

온라인 자동 평가 시스템을 이용한 비선형 탐색 문항의 개발과 적용

Development and Application of Non-linear Search Questions Using an Online Judge System

정상수[†] · 고태능^{**} · 이영준^{***}Sangsu Jeong[†] · Hakneung Go^{**} · Youngjun Lee^{***}

요 약

현재까지 진행된 온라인 자동 평가 시스템 교육 관련 연구 성과에는 온라인 자동 평가 시스템 개발 및 학생의 흥미 유발에 관한 내용 중심이어서 난이도와 성취기준에 따라 수업 장면에서 사용할 수 있는 문제가 거의 없는 실정이다. 그리고 자율 학습을 하는 학생들이 어떤 방향으로 학습을 진행해야 하는지 안내하는 자료도 부족하다. 따라서 본 연구에서는 온라인 자동 평가 시스템을 적용한 프로그래밍 수업 장면에서 활용할 수 있는 탐색 교육프로그램을 개발하였다. 개발한 교육프로그램은 상태 정의를 통한 탐색 학습 프로그램으로 다양한 프로그래밍 경진대회에 출제된 동적 계획법 및 탐색 관련 문항을 분석하여 정보과학을 수강하는 학생의 수준에 맞도록 4차시 분량으로 설계 및 개발하였다. 본 연구를 통해 개발한 프로그램을 전문가 검토를 통해 타당도를 확보하였다. 전문가 설문 결과를 Item Content Validity Index를 이용하여 값을 산출하였고, 그 결과 모든 항목이 .80 이상으로 타당도를 확보하였다. 또한 전문가의 세부 의견을 반영하여 수정 및 보완하여 완성하였다. 본 연구를 통해 개발된 프로그램은 정보 과학 수업에서 온라인 자동 채점 시스템에 적용할 수 있는 문항으로 학생이 제출한 소스 코드에 대해 즉각적인 피드백을 할 수 있으며, 소스 코드의 정답 유무를 실시간으로 확인할 수 있어 교사의 수업 부담을 낮출 수 있을 것이다.

주제어: 컴퓨터 교과교육, 프로그래밍 교육, 알고리즘 교육, 온라인 자동 평가 시스템, 정보 과학

ABSTRACT

The research results related to the online judge system education conducted so far are focused on the development of the online judge system and the interest of students, so there are few problems that can be used in class scenes according to the difficulty and achievement standards. Also, there are insufficient materials to guide self-learning students in which direction they should proceed. Therefore, in this study, an exploration education program that can be used in the programming class scene to which the online judge system is applied was developed. The developed educational program is a search program through state space definition. It was designed and developed in 4 sessions to match the level of students taking information science by dynamic programming and search problem in various programming competitions. The validity of the program developed through this study was secured through expert review. The value of the expert survey results was calculated using the Item Content Validity Index, and as a result, validity was secured for all items to be .80 or higher. In addition, the detailed opinions of the experts were reflected, and the values were corrected and supplemented. The program developed through this study is a question that can be applied to the online automatic scoring system in the information science class, and can provide immediate feedback on the source code submitted by the student. It will be possible to reduce the burden of class.

Keywords: Computer Education, Programming education, Algorithm education, online Online Judge system, computer science

[†]중심회원: 한국교원대학교 컴퓨터교육과 박사과정

^{**}정 회 원: 한국교원대학교 컴퓨터교육과 박사수료

^{***}중심회원: 한국교원대학교 컴퓨터교육과 교수(교신저자)

1. 서론

제 4차 산업혁명으로 소프트웨어와 인공지능이 사회에 미치는 영향이 점차 커짐에 따라 미래 사회의 핵심이 될 학생들의 소프트웨어와 인공지능 역량이 중요해지고 있다. 따라서 학생들이 사회 변화를 인식하고 대처하기 위해 소프트웨어와 인공지능 역량을 함양할 수 있는 교육의 역할이 강조되고 있다. 이에 정부는 교육 분야 국정 과제 중 하나인 디지털 100만 명을 양성하기 위해 소프트웨어·인공지능 교육을 포함하였다[1]. 뿐만 아니라 2015 개정 교육과정과 2022 개정 교육과정의 인공지능 기초와 정보 과학에서 탐색의 원리를 다루고 있다[2][3][4][5]. 하지만 정보과학과 프로그래밍 과목에서 함수와 재귀 함수의 원리를 학습하기 위해 제공되는 문항은 n 번째 피보나치수열 및 $n!$ 계산 등이 있다. 이는 오래전부터 학습해 오던 내용이지만, 재귀 함수를 이용하여 동적 계획법 없이 n 번째 피보나치수열의 값을 구하기 위해서는 모든 상태를 방문해야 답을 구할 수 있으므로 다항 시간에 처리할 수 없는 문제점이 있다. 또한, 재귀 함수를 이용한 $n!$ 계산은 overflow가 발생하여 정확한 값을 찾을 수 없는 문제점이 있다[21]. 따라서 상태와 수행 작업의 정의한 후 탐색을 통해 값을 계산하는 과정에는 적절하지 않은 문항이다.

고등학교 정보과학 수업 시간에 함수와 재귀 함수를 배우고 재귀 함수를 이용하여 n 번째 피보나치수열을 찾는 문제를 학생들에게 제시하면 n 이 1~20까지는 1초 안에 답을 확인할 수 있지만, n 의 값이 100이면 수업이 끝날 때까지 값을 확인하지 못하는 문제점이 있다. 또한 재귀 함수의 모양이 일정하여 심화학습에 사용하기 부족하다. 또한 2022 개정 정보 교육과정 정보 및 정보과학의 평가 방법에 "평가 내용이나 방법에 따라 다양한 디지털 도구(프로그램 자동 평가시스템(online judge 등), 학습관리시스템(LMS) 등)를 활용할 수 있으며," 라고 명시되어 있다[3]. 따라서 정보과학 비선형 탐색 과정 학습을 위해 다양한 문항이 개발되고 제시되어야 한다.

"자료구조 수업에서 온라인 자동평가용 문제해결 프로그래밍 문제은행 개발 및 적용", "온라인 평가 도구를 활용한 프로그래밍 교육에서 학습 동기와 사고력 간 인과 관계", "프로그래밍 교육에서 온라인 저지 시스템의 활용이 학습동기와 사고력에 미치는 영향" 등 온라인 자동 평가 시스템을 교육에 적용하는 연구는 대다수가 온라인 평가 시스템 개발 및 학생

의 흥미와 관련된 연구가 있지만, 수업 장면에서 활용할 수 있는 문항 개발과 관련된 내용은 부족하다.[7][8][9].

이에 본 연구에서는 다양한 국내의 프로그래밍 경진대회의 동적 계획법과 탐색을 통해 해결할 수 있는 문제를 분석하여 온라인 자동 평가 시스템에 적용할 수 있는 문항을 개발하였다[10]. 개발한 문항은 문제 상황을 분석하여 상태 정의와 수행 작업의 정의, 초기 상태와 현재 상태, 목표 상태를 정리하고 탐색 과정을 통해 해결할 수 있는 문항으로 인공지능 기초의 탐색 단원과 정보 과학의 알고리즘 단원, 동아리 활동 및 방과후 활동의 문제 해결 프로그래밍 과정에서 사용될 수 있을 것이다[2][3]. 그리고, 수업에 참여한 학생들은 문항을 해결하는 과정에서 비선형 탐색 원리를 이해할 수 있도록 하였다. 또한 프로그래밍 수업 장면에서 온라인 자동 평가 시스템을 이용하기 위한 문항을 개발과정을 체계화하기 위한 기초 자료를 마련하고자 한다.

2. 이론적 배경

2.1 계산 문제

계산 문제(Computational problem)는 이론 컴퓨터 과학의 주요 연구 대상 중 하나로 알고리즘으로 해결할 수 있는 문제를 말한다. 계산 문제에는 결정 문제(Decision problem), 탐색 문제(Search problem), 계수 문제(Counting problem), 최적화 문제(Optimization problem) 등이 있다[11].

결정 문제는 답을 '예' 또는 '아니오'로 판별할 수 있는 문제를 말한다. 예를 들어, '3은 홀수인가?'와 같은 문제이다. 탐색 문제는 주어진 범위 내에서 원하는 조건을 만족하는 해를 찾는 문제를 말한다. 예를 들어, '10이하의 자연수 중 홀수는?'와 같은 문제이다. 계수 문제는 탐색을 통해 찾은 해의 개수를 파악하는 문제를 말한다. 예를 들어, '10 이하의 자연수 중 홀수는 모두 몇개인가?'와 같은 문제이다. 최적화 문제는 탐색을 통해 찾은 해 중 가장 좋은 해를 따져 보는 문제를 뜻하며, 최적해 탐색 문제라고도 한다. 예를 들어, '10 이하의 자연수 중 가장 큰 홀수는 무엇인가?'와 같은 문제이다. 인공지능 탐색에서는 탐색 문제와 관련이 있고 이 연구는 계수 문제와 최적화 문제와 관련이 있다.[2]

2.2 탐색을 통한 문제 해결

심리학에서 문제 해결은 정신적 과정, 컴퓨터 과학에서 문제 해결은 컴퓨터화되는 과정이다[12]. 컴퓨터 과학에서 문제를 해결하기 위해서는 문제를 초기 상태(initial state), 목표 상태(goal state), 수행 작업의 3 가지로 표현해야 한다.[12] 초기 상태는 아직 문제가 해결되지 않은 일련의 조건들을 말한다. 목표 상태는 수행 작업을 통해 문제의 목표인 상태를 말한다. 수행 작업은 초기 상태에서부터 목표 상태에 이르기까지 실행해야 하는 단계들을 말하며, 문제 해결은 초기 상태에서 일련의 조작들로 목표 상태에 이르는 것을 말한다. 컴퓨터 과학에서의 탐색알고리즘은 문제 상황을 입력해서 그 문제에 대한 해결 방법을 찾는 알고리즘이다. 또한 문제를 해결할 방법을 모아둔 집합을 탐색 공간이라 한다.

2.3 상태 정의

상태 정의는 문제 해결(Problem Solving) 해결 과정에서 매우 중요하다. 상태는 문제의 해결 정도를 나타내며 문제 해결을 위해 처음 당면한 상황을 초기 상태, 문제의 목표를 향해 가고 있는 상태를 현재 상태, 그리고 목표에 도달한 상태를 목표 상태라고 한다. 문제의 해결은 문제 공간의 초기 상태에서 목표 상태에 이르기까지의 수행 작업을 찾는 것이다. 문제의 상태 공간을 수행 작업을 이용하여 선형과 그래프 트리 등을 이용하여 구조화할 수 있고 탐색 과정을 통해 문제를 해결하는 과정을 구할 수 있다. 문제의 상태 정의와 수행 작업을 구조화하여 문제를 해결하는 것은 문제 해결 프로그래밍과 인공지능의 탐색 분야에서 사용할 수 있다. 실제적인 문제를 대부분은 상태 공간이 너무 크기 때문에 명시적인 그래프와 트리로 표현하기 힘들다. 이 경우에는 문제를 추상화하여 상태와 상태들 간의 관계를 설정하여야 한다. 또한 상태들을 어떤 방법으로 탐색할지를 결정해야 한다[12].

2.4 탐색 기반 설계

탐색 기반 설계는 2015 개정 정보 과학 교육과정 성취기준 중 "[12정과 03-06] 관계 기반 알고리즘을 이해하고 전체 문제와 부분 문제의 재귀적 관계를 정의한다." 와 2022 개정 정보과학 교육과정 성취기준 중 "[12정과03-01]문제를 분석하여 상태 공간으로

구조화하고, 상태 공간을 모두 탐색하여 문제를 해결하는 알고리즘을 설계한다." 에 명시되어 있다[2][3].

2015개정 교육과정 정보과학 교과서에서 탐색 기반 설계는 컴퓨터의 빠른 계산 능력을 기반으로 다양한 상태 공간 탐색을 통해 문제를 해결하는 것으로 설명하고 있다. 또한 이러한 알고리즘을 효율적으로 설계하는 방법을 탐색 기반 알고리즘 설계라고 하고 있다. 탐색을 통해 문제를 해결한다는 것은 선형 자료 구조인 배열 또는 리스트 등에서 원하는 자료를 찾거나, 선형으로 이루어진 상태 공간에서 목표 상태를 찾는 것을 예로 들 수 있다. 또한 비선형 자료 구조인 트리 또는 그래프 등에서 원하는 자료를 찾는 행위, 또는 주어진 문제를 해결하는 과정 중 문제 공간에서 해를 찾는 것도 포함된다. 그리고 탐색한 해의 개수를 세는 계수 문제, 탐색한 해 중 가장 품질이 좋은 해를 찾는 최적화 문제 등도 이에 포함된다.

2.4.1 선형 탐색

선형적으로 구조화된 상태 공간을 탐색하여 문제의 해결 방법을 찾는 것을 말한다. 일반적으로 선형 탐색 방법을 설계하기 위해서는 상태 공간을 선형적으로 구조화하여 표현하고, 이를 순차 탐색 또는 2진 탐색을 통해 답을 계산할 수 있다.

2.4.2 비선형 탐색

비선형적으로 구조화된 탐색 상태 공간을 탐색하여 문제의 해결 방법을 찾는 것을 말한다. 따라서 비선형 탐색 과정을 설계하기 위해서는 제일 먼저 문제 상태를 분석하여 상태 공간을 비선형적으로 구조화해야 한다. 비선형 자료 구조의 형태로 구조화된 상태 공간은 트리 또는 그래프의 형태로 나타낼 수 있다. 따라서 이 공간을 탐색하기 위해서는 깊이 우선 탐색 또는 너비 우선 탐색을 적용할 수 있다.

2.5 자동 평가 시스템

코드업, 코이스터디 등 중고등학생들이 문제해결 프로그래밍을 학습할 수 있도록 온라인 자동 평가 시스템을 현재 정보교사들이 학생들을 위해 운영하고 있다.

자동 평가 시스템은 학습자가 문제를 해결하기 위한 프로그래밍 소스 코드를 시스템에 제출하면 저장

하고 문제 상황을 해결할 수 있는 프로그램 인지를 확인하여 바로 학습자에게 적절한 피드백을 제공하는 시스템이다. 또한 학습자가 제출한 소스 코드를 시스템에 보관하여 학습자가 원할 때 인터넷이 연결된 공간이라면 시간과 장소의 제약 없이 제출된 프로그램 소스 코드를 확인할 수 있다. 또한 교사는 학생의 소스 코드를 확인하여 어떤 부분에서 어려움을 겪고 있는지 확인할 수 있다[14]. 현재 평가할 수 있는 프로그래밍 소스 코드로는 C/C++, JAVA, Ruby, Python 등이 있다[10][15][16][17].

2.5.1 코드업(codeup.net)

코드업은 2012년에 고등학교 프로그래밍 및 알고리즘 수업 장면에서 활용하기 위해 개발되어 현재까지 운영되고 있는 알고리즘 자동 평가를 지원하는 사이트로 프로그래밍 입문자들이 쉽게 접근하여 프로그래밍 및 알고리즘 문제를 접하고 해결할 수 있도록 지원하고 있다. 프로그래밍에 관심이 있는 초·중·고등학교 학생들이 해결할 수 있는 다양한 문제가 탑재되어 있어 초중고등학교 수업에 활용되고 있다[15].

2.5.2 코이스터디(koistudy.net)

영재학교 학생들과 고등학교 학생들을 위해 운영되고 있는 사이트로 코드업에 비해 문제 해결 프로그래밍 실력이 높은 학생들을 위한 문제가 다수 탑재된 사이트이다. 국내외 프로그래밍 경진대회 문제 및 알고리즘 문제들이 탑재되어 있다. 현재 영재학교와 일반고 정보 및 정보과학 수업에 활용되고 있으며, 정보올림피아드를 준비하는 학생들도 다양한 알고리즘 문제를 해결하고 있다[16].

2.5.3 저지온(judgeon.net)

경남에 있는 과학고에서 운영하는 온라인 자동 평가 시스템으로 과학고 정보과학 수업에 활용되고 있다. 난이도가 낮은 문제부터 난이도가 높은 문제까지 적절한 비율로 구성되어 있다[17].

2.5.4 기타 국내 알고리즘 자동 평가 시스템

이외에도 다양한 영재학교와 과학고등학교, 일반고등학교에서 선생님 개인 온라인 자동 평가 시스템을 개설하여 활용하고 있다. 또한 국내 최대 규모이며 다

양한 기능을 제공하고 있는 백준 온라인 저지 등도 있다[10].

2.5.5 AtCoder

일본의 프로그래밍 경진대회 플랫폼으로 보통 주말에 열리며, 다양한 수준의 참가자를 위한 여러 레벨의 문제가 제공된다. 알고리즘과 데이터 구조와 관련 문제가 출제된다[34].

2.5.6 Codeforces

러시아의 프로그래밍 경진대회 플랫폼으로 전 세계의 프로그래머들이 참여하고 있다. 다양한 유형의 대회가 정기적으로 개최되며, 알고리즘과 프로그래밍 기술에 초점을 맞추고 있다[35].

자동 평가 시스템에는 다양한 알고리즘 문항이 존재하지만 학습자의 수준에 맞춰 체계적인 학습을 할 수 있는 과정이 부족하다. 본 연구에서는 다양한 비선형 문항을 분석하여 체계적으로 학생들에게 제공하고자 한다.

2.6 ADDIE 모형

ADDIE 모형은 교수 체제 설계 과정의 일반적 형태를 나타낸 모형으로 보편적으로 활용되고 있다. ADDIE 모형은 5단계로 구성되어 있으며, 각 단계는 분석(Analysis), 설계(Design), 개발(Development), 실행(Implementation), 평가(Evaluation)로 구성되어 있다. 각 단계는 선형적으로 진행되기도 하지만, 때로는 필요에 따라 환류되기도 한다[18][19].

Table 1 ADDIE model

| 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 |

| Process | Content |
|------------|---|
| | · Application to information science classes and club activities |
| Evaluation | · Proceed with expert reviews · Study material validity statistics analysis · Revised based on expert review comments |

2.6.1 분석(Analysis)

학생의 수준, 학생의 특성, 학습 후 변화 등을 분석하는 단계이다. 교육에 사용할 수 있는 물적 자원과 학습 공간 등 물리적 환경도 분석한다.

2.6.2 설계(Design)

분석 단계에서 얻은 결과를 바탕으로 수업을 설계하는 단계이다. 수행 목표를 구체적으로 설정하고 이를 평가하기 위한 도구도 선정한다.

2.6.3 개발(Development)

설계를 바탕으로 교수 학습 자료를 개발하는 단계이다. 이 단계에서는 초안 개발, 형성 평가, 프로그램 수정, 최종 산출물 제작 등이 포함된다.

2.6.4 실행(Implementation)

설계되고 개발된 교수 학습 자료를 실제 수업 현장에 적용하고, 수정 보완하는 단계이다.

2.6.5 평가(Evaluation)

교수 학습 자료를 개발하고 실행한 모든 과정을 평가하는 단계이다. 교수 학습 자료와 프로그램의 적합성과 효율성 평가, 지속 가능성 검토, 발생한 문제의 수정 사항 등을 검토하는 것을 포함한다.

ADDIE 모델은 분명한 단계별 구조로 프로젝트를 체계적으로 관리할 수 있으며, 분석 단계에서 학습자의 필요와 배경을 깊이 파악함으로써, 학습자 중심의 맞춤형 교육 경험을 제공한다. 따라서 이 연구에서는 ADDIE 모델을 선정하였다.

3. 프로그램 개발

3.1 개발 의도

자동 평가 시스템에 있는 문항들은 정보 올림피아드 등 프로그래밍 경쟁 시스템에서 사용되는 문항들이 많아 현재 재귀 학습 및 탐색을 처음 학습하는 학생들을 위한 체계적인 문제가 부족하다[10][15][16][17][20][21]. 따라서 상태 공간 정의 및 수행 작업 분석을 통한 상태 공간 탐색을 학습할 수 있는 학습 프로그램이 필요하다. 개발한 문제는 동적 계획법을 적용하지 않아도 되는 문제로 상태 정의 및 수행 작업 정의, 초기 상태, 목표 상태, 현재 상태를 학습할 수 있는 문항으로 구성하였다[2][3].

3.2 개발 절차

고등학교 인공지능기초 탐색 또는 정보과학 알고리즘 단원에서 사용할 수 있도록 3가지 교육 내용 선정 기준을 개발하여 적용하였다. (1) 학습자의 수준 (2) 창의적 문제해결력 신장을 위한 내용 선정 (3) 교육 시간 등을 고려하여 선정하였다. 개발 절차는 Table 1에 제시한 것과 같이 교수 체제 설계 모델인 ADDIE를 기초로 하여 진행하였다[18][19].

3.3 자동 평가 시스템 적용 문제 개발 내용

본 연구에서는 첫째, 탐색 문제 해결을 위한 상태 공간과 수행 작업의 개념과 원리와 둘째, 수행 작업을 통한 상태 공간 탐색 방법, 셋째, 초기 상태, 현재 상태, 목표 상태의 표현 방법을 동시에 학습이 이루어질 수 있도록 학습 자료를 개발하였다.

상태 정의를 통한 탐색 학습 프로그램 개발은 2015 개정교육과정과 2022 개정 교육 과정을 바탕으로 교수학습 모델과 전략 및 교수학습 문항을 개발하였다.

비선형 탐색 문항을 개발하기 위해 Figure 1과 같이 한국 정보 올림피아드, USA Computing Olympiad, Japanese Olympiad in Informatics, Baltic Olympiad in Informatics 등 국내외 정보 올림피아드 문제와 google codejam, 대학교 프로그래밍 경시대회 문제 중 비선형 탐색 수업에 활용할 수 있도록 동적 계획법과 탐색 관련 있는 160여개의 문제를 제목, 내용, 계산 문제 유형, 동적 계획법 필요 여부, 재귀 함수 필요 여부,

수행 작업의 조건 필요 여부, 문제에서 수행 작업 제시 여부, 출제된 대회명, 출제년도, 난이도를 정리하였다. 또한 각 문제에 대한 해답 코드를 작성하였다.

육 및 정보과학 교육 경험이 있는 현장 교사 12명으로 구성하였다.

Table 2 Search problems with status definitions

| 순 | 제목 | 내용 | 문제 | DP 필요 | 재귀 | 주요 조건 |
|----|---------------------------|---|-------|-------|----|-------|
| 1 | Equal Sum Sets | 1-k의 자연수중 서로 다른 n개의 수를 뽑아 | 경우의 수 | 0 | 1 | 0 |
| 2 | Points | 수행 작업 + 그린 보상을 이용하여 계산 | 최대값 | 1 | 1 | 1 |
| 3 | LCS | 순서가 동일한 공통 부분 문자의 최대 길이 | 최대값 | 1 | 0 | 0 |
| 4 | 동전 | a원 동전으로 k원 만드는 방법의 수 (동전순) | 경우의 수 | 1 | 1 | 1 |
| 5 | Losy | n개를 골라 k를 만들때 비용 최소화 | 최소값 | 1 | 1 | 0 |
| 6 | Lecture Halls Reservation | 스케줄 + 최대 사용 시간 출력 | 최대값 | 1 | 1 | 1 |
| 7 | Spam | 문자열 비교 | 최대값 | 1 | 1 | 1 |
| 8 | Magazines Delivery | 3가지 수행작업 | 최소값 | 1 | 1 | 0 |
| 9 | Segments | 벽장로 변형 | 최소값 | 1 | 1 | 0 |
| 10 | Mouse Journey | 탐색 있는 격자판 이동 최솟 | 경우의 수 | 1 | 1 | 0 |
| 11 | 엘도라도 | 증가 수열의 개수 + 각 칸의 값이 주어짐 | 경우의 수 | 1 | 1 | 0 |
| 12 | Cow Digit Game | 게임 이론 + 증가 알고리즘 | 게임 이론 | 1 | 0 | 0 |
| 13 | Dairy Queen | 동전 문제 | 경우의 수 | 1 | 1 | 0 |
| 14 | Bulls and Cows | 가운데 중간 필요 | 경우의 수 | 1 | 1 | 0 |
| 15 | Supermarket | 필요한 물건들 고를 때, 최소값 물건들의 집는 | 최소값 | 1 | 1 | 1 |
| 16 | 공통 부분 문자열 | 최대 공통 연속 부분수열의 길이 | 최소값 | 1 | 0 | 1 |
| 17 | 1학년 | + , -를 조합하여 원하는 결과가 나오는 경우 | 경우의 수 | 1 | 1 | 0 |
| 18 | Exotic Foods | 인접 동전을 주울 수 없을 때, 최대값 | 최대값 | 1 | 1 | 0 |
| 19 | 미로에 갇힌 상근 | n번만에 돌아오는 경우의 수 + 육각형 | 경우의 수 | 1 | 0 | 1 |
| 20 | Robots on a grid | 격자판 최단 거리로 도착하는 경우의 수 + 경우의 수가 0 이면 탐색을 통해 경로가 있는지 판단 | 경우의 수 | 1 | 1 | 0 |
| 21 | 알약 | 2가지 수행작업중 하나를 선택, 상태는 이전 | 경우의 수 | 1 | 1 | 0 |
| 22 | 사탕가게 | 금액가진 금액보다 작은 사탕들을 구입하였 | 최대값 | 1 | 0 | 0 |
| 23 | 최대 정사각형 | 1로만 이루어진 가장 큰 정사각형 찾기 | 최대값 | 1 | 0 | 0 |
| 24 | 라그랑주의 네 제곱수 정리 | 동전문제 변형 + 최대 4개의 동전 | 경우의 수 | 1 | 1 | 0 |
| 25 | Common Subsequence | 공통 길이의 개수 + 이중 for | 최소값 | 1 | 0 | 0 |
| 26 | Piggy-Bank | 금액이 같을 때, 최소 금액 | 최소값 | 1 | 1 | 0 |
| 27 | Coins | 동전 문제 | 경우의 수 | 1 | 1 | 0 |
| 28 | 줄어들지 않아 | 길이 n인 증가수열의 개수 | 경우의 수 | 1 | 1 | 0 |
| 29 | 벽장문의 이동 | 벽장문 | 경우의 수 | 1 | 1 | 0 |
| 30 | 플레이우기 | 최장 증가 수열 | 최대값 | 1 | 0 | 0 |
| 31 | 동전 바꿔주기 | 동전 문제 | 경우의 수 | 1 | 1 | 0 |
| 32 | 숫자카드 | 단순 탐색 + 수행작업 가능? //암호코드 동 | 경우의 수 | 1 | 1 | 1 |
| 33 | 전깃줄 | 최장 증가 수열 | 최소값 | 1 | 0 | 0 |
| 34 | 지급률 | 단순 탐색 + 수순 | 경우의 수 | 1 | 1 | 0 |

Figure 1 Dynamic programming and search problem analysis

정리된 자료와 해답 코드를 이용하여 상태 정의, 수행 작업 정의, 초기 상태의 개수, 목표 상태를 판단하는 조건 등으로 분석하여 상태 정의를 위해 필요한 변수의 개수, 수행 작업의 개수, 목표 조건, 초기 상태, 목표 조건을 만족하는 상태 등을 고려하여 온라인 자동 평가 시스템에 탑재할 수 있는 4차시 분량의 문제를 Table 2와 같이 개발하였다.

| Step | Contents | time |
|--|---|------------|
| Understand state definitions and actions to be performed | · 1 variable required for state space · 2 actions · simple goal state | 1st lesson |
| | · 1 variable required for state space · 2 actions · simple goal state | |
| | · 2 variable required for state space · 2 actions · simple goal state | |
| Understand the actions | · 2 variable required for state space · 3 actions · simple goal state · Restrictions on action | 2nd lesson |
| | · 3 variable required for state space · 3 actions · simple goal state · Restrictions on action | |
| Understand the state space | · 1 variable required for state space · 3 actions · goal test and Search end condition | 3rd lesson |
| Understand the goal state | · 2 variable required for state space · 3 actions · goal test and Search end condition | |
| | · 4 variable required for state space · 3 actions · goal test and Search end condition | |
| Understand the initial state | · 1 variable required for state space · 2 actions · simple goal state · 2 initial states | 4th lesson |
| | · 1 variable required for state space · 2 actions · simple goal state · 2 initial states | |

4. 결과

4.1 전문가 검토 결과

이 연구에서 개발한 상태 공간 탐색을 이용한 재귀 학습 프로그램의 타당도를 검증하기 위해서 정보과학을 직접 학교 현장에서 학생을 지도하는 중등 교육 전문가의 검토 과정을 거쳤다. 중등 교육 전문가는 국내 온라인 자동 평가 시스템 사이트를 직접 운영하거나 온라인 자동 평가 시스템을 이용하여 프로그래밍 교

상태 정의를 통한 탐색 학습 프로그램에 대한 타당도 검토 문항은 중등 교육 전문가의 협의를 통해 구성하였고, 검토 문항은 Table 3와 같다. 중등 교육전문가 타당성 검토 문항은 적용 가능성(8문항), 유용성(2문항), 정보과 핵심역량 반영(3문항), 학생 요구(2문항), 학습 만족도(3문항) 영역으로 구성하여 총 18문항이었다. 각 문항은 4점 리커트 척도(매우 긍정, 긍정, 부정, 매우 부정)으로 구성하였다.

Table 3 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. State definition and search principle are well reflected. |
| 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 계산하기 위해 각 문항에 대하여 “매우 긍정”, “긍정”으로 응답한 경우 1으로, “부정”, “매우 부정”으로 응답한 경우 0으로 치환하여 평균하였다. CVI의 절단점은 .80으로 두었다. 결과는 **Table 4**와 같다.

Table 4 Expert Advisory Results

| Area | Item | Mean | SD | I-CVI |
|---------------|------|------|------|-------|
| Applicability | 1 | 3.92 | 0.29 | 1 |
| | 2 | 4.00 | 0.00 | 1 |
| | 3 | 3.92 | 0.29 | 1 |
| | 4 | 3.58 | 0.67 | .92 |
| | 5 | 3.92 | 0.29 | 1 |
| | 6 | 3.75 | 0.45 | 1 |
| | 7 | 3.83 | 0.39 | 1 |
| | 8 | 3.83 | 0.39 | 1 |

| Area | Item | Mean | SD | I-CVI |
|----------------------------------|------|------|------|-------|
| usability | 9 | 4.00 | 0.00 | 1 |
| | 10 | 4.00 | 0.00 | 1 |
| Competencies in computer Science | 11 | 3.92 | 0.29 | 1 |
| | 12 | 4.00 | 0.00 | 1 |
| | 13 | 3.75 | 0.45 | 1 |
| student needs | 14 | 3.83 | 0.39 | 1 |
| | 15 | 3.50 | 0.52 | 1 |
| Learning Satisfaction | 16 | 3.58 | 0.51 | 1 |
| | 17 | 3.50 | 0.52 | 1 |
| | 18 | 3.50 | 0.52 | 1 |

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

전문가 A: 동아리 학생들에게 적용해보니 일반고등학교 학생들 수준에 적합한 내용으로 구성되었다. (후략).

전문가 B: 탐색 문제에서 탐색을 위한 상태와 수행 작업을 정의하는 데 많이 도움이 되는 문항이다. 이 문제를 통해 탐색에 대해 이해할 수 있었다. (후략).

전문가 C: 초기 상태, 현재 상태, 목표 상태를 이해할 수 있는 좋은 문항이다. 이다음 과정이 만들어졌으면 좋겠다. (후략).

반면, 전문가 D는 다음과 같은 의견도 제시하였다.

전문가 D: 탐색 문항을 푸는 방법은 4가지가 있다. 이중 첫 번째 방법에 문항 개발이 맞춰진 것 같다. 다른 방법도 학습할 수 있는 문항이 개발되었으면 좋겠다. 그리고 본문이 딱딱한 것 같다. 조금 더 직관적이었으면 좋겠다.

이 밖에도 전문가 중 일부로부터 영재학교 학생이나 과학고 학생들을 위해 난이도 있는 문항이 개발되어야 한다는 개선 의견도 수집할 수 있었다.

4.2 개발 결과

한국 정보 올림피아드, USA Computing Olympiad, Japanese Olympiad in Informatics, Baltic Olympiad in Informatics 등 국내외 정보 올림피아드 문항과 google

codejam, 대학교 프로그래밍 경시대회 문제 중 동적 계획법과 탐색 문제를 분석하여 상태 정의를 통한 탐색 학습 프로그램에 적용할 문제를 개발 완료하였다.

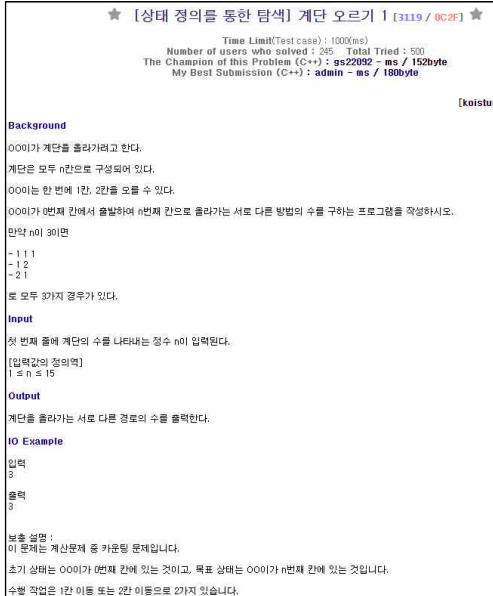


Figure 2 Search problems with status definitions 1

Figure 2는 개발된 문항 중 1차시에 사용하였다. 문항은 상태 정의를 위해 필요한 변수 1개, 수행 작업이 2개이다.

Table 5. Python code to solve Figure 1

```
def f(c) :
    # goal test
    if c == n :
        global ans
        ans += 1
        return
    if c > n :
        return
    f(c+1) # operation
    f(c+2) # operation
ans = 0
n = int(input())
f(0) # initial state
print(ans)
```

Table 5는 Figure 1에 제시된 문제를 해결하는 파이썬 프로그램으로 초기 상태, 수행 작업, 현재 상태가

목표 상태인지 판단하는 조건 등으로 구성하였다.

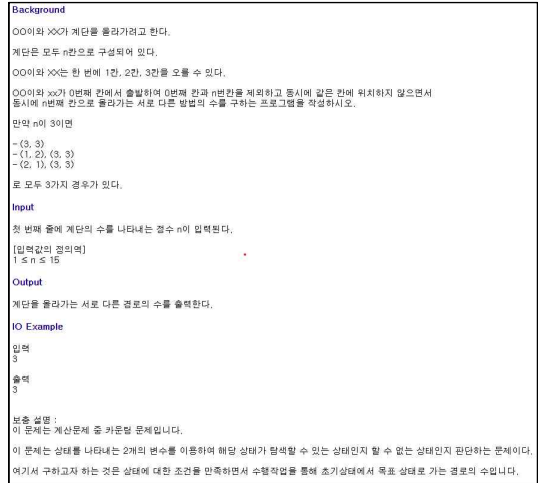


Figure 3 Search problems with status definitions 8

Table 6. C language code to solve Figure 2

```
#include<stdio.h>
int ans, N;
void f(int n, int m) {
    // goal test
    if (n == 0 && m == 0){
        ans++;
        return;
    }
    if (n <= 0 || m <= 0) return;
    if (n == m && n != N) return;
    // 9 operations
    for (int i = 1; i < 4; i++)
        for (int j = 1; j < 4; j++)
            f(n - i, m - j);
}
int main() {
    int n;
    scanf("%d", &n);
    N = n;
    f(n, n); // initial state
    printf("%d", ans);
    return 0;
}
```

Table 6은 Figure 3에 제시된 문제를 해결하는 C언어 프로그램으로 초기 상태를 표현하는 변수가 2개 필요하고, 9개의 수행 작업, 현재 상태가 목표 상태인지 판단하는 조건 등으로 구성하였다.

Figure 4는 상태와 수행 작업의 관계를 표현한 것으로 현재 상태와 수행 작업을 설명하기 위한 학습 자료

로 현재 상태에 오기 위한 수행 작업에 따라 현재 상태에서 할 수 있는 수행 작업을 나타낸 것이며, 현재 상태에서 할 수 있는 수행 작업을 통해 다음 상태가 될 수 있는 상태를 표현한 자료이다.

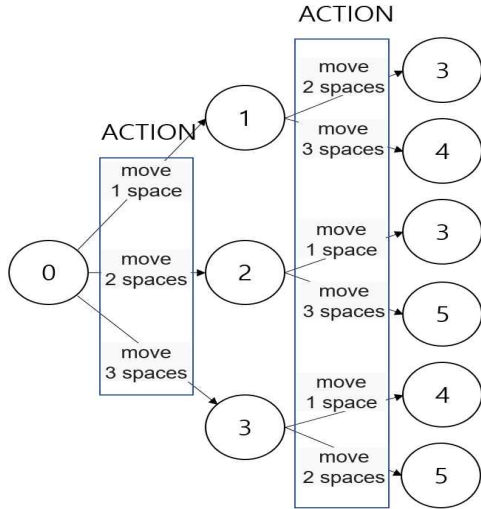


Figure 4 Relationship between state and action

Figure 5는 동아리 시간을 활용하여 개발한 프로그램을 적용한 사례이다. lpts는 문제를 해결하는 코드를 작성하여 점수를 획득한 것이며 (-1)은 틀린 횟수를 나타낸다. 학생들이 제출한 소스 코드를 실시간으로 평가하여 피드백을 제공하여 문제 상황을 해결할 수 있도록 한다.

| A(I) | B(I) | C(I) | D(I) | E(I) | F(I) | G(I) | H(I) | I(I) | J(I) | K(I) | L(I) | M(I) |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts |
| [00:09:59] | [00:08:06] | [00:22:59] | [00:24:51] | [00:27:43] | [00:29:42] | [00:39:31] | [00:42:44] | [00:46:52] | [00:48:33] | [00:54:03] | [00:56:10] | [00:58:53] |
| lpts | lpts | lpts | lpts (-1) | lpts | lpts | lpts | lpts | lpts (-1) | lpts | lpts | lpts | lpts (-1) |
| [00:06:47] | [00:11:27] | [00:03:02] | [00:18:07] | [00:03:03] | [00:35:01] | [00:36:07] | [00:31:00] | [01:11:31] | [01:13:45] | [01:15:11] | [01:17:35] | [01:24:38] |
| lpts | lpts | lpts | lpts | lpts (-1) | lpts | lpts (-1) | lpts | lpts (-1) | lpts | lpts | lpts | lpts (-1) |
| [00:03:43] | [00:04:42] | [00:08:31] | [00:12:59] | [00:19:07] | [00:22:33] | [00:39:45] | [01:34:04] | [01:54:53] | [02:01:25] | [02:05:45] | [02:10:47] | [02:22:42] |
| lpts | lpts | lpts | lpts | lpts (-1) | lpts | lpts (-1) | lpts | lpts | lpts | lpts | lpts | lpts (-1) |
| [00:05:24] | [00:05:21] | [00:13:34] | [00:19:14] | [00:05:05] | [00:30:35] | [01:25:47] | [00:31:30] | [01:51:24] | [01:56:44] | [01:58:07] | [02:05:11] | [02:12:23] |
| lpts | lpts | lpts | lpts (-1) | lpts (-1) | lpts (-1) | lpts (-1) | lpts (-1) | lpts (-1) | lpts | lpts | lpts | lpts |
| [00:03:08] | [00:06:24] | [02:31:28] | [00:28:34] | [02:26:34] | [01:16:20] | [02:21:43] | [03:06:07] | [03:48:18] | [03:40:47] | [01:19:36] | [01:44:44] | [02:45:51] |
| lpts | lpts | lpts (-1) | lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts | lpts (-1) | lpts |
| [00:04:18] | [00:06:12] | [01:38:39] | [00:21:42] | [00:26:42] | [00:35:10] | [00:42:21] | [00:56:18] | [06:23:46] | [06:26:23] | [00:26:07] | [01:31:44] | [01:51:17] |
| lpts | lpts | lpts | lpts | lpts | lpts (-1) | lpts (-1) | lpts (-1) | lpts | lpts | lpts | lpts | lpts |
| [04:51:54] | [04:53:18] | [04:56:48] | [00:03:28] | [00:22:43] | [00:29:10] | [00:21:03] | [10:54:22] | [14:18:42] | [07:43:12] | [07:46:51] | [07:50:20] | [08:01:42] |
| lpts | lpts | lpts | lpts (-1) | lpts | lpts | lpts | lpts (-1) | lpts | lpts | lpts | lpts | lpts |
| [00:04:12] | [00:25:51] | [00:35:32] | [17:01:51] | [17:17:42] | [17:25:24] | [17:28:10] | [04:26:30] | [01:28:45] | [01:34:00] | [01:33:46] | [04:19:54] | [04:21:13] |
| lpts | lpts | lpts | lpts (-1) | lpts (-1) | lpts (-1) | lpts (-1) | lpts (-1) | lpts | lpts | lpts | lpts | lpts |
| [00:18:36] | [00:20:35] | [23:44:14] | [06:51:32] | [08:23:10] | [10:08:34] | [02:03:19] | [09:50:18] | [02:47:43] | [02:28:03] | [03:24:15] | [03:33:33] | [04:40:11] |
| lpts | lpts | lpts | lpts (-1) | lpts (-1) | lpts (-1) | lpts (-1) | lpts (-1) | lpts | lpts | lpts | lpts (-1) | lpts |
| [00:06:53] | [00:31:38] | [18:28:39] | [18:56:32] | [00:01:49] | [00:08:03] | [00:39:24] | [17:46:24] | [17:23:10] | [04:28:32] | [06:31:20] | [06:00:04] | [17:32:11] |

Figure 5 Apply to programming class

상태 정의를 통한 탐색 학습 프로그램을 학생에게 적용해본 결과 충분히 수업 장면에서 활용할 수 있는 난이도임을 확인하였다. 또한 일반고 인공지능기초 탐색 단원에 문항을 적용하여 보았다. 총 4차시의 수업이 필요하였고, 수업 장면에서 교사의 적절한 안내도 필요하였다.

5. 결론

본 연구에서는 중고등학교 정보에 제시된 상태 정의와 정보과학에 제시된 전 탐색 및 관계 기반 설계 교육을 위해 2015, 2022 개정 정보교육의 목표 및 성취 수준을 분석한 후, 분석 내용을 기반으로 고등학교 학생을 위한 탐색 교육내용 개발과 활동지 개발 기준을 제안하였다.

본 논문에서 제안한 상태 정의를 통한 탐색 학습 프로그램의 타당성을 검증하기 위하여 코드업과 코이스터디 개발 및 운영하는 정보교육 전문가를 포함하여 온라인 자동 평가 시스템을 이용한 수업 경험이 있는 중등 정보교육 전문가 그룹을 활용한 통계 분석을 시행하였다. 개발된 프로그램의 전문가 검토 결과, I-CVI 값은 모두 .80 이상 (.91 ~ 1.00)으로 절단 점 이상이다. 따라서 모든 항목에 대한 내용 타당도를 갖춘 것으로 확인되었다.

또한, 이 연구에서 개발된 문항을 온라인 자동 평가 시스템에 탑재하여 정보과학 및 인공지능 탐색 수업에 활용하여 효과적인 수업에서 활용할 수 있다. 온라인 자동 평가 시스템을 통해 학생들이 제출한 소스 코드의 정확성을 실시간으로 확인할 수 있으며, 이는 즉각적인 피드백을 제공하여 학습자의 흥미를 지속적으로 유지시키고 컴퓨팅 사고력을 향상시킬 수 있을 것으로 기대한다. 본 연구의 상태 정의와 탐색 학습 방식을 통합한 이 온라인 자동 채점 시스템의 문항은 정보과학 및 인공지능 교육에 큰 이점을 제공할 것으로 기대한다. 또한 교사는 실시간으로 학생의 성취도를 확인함으로써 평가 부담을 경감할 수 있다[7][8][9].

본 프로그램의 내용을 토대로 문항 개발을 발전시킨다면 다른 알고리즘 교육 활동에도 적용할 수 있을 것이다.

다만, 본 연구에서 개발한 프로그램을 수업 현장에 실제 투입하여 효과성 검증은 이루어지지 않았다. 추후 연구에서는 정보과학 및 알고리즘을 학습하는 학습자를 대상으로 사전-사후 실험 설계를 하여 학습

흥미, 컴퓨팅 사고력, 학습 이해 등에 대한 효과성을 검증해야 한다. 또한, 개발한 프로그램은 동적 계획법 등의 학습에 적용하기에는 무리가 있다. 동적 계획법 등을 위한 학습 프로그램도 계속 개발되어야 할 것이다.

참고문헌

- [1] Ministry of Education (2022). *We train 1 million talented people who will become the protagonists of the digital age*.
- [2] Ministry of Education. (2015). *The 2015 Revised General Guidelines of National Curriculum*.
- [3] Ministry of Education. (2022). *The 2022 Revised General Guidelines of National Curriculum*.
- [4] Ministry of Education. (2015). *Software Education Operational Guidelines*.
- [5] Ministry of Education. (2020). *Comprehensive Plan for Information Education*.
- [6] 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.
- [7] 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. DOI : 10.32431/kace.2018.21.4.002.
- [8] 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. DOI : 10.14352/jkaie.2020.24.4.379.
- [9] 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. DOI : 10.32431/kace.2021.24.5.001.
- [10] Baekjoon Online Judge. (2022, October 25). Retrieved from <https://www.acmicpc.net/>.
- [11] Even, Shimon; Selman, Alan L.; Yacobi, Yacov (1984). The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2), 159-173. DOI: 10.1016/S0019-9958(84)80056-X.
- [12] Russell, Stuart J. & Norvig, Peter (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.
- [13] 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.
- [14] 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.
- [15] codeup. (2022.October 25). Retrieved from: <https://codeup.kr/>
- [16] koistudy. (2022.October 25). Retrieved from: <https://koistudy.net/>
- [17] judgeon. (2022. October 25). Retrieved from: <http://www.judgeon.net/>
- [18] ADDIE Model. (2022, October 25). Retrieved from https://en.wikipedia.org/wiki/ADDIE_Model.
- [19] ADDIE Model. (2023. November 15). Retrieved from: <https://educationaltechnology.net/the-addie-model-instructional-design/>
- [20] 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.
- [21] 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. DOI : 10.32431/kace.2017.20.1.002.
- [22] State space search. (2022. October 25). Retrieved from: https://en.wikipedia.org/wiki/State_space_search
- [23] Dynamic positioning. (2022. October 25). Retrieved from: https://en.wikipedia.org/wiki/Dynamic_positioning
- [24] Song, J.H.(2011). *An Automated Assessment based Programming Education System for Self-Directed Learning*. Doctoral dissertation, Soongsil University.
- [25] 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. DOI : 10.1145/1517632.1517643
- [26] Andy Kurnia.& Andrew Lim.& Brenda Cheang. (2001). Online Judge. *Computers & Education* 36, 299 - 315 DOI : 10.1016/S0360-1315(01)00018-5.
- [27] 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. DOI : 10.32431/kace.2014.17.4.005.
- [28] Lim, H.S. (2007). ACM-ICPC problem formulation and

grading process.. *Communications of the Korean Institute of Information Scientists and Engineer*. 23(7), 52-55.

- [29] Kim, M.J.(2007). Design and Implementation of an Automatic Grading System for Programming Assignments. *Journal of Internet Computing and Services*, 8(6), 75-85.
- [30] Jung, E.M.(2009). *Design and Implementation of Automatic Marking System for a Subjectivity Problem of the Program*. Master's thesis, Andong University.
- [31] 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. DOI : 10.32431/kace.2020.23.4.003.
- [32] 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*. 23(6), 961-972. DOI : 10.14352/jkaie.2021.25.6.961.
- [33] 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. DOI: <http://dx.doi.org/10.1002/nur.20199>
- [34] AtCoder. (2023. November 15). Retrieved from: <https://atcoder.jp/>
- [35] codeforces. (2023. November 15). Retrieved from: <https://codeforces.com/>



정 상 수

2009년 경상국립대학교
컴퓨터교육과(이학사)
2019년 한국교원대학교 교육대학원
컴퓨터교육전공 (교육학석사)

2020년 ~ 현재 한국교원대학교 컴퓨터교육전공 박사과정
관심분야: 프로그래밍 교육, 인공지능 교육, 교과교육
E-Mail: whiteedu84@gmail.com



고학능

2015년 광주교육대학교 과학교육과 (학사)
2018년 광주교육대학교 교육대학원
컴퓨터교육전공(교육학 석사)
2023년 한국교원대학교
컴퓨터교육과 박사수료

2018년 ~ 현재 가톨릭대학교 교사
관심분야: 프로그래밍 교육, 온라인 저지, 인공지능 교육, 데이터 과학
E-Mail : snddl3@outlook.kr



이 영 준

1988년 고려대학교 전산과학과(이학사)
1994년 미국 미네소타대학교 전산학과
(Ph.D)

2003년 ~ 현재 한국교원대학교 컴퓨터교육학과 교수
관심분야: 지능형시스템, 학습과학, 정보교육, 인공지능교육
E-Mail: yjlee@knue.ac.kr