的排名、银行的征信评分、招聘时的简历筛选, 背后的设计选择直接牵动着社会资源怎么分配、个人权利能不能落实。课程里绕不开的“稳定性”“最优性”“公平性”“资

***基金资助:** 内蒙古自治区级一流本科课-校园网组网与配置虚拟仿真实验, 校级课程思政标杆课—数据结构与算法(内大教字〔2024〕30号)。

****通讯作者:** 王燕 cswy@imu.edu.cn

源消耗”这些概念，都能当作思政的切入口。但学生写算法时，眼睛通常只盯着正确性与时间复杂度，很少停下来问一句：这个排序规则公平吗？这个递归深度会不会把系统资源耗光？这个数据结构选得是不是浪费了宝贵的内存？恰恰是这些追问，构成了价值引领最好的起点^[1]。

2 教学中的价值缺位及其问题

目前数据结构与算法的教学有几个比较突出的问题。首先是算法伦理这一块关注得太少。拿排序算法举例，学生关心的通常只有时间复杂度有没有达到 $O(n\log n)$ 、空间复杂度是多少，很少有人会停下来琢磨“稳定性”这个技术属性究竟有什么社会意涵——两个元素关键字相等的时候，算法保不保留它们原来的先后？这个听上去纯技术的选择，一旦放进高考录取同分排序、招聘简历排序、搜索结果排名这类场景，牵扯的其实是公平与正义^[2]。我们把学生训练成了“效率至上”的算法工程师，却很少要求他们回过头想想算法带来的社会后果。

其次，“严谨性”与“责任意识”的培养一直偏弱。递归是这门课公认的难点，学生写出无限递归把栈撑爆，或者因为递归太深让程序直接崩掉，都不算稀奇事。可在线评测系统的测试用例规模一般都不大，这类错误往往就被遮过去了。栈空间是有限的，一个递归函数在最坏情况下会调用到多深，学生心里没数，更谈不上主动去估算系统资源的边界^[3]。“只要样例能过就行”的心态，和工程实践中“对资源负责”的要求正好相反。

还有一点出在评价体系上：它偏重功能是否正确、效率指标是否达标，过程规范与伦理反思基本不在考核范围内。实验报告往往贴上代码和运行截图就算交差，很少有教师会要求学生分析自己的算法在公平性、资源消耗上做了哪些取舍。时间一长，“跑通即胜利”的功利心态也就形成了^[4]。

这几个问题看着各自独立，根子其实是同一个：教学中缺少对“算法价值负载”“工程责任”“资源伦理”的系统引导。而这恰好是课程思政能够发力的地方^[5-6]。

3 思政教学案例设计

本文从数据结构与算法课程里挑出两个核心知识点——“排序算法的稳定性”和“递归算法的栈空间管理”，分别对应公平正义意识、科学精神与责任担当这两组思政主题。第一个案例落在排序的“稳定性”上，借高考录取同分排序这个情境展开讨论，让学生看清算法设计选择背后藏着的公平性判断^[7]；第二个案例讲递归函数的栈溢出风险，用真实案例分析加代码走查的方式，把严谨态度和对系统资源的责任感带

出来。下面按教学目标、流程、场景设计、学生反馈与反思的顺序，逐一展开。

3.1 案例 1：排序算法中的“稳定性”与公平性讨论

(1) 学情分析与教学目标

排序算法是这门课的教学重点之一，学生要掌握冒泡、插入、归并、快速、堆排序等一系列算法的原理和复杂度分析。“稳定性”是其中一个重要概念：关键字相等时，稳定排序会保持元素原有的相对顺序，不稳定排序则不作这个保证。可实际教学中，教师往往只要求学生记住哪几种排序稳定、哪几种不稳定，至于稳定性在真实应用里究竟意味着什么，很少展开。

课前的课堂问答印证了这一点：面对“假如让你设计高考录取系统，几名考生总分完全相同，排序规则该怎么定”这个问题，相当一部分学生的回答是“随便排，反正不影响录取结果”。这个回答本身就说明，公平性这一维度在学生的算法设计意识里基本是缺席的。

本案例的教学目标分知识技能和思政两条线。知识技能这条线，要求学生弄清稳定排序与不稳定排序在技术上的区别，能结合应用场景选出合适的算法，并掌握通过改写比较函数来保持或破坏稳定性的做法。思政这条线指向另外几点：算法设计不是价值中立的，工程师的每一次选择都带着社会影响；技术决策中要养成主动权衡公平性的习惯；程序正义与实体正义之间是什么关系，也值得想一想。

(2) 教学流程设计

本案例嵌在“排序算法”这一章里，大约占用 1 学时（45 分钟）。整个流程走五个阶段，各阶段与排序算法选择的对应关系见图 1。

①问题导入（约 5 分钟）

课堂一开始，教师并不直接给稳定性下定义，而是先抛出一个情境：“某高校进行自主招生，综合成绩相同的考生要按‘面试成绩’排序来决定录取先后。你作为系统开发人员，这个排序怎么实现？要是用了快速排序，会出什么岔子？”学生讨论一轮之后，教师摆出一个很简单的例子：原始数据本来就按面试成绩从高到低排好，也就是说原始顺序已经体现了面试的优先级。走一遍稳定排序（比如归并排序），综合成绩相同的考生之间仍然保持着面试成绩的降序；换成不稳定排序（比如快速排序），同分考生的顺序就可能被打乱，面试成绩高的反倒排到了后面。教师接着追问：“这公平吗？如果有学生因为你选的算法被排到后面、丢掉了录取机会，这个责任谁来担？”到

这一步，学生开始意识到，一个看起来“无关紧要”的技术选择，放到特定场景下是有真实伦理后果的。

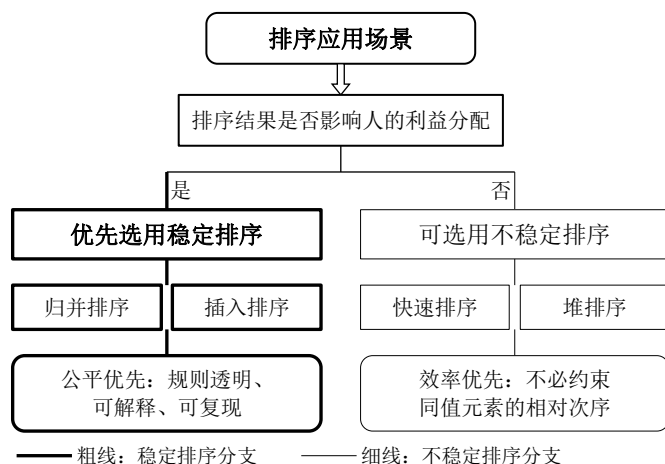


图1 排序算法稳定性与公平性决策图

②技术讲解（约15分钟）

接下来教师系统讲解排序稳定性的技术原理，借助图1把稳定排序（冒泡、插入、归并）与不稳定排序（快速、堆、选择）的适用场景摆在一起对比，重点讲清楚快速排序为什么会在分区时交换非相邻元素、稳定性又是怎么被破坏掉的。这里还要补一条工程上的常识：不少实际应用中稳定性属于硬性需求，像电子表格的多级排序（先按A列排序，再按B列排序时保持A列相对顺序），就必须用稳定排序。

随后教师把问题再往前推一步：既然稳定排序做得出来，为什么还要保留不稳定排序？效率与公平之间，哪些场合可以让步，哪些场合必须先保公平？目的是让学生明白，工程里的权衡从来不是单纯的技术问题，它同时也是价值判断。举例来说，金融交易系统里公平性的优先级高于那点微小的性能提升；而在内部的数据处理任务中，效率反倒可能排在第一位。

③角色扮演（约20分钟）

这一环节是本案重心。教师设计了一个模拟场景：某省教育考试院打算开发新的录取系统，排序规则上有两种备选方案。方案A是总分相同的考生按随机顺序录取，顺序由计算机随机生成；方案B则是总分相同的考生按单科成绩（如数学、语文）依次排序录取。接下来召开一场“规则听证会”，听各方怎么说。

学生分成四个角色小组：教育公平专家，主张规则要透明、可解释；学生家长代表，担心随机性带来不公；技术开发人员，关心实现复杂度和性能；教育行政部门官员，关注执行成本与可操作性。

听证会按各方陈述立场、自由辩论、投票表决的顺序推进。教师担任主持人，会在关键处插问几句：

“随机排序是不是就等于每个人机会均等？”“按单科排序，会不会引入新的歧视？”“实现起来麻烦一点，是否足以否掉一个更公平的方案？”

辩着辩着，学生慢慢意识到一件事：所谓“绝对公平”的排序规则并不存在，但工程师有责任让规则公开、透明、说得清楚，规则被质疑时也拿得出依据。最后投票的结果见4.2节，学生倾向于方案B，理由是按单科排序结果确定、可复现，也不会像随机那样让人觉得不公。

④教师点评与技术关联（约5分钟）

教师作结：这个案例想说明的是，排序算法怎么选（尤其是稳定性的取舍），并不只是个技术问题。凡是牵涉到人的利益分配的系统，工程师就得把规则的可解释性、公平性以及利益相关方能否接受一并考虑进去。今天这场讨论并没有唯一正确的答案，要紧的是学生开始意识到：代码里的每一行选择，都可能承载着价值判断。

⑤实验报告中的“伦理反思”写作（课后融入实验）

与之对应的排序实验要求学生实现三种排序算法并比较性能，实验报告里另加一个“算法伦理反思”部分，让学生回答：假如你的排序算法被用在下面三种场景里——学校奖学金评定，成绩相同时按综合素质排序；电商平台的商品搜索结果排序；数据分析任务中的临时排序，你分别会选稳定排序还是不稳定的快速排序？理由是什么？请从公平性、效率、可解释性这三个维度写出你的分析。该部分的评阅结果见4.3节。

3.2 案例2：递归算法中的科学精神与责任意识

（1）学情分析与教学目标

递归是这门课的核心思想之一，树的遍历、分治算法、动态规划里到处都用得上。可它同时也是学生最容易栽跟头的知识点：递归基漏写导致无限递归、递归深度过大把栈压爆、重复计算拖垮效率，这些错误反复出现。但在线评测平台的测试数据规模有限，问题的严重性显不出来——递归函数在样例上跑得好的，换成极端输入就会悄无声息地崩掉^[8]。

更让人担心的是学生遇到栈溢出后的反应。大多数数人第一时间想到的是“把栈空间调大”或者“干脆改成循环”，很少有人回头审视自己的算法设计对系统资源是否负责。这种治标不治本的思维习惯，不符合科学精神所要求的严谨分析、责任意识。

本案例的教学目标同样分两个层面。知识技能上，学生要理解递归的工作原理，也就是调用栈、栈帧和

栈空间限制等知识点,掌握尾递归优化的做法,并学会估算递归深度与空间复杂度。思政层面则希望达到三点:写递归函数时主动分析最坏情况下的深度,这是工程态度的严谨;不滥用递归以致资源耗尽,这是对系统资源的责任感;碰上栈溢出先系统排查再动手,而不是粗暴地改,这是科学方法的体现。

(2) 教学流程设计

本案例放在“递归与分治”这一章,占用大约1学时(50分钟),另外还要借用实验课的部分环节。先说集中讲授的部分。

①问题导入(约10分钟)

教师先摆出一段看上去毫无问题的递归代码——用最朴素的递归求斐波那契数列的第 n 项,然后问:“这段代码 $n=10$ 时跑得很正常, $n=30$ 时有点慢,那 $n=100$ 会怎么样?”大部分学生的回答都是“更慢”。教师当场把 $n=100$ 跑起来,程序瞬间崩了,栈溢出。学生愣住:“怎么会崩?不就是慢一点吗?”

教师随即解释:每递归一次,调用栈上就要压进一个栈帧; $n=100$ 时深度大约100层,其实还在限制以内,默认栈大小通常是8 MB,每帧几十个字节,100层根本不算什么。接着把 n 改成10000再跑,又崩了。到这里学生才开始琢磨:递归深度并非没有上限,栈空间是一种有限的系统资源。对不少人来说,这是第一次意识到“递归是有深度限制的”。

②技术讲解(约15分钟)

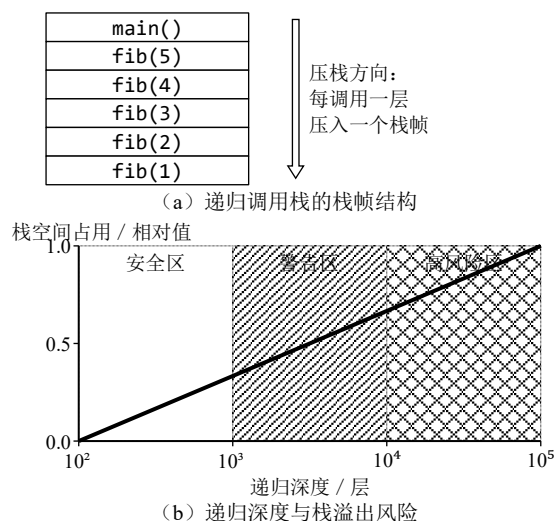


图2 递归调用栈与栈溢出风险示意图

教师配合图2把递归调用中栈帧压入与弹出的过程讲一遍。要点有三个:每次函数调用都会在栈上占一块空间,用来存参数、局部变量和返回地址;触及递归基之后栈帧开始逐层弹出;一旦递归深度超出承受范围、栈空间被耗尽,操作系统就会抛出段错误或者栈溢出信号。

然后是估算方法。先看单个栈帧大概占多少字节,这取决于局部变量的数量;再看栈空间的上限是多少,两者一除,最大安全递归深度就出来了。比如每帧64字节、栈空间8 MB,算下来大约是131 072层。像二分查找这类深度为 $\log_2(n)$ 的递归,深度完全可控;可换成斐波那契朴素实现这样的线性递归,深度跟 n 成正比,风险就相当高了。

③代码走查与风险分析(约15分钟)

教师给出几个递归函数,让学生分组去算它们最坏情况下的递归深度,判断有没有栈溢出的风险。这几个函数分别是:二叉树的中序遍历,树高为 h ,深度就是 h ;快速排序的递归版本,最坏的输入会把深度顶到 n ;归并排序的递归版本,深度是 $\log n$ 。算完之后学生会发现,快速排序若不做随机化,遇上已经排好序的输入,递归深度正好等于数组长度,数据量到100万时栈溢出是必然的。这一点此前几乎没有学生留意过。

教师顺势再问一句:“假如你写的递归函数上线以后让服务器频繁崩溃,领导让你‘把栈空间调大’了事,你会怎么办?”这个问题是想把学生引到正确的处理顺序上去:先弄明白递归深度为什么会那么大,再看能不能用尾递归优化、改成迭代,或者从算法本身改进,比如给快速排序加上随机化。一味调大栈空间只是把问题盖住,算不上负责任的做法。

④实验中的“责任声明”(约5分钟讲授+实验环节)

实验课上,学生要实现二叉树的三种递归遍历,也就是前序、中序、后序,同时给出用栈模拟的非递归版本,另外还要写一份“递归责任声明”。声明里需要交代三件事:最坏情况下递归函数的深度自己估算过;在输入规模的上限下验证过不会栈溢出;如果最终仍然选了递归而不是迭代,得说明理由,比如可读性上的收益盖过了风险。这份声明随实验报告一并提交,评阅结果见4.2节。

⑤反思与总结(约5分钟)

下课前,教师把话收拢:递归是件趁手的工具,同时也是一把双刃剑。科学精神要求我们不光知道“怎么用”,还得知道“边界在哪儿”。工程师的责任,就在于动用任何系统资源之前先主动评估风险,而不是等崩了再去补。希望上过这门课以后,学生每次写递归函数都会多问自己一句:我的递归能有多深?会不会爆栈?

4 思政融入的实施成效与评价

4.1 观测设计与数据来源

本节数据取自学院《数据结构与算法》课程的一轮完整教学实施,选课学生共45人,代码走查环节按每组5人分为9组。为了不给学生额外增加负担,观测材料全部取自教学过程中本就会产生的记录,没有另行组织问卷调查。材料有三个来源:一是课堂环节的现场记录,包括3.1节问题导入时的举手表决、角色扮演听证会的投票结果,以及3.2节代码走查环节各小组提交的深度分析单;二是排序实验报告中“算法伦理反思”部分的评阅结果;三是递归实验中“递归责任声明”的评阅结果。这类以学生实际产出为依据的表现性数据,在面向产出的课程评价中已有较多应用^[9]。需要说明的是,本轮教学没有设置平行对照班,下文给出的都是同一批学生在教学过程中的表现性数据。

4.2 课堂与实验环节的过程性观测

按上述口径统计,两个案例各环节的观测结果见表1,主要指标的占比见图3。其中前两行是同一个问题在教学前后的两次举手表决,问题的措辞与选项完全一致,只是提问时机不同,因此两次结果可以直接对照。

表1 思政融入环节的过程性观测结果

观测项	数据来源	计数	占比/%
课前认同“同分随便排”	举手表决	19 / 45	42.2
课后认同“同分随便排”	举手表决	3 / 45	6.7
课后能说明稳定性取舍	举手表决	42 / 45	93.3
听证会支持可复现方案	投票记录	45 / 45	100.0
走查主动识别深度风险	分析单	8 / 9	88.9
责任声明含深度估算	报告评阅	42 / 45	93.3
责任声明含上限验证	报告评阅	34 / 45	75.6

表1第5行反映的是学生会不会主动把复杂度分析往前推一步,推到“最坏情况下究竟要占用多少系统资源”这一层。按以往的教学经验,这一步几乎没有学生会自发想到,通常要等到程序真的崩掉才被动补课,本轮9个小组中有8组在走查环节主动提出了这一风险。最后两行则用来检验“递归责任声明”有没有被写成一份走过场的材料:只有当深度估算与上限验证同时完成,声明才算真正落到了工程判断上,只写结论不给依据的不计入统计。两项之间17.7个百分点的落差说明,学生把“估算”做成习惯要比把“验证”做成习惯来得容易,后者仍是下一轮教学需要着力的一环。

4.3 伦理反思写作的质量分布

排序实验报告中的“算法伦理反思”部分按三档评阅,评阅标准在布置实验时就随任务书一并公布,45份报告的评阅结果见表2。

表2 实验报告“算法伦理反思”部分的评阅结果

档次	判定标准	人数/人	占比/%
A	三维度分场景权衡并给出理由	37	82.2
B	联系具体场景仅单一维度分析	7	15.6
C	只复述技术差别,未涉及场景	1	2.2

这三档之间的差别,正好对应价值引领落到行为层面的三个阶段:C档的学生仍停留在算法性质的复述上,B档已经能把技术属性与具体应用场景联系起来,只有A档才真正完成了多个维度之间的权衡。A、B两档合计占97.8%,说明伦理反思并没有停在口号层面,多数学生能够把价值判断落到具体场景的取舍上。三档的比例关系比任何一个平均分都更能说明问题,也便于在后续学期之间做纵向比较。

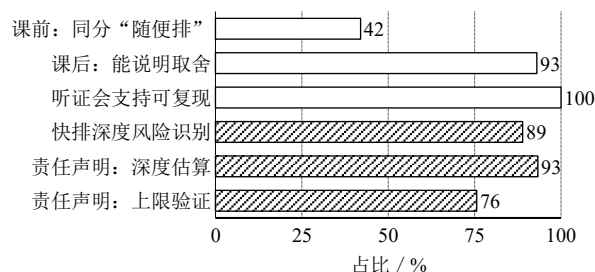


图3 思政融入环节主要观测指标的占比

4.4 讨论与局限

谈到课程思政,教师最大的顾虑通常是“会不会挤占核心内容的时间”。就数据结构与算法而言,这个担心可以打消:稳定性本身就是排序算法的重点,栈溢出风险分析原本也在递归教学应覆盖的范围内,而恰恰是这些内容,在传统教学中最容易被一带而过^[10]。本文设计的这两个嵌入式思政案例,一头把“为什么需要稳定性”“如何评估栈资源消耗”讲透了,另一头也把算法设计中的工程伦理与责任意识立了起来^[11]。

从实施后的情况看,意识层面的变化最直接。表1前三行显示,认同“同分随便排”的比例由42.2%降到6.7%,能结合场景说明稳定性取舍的比例达到93.3%,说明相当一部分学生是头一回意识到排序算法背后还牵连着公平性问题,“算法中立”这个默认假设受到了实质性冲击。学生普遍反映这样上课比传统讲授更有吸引力,还有人主动建议多引入一些真实场景,比如简历筛选中的排序歧视——这说明技术与社会责任之间的那座认知桥梁,课程已经搭起来了。技术层面上,学生的资源安全意识也有所增强^[12]。图3中后三项指标都在75%以上,说明代码走查与责任声明这两个环节确实把“先估算、再动手”的顺序固定了下来。这种意识还可以在后续课程中延续下去,比如操作系统、软件工程这些课都能接得上。有学生在反思里写道,自己过去一味追求递归的“技术美感”,如

今算清了递归的实际成本,才明白某些场景下迭代反而是更负责任的选择^[13-14]。总的来看,课程完成了从“会写代码”到“会想代价”的转变,责任意识就此内化成了技术决策的一部分。

这一轮实践也有几处不足。其一,未设置平行对照班,表1的前后对照只能说明同一批学生在教学过程中的变化,尚不能完全排除课程进度本身带来的影响;其二,观测只覆盖一个学期、一个教学班,45人的样本规模有限,结论的稳定性有待后续学期重复验证;其三,伦理反思与责任声明都由任课教师评阅,分档难免带有主观成分,后续可以增补锚定样例、引入双人独立评阅,并把往届同类题目的卷面得分一并纳入纵向比较。

5 结束语

对计算机专业的学生来说,数据结构与算法既是一套“思维体操”,也是学生工程价值观成型的关键阶段。本文拿排序算法的稳定性和递归算法的栈空间管理作载体,设计了两个融入公平正义意识、科学精神与责任担当的思政案例,让价值引领伴随算法学习自然发生^[15]。一轮完整实施的过程性数据表明,这种嵌入式的思政设计对学生的伦理敏感度、严谨态度和资源责任感都有实实在在的推动:认同“同分随便排”的比例由42.2%降至6.7%,能结合场景说明稳定性取舍的比例升至93.3%,递归责任声明中给出深度估算的占93.3%、完成上限验证的占75.6%,伦理反思达到多维权衡档次的占82.2%。今后,我们打算探索跨课程的协同路径,把该课程与操作系统、软件工程等后续课程联动起来,让资源意识和工程伦理责任感在整条技术学习链上持续得到强化,从“单点嵌入”真正走向“全程贯通”。

参考文献

- [1] 傅继彬. 构建计算机网络课程中的思政教育协议栈[J]. 计算机技术与教育学报, 2022, 10(5): 23-26.
- [2] 赵丽, 张志强. 《数据结构与算法》课程思政教学的探索与实践[J]. 河西学院学报, 2023, 39(2): 108-112.
- [3] 方维, 袁宝库, 梁峰绮. 基于PTA平台的程序设计类课程教学改革实践[J]. 计算机技术与教育学报, 2022, 10(1): 97-100.
- [4] 夏洁武, 李金忠, 彭蕾. OBE理念下“软件工程”课程思政教学案例设计[J]. 台州学院学报, 2023, 45(2): 87-92.
- [5] 王昭. 融入课程思政的数据结构教学改革探讨[J]. 电脑知识与技术, 2024, 20(15): 141-143.
- [6] 王兵书, 冯喜康, 马春燕. 文化元素融入“数据结构”课程思政的教学探索[J]. 计算机技术与教育学报, 2022, 10(3): 60-64.
- [7] 王昭. 数据结构与算法课程思政教学探索与实践[J]. 计算机教育, 2024(2): 16-19.
- [8] 陈明, 王勇, 张剑. 军校计算机类课程思政教学探索——以数据结构与算法课程为例[J]. 电脑知识与技术, 2023, 19(19): 156-158.
- [9] 谭貌, 段斌, 周彦, 旷怡. 面向产出落实工程教育认证标准的院系机制与实践[J]. 计算机技术与教育学报, 2023, 11(5): 16-20.
- [10] 于晓敏, 耿蕊, 张银霞. 基于OBE理念的数据结构课程思政教学[J]. 高师理科学刊, 2025, 45(7): 105-108.
- [11] 刘洪波, 李桂玲. 基于任务驱动的数据结构课程思政混合式教学设计与实践[J]. 中国信息技术教育, 2025(1): 92-95.
- [12] 初人杰, 罗梦贞, 杨嫒, 廖媛媛. 课程思政在大数据教学中的探索与研究——以“数据挖掘算法基础”课程为例[J]. 工业和信息化教育, 2024(3): 45-49.
- [13] 马晓波, 吕岩, 仁庆道尔吉. 线上线下混合式教学模式下的数据结构与算法课程思政教学探索[J]. 西部素质教育, 2025, 11(18): 61-64.
- [14] 王静, 李斌. 人工智能类课程思政的价值意蕴与实践路径——基于科技伦理的视角[J]. 江苏高教, 2024(9): 89-95.
- [15] 邹世辰, 郝文宁, 程恺, 余晓晗. 数据分析与挖掘课程思政设计与实践[J]. 计算机技术与教育学报, 2023, 11(2): 117-122.