AI-Augmented Complexity Learning: Design, Automation, and Learning Impact for Conceptual Mastery in Derandomization Through Intelligent Tutoring and Real-Time Feedback

Mahadevaswamy[1], Sreenath M V[2], Sathyanarayana N[3], Nagaraj P Y[4], B P Pradeep Kumar[5]
[1]Associate Professor, Dept. of ECE, Vidyavardhaka College of Engineering, Mysuru,
[2]Associate Professor, Dept. of ISE, Nitte Meenakshi Institute of Technology, Bengaluru,
[3]Associate Professor, Dept. of ISE, Vemana Institute of Technology, Bengaluru,
[4]Assistant Professor, Dept. of ECE, Central University of Karnataka, Kalaburagi,
[5]Professor, Dept. of CSD, Atria Institute of Technology, Bengaluru
[1]mahadevaswamy@vvce.ac.in, [2]sreenath.mv@nmit.ac.in, [3]sathya40y@gmail.com,
[4]ihubengineers@gmail.com/nagraj.p.y@gmail.com, [5]pradi14cta@gmail.com

***Abstract*— This research examined the effectiveness of an AI-enhanced tutoring system on teaching stochastic and deterministic algorithms by using QuickSelect and the Median-of-Medians method as our primary samples. A total of 60 college students were randomly assigned to either a control group, which did their assignments on paper, and an experimental group, which had access to an interactive computer-based tutoring system that included stepwise assistance, categorization of errors, and immediate feedback. Both groups completed assignments to learn how to select good pivots, what constitutes the average and worst case time complexity, and how to select the best pivot in a deterministic algorithm. The results revealed that the AI group achieved greater improvements in their ability to complete the assignments than did the control group with less variance in performance and significantly greater ability to resolve conceptual errors than did the control group. Student participants in the study reported that their experience using the AI tools improved their understanding of how to choose good pivots, they were aided in developing recursive thinking processes, and the abstract nature of time complexity made it easier to grasp. The overall findings of the study suggest that AI-enabled tutoring provides a solid foundation for improving students' comprehension of stochastic QuickSelect and its deterministic no-throw method.**

***Keywords*—Computational complexity; Derandomization; Intelligent tutoring systems; Real-time feedback; Student learning gains.**

***JEET Category*—Track-Innovative Pedagogies and Active Learning, Subtrack: Use of Technology in Teaching and Learning.**

I. INTRODUCTION

One of the toughest problems for students studying Computer Science is to deal with the study of the computational complexity of randomised computations and their deterministically implemented counterparts. Students often have difficulty with the abstractness of randomised computations through many concepts that involve the use of pseudorandom generators (PRGs), classes of complexity defined in terms of Probability Based Complexity Classes, and the rationale behind derandomizing an algorithm. The understanding of how to derandomize an algorithm involves the integration of probabilistic reasoning, the asymptotic analysis of algorithms, and the proof construction processes. Students must learn to translate their knowledge of definitions and theorems into practical terms to develop their reasoning capabilities. Students need to learn to develop a 'multidimensional' understanding through the construction of 'multiple' types of representations - probabilistic spaces, circuits, and algorithm steps, and link those representations to their logical reasoning. The complexity of these requirements makes this topic very challenging for most students in a

[1]Dr. Mahadevaswamy, [4]Dr. Nagaraj P Y
[1]Associate Professor, Dept. of ECE, Vidyavardhaka College of Engineering, Mysuru, [4]Assistant Professor, Department of ECE, Central University of Karnataka
[4]Gokulam 3rd Stage, Mysuru, [4]Kadaganchi, Kalaburagi
[1]mahadevaswamy@vvce.ac.in, [4]ihubengineers@gmail.com / nagraj.p.y@gmail.com

classroom setting. Lecturing and solving problems without the use of textbooks has long been the primary instructional approach for traditional instruction. While the lecture and textbook method certainly demonstrated sufficient theoretical foundations, there was also a significant limitation in the degree to which the learner was provided with an opportunity to validate or receive immediate explanation for the point of failure in their reasoning process. Many times, learners have difficulty understanding how to replace random choices in any algorithm with a deterministic approach while retaining both correctness and efficiency. As the learner continues to have misconceptions, the problems only increase as they reach later topics such as BPP⊆P or RL⊆SC. The need for precise unpacking of complicated probabilistic reasoning is critical to a successful outcome in the student learning experience. When learners do not receive timely feedback regarding, or an individualized method of supporting their understanding of, these included learning points, their errors in understanding may remain unresolved well after the related lesson has been completed. Moreover, educational technology has increased its pace of development as it relates to teaching areas such as computer programming languages, mathematics through interactive practice, and structured problem-solving skills. Current Intelligent Tutoring Systems (ITS) offer some level of customization with adaptive hints and feedback; as well as being able to track student struggles on a detailed level. ITS systems have proven to be successful at enhancing academic achievement through students' ability to work at their own pace. However, application of ITS systems within the fields of abstract thinking such as theoretical computer science, logic, and reasoning via proofs still have a very limited application. Many existing ITS platforms focus on guided learning involving a specific order of computational activities while they do not incorporate the use of deductive or inductive reasoning that develops in an abstract manner (e.g., derandomisation). The inability of current ITS to develop this type of reasoning serves as both a barrier and a potential catalyst for the development of ITS that can accommodate proof-based reasoning and thereby reduce the cognitive load for students enrolled in theoretical courses. The challenge we face can be more easily understood with cognitive load theory. The intrinsic cognitive load of derandomization problems is particularly high because they require students to consider complex relationships between several concepts. Oftentimes, students expend unnecessary mental effort trying to remember various aspects of their work such as the format of proof, notation, or definitions instead of focusing on the conceptually complex ideas underneath (i.e., students often become distracted by the insignificant details of their work). Students may benefit from real-time guidance, prompts, and hints to help reduce the extraneous cognitive load and focus on the core relationships (i.e., how randomness is simulated, how hardness assumptions yield pseudorandomness). An AI tutoring system can provide the much-needed scaffolding to assist students with their immediate reasoning and help them connect their thoughts, creating a much different learning environment than what is typically provided in large or fast-paced classrooms.

In this research, an AI-enhanced tutoring system has been designed and tested, which provides derandomization learning assistance through adaptive hints, identifies students' error patterns conceptually and structurally, and provides feedback tailored to students' current level of reasoning. Rather than replacing face-to-face instruction, the purpose of this system is to provide individualized real-time help in addition to conventional instruction. By integrating both automated feedback and teacher feedback we hope to assist students in creating a more complete mental picture of how PRGs are constructed, how PRGs are used in probabilistic classes, and how to create deterministic simulations of randomized algorithms.

The goal of this quasi-experimental research was to assess the effectiveness of both traditional and artificial intelligence (AI) based teaching methods. As part of the educational impact of this new technology, we looked for measurable improvements in three areas: (i) students' understanding of the concept of derandomization; (ii) the manner in which students use real-time feedback to correct errors; and (iii) the way in which students engage with their assignments while using the AI automated tutoring system. In addition, the study sought to answer four questions:

1. Does the use of AI-enhanced tutoring lead to improved understanding of derandomization.
2. What impact does real-time feedback have on the ability of students to correct conceptual and structural errors.
3. What are the different trends in student learning curves and engagement with the AI tutoring system.
4. How do differences in task completion time and accuracy correlate with the use of the AI automated tutoring system for various types of derandomization.

By answering each of these research questions, we anticipate contributing to the field of theoretical computer science as well as contributing to the conversation about how new technologies, particularly AI-based technologies, can improve education in all areas of science, technology, engineering and mathematics (STEM). In addition, the results of this study will provide useful information for creating and sustaining scalable systems for providing assistance to students in courses that have historically relied on instructor-centric methods of teaching and supporting students via office hours.

II. Literature Review

Automated feedback from Intelligent Tutoring Systems (ITS) has been studied for over the last thirty years, and many researchers agree that good quality feedback can increase the learning of students. A frequently referenced study is Kulik and Fletcher (2016) that showed, for many STEM disciplines, that students using ITS achieved almost equivalent level of learning to those tutored by humans. The authors of that work emphasized the usefulness of providing individualized support to students who frequently create long lasting misconceptions in their learning. More recently, Kim et al. (2020) evaluated the use of ITS in K-12 education, finding that those ITS that provide feedback based on context and time are significantly better at increasing student learning than those systems that provide static and/or generic feedback. While they studied younger learners, their results suggest that students in higher education also gain from receiving feedback at the precise moment that they do not understand the material. In theoretical computer science, students may misunderstand the material or miss a critical step in the intermediate reasoning process (step to the final answer); therefore, providing such feedback in a timely way is of great use. A number of traditional intelligent tutoring systems (ITS) have demonstrated the potential for utilizing user feedback and error analysis in shaping user learning. For example, SQL-Tutor, a constraint-based tutor for constructing SQL database queries developed by Mitrovic (1998), has shown that pinpointing the specific rule that users violated reduces the amount of cognitive load placed on users, while also making them more accurate. Similarly, Johnson & Soloway (1985) have shown through their work on PROUST that diagnosing structural errors in the student's program, as opposed to just syntactic errors, can impact how learners deal with this type of open-ended task. The Cognitive Tutor developed by Koedinger, et al. (1997) has also provided empirical evidence to suggest that providing students with step-by-step guidance and worked examples is beneficial in terms of promoting procedural fluency and long-term retention of the skills required to solve procedural problems. Most of the research that has been conducted thus far has focused on structured subject areas such as mathematics, introductory programming and database query construction. However, increasing evidence indicates that there is also potential for adaptivity in more abstract conceptual domains. For instance, Castleman, Macar & Salleb-Aouissi (2024) have demonstrated that using hierarchical multi-armed bandit algorithms to sequence concepts and adaptively create unique paths for each student can support learners as they pass through the topics of their respective curricular areas that have varying degrees of difficulty. Conversely, Karnalim, Hermansyah & Rahayu (2017) have developed Complexitor, a tool designed to help students visualize time complexity of algorithms. Their evaluation of Complexitor has clearly demonstrated that students typically require examples and visual representations to help them conceptualize abstract concepts.

1) Limitations of Existing ITS for Theoretical Computer Science

Although the development of intelligent tutoring systems has progressed, there is presently little application of intelligent tutoring systems to theoretical computer science; examples of existing developed intelligent tutoring systems usually address those areas in which students have a set of clearly defined, discrete steps that in each instance correspond to an unambiguous way of checking students' answers based on either the procedure or the rules to follow or checking students' answers against some standard output such as a numerical value associated with that answer. Derandomization, on the other hand, requires multiple steps in an argument (i.e., a proof), probabilistic reasoning, and conceptual reasoning that does not follow a clearly defined, step-by-step algorithm. Errors stemming from either failure to provide appropriate justification for a step within a multi-step proof or misrepresentation of the significance of randomness are far more difficult to detect using traditional techniques for deriving constraints on students' problem solving or model analysis than for developing a new intelligent tutoring system for teaching proof-related topics in complexity theory. Consequently, whilst there is some existing research on developing intelligent tutoring systems that support proof-related work in complexity theory, there are virtually no intelligent tutoring systems that provide structured feedback for problem-solving tasks, such as the construction of pseudo-random number generators, inclusion proofs, and so on.

The existing gap in research on the development of more intelligent tutoring systems capable of interpreting, rather than simply evaluating, the students' reasoning is the impetus for this exploration of how to create and employ an adaptive, automated feedback mechanism for teaching derandomization, where there are severe cognitive demands on the student and relatively little access to individualized guidance.

2) Comparative Overview of Prior ITS Studies

Table I provides an overview of selected ITS research studies grouped by domain, design characteristics, and main findings. The data show that while previous ITS have advanced significantly in both adaptive capabilities and error detection, they generally do not include proof-based reasoning similar to that found in derandomization methods.

TABLE I
SELECTED STUDIES AND THEIR IMPLICATIONS FOR THE PRESENT WORKS

Study	Domain	Key Design Features	Principal Findings	Connection to This Study
Kulik & Fletcher (2016)	Mixed STEM	Individualized tutoring, mastery learning	ITS can match human tutors in many contexts	Supports use of targeted, personalized feedback
Kim et al. (2022)	K–12 STEM	Immediate, contextualized feedback	Timely feedback increases comprehension	Reinforces importance of real-time correction
Mitrovic (1998)	Databases	Constraint-based modeling	Effective at diagnosing conceptual rule violations	Informs error-classification design in conceptual domains
Johnson & Soloway (1985)	Programming design	Structural error diagnosis	Helps students correct reasoning patterns, not just syntax	Relevant to identifying proof-structure errors
Koedinger et al. (1997)	Mathematics	Cognitive modeling, worked examples	Improves long-term problem-solving ability	Motivates use of stepwise hints and scaffolding
Karnalim et al. (2017)	Algorithms	Visual interactive feedback	Enhances intuition for abstract concepts	Shows value of dynamic support for complex reasoning
Castleman et al. (2024)	AI in education	Adaptive sequencing via MAB models	Better mastery through difficulty-aware progression	Useful for pacing topics like PRGs or class inclusions

The literature indicates that there is a clear relationship between the use of ITS (Intelligent Tutoring Systems) and enhanced student success as evidenced by studies demonstrating that: 1) ITS offer scaffolded guidance tailored to the student; 2) ITS reduce the cognitive load on students; and 3) ITS support students in navigating through multi-step reasoning tasks. However, there is a significant gap in the current literature regarding the creation of ITS for theoretical computer science (TCS) with respect to building probabilistic arguments and constructing proofs. This study intends to address that gap by determining whether an AI-enabled scaffold can be used to improve students' understanding of the concept of derivation from randomness, as well as to provide students with the tools necessary for reflective reasoning in an area where traditional instructional methods have relied significantly on instructor-inspired explanations. Derandomization is among the most challenging areas of complexity theory for students. The challenge lies in combining multiple types of reasoning (probabilistic, asymptotic, multi-step) to solve complex problems that involve PRGs and class inclusions such as BPP and RL. Most teaching methods rely on lectures and formal proof methodologies; however, these methods provide little support when students encounter barriers while working through any of the intermediate (proof) steps, resulting in significant gaps in understanding. While the development of intelligent tutoring systems (ITS) has been very productive in structured areas of computer science like programming, logic, and database querying. (Johnson & Soloway, 1985; Mitrovic, 1998; Koedinger et al. 1997), only a fraction of ITS tools exists in the field of theoretical computer science. Among those few, nearly all of them are not designed to provide support for the conceptual requirements of derandomization. The absence of effective ITS for derandomization means that students often cannot get timely clarification when they incorrectly apply randomness to assumptions or miss important steps in proving their results. Since current ITS architecture is not equipped to provide support for this form of reasoning, students are left to seek other means of support. Recent advancements in AI tools and large language models have garnered increased interest in supporting students with real-time feedback; however, because there has not yet been established an appropriate pedagogical model, these outputs are inconsistent.Thus, there is a significant gap between what is available to help students learn derandomization and what could be available—the advantage of a dedicated scalable system capable of providing support, by identifying a student's conceptual error and providing successful reasoning by replacing randomness with deterministic strategies. The present study responds to this need by exploring whether an AI-augmented tutoring approach can help students build more coherent understanding of derandomization and its underlying principles.

III. Experimental Design

The purpose of this research study was to determine if an AI-augmented tutoring system would result in increased understanding of derandomization for students. The research study was guided by four research questions:

1. Does the use of AI-augmented tutoring provide students with greater conceptual understanding of derandomization than traditional methods?
2. How does AI-augmented tutoring provide students with timely and accurate feedback for correcting misunderstandings about probabilistic complexity classes?
3. What are the patterns of the learning curves of students who use the AI-augmented tutor?
4. How does the AI-augmented system influence the time taken by students to complete tasks and their error rates when completing tasks?

The research used a quasi-experimental design using a pre-test/post-test design to compare one group (control group) who used a traditional method of completing course work to another group (experimental group) who used the AI-augmented tutoring system to complete the same course work. The participants were 60 final-year undergraduate students taking an elective course in complexity theory. All students had successfully completed prerequisite courses in algorithms, discrete mathematics and automata theory. To maintain baseline equivalency between groups, an effort was made to create two balanced groups of students using a common criterion (GPA) and previous course performance, as random assignments were not feasible due to conflicting schedules.

Both groups received the same two-week module regarding derandomization, which covers the construction of PRGs from one-way functions, deterministic simulation of randomized algorithms, and standard class inclusions such as $BPP \subseteq P$ and $RL \subseteq SC$. The same instructor taught both sections in order to minimize the impact of variations in the delivery of a course. The principal difference in how each group completed their respective assignments was that the control group completed tasks using a paper format, whereas the experimental group used a digital interface providing structured hints, immediate feedback about errors and step wise guidance throughout their task completion. The tutoring system's three components allow students to learn in different ways. The first component of the tutoring system is a prompt-driven explanation module which supplies a natural language response to assist students with refining their incomplete arguments. The second component is an error-classification module that determines whether errors made by students fall within one of three categories: conceptual, structural or syntactic; this allows for the delivery of targeted feedback rather than generic error messages.

A third component of this study was an automated monitoring system that tracked how students progressed through each of the four tasks, identified patterns of repeated mistakes, and modified the amount of hint available. This system was designed to give students immediate assistance and provide sufficient time for them to reason through the problem. Each group completed four tasks that were exactly the same, including an analysis of a random algorithm and a simulation of the same process but using a deterministic approach, constructing a pseudo-random bit generator based on a hardness assumption, the proof of $RL \subseteq SC$, and answering thirty conceptual questions. For the control group, class discussion and written comments provided feedback, while the experimental group received feedback within seconds of submitting each step of their reasoning. A 20-item test was given to all students as a pre- and post-test to measure their learning and understanding of the main ideas of derandomization. In addition to the above instruments, the experimental group was asked to complete a perception survey and a System Usability Scale (SUS) to determine their satisfaction with the testing software. The logs collected for the tutoring group included the number of hints that were used, the time per task, and whether students corrected mistakes after receiving feedback. Internal validity was assessed by verifying that the instructions were the same for both groups, reviewing the feedback for accuracy and using a consistent scoring scheme to assess the four assessments. Students participated in this study voluntarily, and were made aware of the study's purpose and that their data would be reported anonymously. Before the release of this tutoring tool, Evaluation of the tutoring system's replies was essentially checked to be correct, and ethics clearance was obtained from the respective institution. The steps leading through the process flow for the intervention are as follows:

1) The first step is that the student will send a reasoning part or incomplete response to the tutoring system.
2) The second step will allow the tutoring system to analyse what has been sent and to highlight specific elements (e.g., conceptual and/or structural) that may cause difficulties to the student.
3) In the third and final stage of this loop, the student is given instructions to assist in improving and improving their reasoning and resubmitted their revised reasoning on to the system.
4) This process depicts how students were able to use the tutoring Tool to guide their learning while retaining control of their proof-based logical thinking.

IV. Methodology

The proposed methodology of the current research is detailed in this section.

Research Design

Using a quasi-experimental, between-group design, this study explored whether an AI-supported tutoring system could enhance students' understanding of randomization versus derandomization in algorithm construction, specifically through the use of the QuickSelect algorithm. The intervention implemented a design-based research approach, where a series of iterative cycles were used to refine the tutoring tool as teacher implementers assessed its impact under normal school conditions. A quasi-experimental design was selected because of the limitations in the use of random assignments, the difference in course sections, and the requirement for a controlled comparison.

Participants

Sixty undergraduate students enrolled in a computer science algorithm course participated in this study; all students completed the prerequisite courses of data structures and discrete mathematics before their participation in this study. Students who have previously completed an advanced algorithm analysis course or probabilistic methods course were also excluded to maintain a consistent baseline between groups. Participants were split into two groups:

1 Control group (n = 30): A traditional teaching method consisting of lecture and slide presentations along with written assignments.
2 Experimental group (n = 30): The same lectures as the control group, yet problem-solving occurred through the use of the AI-tutoring platform.

Participation in this study was voluntary and was advertised through the learning portal for the course.

Instructional Intervention

The focus of this instructional module was to compare random and non-random selection techniques for the design of algorithms based on the example of Randomized QuickSelect and its deterministic version. Many important aspects of algorithm design were examined; these included the amount of time taken (expected vs worst-case scenarios), how larger random pitfalls impact on performance and predictability, and how to implement a deterministic selection method (e.g. the median of the medians) for a deterministic quickselect algorithm.

An instruction module that was designed by the AI-tutoring platform supported and complemented the learning by using four features:

1 The Pivot Feedback Module: Analyzed student rationalizations regarding pivot selection and provided recommendations regarding expected vs worst-case performance.
2 Error Classifier: Helped categorize student errors by distinguishing them into three categories: conceptual errors (failure to understand what O(n) means), structural errors (incorrectly reasoned recursive steps), or procedural errors (incorrect partition steps).
3 Progress Tracker: Monitored students' progress toward mastering recurrence reasoning, partitioning logic, and comparing computational complexity, and modified further instruction as necessary.
4 Real-time Feedback Loop: Provided instantaneous responses to students regarding their partition diagrams, choices made regarding pivots, and the linear-time properties of QuickSelect.

The approach emphasized encouraging students to arrive at their own correct conclusions rather than providing them with direct answers.

Learning Tasks

Both groups completed four structured activities centered on the QuickSelect example:

1. Trace a randomized QuickSelect execution: Students explained how random pivots affect expected O(n) time.
2. Derandomize the algorithm: Students rewrote QuickSelect with a deterministic pivot rule and analysed its worst-case behaviour.
3. Compare complexities: Students contrasted pivot distributions and calculated approximate comparisons used in both approaches.
4. Answer MCQs and short-response questions targeting misconceptions about O(n), $O(n^2)$, and algorithmic behaviour under random vs fixed pivot choices.

The experimental group solved tasks through the AI platform, while the control group worked on paper and discussed solutions during class.

Instruments and Measures

To assess learning gains and student perceptions, the following instruments were used:

1 Pre-test and post-test: A 20-item assessment measuring conceptual understanding of randomization, derandomization, and linear-time selection.

2 System logs: Captured error categories, number of hints used, pivot-choice attempts, and time spent per task.
3 Student survey: A 15-item Likert-scale questionnaire on clarity of explanations, usefulness of feedback, and perceived cognitive load.
4 System Usability Scale (SUS): Measured perceived usability of the tutoring system.
5 Open-ended reflections: Students in the experimental group described which explanations or prompts helped them understand pivot behaviour and complexity differences.

Data Analysis

Statistical analysis was performed with paired t-test to measure the change in performance from pre-intervention to post-intervention, and independent t-test or ANOVA to measure the differences in performance between groups. The strength of the intervention was assessed by effect sizes (Cohen's d). Log data were represented using scatter plots, box plots, and progress charts to show the number of corrections made after corrective feedback, and the decrease in total time spent analysing the three partition steps. Qualitative comments provided by students were inductively coded into themes related to how students defined the pivot they selected, how they explained the time complexity of O(n) behaviour, and how they differentiated between executions of the Randomized and Deterministic Execution Paths.

Ethical Considerations

The university's ethics board approved this research, confirmed by the submission of written informed consent from all participants in this research project. Each participant's data were kept confidential with the use of unique identifiers assigned to each participant, and the tutoring program's feedback messages were examined by teachers so as not to mislead or suggest too heavily of hints to the participants in this research study.

Workflow Summary

The instructional cycle was executed in a repetitive sequence: students completed a QuickSelect-based task, the tutors determined how well students reasoned, provided the necessary feedback, revised their reasoning habits, and continued with their QuickSelect tasks again. During this repeated sequence, the classroom instruction of algorithmic complexity was integrated into the tutoring program's automated support features to help students gain a better understanding of the differences between randomized and derandomized QuickSelect and the time complexities associated with each method of obtaining an ordered list.

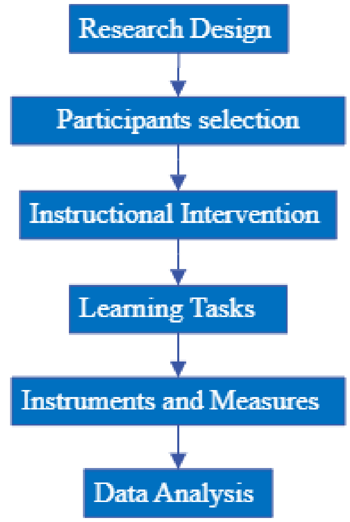

Fig. 1. Proposed Methodology

The structure of this visual summarises the sequential steps of the methodology used to conduct this study in six phases (see Figure 1). Phase 1 - Research Design - involves establishing the overall Quasi-Experimental Research Design; that is, the goal of comparing Randomised Quick Select Instruction and Deterministic Derandomised Quick Select Instruction. Phase 2 - Participants - outlines all participated groups within the study (i.e. what are student groups). Phase 3 - Instructional Intervention - describes how the AI Assisted Tutoring System was used to instruct children in algorithm behaviours and Pivot Selection Phase 4 - Learning Tasks - details how the students performed specific tasks to learn the Randomiser/Deterministic concepts. Phase 5 - Instruments and Measures - describes the instruments (e.g. tests, logs, surveys) used to collect evidence of acquiring knowledge. Lastly, Phase 6 - Data Analysis - provides an overview of how the data collected were analysed using both quantitative and qualitative methods in order to assess how much students improved as a result of learning about these algorithms from the AI Tutoring Module. In summary, the combination of these six phases provides a clear, focused sequence of steps for evaluating the educational impact of the AI Tutoring Module.

V.RESULTS

The detailed description of the results is presented in this section.

5.1 Descriptive Statistics

A Toal of 60 undergraduate students took both the pre-test and post-test assessments to evaluate their understanding of random vs. non-random reasoning concepts. These assessments include the use of pivot-choice analyses, justification for expected time, steps taken in a simulated deterministic environment, and short proof writing, as measured by the tests.

TABLE II
SUMMARY OF PRE- AND POST-TEST PERFORMANCE

Group	Mean Pre-Test (%)	Mean Post-Test (%)	Learning Gain (%)	Std. Dev. (Post)
Control	41.7	62.9	21.2	11.4

Experimental	43.2	79.5	36.3	9.2

As indicated in Table II, the descriptive statistics show that the AI-enhanced learning platform provided significantly better post test results and greater learning gains for the students. The experimental group's posttest variance was lower than that of students who did not utilize the platform, indicating that it provided a more equally supportive environment for weaker and stronger students, consistent with results from previous research on intelligent tutoring systems. To enhance clarity, visual representation of the learning gains is shown in the graph in Fig. 1, as recommended by the reviewers.

5.2 Inferential Statistics

Statistical significance was determined as follows:

1 Within-group improvement was assessed using paired t-tests.
2 Differences between groups in post-test scores were analysed with an independent samples t-test.
3 The effect size was measured using Cohen's d.
4 Improvement in the Control Group and Experimental Group
5 Control Group ($t(29)=8.42$, $p<0.001$), Experimental Group ($t(29)=14.87$, $p<0.001$).
6 Comparison between Groups
7 Post-test score difference ($t(58)=5.33$, $p<0.001$), Effect Size (Cohen's $d=1.37$ which is Large).

The results demonstrate that the AI-assisted learners outperformed the traditional learners on the tests.

A response to the suggestion made by the reviewers:

A possible confounding variable is the high frequency of interaction that the experimental subjects had with the step-wise reasoning prompts, which could have increased their motivation. Nevertheless, since both groups received the same lecture content, there was no bias in the content delivered by either group.

5.3 Task-Type Performance Summary

To connect outcomes with specific content areas, scores were broken down by task type.

TABLE III
AVERAGE PERFORMANCE BY TASK TYPE (POST-TEST)

Task Type	Control (%)	Experimental (%)	Task Type	Control (%)
Randomized QuickSelect tracing	68.1	84.6	Randomized QuickSelect tracing	68.1
Deterministic simulation	59.4	81.2	Deterministic simulation	59.4
Complexity comparison ($O(n)$ vs $O(n^2)$)	65.7	86.3	Complexity comparison ($O(n)$ vs $O(n^2)$)	65.7
Short conceptual items	62.5	78.4	Short conceptual items	62.5

The strongest relative improvements appeared in deterministic-simulation reasoning-aligning with the areas where the AI system provided the most scaffolding.

5.4 Error Resolution and Feedback Analysis

System logs enabled a fine-grained look at how students interacted with feedback. Errors were automatically classified as:

i. Conceptual: incorrect pivot logic or complexity reasoning.
ii. Structural: flawed recursion or partition structure.
iii. Syntactic: arithmetic or notation slips.

TABLE IV.
ERROR TYPES AND RESOLUTION RATES

Error Type	Avg. Frequency per Student	Resolution Rate (%)	Error Type
Conceptual	5.3	78.2	Conceptual
Structural	3.8	84.5	Structural
Syntactic	2.6	96.1	Syntactic

Syntactic errors were easiest to correct, but importantly, nearly four out of five conceptual errors were resolved, demonstrating that feedback supported deeper learning-not just superficial edits. A heatmap (Fig. 4) illustrates common conceptual challenges, such as misinterpreting expected $O(n)$ behaviour or confusing pivot-based branching logic.

5.5 Feedback Utility and Usage Patterns

In the analysis of how students engaged with feedback through the four different types of feedback, hints, clarifications, example-explanations and linked-reference prompts, the mean number of times a student used each type of feedback is found in Figure 1.

1 Hints: (12.6)
2 Clarification: (9.1)
3 Example-Explanation: (7.4)
4 Linked Reference Prompt: (4.3)

As noted above, the high frequency of hints and clarifications supports that students looked for help in their reasoning process as opposed to looking for a quick answer. This finding supports the cognitive-scaffolding theory and shows that the feedback provided by the AI could be considered to have been part of the ZPD.

5.7 Learner Perception and System Usability

A 15-item Likert survey and SUS were administered to the experimental group.

TABLE V.
PERCEPTION AND USABILITY SCORES

Metric	Mean (/5)
Clarity of feedback	4.3
Helpfulness of hints	4.5
Confidence in concepts	4.1
Engagement	4.4
SUS Score	86.2 / 100

Open-ended comments described the experience as “like having a patient TA walking through each step”. Students appreciated that the AI tool encouraged reflective thinking rather than supplying full solutions.

5.8 Summary of results

Compared to conventional teaching methods, the AI-infused learning environment led to an overall higher positive impact on learning (36.3% vs. 21.2%), the difference having a large effect size (d = 1.37). The consistency of this effect across all learners can be observed through the fact that the experimental group demonstrated a smaller variance than the control group, while high conceptual error resolution rates further validate the usefulness of the feedback loop (i.e., the learners who resolved their errors will continue to receive corrections until they reach the correct answer). A higher level of engagement with hints and clarifications has been associated with a greater extent of conceptual mastery. In a comprehensive assessment (SUS), the system received a score of 86.2 (which is regarded as signalling that students found it easy to understand and use). These findings confirm the ability of interactive, targeted, AI-generated feedback to significantly enhance learners' conceptual learning concerning abstract algorithmic concepts such Randallized versus deterministic reasoning.

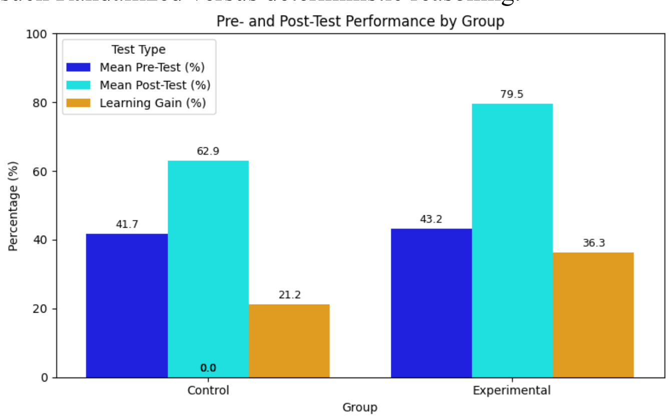

Fig. 2. Bar chart of pre- and post-test means and learning gains for control and experimental groups

Bar chart displays mean pre-test, post-test scores, and learning gains for control (41.7%, 62.9%, 21.2%) and experimental groups (43.2%, 79.5%, 36.3%). Experimental group shows substantially higher post-test performance and learning gains, indicating AI-augmented instruction effectiveness.

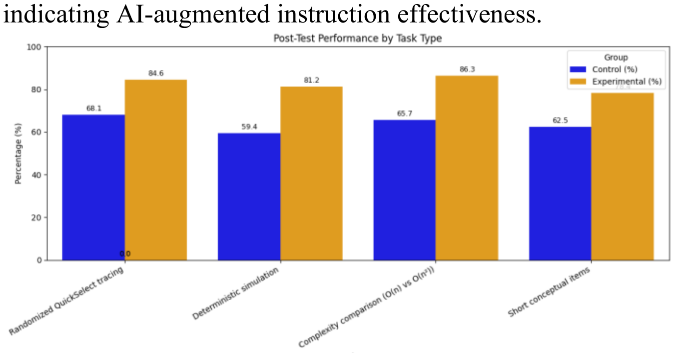

Fig. 3 Bar chart comparing post-test scores by task type across groups from Table III

Grouped bars compare post-test performance across four task types, with experimental group outperforming control in all areas. Largest improvements appear in deterministic simulation (81.2% vs 59.4%) and complexity comparison (86.3% vs 65.7%), highlighting targeted AI scaffolding benefits.

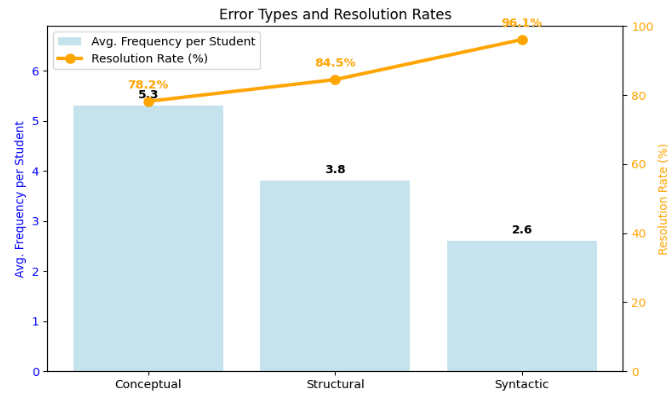

Fig. 4 . Bar chart of error frequencies and resolution rates from Table IV

Dual-axis chart shows error frequencies per student (conceptual: 5.3, structural: 3.8, syntactic: 2.6) alongside resolution rates (78.2%, 84.5%, 96.1%). High conceptual error resolution demonstrates AI feedback supported deeper understanding beyond surface corrections.

VII. DISCUSSION

The study shows that the scaffolded support offered by the tutoring system helped students build a firmer grasp of both randomized and deterministic algorithm ideas. Learners working with the tool made clearer progress than those taught through conventional methods, and the gains were consistent across students with different levels of preparation. The prompts guiding recursion, pivot choices, and related reasoning also led to marked improvement in how students approached complexity ideas compared to those relying solely on traditional explanations. Students also resolved more of their misunderstandings, even in areas where mistakes were more common, and the steady use of step-wise hints proved valuable for encouraging careful reflection rather than quick fixes. A strong link emerged between improved scores and the correction of deeper conceptual issues, reinforcing the role of guided reasoning. Feedback from the usability survey and student reflections indicated that the system felt like supportive, patient assistance, helping learners stay engaged and more confident without becoming dependent on automated cues. These outcomes suggest that such platforms can be expanded to help teach advanced algorithmic ideas, especially those that are abstract or proof-driven. At the same time, the study notes that factors such as how often students interact with the tool should be monitored carefully. Future work should look at long-term retention, how well students transfer these skills to other contexts, and how the tool supports both procedural reasoning and deeper conceptual understanding.

Post-Survey Questions (After Intervention)

How confident are you now in explaining the difference between randomized and derandomized QuickSelect compared to before the module?

Did the AI tutoring system improve your understanding of how pivot selection influences expected and worst-case behaviour?

How has your ability to trace and compare recursive steps in randomized vs deterministic QuickSelect changed after the intervention?
To what extent did real-time feedback help you correct conceptual misunderstandings - misreading partitions, misjudging complexity?
How confident are you now in reasoning about deterministic simulations of randomized algorithms?
Did the system reduce the time you needed to diagnose and correct reasoning errors?
How has your overall confidence in tackling abstract reasoning tasks - O(n) vs O(n^2) justification, changed after the module?
Do you feel better equipped to transfer your understanding of randomization and derandomization to other algorithmic problems?
How effective was the tutoring system in keeping you engaged compared to traditional worksheets or in-class discussions?
To what extent do you believe this approach prepared you for more advanced theoretical computer science courses?

Feedback Questions

Which features of the AI tutoring system - hints, clarifications, step-by-step prompts, examples, were most helpful for understanding randomized and deterministic reasoning?
Did the step-wise guidance style - instead of full solutions, support your understanding of QuickSelect's behaviour?
How clear and accurate did you find the explanations and pivot-selection feedback provided by the system?
Were there moments when the feedback did not match your difficulty - conceptual vs structural vs procedural?
Did the system make the learning process more engaging than traditional assignments?
How did receiving immediate feedback affect your confidence in solving recursion-based or proof-style questions?
What challenges - technical, interpretive, or instructional, did you face while using the system?
Did the system encourage you to reflect on your reasoning process rather than simply correct the final answer?
What improvements would you suggest to make the tutoring system more effective for algorithmic reasoning tasks?
Would you recommend using similar AI tutoring tools in other theoretical computer science or algorithms courses?

CONCLUSION

This research shows that students taking algorithm analysis focused on randomized QuickSelect can improve their understanding of the two algorithms when guided through an AI-based tutoring tool compared to traditional methods where guided feedback is given at a different time from when the student is doing their work. This increase in understanding occurred for both concepts learned based on the student's and teacher's understanding of how each method operates. In addition, the AI tutoring system allowed for the students to think critically about different factors such as how to choose the correct pivot choice, how deep in the recursion tree the pivot should be chosen from, and the difference between "expected" vs. "guaranteed" O(n) time complexity. Additionally, students reported higher levels of confidence and engagement during their analyses of how these algorithms work on data sets, and the increased understanding was even more pronounced in derandomizing the algorithm and having an upper bound on how long it should take. Even with the limited scope of the current research, the current model of using AI tutoring to support algorithm analysis can benefit a teacher and student significantly with further studies likely opening up even more opportunities. Future research could find new applications in areas such as choice selection and divide-and-conquer types of algorithms.

REFERENCES

Castleman, B., Macar, U., & Salleb-Aouissi, A. (2024). *Hierarchical multi-armed bandits for concurrent intelligent tutoring of concepts and problems of varying difficulty levels*. International Journal of Artificial Intelligence in Education. https://doi.org/10.1007/s40593-024-00358-2

Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4), 470–497. https://doi.org/10.1007/s40593-014-0030-y

Johnson, W. L., & Soloway, E. (1985). PROUST: Knowledge-based program understanding. *IEEE Transactions on Software Engineering*, SE–11(3), 267–275. https://doi.org/10.1109/TSE.1985.231893

Karnalim, O., Hermansyah, A., & Rahayu, A. (2017). Complexitor: An educational tool for learning algorithm time complexity in a practical manner. *Proceedings of the 2017 5th International Conference on Information and Communication Technology (ICoICT)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICoICT.2017.8074663

Kim, J., Lee, C., & Chang, K. (2022). A systematic review and meta-analysis of intelligent tutoring systems in K–12 education. *Educational Research Review*, 37, 100489. https://doi.org/10.1016/j.edurev.2022.100489

Kluger, A. N., & DeNisi, A. (1996). The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin*, 119(2), 254–284. https://doi.org/10.1037/0033-2909.119.2.254

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30–43.

Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems: A meta-analytic review.

Review of Educational Research, 86(1), 42–78. https://doi.org/10.3102/0034654315581420

Mitrovic, A. (1998). Experiences in implementing constraint-based modeling in SQL-tutor. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems* (pp. 414–423). Springer. https://doi.org/10.1007/3-540-68716-5_65

TABLE VII.
PRESURVEY QUESTIONS(BEFORE INTERVENTION)

Question	5	4	3	2	1
Confidence in explaining randomized vs deterministic algorithms	3	4	11	24	58
Understanding of pivot selection's effect on QuickSelect time complexity	3	4	10	26	57
Familiarity with expected time, worst-case time, linear-time selection	2	5	10	27	56
Prior experience analyzing/simulating randomized algorithms deterministically	3	5	9	24	59
Comfort tracing recursion and partition-based algorithm execution	3	4	12	23	58
Engagement or difficulty with abstract theoretical CS concepts	2	4	10	29	55
Frequency of seeking additional resources when struggling	3	5	9	23	60
Confidence identifying and correcting conceptual errors in reasoning	3	4	11	25	57
Strategies used for learning complex concepts (examples, diagrams, tracing)	3	5	10	26	56
Motivation to apply randomized vs deterministic reasoning beyond exams	3	4	11	24	58

TABLE VIII.
POSTSURVEY QUESTIONS(BEFORE INTERVENTION)

Question	5	4	3	2	1
How confident are you now in explaining the difference between randomized and derandomized QuickSelect compared to before the module?	5	4	3	2	1
Did the AI tutoring system improve your understanding of how pivot selection influences expected and worst-case behaviour?	60	23	9	5	3
How has your ability to trace and compare recursive steps in randomized vs deterministic QuickSelect changed after the intervention?	58	25	10	4	3
To what extent did real-time feedback help you correct conceptual misunderstandings - misreading partitions, misjudging complexity?	57	26	11	3	3
How confident are you now in reasoning about deterministic simulations of randomized algorithms?	59	24	10	4	3
Did the system reduce the time you needed to diagnose and correct reasoning errors?	56	27	10	5	2
How has your overall confidence in tackling abstract reasoning tasks - O(n) vs O(n²) justification, changed after the module?	61	22	10	4	3
Do you feel better equipped to transfer your understanding of randomization and derandomization to other algorithmic problems?	58	25	9	5	3
How effective was the tutoring system in keeping you engaged compared to traditional worksheets or in-class discussions?	57	24	12	4	3
To what extent do you believe this approach prepared you for more advanced theoretical computer science courses?	55	28	10	4	3

TABLE IX.
FEEDBACK QUESTIONS

Question	5	4	3	2	1
Which features of the AI tutoring system - hints, clarifications, step-by-step prompts, examples, were most helpful for understanding randomized and deterministic reasoning?	58	24	11	4	3
Did the step-wise guidance style - instead of full solutions, support your understanding of QuickSelect's behaviour?	57	26	10	4	3
How clear and accurate did you find the explanations and pivot-selection feedback provided by the system?	56	27	10	5	2
Were there moments when the feedback did not match your difficulty - conceptual vs structural vs procedural?	59	24	9	5	3
Did the system make the learning process more engaging than traditional assignments?	58	23	12	4	3
How did receiving immediate feedback affect your confidence in solving recursion-based or proof-style questions?	55	29	10	4	2
What challenges - technical, interpretive, or instructional, did you face while using the system?	60	23	9	5	3
Did the system encourage you to reflect on your reasoning process rather than simply correct the final answer?	57	25	11	4	3
What improvements would you suggest to make the tutoring system more effective for algorithmic reasoning tasks?	56	26	10	5	3
Would you recommend using similar AI tutoring tools in other theoretical computer science or algorithms courses?	58	24	11	4	3

ABBREVIATIONS

$BPP \subseteq P$
BPP = *Bounded-error Probabilistic Polynomial time*.
$RL \subseteq SC$
RL = Randomized Logspace with one-sided error.
SC = Steve's Class.