



자동 채점 시스템을 활용한 스택과 큐 자료 구조 학습 자료 개발

Development of Stack and Queues Data structure learning Materials Using Online Judge system

정상수[†] · 고헌능^{††} · 이영준^{†††}

Sangsu Jeong[†] · Hakneung Go^{††} · Youngjun Lee^{†††}

요약

인공지능과 컴퓨터 과학의 발전에 따라 프로그래밍 교육이 중요해지고 있다. 정보과학을 학습하는 고등학생 및 정보 영재는 자료 구조, 특히 스택과 큐를 정확히 이해하고 문제 해결에 사용하는 것이 중요하지만, 현재 교육 현장에서는 이에 대한 체계적이고 심층적인 학습 자료가 부족한 상태이다. 이 연구는 고등학교 정보과학 교육에서 스택과 큐 자료 구조의 이해와 활용을 돕기 위해 개발된 학습 자료에 대해 다루고 있다. 2022 개정 교육과정을 기반으로, 스택과 큐의 원리를 체계적으로 학습하고 문제를 해결할 수 있는 자료를 제공함으로써 학생들의 컴퓨터 과학에 대한 흥미와 문제 해결 능력을 향상하는 것을 목표로 한다. 자동 채점 시스템을 활용하여 학생들은 이론 학습과 실습을 동시에 진행할 수 있고, 실시간으로 코드를 제출하고 결과를 즉시 확인할 수 있다. 개발한 학습 자료의 타당성을 검증하기 위해, 정보교육 전문가 그룹과 함께 통계 분석을 시행하였고, I-CVI 값은 .80 이상으로 나타나 내용 타당도가 확보된 것으로 평가되었다. 또한, 본 프로그램은 다른 자료 구조나 알고리즘 교육 활동에도 확장 적용할 수 있다.

주제어 컴퓨터 교과교육, 알고리즘 교육, 온라인 자동 채점 시스템, 정보 과학, 스택과 큐

ABSTRACT

As artificial intelligence and computer science advance, the importance of programming education is increasing. For high school students and gifted individuals in information science, it's crucial to accurately understand and use data structures, particularly stacks and queues, for problem-solving. However, there is currently a lack of systematic and in-depth learning materials on this topic in the educational field. This study discusses a learning program developed to help understand and utilize the data structures of stacks and queues in high school information science education. Based on the revised curriculum of 2022, it aims to enhance students' interest in computer science and their problem-solving abilities by providing materials that systematically teach the principles of stacks and queues and solve problems. With the use of an automatic grading system, students can simultaneously proceed with theoretical learning and practice, submit their codes, and check the results in real time. To verify the validity of the developed learning program and materials, a statistical analysis was conducted with a group of information education experts, and the I-CVI value was found to be 0.80 or higher, indicating that content validity was secured. Furthermore, this program can be extended and applied to other data structure or algorithm education activities.

Keywords Computer Education, Algorithm Education, online Online Judge system, computer science, Stacks and Queues

†중심회원	한국교원대학교 컴퓨터교육과 박사수료
††정 회원	한국교원대학교 컴퓨터교육과 박사졸업
†††중심회원	한국교원대학교 컴퓨터교육과 교수(교신저자)
논문투고	2024년 04월 05일
심사완료	2024년 08월 16일
게재확정	2024년 08월 19일
발행일자	2024년 09년 04일

1. 서론

4차 산업혁명의 도래로 산업 구조는 소프트웨어 중심으로 재편되고 있다. 이 변화는 빅데이터, 초연결성, 인터넷, 그리고 인공지능 기반의 서비스 산업의 등장으로 이어지며, 기존의 제조 산업도 소프트웨어 기반으로 융합 및 재구조화되고 있다. 따라서, 소프트웨어 역량의 중요성이 점점 증가하는 가운데, 현재의 소프트웨어 교육은 단순한 일방적 강의를 넘어서 학습자와 교사 간의 상호작용, 인공지능 기반 학습 분석 및 관리 시스템, 그리고 학습자 간 커뮤니티를 통한 학습 효과 극대화를 추구하고 있다. 그러나 초, 중, 고등학생을 대상으로 하는 소프트웨어 교육은 주로 기초 프로그래밍 및 코딩 실습에 초점을 맞추고 있어 자료구조와 알고리즘 구현 역량을 키울 기회가 부족하다[1, 3, 20]. 따라서 자료구조 및 알고리즘 구현 역량을 키울 수 있는 교육 자료가 필요하다.

문제 해결에 필요한 자료 구조와 알고리즘을 문제 해결을 위해 선택하는 방식으로 접근함으로써, 학습자는 학습 목표의 본질을 더 깊이 이해할 필요가 있다[3]. 하지만 현재 고등학교 교육 현장에서는 스택, 큐와 같은 핵심 자료 구조에 대해 체계적이고 심층적인 학습 자료가 부족한 상황이다. 기존 정보과학 교과서나 참고 자료들은 대부분 이론적인 설명에 치중하며, 학생들에게 스택과 큐를 직접 구현하고 활용할 문제 해결 기회를 충분히 제공하지 않는다. 이로 인해 학생들은 스택과 큐의 개념을 추상적이고 어렵게 느끼며, 실제 프로그래밍 상황에서의 활용법을 명확히 이해하지 못하는 경우가 많다[2, 4].

이러한 문제를 해결하기 위해 자동 채점 시스템이 중요한 역할을 할 수 있다. 자동 채점 시스템은 학생들이 프로그래밍 언어의 문법을 학습한 후, 다양한 문제를 창의적으로 해결하고 자신의 프로그램이 정확성과 효율성을 확인하는 데 도움을 준다. 이는 교수자와 학습자 모두에게 큰 이점을 제공한다. 자동 채점 시스템을 도입함으로써, 학생들은 스택과 큐와 같은 자료 구조를 실제로 구현하고 활용해보는 기회를 얻게 되어 개념을 보다 구체적이고 명확하게 이해할 수 있다[3, 5, 6]. 또한, 자동 채점 시스템은 고등학교에서 정보 과학을 활용한 문제 해결 능력을 키우는 교수-학습 도구로 활용될 수 있으며, 문제의 난이도를 조정하여 정보과학 교육에서 자료 구조와 알고리즘의 심화 학습용으로도 쓰일 수 있다[6].

이에 본 연구는 자동 채점 시스템을 활용하여 정보과학을 학습하는 학생들이 스택과 큐를 더 쉽고 흥미롭게 학습할 수 있는 자료를 개발하는 것을 목표로 한다. 자동 채점 시스템은 학생들이 직접 코드를 작성하고 제출하여 결과를 즉시 확인할 수 있는 환경을 제공함으로써, 이론적인 학습과 실습을 동시에 진행할 수 있다. 이를 통해 학생들은 스택과 큐의 구조와 작동 원리를 더 깊이 이해하고, 실제 프로그래밍 상황에서 어떻게 활용할 수 있는지 체득할 수 있을 것이다.

본 연구가 제공하는 스택과 큐 프로그래밍 학습 자료는 정보과학을 학습하는 고등학생 및 정보 영재들의 컴퓨터 과학에 대한 흥미를 증진시키고, 더 나아가 프로그래밍 능력과 문제 해결 능력을 향상시키는 데 기여할 것으로 기대된다.

2. 이론적 배경

2.1. 선형 자료 구조

선형 자료 구조는 데이터 요소들이 선형 순서로 되어 있는 자료 구조를 말한다. 여기에는 배열, 연결 리스트, 스택, 큐 등이 포함되며, 각각의 데이터 요소는 그 전과 그 후의 데이터 요소와 연결된 구조로 되어 있다. 선형 자료 구조는 데이터의 저장과 접근을 효율적으로 할 수 있도록 돕는다[2, 4, 7]

2.1.1 스택(Stack)

스택은 데이터를 저장하거나 접근하는 데 사용되는 추상적인 데이터 타입으로, 나중에 들어간 데이터가 먼저 나오는 '후입선출'(LIFO, Last In First Out) 구조를 가진다. 스택 연산자는 Table 1와 같다. 스택은 일상생활에서도 쉽게 찾아볼 수 있는 구조로, 예를 들어 쟁반을 쌓아 놓은 모습과 유사하다. 제일 위에 놓인 쟁반을 먼저 가져가는 것처럼 스택에서도 가장 최근에 추가된 데이터를 가장 먼저 꺼낸다[2, 4, 7, 8].

Table 1. Stack Operations

Operations	Explanation
Push	An operation to add data to the stack.
Pop	An operation to retrieve data from the stack.
Top	An operation to view the data at the top of the stack without removing it from the stack.
IsEmpty	An operation to check if the stack is empty.
Size	An operation to return the number of data items stored in the queue.

스택은 프로그래밍에서 Table 2와 같이 다양한 상황에서 활용된다[2, 4, 7, 8].

Table 2. Stack Usage in Programming

Usage in Programming	explanation
Function Call	In programming, when a function is called, the information of the function is stored in the computer's memory stack. Once the function completes its execution, the information is popped off the stack, and control returns to the previous function.
String Reversal	A stack can be used to reverse a string. Each character of the string is pushed onto the stack, and then they are popped off in reverse order, achieving the reversed string.
Bracket Matching	In programming, stacks can be used to check if the brackets in an expression are balanced. Upon encountering an opening bracket, it is pushed onto the stack, and when a closing bracket is encountered, a bracket is popped from the stack to check for a match.
Reverse Polish Notation Calculation	Stacks can be used in the implementation of calculators or for evaluating mathematical expressions in Reverse Polish Notation.

2.1.2 큐 (Queue)

큐는 데이터를 저장하거나 접근하는 데 사용되는 추상적인 데이터 타입으로, 먼저 들어간 데이터가 먼저 나오는 '선입선출' (FIFO, First In First Out) 구조를 가진다. 큐 연산자는 Table 3과 같다. 큐는 일상생활에서도 쉽게 찾아볼 수 있는 구조로, 예를 들어 사람들이 줄을 서서 기다리는 모습과 유사하다. 줄의 앞에서부터 한 명씩 나가듯이 큐에서도 가장 먼저 추가된 데이터를 가장 먼저 꺼낸다[2, 4, 7, 9].

Table 3. Queue Operations

Operations	Explanation
Push	An operation to add data to the back of the queue.
Pop	An operation to retrieve data from the front of the queue.
Front	An operation to view the data at the front of the queue without removing it from the queue.
IsEmpty	An operation to check if the stack is empty.
Size	An operation to return the number of data items stored in the queue.

큐는 프로그래밍에서 Table 4와 같은 다양한 상황에서 활용된다[2, 4, 7, 9].

Table 4. Stack Usage in Programming

Usage in Programming	Explanation
Data Buffering	Queues can be used for data buffering. For example, they can be utilized to temporarily store and sequentially process data packets in network communication.
Task Scheduling	Queues are employed in operating systems to manage and schedule processes or tasks. Processes are added to a queue and processed in a First In First Out (FIFO) manner.
Order Processing	Queues can be used in order processing systems to manage customer orders. Orders are placed in a queue and processed in order as they are prepared.
Breadth-First Search	Queues can be utilized to implement the Breadth-First Search (BFS) algorithm. BFS is used to level-wise traverse nodes in a graph or tree structure.

2.2. 스택과 큐 교육의 중요성

스택과 큐는 컴퓨터 과학에서 매우 기본적인지만 핵심적인 자료 구조로, 프로그래밍의 다양한 분야에서 중요한 역할을 수행한다. 이들 자료 구조의 이해는 학생들에게 복잡한 프로그래밍 문제를 해결할 수 있는 능력을 개발하는데 도움을 주며, 논리적 사고와 추상화 능력을 향상시킨다. 또한, 스택과 큐를 사용하는 경험은 데이터를 효율적으로 관리하고 조작하는 방법을 배우는 데에도 기여하며, 이는 데이터 과학과 데이터베이스 관리 같은 분야에서 중요한 기술이 된다[2, 4, 7].

2.3 2022 개정 정보과학 교육과정

2022년 개정된 교육과정에서 정보과학 과목은 학생들이 다양한 문제 상황에서 적절한 데이터 구조를 선택하고 활용할 수 있는 능력을 함양하는 단원이 있다[2]. 이 단원에서는 데이터 간의 관계를 명확하게 파악하고, 그들의 특성에 알맞게 구조화의 중요성을 학생이 이해할 수 있도록 하고 있다. 2022 개정 교육과정 정보과학 과목에서 스택과 큐에 관련된 내용 체계는 Table 5와 같이 체계화되어 있다[2].

Table 5. Data Structures

Key Idea	It is crucial to choose the appropriate data structure according to the problem at hand. Understanding the relationships between data and structuring them accordingly helps to solve problems efficiently.
Categories:	Content Elements
Type	
Knowledge and Understanding	Stacks and Queues Trees and Graphs
Processes and Functions	Implementing stacks and queues using sequential data structures Exploring problems that can be solved using stacks and queues Implementing trees and graphs using adjacency matrices and adjacency lists Exploring problems that can be solved using trees and graphs
Values and Attitudes	Developing an attitude of solving problems through data structuring Recognizing the importance of data structuring for efficient problem solving

2.4. 자동 채점 시스템의 구성 요소

자동 채점 시스템은 일반적으로 Table 6과 같이 4가지의 구성 요소로 이루어져 있다.

Table 6. Components of the System

Component	Explanation
Problem	The problem that the user needs to solve.
Test dataset	A series of test data pairs consisting of input data and correct answers, used to measure the accuracy and efficiency (in terms of time and space) of algorithms for the given problem.
Automatic evaluation program	A program that compiles the source code submitted by the user into an executable file, then inputs test data to compare the output with the correct answers. It also measures the execution time and memory usage of the executable file.
User service environment	A user environment that provides services such as problems, grading status for each problem, automatic grading results, and ranking calculations.

자동 채점 시스템을 운영하기 위해선 구조화된 문제와

그에 맞는 채점데이터 세트가 필요하다. 채점데이터 세트는 알고리즘의 정확성과 효율성을 평가하기 위해 입력과 정답 데이터를 포함한 여러 쌍으로 구성된다. 또한 자동 채점 시스템은 채점데이터 세트를 이용하여 시간과 공간 복잡도를 테스트 한다. 각 문제는 알고리즘의 기본 조건을 고려하여 설계되어야 하며, 입력 데이터의 체계적인 구성을 통해 알고리즘의 정확성, 입력 데이터의 다양한 범위를 통해 알고리즘의 효율성을 평가한다.

2.5. 자동 채점 시스템의 특징

자동 채점 시스템은 문제를 제시하고 사용자로부터 그 문제를 해결하는 소스 코드를 제출받아, 컴파일 오류와 실행 오류에 대한 정보를 제공하며, 소스 코드에 내재된 알고리즘이 해당 문제를 해결하는 데 있어 정확하고 시간적, 공간적으로 효율적으로 구현되었는지를 자동으로 평가하는 시스템이다. 이러한 시스템은 평가의 공정성과 객관성을 담보해야 하는 프로그래밍 경시 대회에서 많이 활용되고 있다[3]. Table 7은 현재 정보 교사가 운영하는 ‘알고리즘 자동 채점’ 사이트이다. 2022년 개정 정보 교육과정과 2015년 개정 정보교육 과정의 교수, 학습 및 평가 방법에서 자동 채점 시스템의 활용을 권장 있다[2, 16]. 이에 따라 많은 학교에서 프로그래밍 수업과 수행 평가에 자동 채점 시스템을 활용하고 있다. [33, 34]. 자동 채점 시스템은 일관성 있고 정확한 평가, 즉각적인 피드백, 효율성 향상, 접근성 증대, 안정성과 확장성, 학생의 학습 데이터 수집 등을 제공함으로써 학생과 교사에게 다양한 피드백을 제공한다[6, 10, 11, 12, 13]

Table 7. Domestic online judge operated by teacher

URL	Language
codeup.kr	C, C++, Python, Java
koistudy.net	C, C++, Python
judgeon.net	C, C++, Python, Java
codingfun.net	C, C++, Python, Java
biko.kr	C, C++, Python

2.6 ADDIE 모형

ADDIE 모형은 교육과 학습 프로그램을 개발, 평가 및 실행하는 프로세스를 체계화한 모델이다. 이 모델은 Table 8과 같이 Analysis (분석), Design (설계), Development (개발), Implementation (시행), Evaluation (평가)의 다섯 단계로 구성되어 있으며, 효과적인 학습 경험을 설계하기 위해 널리 사용된다[14]. 본 연구에서는 ADDIE 모형을 이용하여 스택과 큐에 관한 학습 자료를 개발하고 평가하였다.

Table 8. ADDIE

Process	Content
Analysis	<ul style="list-style-type: none"> · Analysis of related research and training materials · Analysis of Information Curriculum Achievement Criteria · Automatic Evaluation System Analysis
Design	<ul style="list-style-type: none"> · selection of teaching strategies · Evaluation tool design
Development	<ul style="list-style-type: none"> · development of teaching materials · Pilot Testing and Modification
Implementation	<ul style="list-style-type: none"> · Proceed with expert reviews · Application to information science classes and club activities
Evaluation	<ul style="list-style-type: none"> · Proceed with expert reviews · Study material validity statistics analysis · Revised based on expert review comments

3. 자동 채점 시스템을 활용한 스택 학습 자료 개발

3.1. 학습 자료 개발의 기본 원칙

본 연구는 컴퓨터 과학에서 광범위하게 활용되는 스택과 큐 데이터 구조에 대한 학습 자료 개발의 기본 원칙과 의도를 탐구한다. 문제 개발은 스택과 큐의 기본 구조와 작동 원리를 정확히 이해하고, 이를 다양한 상황에서 적절히 활용할 수 있는 능력을 함양하는 데 초점을 맞춘다. 또한, 문제 해결 능력을 향상시키고, 학생들이 창의적이고 비판적인 방법으로 문제에 접근할 수 있도록 유도한다. 문제는 실생활과 연결되어 학생들이 학습 내용의 실용성을 인식하고, 이론과 실천을 연결하며 학습에 대한 흥미와 동기를 증진시킬 수 있도록 설계되어야 한다. 마지막으로, 학생들의 반응과 성과 분석을 통해 문제를 지속적으로 개선하고 학습 효과를 극대화하는 것이 중요하다. 이러한 접근 방식을 통해 개발된 학습 자료는 학생들이 이론적 지식을 넘어 실제 문제 해결 능력을 함양하는 데 크게 기여할 것이다.

3.2 학습 자료 개발 절차

프로그램 개발을 위해, 우리는 세 가지 주요한 교육내용 선정 기준을 도입하고 적용하였다. 첫째, 학습자의 수준을 고려하여 모든 학습자가 교육내용을 이해하고 소화할 수 있도록 하였다. 둘째, 창의적 문제 해결 능력을 향상하기 위해 적절한 교육내용을 선별하였다. 셋째, 교육 시간과 기타 관련 요소들을 고려하여 교육내용을 최적화하였다. 이러한 개발과정은 Table 8에 명시된 바와 같이 ADDIE 교수 체제 설계 모델을 기반으로 진행하였다.

3.3 학습 자료의 구성

학습 자료는 스택의 원리 이해, 스택을 활용, 큐의 원리 이해, 큐의 활용이 순차적으로 이루어질 수 있도록 학습 자료를 개발하였다. 이때, 스택과 큐의 구조를 학습하는 문제

부터 스택과 큐를 이용하여 해결하는 난이도 높은 문제 순으로 개발하였다. 스택과 큐 학습 자료 개발은 “[12정과02-01]스택과 큐의 원리를 이해하고, 순차적인 데이터 구조를 이용하여 스택과 큐를 구현한다.”와 “[12정과02-02] 스택, 큐를 활용하여 문제를 효율적으로 해결하는 프로그램을 작성한다.”에 맞춰 Table 9과 같이 8차시 분량의 교수학습 모델과 전략 및 학습 자료를 개발하였다.

Table 9. Development Items

Step	Difficulty	Contents	Time
Understanding Stack	Essay	Utilizing Parenthesis Problems Insertion and Deletion in a Stack Counting Elements Stored in the Stack Top Element of the Stack	1 hour
Applications of Stack	Hard	1D(1*n) 2048 Game 2D(n*n) 2048 Game	3 hours
Understanding Queues	Normal	Using 1D and 2D Arrays Insertion and Deletion in a Queue Using a Check Array to Confirm Search Completion	1 hour
Applications of Queue	Hard	Knight's Movement Propagation of News Scheduling	3 hours

4. 자동 채점 시스템을 활용한 스택 학습 자료 개발

선형 자료 구조 학습 자료의 타당도를 검증하기 위해서 정보 과학 또는 알고리즘 및 자료 구조를 지도한 경험이 있는 전문가의 검토 과정을 거쳤다. 교육전문가는 자동 채점 시스템을 수업에 직접 활용하여 프로그래밍 교육 또는 정보 과학 교육 경험이 있는 현장 교사 8명으로 구성하였다.

4.1 전문가 검토 결과

자동 채점 시스템을 활용한 효과적인 선형 자료 구조 학습 자료에 대한 타당도 검토 문제는 선행연구[36]를 토대로 중등 정보 교육 전문가의 협의를 통해 구성하였고, 검토 문제는 Table 10와 같다. 중등 교육전문가 타당성 검토 문항은 적용 가능성(7문항), 유용성(2문항), 정보과 핵심역량 반영(3문항), 학생 요구(2문항), 학습 만족도(3문항) 영역으로 구성하여 총 18문항이었다. 각 문항은 4점 리커트 척도로 구성하였다.

Table 10. Questionnaire of Content Validity

Area	Contents
Applicability	1. Produced according to the national curriculum.
	2. It was made according to the learning objectives.
	3. Teaching and learning methods are suitable for use in the classroom.
	4. Provide appropriate guidelines for student-teacher behavior.
	5. The content presented is systematic.
	6. It is likely to be applied in the classroom.
	7. Utilizing learning materials helps to improve existing teaching and learning methods.
	8. It was developed to facilitate evaluation after application of the class.
usability	9. Suitable for utilizing automatic evaluation system.
	10. This learning material well reflects the principles of linear data structures (stacks and queues).
Competencies in computer Science	11. Improve learners' computational thinking skills.
	12. Improve learners' problem-solving skills.
	13. Improve the creativity of learners.
student needs	14. It is suitable for cultivating the principles and concepts of AI/SW.
	15. It will be helpful in nurturing convergence talent.
Learning Satisfaction	16. Students will be satisfied with their lessons using this learning material.
	17. The content level is appropriate for the target student.
	18. This study material will arouse the interest of the students.

전문가들의 검토 결과를 수집하여 통계 분석 방법인 I-CVI(Item Content Validity Index) 산출식을 활용하여 분석하였다. I-CVI의 계산을 위하여 각 문항에 대하여 “매우 긍정”, “긍정”으로 응답한 경우 1으로, “부정”, “매우 부정”으로 응답한 경우 0으로 치환하였다. CVI의 절단점은 .80으로 두었다. 결과는 Table 11과 같다.

Table 11. Expert Advisory Results

Area	Item	Mean	SD	I-CVI
Applicability	1	3.88	0.35	1
	2	3.88	0.35	1
	3	3.88	0.35	1
	4	3.50	0.53	1
	5	3.88	0.35	1
	6	3.75	0.46	1
	7	3.75	0.46	1
	8	4.00	0.00	1
Usability	9	4.00	0.00	1
	10	4.00	0.00	1
Competencies in computer Science	11	4.00	0.00	1
	12	4.00	0.00	1
	13	3.88	0.35	1

Area	Item	Mean	SD	I-CVI
Student needs	14	3.75	0.46	1
	15	3.88	0.35	1
Learning Satisfaction	16	3.88	0.35	1
	17	3.63	0.52	1
	18	3.75	0.46	1

전문가 자문 결과 모든 문항에 대해 I-CVI 값이 .80 이상으로 나타났기 때문에 본 연구에서 개발한 교육프로그램은 내용 타당도가 통계적으로 확보되었다고 할 수 있다. 또한 검토에 참여한 전문가의 다양한 의견을 수집할 수 있었다.

전문가 A: 문제 제시에 체계적인 예시 자료 제공이 되었으면 학습하는 학생에게 도움이 될 것 같다.

전문가 B: 기본적으로 자료구조에 대한 이론만 다루는 경우가 많은데 이 자료는 자료구조의 본질을 이해한 후 적용하여 문제를 해결하는 도구로 활용할 수 있는 역량을 기를 수 있도록 구성된 점이 독보적임. 매우 참신한 콘텐츠로 과학고 영재학교 학생들을 대상으로 한다면 매우 가치 있는 콘텐츠로 판단됨.

전문가 C: 학생들이 문제를 정확하게 해결하기 위해 더 많은 시간이 필요할 수 있음.

이 밖에도 학생 수준에 맞는 단어를 선택하여 지문을 구성해야 한다는 개선 의견도 수집할 수 있었다.

4.2 개발 결과

정보 올림피아드, atcoder.jp, codeforces.com 등에 출제된 문제 중 스택과 큐 자료 구조와 관련된 문제를 분석하여 정보과학 수업에 사용할 수 있는 문제를 Figure 1과 같이 개발하였다.

Background
OOO은 짝 맞추기 게임을 즐겨한다.

짝 맞추기 게임은 3장의 카드('a', 'b', 'c')가 있고 2개의 동일한 카드가 연속적으로 있으면 다음 카드 1장으로 바꾸는 게임이다.

'a'와 'a'가 이웃 하면 'b' 카드 한 장으로 바꿈
'b'와 'b'가 이웃 하면 'c' 카드 한 장으로 바꿈
'c'와 'c'가 이웃 하면 'a' 카드 한 장으로 바꿈

만약 바꿀 수 있는 카드의 짝이 여러개이면 앞에서부터 순서대로 바꾼다. 또한 바꿀 수 없을 때까지 계속해서 짝을 맞추어 카드를 바꾼다.

예를 들어 카드 배열이 "aabccc"이면 'b'가 된다.

1. aabccc -> bbccc
2. bbccc -> cccc
3. cccc -> acc
4. acc -> aa
5. aa -> b

Input
첫 째 줄에 카드 배열을 나타내는 문자열(스트링)이 주어진다. 단 그 길이는 1 이상, 30 이하이다.

Output
게임이 끝난후 카드 배열을 출력한다.

IO Example
입력
aabccc

출력
b

Figure 1. Programming Problems

Figure 1은 스택 학습 자료 중 하나로 스택의 원리를 학습할 수 있도록 문제 난이도 등을 고려하여 구성하였다.

Background
OO가 다니는 회사에서 희기적은 A라는 병에 대한 백신을 개발하였다. 이 백신은 다음과 같은 특징을 가지고 있다.

- 백신을 투약 받으면 1분 뒤에 A라는 병에 대한 면역력이 생긴다.
- 병에 대한 면역을 가지지 않은 사람이 면역을 가진 사람 옆에 있으면 1분 뒤에 면역력이 생긴다.

n명이 한 줄로 서 있을 때 m명에게 백신을 투여할 때 어느 정도의 시간이 지나면 모든 사람은 면역을 가진다.

OO이는 m명의 사람에게 백신을 투여할 때, 모든 사람이 각각 몇 분 후에 면역을 가졌는지 궁금증이 생겼다. 경박이를 위해 모든 사람이 각각 몇 분 후에 병에 대한 면역을 가졌는지 출력해보자.

예를 들어 10명을 사람이 있고 3, 7번째 사람에게 백신을 투여하면 1분 후면 3, 7번이 면역을 가지고 2분 후면 3번 사람에 의해 2, 4번 사람이 면역을 가지고 7번 사람에 의해 6, 8번 사람이 면역을 가진다.

따라서 1번부터 10번 까지의 사람이 면역을 가진 시간은 3 2 1 2 3 2 1 2 3 4 이다.

Input
첫 번째 줄에는 n과 m이 공백을 기준으로 입력된다.
두 번째 줄에는 백신의 투약 받는 사람의 위치 x가 공백을 기준으로 입력된다.

[입력값의 정의역]
1 <= m <= n <= 1,000,000
1 <= x <= n

```

Output
면역이 생긴 시간의 공백을 기준으로 출력한다.

IO Example
입력
10 2
3 7

출력
3 2 1 2 3 2 1 2 3 4
    
```

Figure 2. Programming Problems

Figure 2는 큐 학습 자료 중 하나로 큐의 원리를 학습할 수 있도록 문제 난이도 등을 고려하여 구성하였다.

Rank	user	A[100]	B[100]	C[100]	D[100]	E[100]	F[100]	G[100]	H[100]	Score(penal)
1		100	100	100	100 (2)	100	100	100	100 (3)	800.0 (-5)
2		100	100	100 (1)	100	100 (2)	100	100	100 (3)	800.0 (-6)
3		100	100	100 (1)	100 (2)	100 (2)	100 (1)	100	100 (1)	800.0 (-7)
4		100	100	100	100 (2)	100	100 (8)	100	100 (1)	800.0 (-9)
5		100	100	100 (1)	100 (1)	100 (5)	100	100	100 (8)	800.0 (-13)
6		100	100	100 (1)	100 (3)	100 (8)	100	100 (1)	100 (3)	800.0 (-14)
7		100	100	100 (3)	100 (3)	100 (1)	100 (3)	100 (1)	100 (3)	800.0 (-14)
8		100 (1)	100	100 (5)	100 (5)	100	100	100	100 (4)	800.0 (-15)
9		100	100	100 (4)	100 (9)	100 (3)	100	100	100 (1)	800.0 (-17)
10		100	100	100 (7)	100 (3)	100 (2)	100 (4)	100 (1)	100 (3)	800.0 (-20)
11		100	100 (1)	100 (3)	100 (12)	100 (8)	100 (1)	100	100 (3)	800.0 (-26)
12		100 (1)	100 (1)	100 (8)	100 (1)	100 (15)	100	100	100 (9)	800.0 (-30)
13		100	100	100 (8)	100 (18)	100 (4)	100 (4)	100	100 (4)	800.0 (-36)
14		100 (2)	100 (3)	100 (10)	100 (10)	100 (13)	100 (1)	100	100 (5)	800.0 (-42)
15		100 (1)	100	100 (5)	75.0 (16)	92.5 (4)	100 (3)	100	100 (5)	767.5 (-34)
16		100	100	100 (10)	100 (15)	10.1 (8)	100 (2)	100	100 (3)	710.1 (-33)

Figure 3. Apply to programming class

Figure 3은 개발한 문제를 자동 채점 시스템에 통합하여 수업에 적용한 사례를 보여준다. 문제당 만점은 100점이며, 100점 미만의 점수는 해당 문제에서 부분 점수를 받았음을 의미한다. 괄호 안에 표시된 숫자는 학생이 해당 문제를 시도한 횟수를 의미하며, 숫자가 클수록 더 많이 시도한 것이다.

스택과 큐 학습을 목적으로 개발된 8차시 문제를 프로그래밍 과목을 신청한 200명 이상의 고등학교 2학년 학생들에게 적용해 본 결과, 이들 문제가 스택과 큐 수업에 적절히 활용될 수 있는 난이도를 가지고 있음을 확인할 수 있었다. 또한, 수업에 참여한 학생들로부터 다양한 의견을 수집할 수 있었다.

학생 A: 자료구조를 이용하면 복잡한 문제들을 보다 쉽게 풀 수 있다는 사실을 배웠음.

학생 B: 큐는 선입선출, 스택은 선입후출이라는 개념이 직관적으로 다가오지 않았는데, 문제를 해결하

면서 큐와 스택의 개념을 이해하는 데 큰 도움이 되었음.

학생 C: 큐와 스택 문제들을 풀게 되면서 c++의 스택, 큐에 대한 다양한 활용 방법을 알게 되었음. 큐와 스택 모두 일반적인 배열로 구현이 가능한데, 이렇게 간단한 자료구조가 많은 문제에 유용하게 사용된다는 점이 놀라웠음.

5. 결론

이 연구는 인공지능과 컴퓨터 과학의 중요성이 증가함에 따라 고등학교 정보과학 교육에서 필수적인 스택과 큐 자료구조를 체계적으로 학습할 수 있는 프로그램 문제를 개발하였다. 2022년 개정된 교육과정을 분석하여 스택과 큐의 원리 이해 및 문제 해결 능력 향상을 목표로 설정하고, 이를 바탕으로 학생들이 이론과 실습을 병행할 수 있는 자동 채점 시스템 문제를 설계하였다.

본 연구에서 제안된 스택과 큐 학습 자료의 타당성을 검증하기 위해, 프로그래밍 수업을 위해 만들어진 코드 업과 코이스터디를 운영하는 정보 교사들을 포함한 정보교육 전문가 그룹을 활용한 통계 분석을 수행하였다. 개발된 학습 자료 및 프로그램은 전문가의 검토를 거쳐 모든 항목이 절단점 이상의 I-CVI 값 .80을 달성함으로써 내용 타당도를 확보하였다.

이 연구를 통해 개발된 스택과 큐 학습 자료는 자동 채점 시스템을 활용하는 정보과학 수업에 적용될 수 있음을 확인하였다. 또한, 교사는 학생의 성취 수준을 실시간으로 확인할 수 있어 평가의 부담을 줄일 수 있다. 이 학습 자료는 다른 자료 구조 및 알고리즘 교육 활동에도 확장 적용될 수 있을 것이다.

참고문헌

- [1] KAIST Global Institute for Talented Education. (2018) *A Study on Innovative Strategies for Software Talent Education in Preparation for the Fourth Industrial Revolution*. Daejeon.
- [2] Ministry of Education. (2022). *The 2022 Revised General Guidelines of National Curriculum*.
- [3] Chang, W.Y., Kim, S.K.(2017). A review on trends of programming(algorithm) automated assessment system and it s application. *The Journal of Korean association of computer education*. 20(1), 13-26. <https://doi.org/10.32431/kace.2017.20.1.002>
- [4] Jeon, H.S., Kim, B.S., Moon, J.E., Park, J.H., Yoon, S.H., Lim, J.B., Jung, W.Y., Jeong, J.G. (2019). *High School Computer Science Textbook*. SAMYANGMEDIA.
- [5] Lim, H.S. (2007). ACM-ICPC problem formulation and grading process. *Communications of the Korean Institute of Information Scientists and Engineer*. 25(7). 52-55.

- [6] Chang, W.Y., & Kim, S.K. (2014). Development and application of algorithm judging system : analysis of effects on programming learning. *The Journal of Korean association of computer education*, 17(4), 26-34. <https://doi.org/10.32431/kace.2014.174.005>
- [7] Horowitz, E., Sahni, S., & Anderson-Freed, S. (1993). *Fundamentals of data structures in C* (Vol. 1). New York: Computer Science Press.
- [8] Stack. (2024, April 4). Retrieved from [https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type)).
- [9] Queue. (2024, April 4). Retrieved from [https://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type))
- [10] Jeong, K.J. (2010). *Design and Construct of Programming Assessment System based on Online Judge for a Science High School student* [Master's thesis]. Korea National University of Education.
- [11] Jeong, S.S. (2019). *The Effects of Programming Education Using an Automatic Programming Assessment System on Learning Flow of General High School Students* [Master's thesis]. Korea National University of Education.
- [12] Kim, S.K., & Oh, S.H., & Jeong, S.S. (2018). Development and Application of Problem Bank of Problem Solving Programming Using Online Judge System in Data Structure Education. *The Journal of Korean association of computer education*. 21(6), 11-20. <https://doi.org/10.32431/kace.2018.21.4.002>
- [13] Jeong S.S., & Go, J.N., & Lee, Y.J. (2018). Development and Application of Non-linear Search Questions Using an Online Judge System. *The Journal of Korean association of computer education*. 27(2), 1-11. <https://doi.org/10.32431/kace.2024.27.2.001>
- [14] ADDIE Model. (2024, April 4). Retrieved from https://en.wikipedia.org/wiki/ADDIE_Model.
- [15] Ministry of Education. (2015). *The 2015 Revised General Guidelines of National Curriculum*.
- [16] Ministry of Education. (2015). *Software Education Operational Guidelines*.
- [17] Ministry of Education. (2020). *Comprehensive Plan for Information Education*.
- [18] Lee, E.K., & Lee, Y.J. (2007). Analysis of High School Students' Misconceptions on Algorithm Design and Analysis. *The Journal of Curriculum Evaluation*, 10(2), 329-348.
- [19] Chang, W.Y. (2020). Causal relationship between learning motivation and thinking in programming education using online evaluation tool. *Journal of The Korean Association of Information Education*, 24(4), 379-390. <https://doi.org/10.14352/jkaie.2020.24.4.379>
- [20] Chang, W.Y. (2021). The Effects of Online Judge System on Motivation and Thinking in Programming Education: Structural Relationships between Factors. *The Journal of Korean association of computer education*, 24(5), 1-16. <https://doi.org/10.32431/kace.2021.24.5.001>
- [21] Jeon, H.S., Jeong, K.J., & Kim, S.K. (2014). Problem Design & the Application of online judge to Basic C programming language Learning. *The Korean Association of Computer Education*, 18(1), 291-294.
- [22] Jeong, J.K., & Song, J.b., & Lee, T.W. (2009). A Current Status of Korea Olympiad in Informatics and Improvement Strategy. *The Korean Association of Computer Education*, 13(1), 187-196.
- [23] Yingwei L., Xiaolin W., & Zhengyi, Z. (2008). *Programming grid: a computer-aided education system for programming courses based on online judge*. ACM SCE '08 Proceedings, 6. <https://doi.org/10.1145/1517632.1517643>
- [24] Chang, W.Y., Kim, S.K. (2020). Differences in self-efficacy between block and textual language in programming education using online judge. *The Journal of Korean association of computer education*. 23(4), 23-33. <https://doi.org/10.32431/kace.2020.23.4.003>
- [25] Yi, S.Y., Lee, Y.J. (2021). Development of Convergence Education Program for 'Understanding of Molecular Structure' using Machine Learning Educational Platform. *JOURNAL OF The Korean Association of information Education*. 25(6), 961-972. <https://doi.org/10.14352/jkaie.2021.25.6.961>
- [29] Polit, D.F., Beck, C.T. & Owen, S.V. (2007). Is the CVI an Acceptable Indicator of Content Validity? Appraisal and Recommendations. *Research in Nursing & Health*, 30, 459-467. <https://doi.org/http://dx.doi.org/10.1002/nur.20199>
- [30] AtCoder. (2024, April 4). Retrieved from: <https://atcoder.jp/>
- [31] Codeforces. (2024, April 4). *Codeforces*. Retrieved from <https://codeforces.com/>
- [32] Baekjoon Online Judge. (2024, April 4). *Baekjoon Online Judge*. Retrieved from <https://www.acmicpc.net/>
- [33] Codeup. (2024, April 4). *Codeup*. Retrieved from <https://codeup.kr/>
- [34] Koistudy. (2024, April 4). *Koistudy*. Retrieved from <https://koistudy.net/>
- [35] Choi, J. Y. (2014). *Design and Implementation of a Programming Learning Support System Considering Interaction with a Learner* [Master's thesis]. Graduate School of SangMyung University.
- [36] Seong, T.J. (2019). *Understanding and Application of Modern Basic Statistics*. Hakjisa, Pajoo.



정상수

- 2009년 경상국립대학교 컴퓨터교육과(이학사)
- 2019년 한국교원대학교 교육대학원 컴퓨터교육 전공 (교육학석사)
- 2020년 ~ 현재 한국교원대학교 컴퓨터교육전공 박사과정

✚ 관심분야 : 프로그래밍 교육, 인공지능 교육, 교과교육
 ✉ whiteedu84@gmail.com



고학능

- 2015년 광주교육대학교 과학교육과 (학사)
- 2018년 광주교육대학교 교육대학원 초등컴퓨터 교육(교육학 석사)
- 2024년 한국교원대학교컴퓨터교육학과(교육학 박사)
- 2018년 ~ 현재 초등학교 교사

✚ 관심분야 : 프로그래밍 교육, 온라인 저지, 인공지능 교육, 데이터 과학
 ✉ snddl3@outlook.kr



이영준

- 1988년 고려대학교 전산학과(이학사)
- 1994년 미국 미네소타대학교 전산학과(Ph.D)
- 2003년 ~ 현재 한국교원대학교 컴퓨터교육학과 교수

✚ 지능형시스템, 학습과학, 정보교육, 인공지능교육
 ✉ yjlee@knu.ac.kr

부 록

Title

OOO은 짝 맞추기

Background

OOO은 짝 맞추기 게임을 즐겨한다.

짝 맞추기 게임은 3장의 카드('a', 'b', 'c')가 있고 2개의 동일한 카드가 연속적으로 있으면 다음 카드 1장으로 바꾸는 게임이다.

'a'와 'a'가 이웃 하면 'b' 카드 한 장으로 바꿈
 'b'와 'b'가 이웃 하면 'c' 카드 한 장으로 바꿈
 'c'와 'c'가 이웃 하면 'a' 카드 한 장으로 바꿈

만약 바꿀 수 있는 카드의 짝이 여러개이면 앞에서부터 순서대로 바꾼다. 또한 바꿀 수 없을 때까지 계속해서 짝을 맞추어 카드를 바꾼다.

예를 들어 카드 배열이 "aabccc"이면 'b'가 된다.

1. aabccc -> bbccc
2. bbccc -> cccc
3. cccc -> acc
4. acc -> aa
5. aa -> b

Input

첫 째 줄에 카드 배열을 나타내는 문자열(스트링)이 주어진다. 단 그 길이는 1 이상, 30 이하이다.

Output

게임이 끝난후 카드 배열을 출력한다.

IO Example

입력
 aabccc

출력
 b

[그림 1] 스택 학습 자료 예시 1

Title
2024의 신 000

Background
고등학생대 000은 2048의 신으로 불렸다.

소문으로는 2048 게임에서 2^{100} 까지 달성했다는 소문이 있다. (대박 ~~!)

2048 게임은 4*4 격자 판에서 방향키를 이용하여 진행되는 퍼즐 게임으로, <https://play2048.com/> 에서 플레이할 수 있다. 규칙은 다음과 같다.

- 4*4 격자의 각 칸은 비어있거나 2의 거듭제곱(1, 2, 4, 8, ...)이 적혀 있는 타일이 있다.
- 방향키를 누르면 모든 타일이 해당 방향으로 정렬되며, 이웃한 두 타일의 숫자가 같으면 2배 큰 타일로 합쳐진다. 또한, 방향키를 눌러도 격자에 아무런 변화가 없으면 해당 방향키를 누를 수 없다.
- 방향키를 누르면 타일 이동후, 빈 칸에 1와 2타일 중 하나가 빈칸 한 곳에 생성된다.

- 이 문제는.. 한 가지 조건을 변형하였다. 이미 합쳐진 타일이 연쇄적으로 합쳐진다.

n번의 방향키를 누른 후 결과를 출력하여 보자.

Input
1번째 줄에는 방향키를 누른 횟수 n이 주어진다.
2번째줄부터 5번째줄까지 처음 시작시 격자 판이 주어진다.
6번째 줄부터 이동 방법과 빈칸의 위치(x,y), 타일에 적혀 있는 숫자 c가 공백을 기준으로 주어진다.
(1 : 위쪽, 2 : 왼쪽, 3 : 아래쪽, 4 : 오른쪽)
($1 \leq x, y \leq 4$)
($1 \leq n \leq 10$)
($1 \leq c \leq 2$)

Output
n번 방향키를 누른 후 결과를 출력한다.

IO Example
입력
2
1 1 0 0
1 0 1 0
2 0 1 0
1 0 1 1
2 3 4 1
1 4 2 1

출력
8 2 0 1
0 0 0 0
0 0 0 0
0 1 0 0

설명

1 1 0 0 2 0 0 0 8 2 0 1
1 0 1 0 (2 3 4 1) 2 0 0 0 (1 4 2 1) 0 0 0 0
2 0 1 0 => 2 1 0 1 => 0 0 0 0
1 0 1 1 2 1 0 0 0 1 0 0

[그림 2] 스택 학습 자료 예시 2

Title
000은 짝 맞추기

Background
세계 100대 재벌 중 한 명인 000은 취미로 야생 동물들에게 영양가 높은 음식을 제공하고,
아픈 곳을 치료해주기 시작했습니다.

어느 날, m종의 n마리의 야생 동물들이 000을 찾아왔습니다.

000은 조수인 $\triangle\triangle$ 에게 각 종별로 몇 마리씩 찾아 왔으며 어떤 순서로 찾아 왔는지 정리하도록 요청했습니다.

천사보다 더 천사 같은 성격을 가진 조수 $\triangle\triangle$ 을 도와줍니다.

Input
첫번째 줄에는 찾아온 동물의 수 n이 주어진다.
두번째 줄에는 동물 종수 m이 주어진다.
세번째 줄에는 동물의 종 번호가 공백을 기준으로 주어진다. ($1 \leq n \leq 100$), ($1 \leq m \leq 20$)

Output
i번째 줄에 찾아온 i종의 전체 수와 순서를 공백을 기준으로 출력한다.

IO Example
입력
6
2
1 2 1 2 1 1

출력
4 1 3 5 6
2 2 4

[그림 3] 큐 학습 자료 예시 1

Title
주식회사 OOO

Background
주식회사 OOO의 최대 주주는 OOO이다.

주식회사 OOO에는 천사보다 더 천사같은 마음을 가진 $\triangle\triangle\triangle$ 을 비롯하여 n 명의 연구원이 일한다.

주식회사 OOO에는 m 개의 실험 장비가 있고 n 명의 연구원은 각각의 연구 프로젝트를 위해 a_i 개의 실험 장비를 순서대로 이용해야 한다.

n 명의 직원과 연구에 필요한 실험 장비가 순서대로 주어질 때, 각 실험 장비를 이용하는 순서를 출력하여 보자.

만약 같은 시간에 장치를 사용하기 위해 줄을 서야 한다면 이전 실험 장치의 번호가 빠른 사람이 먼저 줄을 선다. 이전에 사용한 실험 장치가 없으면 번호가 빠른 연구원이 앞에 줄을 선다.

연구 장비를 1번 이용하면 다시 줄을 서야 한다.

각 연구원의 장비의 사용 시간은 모두 동일하며 다른 기계로 이동 시간은 고려하지 않는다.

Input
첫번째 줄에는 연구원의 수 n 이 주어진다.
두번째 줄에는 실험 장비의 수 m 이 주어진다.
세번째 줄부터 $n+2$ 번째 줄까지 i 번째 연구원의 사용해야 하는 연구 장비의 총 수 a 와 연구 장비의 번호가 공백을 기준으로 주어진다.
($1 \leq n \leq 100$), ($1 \leq m \leq 20$), ($1 \leq a \leq 100$)

Output
 i 번째 줄에 i 번째 연구 장비가 사용된 총 횟수와 사용 연구원을 순서대로 공백을 기준으로 출력한다.

IO Example

입력
3
2
4 1 2 1 2
3 1 1 1
3 2 2 1

출력
6 1 2 2 1 2 3
4 3 1 3 1

[그림 4] 큐 학습 자료 예시 2