# Research and Practice on Intelligent Teaching Mode of Data Structure Course Empowered by Large Models

Xiaofei Sun[1] Zhenzhen Wang[2] Bin Yang[3] Lei Wu[4] Jie Huang[5] Zeng Li[6]
Wenwen Pan[7]

[1,2,3,4,5,6,7] *College of Information Science and Engineering, Zaozhuang University, Zaozhuang 277160, China*
[7]*Corresponding author.*

**ABSTRACT**
Data structure is a core course in computer related majors, and its experimental teaching is crucial for students to deeply understand and master algorithms and programming skills. However, traditional experimental teaching methods have shortcomings in terms of personalization, learning feedback, and the cultivation of self-directed learning abilities. In recent years, the gradual popularization of big model technology has provided new development ideas for experimental teaching in data structure courses. This article takes the transformation of students' learning habits under the background of big models as the starting point, proposes an experimental teaching mode based on big models, integrates intelligent question answering, code diagnosis, and evolutionary learning, and uses big models to analyze students' learning behavior, dynamically adjust experimental tasks and learning resources that are suitable for learning situations. The research results indicate that the experimental teaching mode based on large models effectively enhances students' learning interest, learning efficiency, and innovation ability.

***Keywords:*** *Large language models, Experimental pedagogy, Learning behavior analysis, Personalized learning.*

## 1. INTRODUCTION

Data structure is an important core course in computer science, which involves efficient data storage and operation method design.[1] It is an important cornerstone of algorithm design and program design. The course covers classic structures and algorithms such as stacks, queues, linked lists, trees, graphs, searches, and sorting, as well as the evaluation and optimization of algorithms, which play an important role in cultivating students' logical thinking and programming abilities.[2] Experimental teaching, as an important part of data structure courses, can help students transform algorithm theory into practical programming techniques, verify the correctness and efficiency of algorithms, and achieve an organic combination of theory and practice.

However, traditional teaching faces many challenges in practice. Firstly, the lack of personalized tutoring is a challenge that traditional teaching cannot overcome.[3] Teachers face a large number of students in their teaching, and often do not have the spare time or energy to provide personalized guidance to students beyond completing the overall teaching task. Secondly, learning feedback is a common problem in traditional teaching.[4] The mastery of course content by students requires teachers to review experimental reports and assignments before feedback can be provided, which has a long cycle and affects students' learning efficiency and teaching effectiveness.[5] Finally, the cultivation of innovative ability has always been a shortcoming of traditional teaching.[6] The learning tasks of traditional teaching are limited by classroom time and scene constraints, and are mostly confirmatory. Students complete established content based on experimental lesson plan templates, lacking open-ended experimental content, which cannot effectively exercise problem-solving skills, let alone stimulate interest in exploring unknown problems.[7] Therefore, improving the personalized level, timely feedback, and innovation of teaching

has become an important goal of data structure curriculum reform.


Figure 1 Knowledge graph (the correlation and hierarchical structure between various knowledge points in this course).

Nowadays, Large Language Models (LLMs) such as ChatGPT and DeepSeek have shown great potential in natural language processing, question answering systems, document generation, and other fields, gradually gaining public recognition and popularity.[8][9][10] Compared with traditional computer technology, large models have shown outstanding performance in related fields.[11][12] Firstly, large models have the advantage of timely feedback. It can respond to user issues in real-time (within seconds), provide detailed solutions, and thus provide reliable guidance for users.[13] Secondly, large models can provide personalized solutions. By learning about users' Q&A habits, application fields, and thinking logic, big models can provide personalized services for that user, and even gradually become unique AI companions only for that user.[14] Finally, the application cost of large models continues to decrease.[15] With the birth and popularization of domestic large-scale models such as DeepSeek and Doubao, their free nature to ordinary users has made the application cost of large-scale models very low, which has enabled the rapid promotion of large-scale model technology.

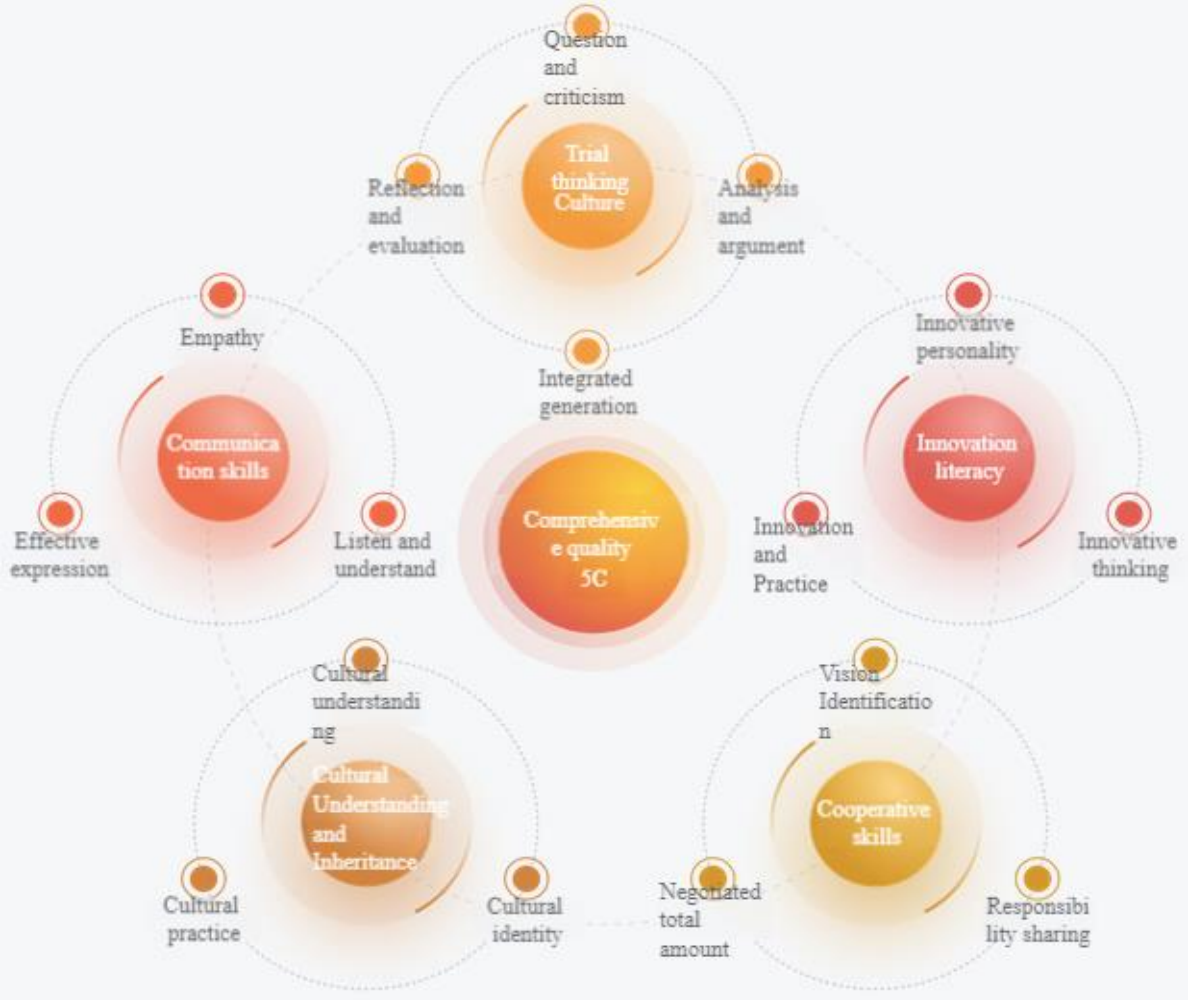


Figure 2 Quality map (Development goals of students in critical thinking, innovative literacy, social responsibility, etc.).

This article closely follows the development trend of big model technology, starting from the changes in students' learning habits, and explores the data structure teaching mode supported by big model technology to promote students' learning effectiveness and innovation ability. By analyzing the learning situation of students before and after applying the big model, summarizing its applicability and promotion prospects in teaching, feasible paths and methods are provided for the teaching reform of data structures.

## 2. DESIGN OF TEACHING MODE BASED ON LARGE MODEL

### *2.1 Knowledge, Ability, and Quality*

This teaching mode uses AI systems to help students comprehensively grasp the systematic knowledge of data structures, including linear tables, stacks and queues, strings, arrays and generalized tables, trees and binary trees, graphs, search and sort, etc. In this teaching reform, this comprehensive knowledge system has helped us correct previous knowledge omissions and biased focus. The researchers also constructed a knowledge graph for this teaching system (as shown in “Figure 2”), analyzing the connections and hierarchical structure between knowledge points and providing visual guidance for students to engage in systematic learning.

This course further focuses on ability development, with 6 core competency points, 57 knowledge units, and 64 learning contents designed. These abilities include using abstract logical thinking to analyze and design data structures, optimizing program performance based on algorithm efficiency theory, using divide and conquer and recursive strategies to solve complex

problems, innovating data structure applications based on practical problems, enhancing system design capabilities through rules and collaborative awareness, and achieving system thinking training through multidimensional technical solution comparison and integration. In addition, the course also focuses on cultivating students' comprehensive qualities, including reflection and evaluation, experimentation and speculation, communication and cultural practice, etc. (the quality map is shown in “Figure 3”), reflecting the three in one cultivation concept of "knowledge ability quality".

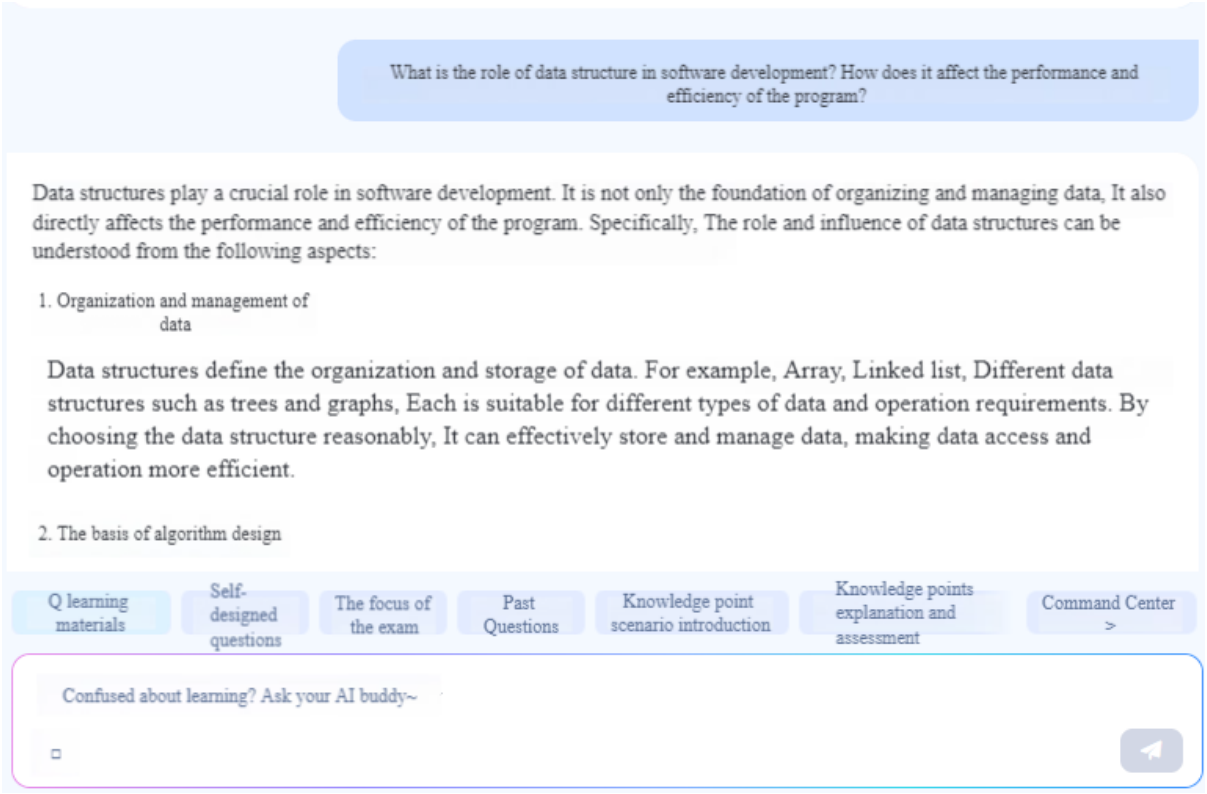


Figure 3 Intelligent Q&A.

## *2.2 Mode Framework*

The teaching model based on large models proposed in this article consists of three parts: intelligent question answering, code diagnosis, and evolutionary learning strategies, providing students with instant feedback, personalized support, and dynamic learning paths. In this mode, the large model provides students with relevant knowledge and algorithm guidance through natural language interaction and code analysis ability; It adjusts learning tasks to adapt to students' different learning levels by analyzing their learning behavior and experimental performance. The core goal of this model is to help students solve problems efficiently, while stimulating exploration interest and innovation potential. Specifically, this mode helps students quickly acquire theoretical knowledge, algorithm implementation, and code optimization suggestions through intelligent question answering modules; It analyzes code issues and provides detailed optimization solutions through the code diagnostic function; The mode uses evolutionary learning strategies to dynamically adjust task difficulty based on student performance, gradually guiding students to transition from basic learning to complex problem-solving.

Specifically, these functions are achieved through intelligent technology, as described below:

### *2.2.1 Intelligent Q&A*

The intelligent question answering module utilizes the natural language processing capabilities of large models such as GPT to provide students with real-time answers and learning support. Students can ask questions to the system through natural language, such as "How to implement queues using linked lists?" or "What is the time complexity of quicksort?" The system will automatically generate relevant algorithm descriptions, pseudocode, or even complete code

implementations and explain them with key points. As shown in “Figure 4”.


Figure 4 Code Diagnosis.

To improve learning efficiency, the Q&A module can use this feature to analyze students’ current needs based on the student’s own questions; for instance, if a student frequently asks about a particular algorithm’s implementation, the system could provide relevant references and expanded content so as to ensure that the student can understand the subject thoroughly. Another advantage of intelligent question answering, however, lies in the system’s interactive characteristics, wherein students can ask the system additional questions regarding their doubts over the explanation given by the system, thus generating a highly efficient ongoing conversation between the human user and the machine.

### *2.2.2 Code diagnosis*

The code diagnosis module makes use of the large model's code generation and analysis functions to let students discover their code's problems fast while proposing a highly thorough and specific optimization solution(“Figure 5”). The three main parts of this are as follows:


Figure 5 Evolutionary learning.

#### 2.2.2.1 *Grammar Error Detection*

Once students have handed over their codes, the system will examine it for syntax, find out where errors occurred, and suggest some possible fixes; if a student has forgotten to put in a semicolon in the program, say, the system is able to locate it and warn of "missing semicolon".

#### 2.2.2.2 *Logical vulnerability analysis*

The system detects logical errors in the program such as recursive function stack overflow risk, and loop invariant errors, detects and suggests a solution to improve the program quality. For example, if the code's recursion depth is too deep and may cause a stack overflow, then the system will suggest refactoring with iteration and provide an alternate way of implementing this.

#### 2.2.2.3 *Performance optimization*

The system can analyze the time and space complexity of the code and recommend more efficient algorithm implementations. For example, for an $O(n^2)$ sorting algorithm, the system will recommend students to use $O(n \log n)$ quicksort and provide optimized code and complexity analysis.

Table 1. Comparison of completion status of representative questions

| Admission Year | Linear Table(%) | Stack and Queue(%) | String (%) | Array and Generalized Table(%) | Tree (%) | Graph(%) | Search(%) | Sort(%) |
|---|---|---|---|---|---|---|---|---|
| 2022 | 97.16 | 96.52 | 97 | 97.21 | 96.26 | 95.27 | 94.82 | 96.7 |
| 2023 | 99.63 | 98.58 | 98.26 | 97.86 | 99.16 | 98.57 | 97.67 | 97.63 |

### 2.2.3 *Evolutionary Learning Strategy*

The evolutionary learning strategy dynamically adjusts the difficulty of tasks by analyzing students' learning behavior data (completion time, error rate, and interaction records), gradually guiding students to transition from basic knowledge learning to complex problem exploration (“Figure 5”).

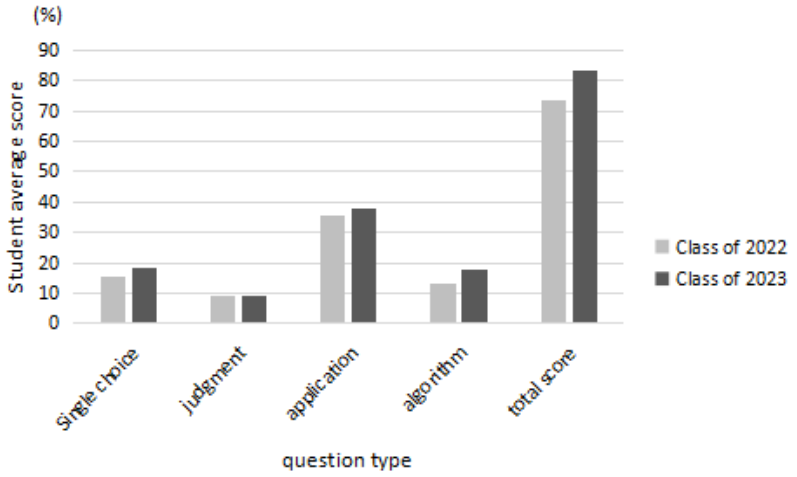


Figure 6 Comparison of Final Exams between 2022 and 2023.

#### 2.2.3.1 *Recommended Basic Tasks*

For students with weaker basic abilities, the system will recommend more fundamental and guiding experimental tasks. For example, when learning sorting algorithms, the system will first recommend tasks for implementing bubble sort and insertion sort, and provide detailed guidance steps and code templates.

#### 2.2.3.2 *Advanced Task Recommendation*

For students with strong abilities, the system will recommend more complex algorithm optimization tasks or open-ended problems. For example, after mastering the sorting algorithm, the system may require students to optimize the implementation of quick sorting or design a priority queue that supports multiple functions.

#### 2.2.3.3 *Real Time Difficulty Adjustment*

The system adjusts the difficulty of tasks in real-time based on the completion status of students. For example, if a student frequently makes mistakes in a certain task, the system will reduce the difficulty of the task and break down the goal into multiple small steps. On the contrary, if students complete tasks quickly and with high accuracy, the system will automatically increase the difficulty of subsequent tasks and encourage students to try more challenging content.

## 3. THE TRANSFORMATION OF LEARNING HABITS IN THE CONTEXT OF LARGE MODELS

In the experimental teaching environment that integrates large models, students' learning behavior and cognitive styles exhibit significant structural changes. The traditional teaching method of teacher

centered and fixed pace knowledge acquisition has gradually been replaced by an adaptive learning mode that focuses on students as the main body and obtains resources on demand. Specifically manifested in the following three aspects:

Firstly, students' questioning style has shifted from "passive response" to "active exploration". With the support of large models, students can ask complex questions such as' How to design a generalized table structure that supports dynamic expansion? 'at any time. The system not only provides answers, but also guides students to think about the advantages and disadvantages of different data structures and their applicable scenarios, thereby cultivating their systematic thinking and critical analysis abilities.

Secondly, students have higher demands for real-time and personalized feedback. They tend to frequently request error checks and performance evaluations from the system during the code writing process, and adjust implementation strategies in real-time based on diagnostic results. This fast iteration mechanism of "encoding feedback optimization" significantly improves debugging efficiency and algorithm understanding depth.

Thirdly, students' learning objectives extend from "completing basic experiments" to "solving open-ended problems". Motivated by the advanced task recommendation mechanism of the system, more students are attempting to challenge comprehensive problems such as "designing LRU cache structures" or "optimizing distributed sorting algorithms". They began actively integrating multi chapter knowledge (such as hash tables and linked lists, divide and conquer and graph algorithms), exploring innovative solutions that demonstrate stronger knowledge transfer and engineering practice capabilities.

## 4. TEACHING EFFECTIVENESS

By applying the proposed AI based teaching model to the teaching of computer major students in 2023, students' enthusiasm and exploration ability have been improved, their hands-on ability has been enhanced, and their self-learning habits have been cultivated, achieving the teaching objectives. In order to test the effectiveness of the teaching mode, significant comparative results were obtained by comparing it with the learning situation of students in the 2022 class. Firstly, the students' autonomous completion of learning tasks was tested. These typical questions comprehensively cover the main content of this course, and by analyzing the completion rate of representative questions, students' enthusiasm can be considered. The analysis results are shown in “Table 1”, indicating that after the reform, students' completion of various types of questions has become more complete, especially their enthusiasm for learning key and difficult knowledge has significantly improved.

In addition, the final grades of students in these two grades were analyzed separately, and the comparison results are shown in “Figure 6”. It can be seen that adopting this teaching model has enabled 2023 students to achieve higher overall scores. Especially in application problems and algorithm design problems that require higher professional quality, it performs better. Therefore, the final total score reflects the improvement of students' comprehensive quality, rather than just rote memorization of theoretical knowledge. Through two comparative experiments, it can be seen that the AI based teaching mode has a significant effect on promoting talent cultivation.

## 5. CONCLUSION

This article studies the transformation of students' learning habits under the background of big models and proposes an artificial intelligence assisted teaching mode based on big models, which optimizes data structure teaching through intelligent question answering, code diagnosis, and evolutionary learning strategies. The research results indicate that this mode significantly improves students' learning efficiency, interest, and innovation ability, providing empirical support for exploring new teaching models. Future research can further optimize the adaptability of large models in teaching scenarios and explore their potential applications in more disciplines, promoting the in-depth development of artificial intelligence technology in the field of education.

## REFERENCES

[1] LIU L, WU Y F, CHEN W W, et al. Challenges and Opportunities of Large Language Models in Programming Courses in Higher Education[J]. Computer Education, 2025, (02): 118–122. DOI:10.16512/j.cnki.jsjjy.2025.02.027.

[2] WANG L, YANG W, HAN Q Y. Research on the Construction of Multimodal Curriculum Maps in Open Education Based on Large

Models and Their Value Logic[J]. Shaanxi Open University Journal, 2025, 27(01): 18–23.

[3] LIU Q. Discussion on the Application of AIGC Technology in Teaching Work[J]. Computer Knowledge and Technology, 2025, 21(10): 134–136+139. DOI:10.14004/j.cnki.ckt.2025.0536.

[4] LI X, LIU X P, DAI H, et al. Innovative Model Design and Practice of Applying Prompt Engineering in Data Structures Course Teaching[J]. Computer Education, 2025, (04): 131–136. DOI:10.16512/j.cnki.jsjjy.2025.04.047.

[5] LIN J, ZHOU Y Z, SONG Y F, et al. Technical Application and Implementation of Large Models in Semantic Standardization of Data Structures[J]. Science and Innovation, 2025, (08): 222–224+228. DOI:10.15913/j.cnki.kjycx.2025.08.059.

[6] LIANG P P, ZHANG Z. Exploration of the Application of Large Language Models in Teaching "Data Structures" to International Students[J]. Technology Wind, 2025, (14): 8–10. DOI:10.19392/j.cnki.1671-7341.202514003.

[7] ZHANG Y. Research on the Construction and Recommendation of Curriculum Content Knowledge Graphs Based on Large Language Models in Smart Education[D]. Sichuan Normal University, 2024. DOI:10.27347/d.cnki.gssdu.2024.001686.

[8] ZHANG Y. Research on the Construction and Recommendation of Curriculum Content Knowledge Graphs Based on Large Language Models in Smart Education[D]. Sichuan Normal University, 2024. DOI:10.27347/d.cnki.gssdu.2024.001686.

[9] YOU S S, DAI H, BAO B K. Exploration of C Language Algorithm and Data Structures Course Reform Driven by Large Models and OBE Educational Philosophy[J]. Chinese Character Culture, 2024, (22): 178–180. DOI:10.14014/j.cnki.cn11-2597/g2.2024.22.052.

[10] GUO N X, DONG Q, XU X F, et al. A Preliminary Study on Teaching Methods for Data Structures Course Based on ERNIE Bot[J]. Journal of Science and Education, 2023, (21): 95–100. DOI:10.16871/j.cnki.kjwh.2023.21.026.

[11] SUN H R, WANG Z H, WU Y F, et al. An Enhanced Q&A Method for Educational Large Models Based on Reranking and Post-Retrieval Reflection[J/OL]. Big Data Research, 1–17[2025-09-04]. https://link.cnki.net/urlid/10.1321.g2.20250709.1504.012.

[12] WANG G X. Construction and Application Research of an Intelligent Q&A Model for Courses Integrating Knowledge Graphs[D]. Hubei Normal University, 2025. DOI:10.27796/d.cnki.ghbsf.2025.000014.

[13] WAN L L, LIU Y J, LONG H J, et al. Automatic Theorem Proving for Linear Data Structures Assisted by Lemmas Based on Large Models[J]. Journal of Jiangxi Normal University (Natural Science Edition), 2024, 48(05): 459–463. DOI:10.16357/j.cnki.issn1000-5862.2024.05.03.

[14] ZHENG W P, LI F J, GUO Y J, et al. Teaching Reform and Practice of Data Structures and Algorithms Course Based on Participatory Teaching Method[C]// China Computer Federation, National University Computer Education Research Society, Ministry of Education Teaching Guidance Committee for Computer Science Majors. Proceedings of the 2024 Chinese Conference on Computer Education in Universities. School of Computer and Information Technology, Shanxi University;, 2024: 187–194. DOI:10.26914/c.cnkihy.2024.074212.

[15] LI X N. Knowledge Graph Completion Algorithm for Multi-Type Graph Data Structures[D]. Dalian Maritime University, 2024. DOI:10.26989/d.cnki.gdlhu.2024.000103.