

# 정보 교육에서 프로그래밍(알고리즘) 자동평가 시스템의 활용 가능성에 대한 고찰

장원영<sup>†</sup> · 김성식<sup>††</sup>

## 요 약

프로그래밍(알고리즘) 자동 평가 시스템은 주어진 문제에 대해 사용자가 제출한 소스코드의 정확성과 알고리즘의 시간/공간 효율성 등에 대한 즉각적인 평가 결과와 교정적인 피드백을 제공한다. 또한, 이러한 실시간 평가 결과를 통해 제공되는 문제별 채점 현황(제출 횟수, 통과 횟수), 랭킹 등의 경쟁적인 요소는 사용자에게 프로그래밍 학습에 대한 동기와 흥미를 제공하는 장점이 있다. 본 연구에서는 프로그래밍(알고리즘) 자동 평가 시스템의 이론적 배경과 선행 연구에 대한 고찰, 국·내외 자동 평가 시스템의 동향에 대해 알아보고, 고등학교와 대학의 학부 과정에서 활용할 수 있는 방안을 제시하였다. 즉, 2015 개정 교육과정의 고등학교 과학 계열 전문 교과인 ‘정보 과학’ 과목에서 자동 평가 시스템의 활용을 제시하고 있으며, 이에 따라 C언어의 문법에 관한 기초적인 내용에서부터 주어진 문제의 알고리즘 설계와 프로그래밍 단계까지 폭넓게 적용할 수 있다. 또한, 대학의 자료구조와 알고리즘 강좌에서 동일 문제에 대한 각 알고리즘의 실제 소요 시간을 직접 비교해 봄으로써 알고리즘의 성능 차이를 확인할 수 있다.

**주제어** : 프로그래밍, 알고리즘, 자동 평가, 자동 평가 시스템, 정보 교육

## A review on trends of programming(algorithm) automated assessment system and it's application

Won-Young Chang<sup>†</sup> · Seong-Sik Kim<sup>††</sup>

## ABSTRACT

The programming(algorithm) automated assessment system is to evaluate automatically the accuracy and time/space efficiency of user's solution to the problem which is provided. This system gives the immediate feedback of the solution, real-time ranking. So, in the course of data structure and algorithm, we can apply the knowledge which we have learned to the problem solving. Especially, in the basic course of learning the syntax of the programming language, the novice student can learn in easy and fun by solving the simple problem. The university students can understand in the easy way the meaning of asymptotic analysis of algorithm in data structure & algorithm course.

**Keywords** : Programming, Algorithm, Automated assessment system, Informatics Education

---

<sup>†</sup> 종신회원: 교육부 교육연구사  
<sup>††</sup> 종신회원: 한국교원대학교 교수(교신저자)  
논문접수: 2016년 11월 23일, 심사완료: 2017년 1월 20일, 게재확정: 2017년 1월 23일

## 1. 서론

초·중등학교에서 소프트웨어(SW)교육을 강화하는 2015 개정 교육과정은 2015년 9월 23일에 고시되어 2018년부터 적용을 앞두고 있는 지금, 각 학교급별 수준에 적합한 SW교육의 목표와 그에 따른 교수·학습 방법에 대한 논의가 활발히 이루어지고 있다. 초등학교는 2019년부터, 중학교는 2018년부터 SW교육이 의무화되고 놀이와 교육용 도구를 활용한 체험 위주의 교수·학습 방법이 적용될 예정이다. 그러나, 고등학교에서는 텍스트 기반 언어를 사용하므로, 자칫 주어진 소스 코드를 그대로 코딩해 보는 단순한 실습에 그칠 우려가 크며, 이에 따라 고등학교에 적합한 교수·학습 방법에 대한 논의가 필요한 시점이다. 특히, 고등학교 과학 계열 전문 교과인 ‘정보 과학’ 과목에서는 보다 높은 수준의 알고리즘과 프로그래밍에 대한 내용요소가 포함되므로 초·중학교와는 다른 교수·학습 방법이 요구된다[1][2][3]. 최근에는 많은 대학들이 전공자 뿐만 아니라 비전공자를 대상으로 한 프로그래밍(알고리즘) 교육을 강화하고 있으므로, 대학에서의 교수·학습 방법에 대한 충분한 연구와 논의도 필요하다.

수학 교육의 목표가 수학자의 양성이 아닌 것처럼, SW교육의 목표 또한 프로그래머의 양성이 아니다. 대학교의 고학년 전공자를 제외하고는 초·중·고등학생 이상을 대상으로 한 SW교육의 목표는 단연 ‘컴퓨팅 사고력(Computational Thinking)의 함양’일 것이다. 2015 개정 교육과정에서는 정보과 교과 역량을 ‘컴퓨팅 사고력’, ‘협력적 문제해결력’, ‘정보문화소양’으로 설정하고, 이를 실현하기 위해 정보과 교육과정에서는 총 4개의 ‘정보문화’, ‘자료와 정보’, ‘문제해결과 프로그래밍’, ‘컴퓨팅 시스템’ 영역과 함께 각 영역별로 그에 적합한 내용 요소와 성취 기준을 제시하고 있다[2]. 2009 개정 교육과정과 비교해서 2015 개정 정보과 교육과정에서는 교과 역량을 ‘컴퓨팅 사고력’ 외에 지식 정보 사회의 구성원으로서 갖추어야 할 정보 문화에 대한 기본 소양과 협력적 태도를 강조하고, ‘문제해결과 프로그래밍’ 영역을 내용과 분량 면에서 강화하였으며, 이진수, 운영체제, 네트워크 등의 어려운 내용은 축소·삭제하였다. 또한, 학교

급별로 학습자의 수준에 맞는 교수·학습 방법을 제시하고 있다. 초등학교에서는 놀이를 통한 알고리즘의 이해와 교육용 도구를 활용한 프로그래밍 체험에, 중학교에서는 교육용 프로그래밍 도구를 활용한 실생활의 문제 해결에 중점을 두고 있다. 고등학교에서는 자신의 진로와 연계하여 심화된 내용을 학습할 수 있도록 C언어나 Python 언어 등의 텍스트 기반 프로그래밍 언어의 활용을 제시하고 있다[2]. 특히, 과학계열 전문 교과인 ‘정보 과학’에서는 실생활과 타학문 분야의 다양하고 복합적인 문제를 해결하기 위해 알고리즘 설계와 분석을 강조하고 있으며, 자기주도적 학습과 수준별 학습이 용이하도록 프로그래밍 자동 평가 시스템의 활용을 제안하고 있다[3]. 이에, 본 논문에서는 이러한 프로그래밍(알고리즘) 자동 평가 시스템의 선행 연구를 고찰하고, 고등학교와 대학에서의 정보 교육 활용 방안에 대해서 논의해 보고자 한다.

## 2. 이론적 배경

### 2.1 프로그래밍(알고리즘) 자동평가 시스템

프로그래밍(알고리즘) 자동평가 시스템(이하, 자동평가 시스템)은 주어진 문제에 대해서 사용자가 그 문제를 해결하는 소스코드를 제출하면, 컴파일 시간 오류와 실행 시간 오류를 확인하고, 소스코드가 정확하고 알고리즘이 시간/공간 효율적으로 구현되었는지의 여부를 자동으로 평가하는 시스템이다. 시스템은 일반적으로 컴파일 과정과 채점 과정의 2단계 자동 평가 과정을 거치게 된다. 컴파일 과정에서는 사용자가 프로그래밍 언어를 이용하여 주어진 문제에 대한 소스코드를 작성하여 제출하면 자동평가 시스템은 이 소스코드를 서버에 파일로 저장하고 이를 자동으로 컴파일하여 실행 파일을 만든다[4]. 컴파일 과정을 통과한 실행 파일은 미리 준비된 입력, 출력 데이터의 세트에 따라 몇 차례 실행을 반복하고, 매 실행시마다 미리 준비된 데이터가 입력되어 출력되는 값과 준비된 정답데이터의 값을 비교한다. 만약 두 데이터의 값이 같고, 시간 제한과 메모리 제한을 초과하지 않았다면 제출한 소스 코드는 올바른 답

으로 받아들여지며, 최소한 미리 준비된 채점데이터 세트에 대해서는 해당 소스 코드의 알고리즘이 정확하고 효율적으로 구현되었음을 확인하게 된다[5][6].

## 2.2 자동평가 시스템의 구성 요소

프로그래밍(알고리즘) 자동평가 시스템은 일반적으로 <표 1>의 4가지 구성요소로 이루어져 있다[7].

### 2.2.1 문제와 채점데이터 세트

프로그래밍(알고리즘) 자동평가 시스템을 운영하기 위해서는 먼저 구조적이고 형식적이며 입력과 출력이 명확한 문제를 제작해야 하고, 그에 상응하는 채점데이터 세트를 만들어야 한다. 채점데이터 세트는 알고리즘의 정확성과 시간/공간 효율성을 평가할 수 있도록 [그림 1]과 같이 15개 내외의 입력데이터와 정답데이터의 쌍을 종적 및 횡적으로 구성하여 개발한다.

### 2.2.2 문제

프로그래밍(알고리즘) 자동평가 시스템에서 사용하는 문제는 기본적으로 [그림 2]의 구성요소로 이루어져 있으며, 각 구성요소들은 알고리즘의 기본 조건인 명확성, 유효성, 유한성, 효율성을 고려하여 설계한다.

<표 1> 시스템 구성요소

| 구성요소        | 관련 설명  |
|-------------|--|
| ① 문제        | 사용자가 해결해야 할 문제   |
| ② 채점데이터 세트  | 해당 문제에 대한 알고리즘의 정확성과 시간/공간 효율성을 측정하기 위해 입력데이터와 정답데이터의 쌍으로 구성된 일련의 채점데이터 세트                 |
| ③ 자동평가 프로그램 | 사용자가 제출한 소스코드를 컴파일한 후 생성된 실행파일에 채점데이터를 입력하여 출력값과 정답데이터를 비교하고, 실행파일의 소요시간과 메모리 양을 측정하는 프로그램 |
| ④ 사용자       | 문제, 문제별 채점 현황, 자동 채점 결과, 랭킹 산출 등의 서비스를 제공하는 사용자 환경   |

### 2.2.3 채점 데이터 세트

채점데이터 세트는 [그림 1]과 같이 15개 내외의 입력데이터와 정답데이터의 쌍으로 구성된 데이터 세트로서 입력데이터와 정답데이터 사이의 종적인 구성을 통해 알고리즘의 정확성을 평가하며, 입력 데이터의 범위가 점점 증가하는 횡적인 구성을 통해 알고리즘의 시간/공간 효율성을 평가한다. 그리고 이러한 평가요소는 [그림 2]의 문제 구성요소에서 ②핵심요구사항, ③제한 시간/메모리, ④입력데이터의 범위와 밀접하게 관련되어 있다[8].



[그림 1] 채점데이터 세트의 구성

| 구성 요소       | 알고리즘의 기본조건 |     | 명확성 |
|-------------|------------|-----|-----|
| ① 배경 설명     |            |     |     |
| ② 핵심요구사항    | 정확성 (유효성)  |     |     |
| ③ 제한 시간/메모리 | 시간/공간      | 유한성 |     |
| ④ 입력데이터의 범위 | 효율성        |     |     |
| ⑤ 입력과 출력의 예 | 입력 출력      |     |     |

[그림 2] 문제 구성요소

### 2.2.4 자동평가 프로그램

자동평가 프로그램은 학습자가 제출한 소스코드를 컴파일하고, 생성한 실행파일을 실행시킨 후 채점데이터 세트의 입력데이터를 순서대로 투입하면서 실행파일이 출력한 값과 정답데이터의 일치 여부를 확인한다. 이때, 입력데이터를 투입하고 실행파일이 결과값을 출력할 때까지 해당 실행파일이 문제에서 제시한 제한시간과 메모리 용량을 초과할 경우, 실행파일의 프로세스를 강제 종료시킨 후 제한시간 초과, 또는 메모리 용량 초과 메시지를 표시한다(알고리즘의 효율성). 제한시간과 제한 메모리의 용량을 초과하지 않으면서 결과값이 출력되는 경우에는 이 값과 정답데이터를 비교해서 일치 여부를 확인하고(알고리즘의 정확성), 그 결과를 사용자에게 알려준다.

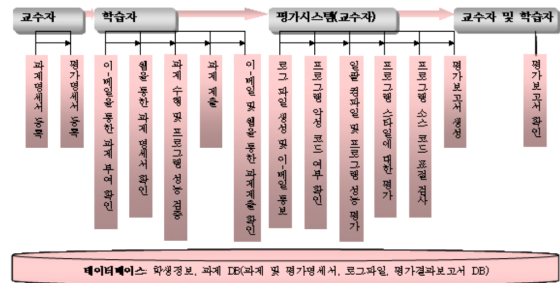
### 2.2.5 사용자 서비스 환경

사용자 서비스 환경은 학습자에게 문제를 제시하고 자동평가 결과와 더불어 지금까지 학습자가 제출한 소스코드를 누적하여 보여준다. 또한, 자동평가 결과를 토대로 문제별 채점 현황과 사용자 간의 실시간 랭킹을 산출한다. 사용자 서비스 환경은 대부분 웹으로 구현하는 것이 일반적이다.

## 3. 선행 연구의 고찰

김미혜(2007)는 수작업으로 이루어지는 프로그래밍 과제 평가가 많은 시간이 소요되는 문제점을 극복하고자, 학습자가 제출한 소스 코드의 정확성과 표절을 일괄적으로 평가할 수 있는 프로그래밍 과제 평가 시스템 PASG(Programming Assignment Grading System) [그림 3]을 개발하였

다. 이 시스템은 알고리즘의 효율성을 평가할 수 없으며, 입력 데이터에 대해서 올바른 출력 데이터를 산출하는지의 여부를 미리 준비한 정답 데이터를 이용하여 비교하는 방식이다. 또한, 2006년부터 미국 스탠포드 대학에서 운영 중인 MOS S(Measure of Software Similarity) 시스템을 통해 제출한 소스 코드의 표절을 검사한다. 시스템의 유용성을 확인하기 위해 2007년도 1학기 D대학교 컴퓨터교육과에 개설된 3개의 프로그래밍 관련 교과목(컴퓨터프로그래밍 C, Java 프로그래밍, 데이터 구조)에 적용하였으며, 기존의 수작업 평가 방법에 비해 과제 평가의 소요 시간이 크게 감소한 것을 확인하였다. 또한, 3개의 강좌에 참여한 학생들 중 평균 76%의 학생이 PASG 시스템을 선호하였다[9].



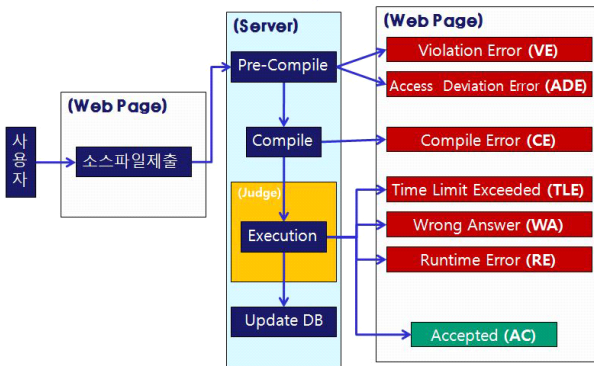
[그림 3] PASG 시스템

정은미(2008)는 프로그래밍 언어 교육이 주관식 평가의 의존도가 높으나, 객관식에 비해 평가의 신뢰도가 낮고, 많은 채점 시간이 소요된다는 문제점을 해결하기 위해 '프로그래밍 언어에 대한 주관식 자동 채점 시스템'을 개발하고, 컴퓨터 공학 전공 2, 3학년 학생 10명을 대상으로 24개 문항에 자동 평가 시스템과 지필 평가를 실시하였으며, 91.7%의 유사도를 보였다. 그러나, 이 시스템은 다음의 문항과 같은 코드 일부를 채우는 주관식 답안을 단순한 텍스트 비교 방식으로 평가하는 방식이므로 소스 코드에 내재되어 있는 '알고리즘'을 평가할 수는 없다[10].

(문항) 대문자를 입력받아 몇 번째 알파벳인지 출력하는 프로그램을 완성하시오.

```
int main()
{
    ( ① ) ch;
    scanf("②", &ch);
    printf("%d", ③);
    return 0;
}
```

정종광(2010)은 프로그래밍 평가의 객관성과 신뢰성을 확보하고, 교사의 평가 업무를 경감하고자 [그림 4]의 Online Judge 기반 프로그래밍 평가 시스템을 개발하였다[5].

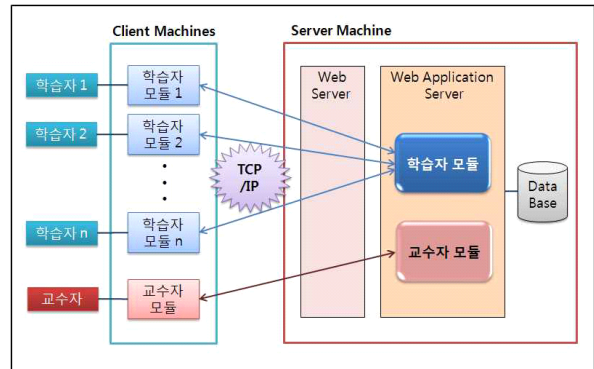


[그림 4] Online Judge 방식의 실행 과정

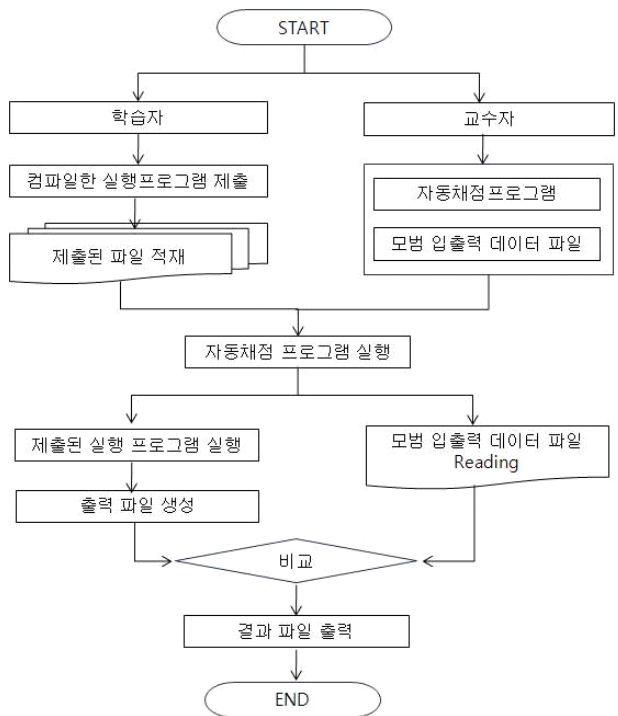
송지희(2011)는 [그림 5]와 [그림 6]의 자기주도 학습을 위한 자동채점기반의 프로그래밍 교육 시스템을 개발하고, 프로그래밍 교육에 있어 학습자의 학업성취도와 수업만족도 향상의 효과성 검증을 위하여 서울 소재 S대학의 학부생으로 구성된 실험 집단(15명)과 통제 집단(15명)을 대상으로 적용, 분석하였다. 그 결과 실험집단의 학업성취도가 통제집단 보다 유의미하게 우수한 것으로 나타났으며, 실험 집단 대상의 수업만족도 조사에서 종래의 수업 방법 보다 자동채점기반의 프로그래밍 교육 시스템을 적용한 수업에서 더 높은 만족도를 보였다[4].

전현석·정종광·김성식(2014)은 기초 C언어를 학습하는 과정에서 과제 집착력과 몰입을 통해 학습할 수 있도록, 각 학습 단계별로 필요한 문법적인 요소를 확인하고, 이러한 문법 요소를 단계별로 적용하는 문제를 개발하였으며, 정종광

(2010)의 자동 평가 시스템에 탑재하여 실제 수업에 적용하였다[11].



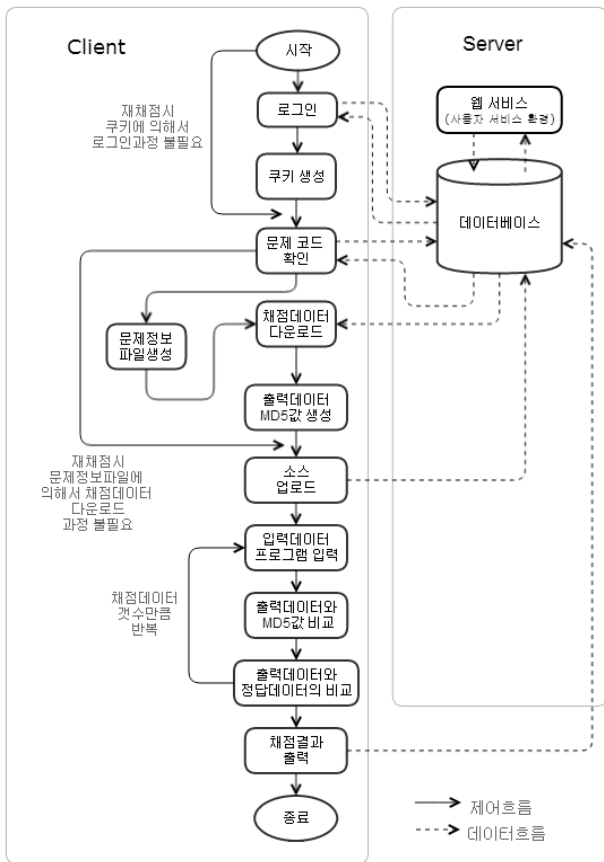
[그림 5] 시스템 구성도



[그림 6] 자동 평가 알고리즘

장원영·김성식(2014)은 지금까지의 자동 평가 시스템이 학습자에게 채점 데이터를 제공하지 않아 교정적 피드백의 원인을 찾기 어렵다는 문제를 인식하고, [그림 7]과 같은 클라이언트-서버기반의 알고리즘 자동 평가 시스템을 개발하였다. 이 시스템의 특징은 채점 데이터가 사용자의 컴퓨터(클라이언트)에 저장되고, 그 곳에서 채점이 이루어진다는 점이다. 자동 평가 시스템의 프로그

래밍 학습 효과를 확인하기 위해 자동 평가 시스템을 초등학생 7명, 중학생 9명, 고등학생 2명으로 구성된 실험 집단에 적용하였으며, 초등학생 9명, 중학생 10명, 고등학생 2명으로 구성된 통제 집단에는 전통적인 교수·학습 방법을 적용하였다. 사후 검사에서 실험 집단의 평균점수가 통제집단에 비해 유의미하게 높은 것으로 나타났으며, 이를 통해 알고리즘 자동평가 시스템을 적용한 교수·학습방법이 전통적인 교수·학습방법에 비해 프로그래밍 학습에 더 효과적임을 확인하였다[7].



[그림 7] 자동평가 프로그램의 실행 과정

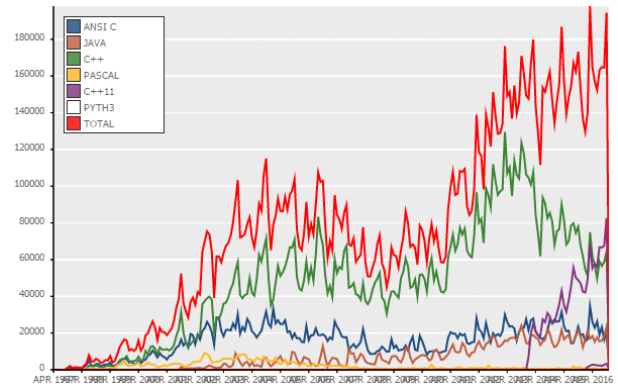
#### 4. 국내외 자동 평가 시스템 동향

프로그래밍(알고리즘) 자동 평가 시스템은 국내외에서 사용자층이 많아지고 있다. 각 시스템의 자동 평가 건수가 지속적으로 증가하고 있고, 국내의 경우 오픈 소스 환경인 hustoj를 활용하여 현직 교사들이 직접 시스템을 구축하여 교실 현

장에서 활용하고 있다[12]. 또한, 최근에는 전국 단위의 교사 연수 프로그램에서도 포함되어 있어 그 적용례가 계속 증가하고 있다.

#### 4.1 국외 현황

국외에서는 비교적 오래 전부터 프로그래밍(알고리즘) 자동 평가 시스템이 개발되어 지속적으로 활용되어 왔다. 예를 들어, UVa Online Judge 시스템은 ACM-ICPC와 같은 세계적인 프로그래밍 콘테스트 대회를 준비할 수 있도록 개발된 사이트로서 [그림 8]은 해당 사이트의 언어별 사용량 추이를 나타낸 것이다.



[그림 8] UVA Online Judge 사용량 추이

이 시스템은 스페인의 University of Valladolid에서 운영되고 있으며, 1997년부터 2016년까지 18,181,566의 자동 채점 횟수가 기록되고 있으며, 매년 지속적으로 증가하고 있는 추세다[13].

#### 4.1.1 Codeforces

러시아 Saratov State University가 운영하고 있는 사이트로서, 모든 사용자는 1주일내 한번씩 2시간 동안 열리는 ‘Codeforces Round’에 참가할 수 있으며, 이미 종료한 ‘Codeforces Round’의 문제를 풀어볼 수 있다. 또한, 다른 사용자가 제출한 소스와 해당 문제의 채점 데이터 일부를 확인할 수 있는 기능을 제공하고 있어 최근 전 세계의 사용자가 꾸준히 증가하여 2015년 말 기준 30만명의 사용자가 가입한 사이트다. 2010년 2월 19일에 첫 번째 ‘Codeforces Round’가 열려 175명

이 참가하였으며, 2016년 1월까지 300개의 Round가 열렸고 평균 5,000명의 회원이 참가하고 있다 [14].

제 대회 문제 등을 제공하고 있다. 시스템을 자체 제작한 점이 특징이다[17].

<표 2> 국외 현황

| 사이트명               | 주소                  | 문제수   | 채점수    | 언어             | 비고    |
|--------------------|---------------------|-------|--------|----------------|-------|
| Codeforces         | codeforces.com      | 3,600 | 2,200만 | C, C++, Java 등 | 자체 개발 |
| 북경대학교 Online Judge | poj.org             | 3,050 | 1,600만 | C, C++, Java 등 |       |
| UVA Online Judge   | uva.onlinejudge.org | 4,300 | 1,800만 | C, C++, Java 등 |       |

#### 4.1.2 북경대학교 Online Judge

중국의 북경대학교에서 운영하는 사이트로서, 2003년에 Ying Fuchen, Xu Pengcheng, Xie Di가 개발한 이후 현재까지 전 세계적으로 두터운 사용자층을 확보하고 있으며, 다양한 온라인 콘텐츠를 자체 운영하고 있다[15].

#### 4.1.3 UVA Online Judge

스페인의 Valladolid 대학의 온라인 채점 사이트로서 10만 여명이 넘는 사용자가 가입해 있으며 1995년에 제작되어 1997년에 공개되었고, ANSI C(C89), C++, Pascal, Java, Python 등의 언어를 사용할 수 있다[16].

### 4.2 국내 현황

#### 4.2.1 KOI Study

경기과학고등학교 정종광 교사가 2009년에 제작하여 운영하고 있으며, 프로그래밍 언어의 기초적인 내용을 학습할 수 있는 문제(C언어 기초 100제)와 더불어, 정보올림피아드 문제, 다양한 국

#### 4.2.2 Coding is fun

장원영이 개발하고(2014, 장원영·김성식), 현재 충북 제천 동명초등학교 문동국 교사와 함께 운영하고 있는 사이트다. C언어의 기초적인 내용을 학습할 수 있는 문제, 자료구조/알고리즘을 학습할 수 있는 문제, 각종 해외 경시대회 문제와 함께 한국 정보올림피아드 기출 문제를 일목요연하게 제시하고 있다. 특히, 다른 사이트와는 다르게 CodeBlocks와 같은 C/C++ 통합개발환경과 연동하여 클라이언트-서버 방식으로 자동 평가가 이루어지는 것이 특징이다[18].

#### 4.2.3 Code Up

경상남도 배준호 교사가 오픈 소스 환경인 Hustoj 시스템과 wordpress 인터페이스를 활용하여 2012년부터 구축, 운영하고 있으며, 교사 권한을 부여 받은 사용자는 자신의 수업을 만들고 그곳에 참여 대상과 문제를 선별할 수 있는 기능을 제공한다. KOI Study 사이트와 함께 프로그래밍 언어의 기초적인 내용을 학습할 수 있는 문제(C언어 기초 100제)를 제공하고 있다[19].

<표 3> 국내 현황

| 사이트명                   | 주소            | 문제수   | 채점수  | 언어             | 비고         |
|------------------------|---------------|-------|------|----------------|------------|
| KOI Study              | koistudy.com  | 1,700 | 146만 | C, C++, Java 등 | 자체 개발      |
| Coding is fun          | codingfun.net | 430   | 10만  | C, C++         | 자체 개발      |
| Code Up                | codeup.kr     | 1,000 | 150만 | C, C++, Java 등 | houstoj 기반 |
| Judge On               | judgeon.net   | 800   | 12만  | C, C++, Java 등 | houstoj 기반 |
| Dovelet                | dovelet.com   | 1,300 | 240만 | C, C++, Java 등 | 자체 개발      |
| Backejoon Online Judge | acmicpc.net   | 1,200 | 400만 | C, C++, Java 등 | 자체 개발      |

#### 4.2.4 Judge On

경상남도 김봉석 교사가 오픈 소스 환경인 Hustoj 시스템을 활용하여 구축한 시스템으로서 콘테스트 기능을 이용하여 교내 수행평가에 적용하는 것이 특징이다[20].

#### 4.2.5 Dovelet

이규호에 의해 2009년에 제작된 사이트로서 문항을 두 그룹으로 나누어 수준 높은 문항들은 유료로 제공하고 있다[21].

#### 4.2.6 Baekjoon Online Judge

최백준이 2013년에 제작한 사이트로 국내외 정보 올림피아드 대회 문제 및 대학에서 개최되는 프로그래밍 대회 문제 등을 풍부하게 제공하고 있고, 사용자 인터페이스가 우수하다[22].

### 5. 정보 교육 활용 방안

#### 5.1 자동 평가 시스템의 장점

전통적인 방식의 프로그래밍(알고리즘) 교육과 비교하여 자동평가 시스템을 활용한 교육은 학습자와 교수자 입장에서 다음과 같은 장점을 가진다[23].

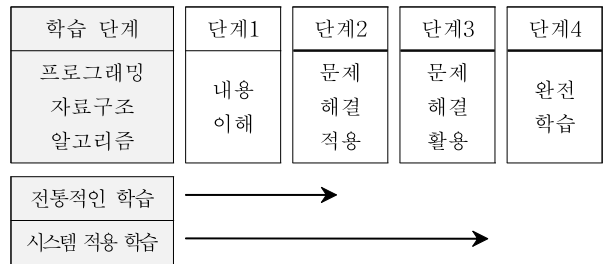
##### 5.1.1 학습 목표의 본질적 접근

프로그래밍, 자료구조, 알고리즘을 학습하는 궁극적인 목적은 ‘주어진 문제의 해결’에 있다. 프로그래밍 언어의 문법을 학습하고, 스택, 큐, 그래프, 검색, 정렬, 분할정복법, 동적계획법 등의 자료구조와 알고리즘을 배우는 이유는 ‘주어진 문제를 효율적으로 해결’하기 위함이다. 그러나 실제 교육 현장에서는 학습 내용의 이해에만 집중하고 그것을 문제 해결에까지 적용하지 못하는 경우가 많다. 학습 목표를 ‘해결해야 할 문제’에 두고, 이를 위한 효율적인 자료 구조와 알고리즘을 적용하는 방식을 통해, 학습자는 학습 목표의 본질에 보다 더 빨리 접근할 수가 있다.

즉, 자동 평가 시스템을 활용하면, [그림 9]와

같이 프로그래밍 언어, 자료구조, 알고리즘에 대한 학습 내용의 이해에 그치지 않고, 학습한 내용을 문제 해결에 적용하고 이와 유사한 문제를 자기 주도적으로 찾아 관련 문제를 해결하는 상황이 쉽게 전개된다.

[그림 9] 학습 목표의 본질



##### 5.1.2 해결해야 할 문제의 상황 맥락화

[그림 2]의 문제 구성 요소의 ‘① 배경 설명’을 통해 <표 4>와 같이 상황 맥락적인 문제를 제시할 수 있다. 이러한 문제 상황은 사용자에게 해결해야 할 맥락과 함께 학습 동기를 부여하는 효과를 제공한다.

<표 4> 상황 맥락적 문제 예시

| 구분        | 문제 예시   |
|-----------|---|
| 일반적인 문제   | 두 정수가 입력으로 주어질 때, 이 두 정수의 곱셈 결과를 출력하는 프로그램을 작성하시오.  |
| 상황 맥락적 문제 | 사각형 모양의 색종이 면적을 구하려면, 30Cm 자를 이용하여 색종이의 가로와 세로의 길이를 구하고, 이 두 길이를 곱해야 한다. 색종이의 가로의 길이와 세로의 길이가 Cm 단위로 주어질 때, 색종이의 면적을 구하는 프로그램을 작성하시오. |

##### 5.1.3 경쟁과 즉각적 피드백의 학습 동기 강화

주어진 문제에 대해 사용자가 자신의 소스코드를 제출하면, 해당 소스의 컴파일 오류와 함께 알고리즘의 정확성과 효율성에 대한 교정적 피드백이 즉각적으로 제시되며, 이러한 채점 결과에 따라 실시간으로 랭킹이 산출되므로 사용자 간에 경쟁적인 요소가 추가되어 학습 동기의 강화를 가져온다[24]. <표 5>는 대부분의 자동 평가 시스템에서 제공하는 교정적 피드백 메시지이다.



<표 5> 교정적 피드백 메시지

| 단계 | 메시지    | 의미   |
|----|--------|--|
| 1  | 컴파일 오류 | 사용자가 제출한 소스 코드 내에 컴파일 오류가 있음                         |
| 2  | 런타임 오류 | 실행 중에 오류가 발생함  |
| 3  | 시간 초과  | 주어진 입력값에 대해 실행과정에서 제한 시간을 초과함                        |
| 4  | 메모리 초과 | 주어진 입력값에 대해 실행과정에서 제한 메모리의 양을 초과함                    |
| 5  | 표현 오류  | 정답 데이터와 출력 데이터 간에 공백, 줄바꿈 등이 불일치함                    |
| 6  | 잘못된 답  | 정답데이터와 출력 데이터 간에 값이 불일치함                             |
| 7  | 올바른 답  | 제한 시간과 제한 메모리를 초과하지 않으면서 모든 채점 데이터 세트에 대해 올바른 답을 출력함 |

자동 평가 시스템은 일반적으로 <표 5>의 단계별로 자동 평가가 진행되고, 그 과정에서 오류가 발견되면 해당 메시지를 출력하고, 그 이후의 평가 과정을 중단한다[13].

#### 5.1.4 평가의 정확성과 수월성

학생들이 제출한 소스코드에 대한 정확하고 객관적인 평가는 교수자에게 많은 부담을 안겨준다. 자동 평가 시스템은 학습자가 제출한 소스코드의 개수와 상관없이 정확하고 객관적인 평가가 가능하다[25].

### 5.2 자동 평가 시스템의 활용 방안

프로그래밍(알고리즘) 자동 평가 시스템은 고등학교와 대학의 기초 프로그래밍 과정에서 적용할 수 있으며, 과학 계열 고등학교의 ‘정보 과학’과 대학의 자료 구조 및 알고리즘 강좌에서 활용할 수 있다.

#### 5.2.1 기초 프로그래밍 과정(고등학교, 대학)

C언어 등의 프로그래밍 언어에 대한 기초적인 학습 후에 해당 차시에서 학습한 문법적 요소를 적용해야 하는 상황 맥락적인 문제를 제시할 수 있다. 자동 평가 시스템을 적용할 수 있는 문제는 [그림 2]와 같은 5가지 구성 요소를 갖추어야 하나, 기초 프로그래밍 과정에 적용하는 문제의 경우 제한

시간/메모리(③), 입력 데이터의 범위(④)는 크게 중요하지 않다. 왜냐하면, 일정 수준의 시간 복잡도가 요구되는 문제가 아닌 경우에는 제한 시간을 초과할 만큼 복잡한 알고리즘이 필요하지 않기 때문이다. 오히려, 학습자에게 도전감을 제공하고 학습 동기를 자극하기 위해서는 상황 맥락적이고 흥미로운 배경 설명(①)과 함께 해결해야 할 문제를 명확하게 제시(②, ⑤)하는 것이 중요하다.

<표 6> 문제 구성요소의 중요도

| 구성 요소       | 기초 프로그래밍 | 자료구조 알고리즘 |
|-------------|----------|-----------|
| ① 배경 설명     | high     | high      |
| ② 핵심요구사항    | high     | high      |
| ③ 제한 시간/메모리 | low      | high      |
| ④ 입력데이터의 범위 | low      | high      |
| ⑤ 입력과 출력의 예 | mid      | mid       |

#### [문제 1] 기초 프로그래밍 문제의 예시

[① 배경 설명]  
사각형 모양의 색종이 면적을 구하려면, 30Cm 자를 이용하여 색종이의 가로와 세로의 길이를 구하고, 이 두 길이를 곱해야 한다.

[② 핵심요구사항]  
색종이의 가로의 길이와 세로의 길이가 Cm 단위로 주어질 때, 색종이의 면적을 구하는 프로그램을 작성하시오.

[③ 제한 시간/메모리]  
프로그램의 실행시간은 1초를 넘을 수 없으며, 메모리공간은 300KB를 초과할 수 없다.

[④ 입력 데이터의 범위]  
입력 형식  
입력 데이터의 첫째 줄에 색종이의 가로와 세로의 길이를 나타내는 정수 2개가 공백을 사이에 두고 주어진다. 단, 주어진 정수는 1~30Cm 이하이다.  
  
출력 형식  
첫째 줄에 색종이의 면적을 Cm 단위로 출력하며, 별도의 'Cm'는 출력하지 않는다.

[⑤ 입출력 형식과 예시]  
입력과 출력의 예 1  
입력 (input.txt)  
3 10  
  
출력 (output.txt)  
30

|                 |      |
|-----------------|------|
| 입력과 출력의 예 2     |      |
| 입력 (input.txt)  | 7 12 |
| 출력 (output.txt) | 84   |

예를 들어 [문제 1]은 프로그래밍 언어의 연산자를 학습한 후 적용할 수 있는 문제다. 학습자는 이러한 상황 맥락적인 문제를 통해 곱셈 연산자와 그에 따른 문법에 대한 이해에 그치지 않고, 초기에 학습 내용을 문제 해결에 적용·활용할 수 있게 된다. 현재, KOI Study, Code Up, Coding is fun 등의 현직 교사가 운영 중인 사이트에서는 이러한 문제들을 꾸준히 개발하여 학교 현장에서 활용하고 있으며, 학생들에게 좋은 반응을 얻고 있다. <표 7>는 Coding is fun에서 운영 중인 기초 프로그래밍 문제 일부의 채점 현황이다[18].

<표 7> 기초 프로그래밍 문제 채점 현황 예시

| 코드    | 문제                 | '올바른 답' 판정 횟수 | 제출 횟수 | 비율     |
|-------|--------------------|---------------|-------|--------|
| 00a56 | Hello, World! 출력하기 | 138           | 311   | 44.37% |
| 00a37 | 정수값 그대로 출력하기       | 817           | 1,488 | 54.91% |
| 00a41 | 대문자로 변환하기          | 608           | 1,214 | 50.08% |
| 00a52 | 두 정수의 합 구하기        | 70            | 94    | 74.47% |
| 00a01 | 사각형의 넓이 구하기        | 1,417         | 2,466 | 57.46% |
| 00a03 | 홀수와 짝수 판별하기        | 1,280         | 2,468 | 51.86% |
| 00a09 | 초 단위의 시간 구분        | 388           | 820   | 47.32% |
| 00a51 | 두 정수의 대소 비교하기      | 72            | 128   | 56.28% |
| 00a48 | 1부터 N까지 짝수합 구하기    | 106           | 191   | 55.5%  |
| 00a12 | 부분 문자열 출력하기        | 399           | 788   | 50.63% |
| 00a30 | 숙제 안 낸 사람은 누구?     | 258           | 921   | 28.01% |

### 5.2.2 고등학교 '정보 과학' 과목(고등학교)

2015 개정 교육과정에서 '정보과학'은 자동 평가 시스템을 활용할 수 있는 대표적인 과목이다. '정보과학'은 <표 8>의 내용 체계로 구성되어 있으며 (내용체계에서 일반화된 지식, 기능은 생략), 자동 평가 시스템을 '프로그래밍', '자료 처리', '알고리즘'

영역에서 활용할 수 있다. 특히, 교육과정에서는 해당 영역의 교수·학습 및 평가 시에 자동 평가 시스템의 활용을 다음과 같이 제시하고 있다[3].

[ 교수·학습 방법 및 유의사항 ]

- 자기주도적 학습과 수준별 학습이 용이하도록 프로그램 자동 평가 시스템을 적극 활용한다.

[ 평가 방법 및 유의사항 ]

- 프로그래밍 과제 평가 시 프로그램 자동 평가 시스템 등을 활용하여 평가의 효율을 높일 수 있다. 단, 프로그래밍 과정에서 발생한 오류에 대한 교정적 피드백을 제공하여 학습자가 프로그램을 수정·보완할 수 있도록 한다.
- 프로그램 자동 평가 시스템을 적용할 때는 학습자가 개발한 프로그램의 수행 시간, 소스 코드의 길이 등을 종합적으로 평가한다.

<표 8> '정보과학' 과목 내용체계표의 일부

| 영역      | 핵심 개념     | 내용 요소                         |
|---------|-----------|-------------------------------|
| 프로그래밍   | 연산 수행     | · 변수와 상수<br>· 연산자             |
|         | 자료 저장     | · 자료형<br>· 다차원 배열             |
|         | 흐름 제어     | · 순차, 선택, 반복 구조<br>· 중첩 제어 구조 |
|         | 모듈화       | · 함수<br>· 변수의 영역              |
| 자료 처리   | 자료구조      | · 선형 자료구조<br>· 비선형 자료구조       |
|         | 정렬과 탐색    | · 자료의 정렬<br>· 자료의 탐색          |
| 알고리즘    | 문제와 알고리즘  | · 문제<br>· 알고리즘 복잡도            |
|         | 탐색기반 알고리즘 | · 전체 탐색<br>· 탐색 공간의 배제        |
|         | 관계기반 알고리즘 | · 관계 정의<br>· 동적 계획법           |
| 컴퓨팅 시스템 | 시뮬레이션     | · 시뮬레이션 설계<br>· 시뮬레이션 구현      |
|         | 피지컬 컴퓨팅   | · 피지컬 컴퓨팅 구성<br>· 피지컬 컴퓨팅 구현  |

5.2.3 ‘자료구조 / 알고리즘’ 강좌 (대학)

프로그래밍(알고리즘) 자동평가 시스템은 대학의 자료구조와 알고리즘 강좌에서도 활용할 수 있다. 스택, 큐, 그래프 등의 기본적인 자료구조를 적용하여 해결할 수 있는 문제를 제시하거나, 검색, 정렬, 그래프 탐색(DFS, BFS), 탐욕적인 방법, 분할 정복법, 동적 계획법 등의 범용적인 알고리즘을 적용하는 문제를 제시하고, 학생들이 제출한 소스 코드의 알고리즘을 자동 평가하는 것이다. <표 9>는 Coding is fun 사이트에서 제공하는 일부 문제들로서, 문제 해결의 주요 열쇠가 되는 자료구조와 알고리즘을 기준으로 이를 통해 해결할 수 있는 문제(군)을 난이도 순서로 제시함으로써 해당 자료구조와 알고리즘을 활용한 문제 해결의 응용력을 향상시킬 수 있다[18].

<표 9> 자료구조/알고리즘 적용 문제(군) (난이도순)

| 자료구조/알고리즘 | 코드    | 문제              | 비고 |
|-----------|-------|-----------------|----|
| 스택        | 00b02 | 기차 교차로          |    |
|           | 15re3 | 쇠막대기            |    |
| 큐         | 00b03 | 요셉(Josephus) 문제 |    |
| 검색        | 00b14 | 어디에 있나?         |    |
|           | 12ne2 | 예산              |    |
| 정렬        | 00b11 | 정렬하기            |    |
|           | 04re1 | 배설공주와 난장이       |    |
| DFS       | 00b46 | 미로 탐색           |    |
| BFS       | 00b07 | 바이러스 감염         |    |
|           | 00b32 | 순간 이동           |    |
| 탐욕적인 방법   | 00b43 | 크레파스            |    |
|           | 00b34 | 회의실             |    |
| 분할정복법     | 00b23 | 쿼드트리            |    |
|           | 00b24 | 히스토그램           |    |
|           | 00b08 | 스카이라인 구하기       |    |
| 동적계획법     | 00b15 | 피보나치수열 n번째 항    |    |
|           | 00b35 | 타일 붙이기          |    |
|           | 00b36 | 천국의 계단          |    |
|           | 00b39 | 숫자삼각형           |    |
|           | 00b40 | 마블페인트 칠하기       |    |
|           | 00b18 | 음료수             |    |
|           | 00b20 | 배낭 문제           |    |
|           | 00b17 | 최소동전교환          |    |
|           | 00b00 | 연속부분최대합         |    |
|           | 00b01 | 최대공통부분문자열       |    |

또한, 자동평가 시스템은 알고리즘의 성능 차이를 확인하는 직관적인 방법을 제공한다. 대부분의

교재에서 알고리즘의 성능 평가는 ‘시간 복잡도’를 활용한다. 그러나, 시간 복잡도는 알고리즘을 처음 배우는 대학의 학부생이 이해하기에 매우 어려운 개념 중 하나다. 이러한 경우에, 각 알고리즘의 실제 수행 시간을 보여줌으로써 알고리즘의 성능과 효율성에 대한 깊이 있는 이해의 실마리를 제공할 수 있다. 예를 들어, ‘배열의 연속된 구간 최대합’을 구하는 [문제 2]는 <표 10>과 같이, ‘무식한 힘의 방법’, ‘분할정복법’, ‘동적계획법’ 등의 알고리즘으로 해결할 수 있는데[26], 이때, 자동 평가 시스템을 통해 각각의 알고리즘을 구현한 소스 코드를 채점해 보고, 실제 수행 시간을 비교해 보는 방식으로 알고리즘의 성능과 효율성을 직접 확인할 수 있다.

[문제 2] 배열의 연속된 구간 최대합

[① 배경 설명]

일차원 배열에 n개의 정수가 저장되어 있을 때, 배열에서 임의의 연속된 구간을 잡아 그 합이 최대가 되도록 하려고 한다. 예를 들어 다음과 같은 일차원 배열이 주어질 때,

|    |     |    |    |     |    |    |     |     |    |
|----|-----|----|----|-----|----|----|-----|-----|----|
| 31 | -41 | 59 | 26 | -53 | 58 | 97 | -93 | -23 | 84 |
|----|-----|----|----|-----|----|----|-----|-----|----|

다음과 같이 연속된 구간이 최대한 187이 된다.

|    |     |    |    |     |    |    |     |     |    |
|----|-----|----|----|-----|----|----|-----|-----|----|
| 31 | -41 | 59 | 26 | -53 | 58 | 97 | -93 | -23 | 84 |
|----|-----|----|----|-----|----|----|-----|-----|----|

[② 핵심요구사항]

n과 n개의 수를 입력받아 합이 최대가 되도록 하는 연속구간을 찾아 그 합을 출력하는 효율적인 프로그램을 작성하십시오.

[③ 제한 시간/메모리]

실행 시간은 0.5초를 넘을 수 없으며, 메모리공간은 300KB를 초과할 수 없다.

[④ 입력 데이터의 범위]

입력 형식  
 입력의 첫째 줄에 1,000,000 이하의 자연수 n이 주어진다. 둘째 줄에는 일차원 배열에 담겨 있는 n개의 정수가 차례로 주어진다. 일차원 배열에 담긴 정수는 -100이상, 100이하이다.

출력 형식  
 출력의 첫째 줄부터 연속된 구간의 합 중 최댓값을 출력한다.

|                 |   |
|-----------------|---|
| [⑤ 입출력 형식과 예시]  |   |
| 입력과 출력의 예       |   |
| 입력 (input.txt)  | <pre>10 31 -41 59 26 -53 58 97 -93 -23 84</pre> |
| 출력 (output.txt) | <pre>187</pre>                                  |

<표 10> 각 알고리즘별 시간복잡도와 의사코드

| 알고리즘         | 시간복잡도        | max_sum 함수 (의사코드)   |
|--------------|--------------|---|
| 무식한<br>힘의 방법 | $O(n^3)$     | <pre>int max_sum(){ for(i=0;i&lt;n;i++){ for(j=i;j&lt;n;j++){ sum=0; for(k=i;k&lt;=j;k++){ sum+=A[k]; } result=MAX(result, sum); } } return result; }</pre>   |
|              | $O(n^2)$     | <pre>int max_sum(){ for(i=0;i&lt;n;i++){ sum=0; for(j=i;j&lt;n;j++){ sum+=A[k]; result=MAX(result, sum); } } return result; }</pre>   |
| 분할<br>정복법    | $O(n \lg n)$ | <pre>int max_sum(int left, int right){ if(left&gt;right) return 0; if(left==right) return A[left]; mid=(left+right)/2 lmax, sum 초기화; for(i=mid;i&gt;=left;i--){ sum+=A[i]; lmax=MAX(sum,lmax); } rmax, sum 초기화; for(i=mid+1;i&lt;=right;i++){ sum+=A[i]; rmax=MAX(sum,rmax); } return MAX(lmax+rmax, max_sum(left, mid), max_sum(mid+1,right)); }</pre> |
| 동적계획법        | $O(n)$       | <pre>int max_sum(){ mSum[0]=A[0]; for(i=1;i&lt;n;i++){ mSum[i]=MAX(mSum[i-1]+A[i],A[i]); result=MAX(result,mSum[i]); } return result; }</pre>   |

<표 11>은 Intel(R) Core2 Quad CPU 2.4GHz에서 [문제 2]에 대한 각 알고리즘의 소요 시간을 측정 한 것이다. 실행하는 컴퓨터의 성능에 따라, 그리고 실행 시점의 CPU 상태에 따라 매번 소요시간이 변동하지만, O표기법의 본질적인 의미를 이해하기 전에 동일한 입력 크기에 대하여 시간복잡도가 서로 다른 알고리즘 간에 실제적인 소요시간의 차이가 발생함을 확인할 수 있다.

<표 11> 알고리즘의 소요시간 비교 (단위: 초)

| n         | $O(n^3)$ | $O(n^2)$ | $O(n \lg n)$ | $O(n)$ |
|-----------|----------|----------|--------------|--------|
| 10        | 0.211    | 0.042    | 0.023        | 0.025  |
| 100       | 0.053    | 0.021    | 0.021        | 0.021  |
| 1,000     | 0.110    | 0.022    | 0.023        | 0.022  |
| 10,000    | 82.002   | 0.069    | 0.032        | 0.027  |
| 100,000   | 측정불가     | 4.353    | 0.157        | 0.053  |
| 1,000,000 | 측정불가     | 465.242  | 2.403        | 0.293  |

즉, <표 11>에서 n이 100인 경우,  $O(n^2)$ ,  $O(n \lg n)$ ,  $O(n)$  알고리즘은 소요시간의 차이가 발생 하지 않지만, 1,000,000인 경우에는 큰 차이가 발생 함을 확인할 수 있다. 반면,  $O(n^3)$  알고리즘은 n이 10,000부터 다른 알고리즘과의 소요시간 차이가 크게 벌어짐을 알 수 있다. 이러한 예시를 통해 교수 자는 학습자에게 시간 복잡도의 점근적 표기법에서 상수항과 계수를 무시하는 이유를 보다 직관적인 방법으로 설명할 수 있다.

## 6. 결론

교육부는 2015년 9월에 지식정보사회가 요구하는 핵심역량을 갖춘 창의·융합형 인재를 양성하기 위한 2015 개정 교육과정을 고시하였다. 특히, 정보과 교육과정의 핵심역량으로 컴퓨팅 사고력, 협력적 문제해결력 등을 제시하고 초등학교 실과에서 17시간 이상, 중학교 정보 과목에서 34시간 이상을 필수로 이수토록 하였으며, 고등학교는 자신의 진로와 연계하여 보다 심화된 학습이 가능하도록 하였다. 초등학교와 중학교에서는 놀이 중심의 다양한 언플러그 활동이나, 스크래치와 엔트리 등의 그래픽 기반 언어를 활용하는 등 학습자의 수준에 맞는 다양한 교수·학습 및 평가 방법에

대한 연구가 한창 진행 중이다. 그러나, 고등학교, 특히 과학계열고등학교의 전문 교과인 ‘정보과학’의 보다 심화된 내용에 걸맞는 교수·학습 및 평가 방법에 대한 구체적인 연구는 아직 부족한 편이다.

프로그래밍(알고리즘) 자동 평가 시스템은 사용자가 제출한 소스코드의 정확성과 알고리즘의 시간/공간 효율성 등을 자동으로 평가하고, 주어진 문제를 해결하기 위해 사용자가 제출한 소스 코드에 대해 즉각적인 평가 결과와 교정적 피드백을 제공한다. 또한, 이러한 실시간 평가 결과를 통해 만들어지는 문제별 채점 현황(제출 횟수, 통과 횟수), 랭킹 등의 경쟁적인 요소는 사용자에게 프로그래밍 학습에 대한 흥미와 동기를 제공하는 장점이 있다. 최근 고등학교의 현장 교사들이 오픈 소스로 공개된 HUSTOJ를 적용하거나, 직접 시스템을 개발하여 학교 현장에서 운영, 활용하고 있으며, 학생들에게 좋은 반응을 얻고 있다.

본 연구에서는 프로그래밍(알고리즘) 자동 평가 시스템의 이론적 배경과 선행 연구에 대한 고찰, 국·내외의 자동 평가 시스템의 동향에 대해 살펴 보았으며, 고등학교와 대학의 학부 과정에서 활용할 수 있는 방안을 제시하였다. 즉, 2015 개정 교육과정의 고등학교 과학 계열 전문 교과인 ‘정보과학’ 과목의 교수·학습 및 평가 방법에서 자동 평가 시스템의 활용을 제시하고 있는 만큼 C언어의 문법에 관한 기초적인 내용에서부터 주어진 문제의 알고리즘 설계와 프로그래밍 단계까지 폭넓게 적용할 수 있으며, 대학의 자료구조와 알고리즘 강좌에서 문제 해결의 주요 열쇠가 되는 자료구조와 알고리즘을 기준으로 이를 통해 해결할 수 있는 문제(군)을 난이도 순서로 제시함으로써 문제 해결의 응용력을 향상시킬 수 있다. 또한 동일 문제에 대한 각 알고리즘의 실제 소요 시간을 직접 비교해 봄으로써 알고리즘의 성능과 효율성을 직관적으로 확인할 수 있으며, 점근적 분석의 의미를 보다 쉽게 이해할 수 있다.

## 참 고 문 헌

- [1] 교육부(2015). **2015 개정 교육과정 총론**. 교육부 고시 제2015-74호 (별책 1)
- [2] 교육부(2015). **실과(기술·가정)/정보과 교육과정**. 교육부 고시 제2015-74호 (별책 10)
- [3] 교육부(2015). **과학 계열 전문교과 교육과정**. 교육부 고시 제2015-74호 (별책 20)
- [4] 송지희(2011). **자기주도학습을 위한 자동채점 기반의 프로그래밍 교육 시스템**. 박사학위논문, 숭실대학교.
- [5] 정종광(2010). **과학고 학생을 위한 Online Judge 기반 프로그래밍 평가 시스템의 설계 및 구현**. 석사학위 논문, 한국교원대학교
- [6] 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.
- [7] 장원영, 김성식(2014). 알고리즘 자동평가 시스템의 개발 및 적용 : 프로그래밍 학습 효과 분석. **한국컴퓨터교육학회 논문지**, 17(4), 26-34.
- [8] 임형석(2007). ACM-ICPC 문제의 출제 및 채점 과정. **한국정보과학회지**, 25(7). 52-55.
- [9] 김미혜(2007). 자동화된 프로그래밍 과제 평가 시스템의 설계 및 구현. **인터넷정보학회 논문지**, 8(6), 75-85.
- [10] 정은미(2009). **프로그래밍 언어에 대한 주관식 자동 채점 시스템의 설계 및 구현**. 석사학위 논문, 안동대학교
- [11] 전현석, 정종광, 김성식(2014). C언어 기초 학습을 위한 문제 설계 및 운영. **한국컴퓨터교육학회 학술발표대회 논문집**, 18(1), 291-294.
- [12] GitHub.(2017). *zhblue/hustoj*. [online] Available at: <https://github.com/zhblue/hustoj> [Accessed 3 Feb. 2017].
- [13] Miguel A. R., Shahriar M., &Rujia L.(2008). Competitive Learning in Informatics: The UVa Online Judge Experience, *Olympiads in Informatics*, 2, 131 - 148
- [14] Mike M.(2017). *Codeforces*. [online] Available at: <http://codeforces.com> [Accessed 3 Feb. 2017].

- [15] Ying F., Xu P., Xie D.(2017). *PKU Judge Online*. [online] Available at: <http://poj.org> [Accessed 3 Feb. 2017].
- [16] Anon(2017). *UVa Online Judge*. [online] Available at: <http://uva.onlinejudge.org> [Accessed 3 Feb. 2017].
- [17] 정종광(2017). *KoiStudy*. [online] Available at: <http://koystudy.net> [Accessed 3 Feb. 2017].
- [18] 장원영(2017). *Coding is fun*. [online] Available at: <http://codingfun.net> [Accessed 3 Feb. 2017].
- [19] 배준호(2017). *CodeUp Online Judge*. [online] Available at: <http://codeup.kr> [Accessed 3 Feb. 2017].
- [20] 김봉석(2017). *JudgeOn*. [online] Available at: <http://judgeon.net> [Accessed 3 Feb. 2017].
- [21] 이규호(2017). *Develet*. [online] Available at: <http://develet.com> [Accessed 3 Feb. 2017].
- [22] Startlink, (2017). *Baekjoon Online Judge*. [online] Available at: <http://acmicpc.net> [Accessed 3 Feb. 2017].
- [23] 장원영, 김성식(2014). 프로그래밍 경시대회 문제를 이용한 알고리즘 지도방법 제안 (과학교등학교 사례를 중심으로). **한국컴퓨터정보학회지**, 22(2), 15-24.
- [24] 임형석, 김희철(2007). 경시대회를 통한 프로그래밍 교육 활성화 방안. **한국정보과학회지**, 125(7). 35-37.
- [25] Raymond S. P., John D. H., & Kayla M. H.(2015). Are automated assessment tools helpful in programming courses?. *2015 ASEE Annual Conference & Exposition*.
- [26] 윤성준, 조상민(역)(2003). **생각하는 프로그래밍**, 서울: 인사이트.



## 장 원 영

1998 충북대학교 회계학과(경영정보학 부전공, 경영학사)  
 2002 충북대학교 컴퓨터교육과(이학사)  
 2015 한국교원대학교 대학원 컴퓨터교육 전공 박사 과정 수료  
 2002~2003 오창고등학교, 음성고등학교 근무  
 2004~2013 충북과학고등학교 정보과학 교사  
 2014 충북교육정보원 파견교사  
 2015~현재 교육부 교육연구사  
 관심분야: Computational Thinking, 알고리즘 교육, 프로그래밍 교육, 프로그래밍(알고리즘) 자동 평가 시스템, Java 언어 교육  
 E-Mail: won0c@naver.com



## 김 성 식

1977 고려대학교 경영학과(경영학사)  
 1977~1991 교육부 및 교육정책 자문위원회 근무(행정고시 19회)  
 1988 미국 오리곤 주립대학교 대학원 컴퓨터학과(이학석사)  
 1992 고려대학교 컴퓨터학과(이학박사)  
 1992~현재 한국교원대학교 컴퓨터교육과 교수  
 관심분야: 교육용콘텐츠, 알고리즘, 원격교육  
 E-Mail: seongkim@knue.ac.kr