

---

# 기초 알고리즘을 활용한 프로그래밍 언어 학습 시스템

박경욱\* · 오경숙\* · 류남훈\* · 이혜미\* · 김응곤\*

Learning System of Programming Language using Basic Algorithms  
Kyoung-Wook Park\* · Kyeong-Sug Oh\* · Nam-Hoon Ryu\* · Hye-Mi Lee\* · Eung-Kon Kim\*

## 요 약

알고리즘을 비롯한 프로그래밍 교육 과정은 전자공학 및 컴퓨터 관련학과를 비롯한 많은 이공계 학과에서 매우 중요한 교과목으로 인식된다. 하지만 과목의 특성 상 많은 학생들이 어려워하고 있으며, 이로 인해 상위 단계의 교과목 수강 시 많은 어려움을 겪고 있다. 순서도는 어떤 문제를 해결하는데 필요한 논리적인 단계들을 그림으로 표현한 것으로, 알고리즘의 흐름을 이해하는데 많이 활용된다. 알고리즘 및 프로그래밍은 구현 과정의 이해를 돕기 위해서 실습 중심의 교육이 매우 중요하다. 더욱이 알고리즘의 개념 이해 및 프로그램 실행 과정에 대한 이해력을 높일 수 있는 보조 프로그램의 필요성이 절실히 요구된다. 본 논문에서는 프로그래밍 과정 중 기본 알고리즘에 대해 쉽게 익힐 수 있도록 기초 알고리즘을 활용한 프로그래밍 언어 학습 시스템을 설계하고 구현하였다.

## ABSTRACT

The curriculum of programming education including algorithm has been recognized as a very important subject to many students majoring in natural sciences and engineering including electronic engineering and computer related departments. However, many students have had difficulties with it due to its characteristics; as a consequence, they have been in trouble taking upper-level subjects. Flow chart is a diagram that expresses logical stages necessary to solve certain problems and has been widely used to have an understanding of the flow of algorithm. The practice-oriented education of algorithm and programming would be very important to assist the understanding of operation processes. Furthermore, it has been desperately required to the necessity of auxiliary programs that could enhance an understanding of the concept of algorithm and program execution process. This study was aimed to design and embody the learning system of programming languages using basic algorithms so as for students to easily learn basic algorithm among the entire programming curriculum.

## 키워드

Flow Chart, Programming Education, Algorithm, GCD

## 1. 서 론

컴퓨터를 이용하여 실생활과 밀접한 관련이 있는 문제를 해결해가는 체계를 익히고, 논리적 사고방식과 문제해결 능력을 길러, 미래의 국가경쟁

력을 육성할 인재를 기를 수 있는 프로그래밍과 알고리즘 교육과정은 매우 중요하다.

IT강국에서 소프트웨어 강국으로 발전해가고 있는 시점에서 국가가 필요로 하는 고급 소프트웨어 인력 규모를 감안할 때 현재 학생들의 프로그

---

\* 순천대학교 컴퓨터학과  
접수일자 : 2009. 11. 24

교신저자, 김응곤(kek@sunchon.ac.kr)  
심사완료일자 : 2010. 1. 28

래밍 기피 현상은 국가적으로 우려스러운 수준이다. 프로그램을 배울 때에 데이터 구조에 대한 사전 지식을 필요로 하고, 수학적 사고 능력을 필요로 하므로 많은 학생들이 어려워하고 있다.

이러한 초보 프로그래머를 위하여 다양한 멀티미디어 요소를 활용하여 알고리즘을 교육하고 있지만, 흥미 유발에 많은 어려움이 있어 알고리즘의 개념을 정립하기 어렵고, 상위 단계의 교과목 수강 시 이해도가 떨어지는 등 어려움이 많은 상황이다. 따라서 구현원리를 이해 할 수 있도록 하면서 알고리즘과 프로그램을 동시에 공부할 수 있는 실기 위주의 학습프로그램이 절실히 요구되고 있다.

본 논문에서는 자료구조와 프로그램에 대한 사전지식이 없는 초보자들도 쉽게 학습할 수 있는 시스템을 개발하였다. 순서도를 이용하여 사용자가 알고리즘을 설계할 수 있으며, 또한 제공 되어진 샘플을 이용하여 순서도를 불러와서 학습할 수 있도록 하며, 사용법에 대한 별도의 교육 없이도 활용이 가능하도록 GUI 기반의 직관적 인터페이스로 구성하였다. 제공되는 샘플은 순서도 뿐만 아니라, C 언어로 된 소스 코드까지 제공하여 이해도를 높였다.

알고리즘의 진행 과정은 이해도에 따라 단계별로 진행할 수도 있으며, 빠르게 또는 느리게 진행할 수 있다. 또한 진행 과정 중 변경되어지는 스택 값, 변수, 실행 결과 등을 확인하면서 알고리즘을 학습 할 수 있도록 하였다.

## II. 관련연구

### 2.1 학부기초 과정의 프로그래밍 교육 내실화 필요성

지식정보사회에 살아가는 우리들에게 중요한 능력은 어떤 사실을 아는 것뿐만 아니라, 새로운 상황에 당면했을 때 아이디어의 창출, 적용을 통해 문제 상황을 효과적으로 해결하는 것이다.

컴퓨터 프로그래밍은 컴퓨터를 활용하여 학습자가 컴퓨터에게 자신이 원하는 것을 수행하도록 하는 작업으로써 일반적인 사고력 신장과 메타 인

지적 측면의 효과, 이해도에 대한 모니터링, 문제 분석 기술 등의 향상뿐만 아니라 컴퓨터 계산 원리 이해와 논리적 사고의 표현 등의 학습 효과를 가지고 있다[1].

최근 프로그래밍 학습이 논리적 사고에 어떠한 영향을 미치는가에 대한 연구가 진행되어 왔다 [2-3]. 그러나 이들 연구 결과들은 특정 언어 하나만을 다루어 부분적이며, 논리적 사고에 대한 긍정적 효과가 인지 발달에 의한 자연적 현상에 기인한 것인지 또는 프로그래밍 학습의 결과에 기인한 것인지를 구분이 명확하지 않다. 또한 학습자의 학업성취나 문제해결능력에 대한 비교 연구는 있으나, 학습자의 메타인지 수준에 따라 프로그래밍 학습이 논리적 사고 발달에 미치는 영향에 대한 연구 결과는 미비한 실정이다.

가장 고민할 점은 프로그램을 어떻게 쉽고 재미있게 인식 시킬 것인지에 있다. 단순 암기 위주의 교육을 받아 왔기 때문에 무엇인가를 스스로 만든다는 것에 대한 익숙하지 않음과 두려움을 가지고 있다는 점이다. 이런 학생들에게 프로그램을 쉽게 배울 수 있도록 하는 것은 중요한 과제이다.

국가가 필요로 하는 고급 소프트웨어 인력 규모를 감안할 때, 현재 학생들의 프로그래밍 기피 현상은 국가적으로 우려스러운 수준이다. 그 이유 중 하나로 대학 1학년 때의 프로그래밍 기피현상을 들 수 있다. 전자, 컴퓨터 분야로 입학한 경우로 한정하여 살펴보면 전자 분야는 중고교 때 공부하던 방식의 수식을 외우고, 풀고, 점수 받는 방식으로 운영되는 반면, 프로그래밍은 무언가를 스스로 만드는 것을 요구하므로 힘들어하는 학생이 적지 않다[4].

### 2.2 순서도를 활용한 프로그래밍 교육의 필요성

프로그래밍 과정은 문제 해결 과정을 연습하고, 창조적이며, 발명적인 문제 해결 방법을 연습할 수 있으며, 지적 사고력을 높일 수 있고, 인지 발달에도 도움을 주며[5], 문제 해결 방법에 대한 실험적 방법을 촉진 시킨다[6].

프로그래밍은 주어진 문제를 해결하기 위해 계

획을 세우고, 프로그래밍 언어의 구문이나 어휘를 기억하였다가 재생하는 것 이상의 인지적 기술을 필요로 할 뿐만 아니라, 프로그래밍 환경이 프로그래머에게 주는 즉각적인 피드백 같은 서비스를 제공함으로써 그 과정에서 요구되어지는 인지적 기술들을 향상시켜 준다.

프로그래밍 교육은 문제 분석 및 해결 능력, 논리적 사고력, 절차적 문제 해결 방식 등을 습득할 수 있으며, 다른 응용 프로그램들을 더욱 깊이 이해하고 잘 활용할 수 있는 기초가 될 수 있다[7].

프로그램은 문제에 대한 지식을 동원하여 문제 해결 전략을 세우고, 알고리즘을 구현한 후 프로그램 언어로 부호화하는 과정[8]으로 논리적 사고력을 배양하고, 오류 검증 및 수정 작업에서 반성적 사고 능력과 같은 고등인지기술을 향상시킬 수 있다.

이러한 프로그래밍 교육은 정보화 시대가 요구하는 각 교과를 초월한 확산적 사고력과 문제해결력을 신장시킬 수 있는 교육으로 아주 중요한 자리를 차지하고 있다.

### 2.3 알고리즘 시각화

시각화를 통한 정보의 제공은 문서 형태의 정보 제공보다도 많은 양의 정보를 효과적으로 전달해 줄 수 있다. 프로그램의 시각화는 프로그램의 동작, 제어의 흐름, 그리고 자료의 상태 변화 등을 시각적 형태로 표현하는 것을 말한다. 이것은 프로그램의 특정 상태를 정적으로 나타내는 것 뿐 아니라 전체적인 혹은 일정 범위의 수행과정을 동적으로 나타내는 것도 포함한다. 특히 후자의 기능을 제공하는 프로그램 시각화를 알고리즘 애니메이션(Algorithm Animation 혹은 Animated Graphical View)이라 한다[9-10].

프로그램을 시각화하고자 할 때에 사용하는 방법으로는 선정의(Predefinition) 방법, 함수 호출(Annotating) 방법, 선언(Declaration) 방법, 조정(Manipulation) 방법 등이 있는데, 이중 함수 호출 방법이 가장 많이 사용된다.

현재 개발된 프로그램 시각화 시스템으로는 BALS(Brown University Algorithm Simulator

and Animator) 알고리즘 애니메이션 시스템 [11-12], Zeus 알고리즘 애니메이션 시스템[13], TANGO(Transition-based Animation Generation) 프로그램 시각화 시스템[14-15], Pavane 시각화 시스템 등이 있으며, 병렬 프로그램에서의 시각화[16], Heterogeneous 환경에서의 시각화 등 여러 방면으로 연구되어 지고 있다. 또한, 자바를 위한 Jeliot 3[17], C++와 Ville와[18], Planani[19]을 위한 VIP[20] 등 많은 프로그램 시각화 도구가 있다.

### III. 프로그래밍 언어 학습 시스템

본 논문에서는 초중고교 및 대학의 저학년 공학도들에게 프로그래밍 언어 관련 과목 강의 시 프로그래밍 언어와 알고리즘의 개념 정립에 활용할 수 있도록 순서도를 활용한 프로그래밍 교육 시스템을 그림 1과 같이 구성 하였다.

프로그래밍 기초 교육 시스템은 순서도 기호 선택 모듈, 순서도 배치 모듈, 진행과정 표현 모듈, 명령어 해석 모듈, 소스코드 실행과정 표현 모듈로 구성된다. 본 시스템을 설계할 때 중점을 둔 사항은 프로그래밍 기초 교육과 알고리즘 교육에 대한 사전 지식이 없는 초보자들이 직접 실행하기에 적합하도록 하였다.

#### 3.1 기호 선택 모듈

순서도 기호 선택 모듈은 사용자에게 편리한 GUI 환경을 통하여 기호를 선택하는 모듈로써, 순서도 기호 선택 시 해당 순서도에 대한 도움말이 제시되며, 각 순서도에 대한 기호 그림이 마우스 포인터의 위치에 표시된다.

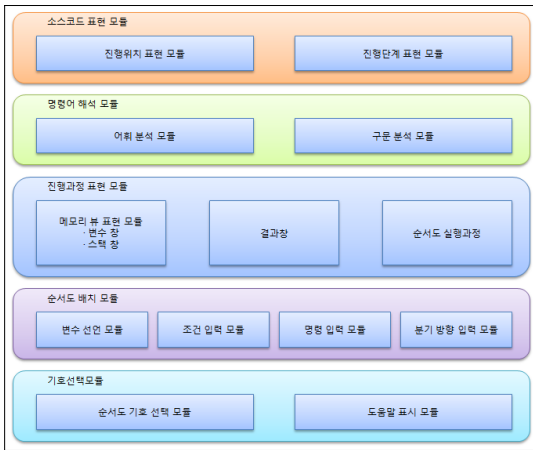


그림 1. 프로그래밍 언어 학습 시스템 구조도  
 Fig. 1 The Block Diagram of Programming Language Learning System

### 3.2 순서도 배치 모듈

순서도 배치 모듈은 작업하고자 하는 알고리즘에 따라 순서도를 선택하여 순서도 배치 영역에 배치하는 모듈이다. 이 모듈에는 변수 선언 모듈, 조건 입력 모듈, 명령 입력 모듈, 분기방향 입력 모듈로 구성된다. 선택한 순서도는 종류에 따라 변수를 선언하거나, 조건을 입력할 수 있으며, 또한 명령을 입력하기도 한다. 이 모듈에서 알고리즘에 대한 전체적인 윤곽을 설정하게 된다.

### 3.3 진행과정 표현 모듈

진행과정 표현모듈은 메모리뷰, 결과 창, 순서도 실행과정의 3가지 표현단계로 구성된다.

메모리 뷰는 변수 창과 스택 창으로 구성된다. 변수 창은 변수 간의 관계 확인과 변수 값의 변화를 각 단계별로 진행하면서 확인하는데 사용된다. 프로그램에 사용된 변수들의 목록을 표시하고, 알고리즘 진행 단계별 변수의 값 변화과정을 실시간으로 보여준다. 이러한 변수 표현은 알고리즘의 핵심이 되는 변수의 변화과정을 시각화하여 표현하기 때문에 초급 입문자들의 알고리즘 개념 정립에 큰 도움을 준다. 스택 창은 스택의 값을 확인

할 수 있도록 하여, 자료 구조 및 자료의 흐름 과정을 효과적으로 이해하도록 돕는다.

결과 창은 알고리즘을 실행하기 위해 필요한 변수 값을 사용자에게 입력 받고, 알고리즘의 흐름에 따라 실행된 결과를 출력한다. 결과 창을 통한 사용자의 직접적인 참여유도와 적절한 시점의 결과 값 표현은 알고리즘의 이해력 향상에 도움을 준다.

순서도 실행 과정은 순서도를 실행시켰을 경우, 현 단계에 사각형 테두리를 표시하여 실행 상태를 쉽게 확인할 수 있도록 한다.

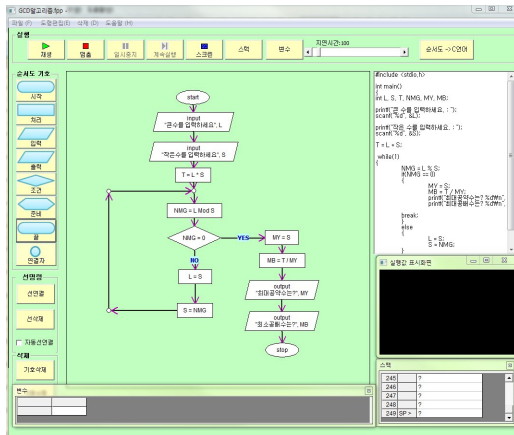
### 3.4 명령어 해석 모듈

명령어 해석 모듈은 토큰을 분리하여 어휘를 분석하고 구문을 분석한다. 어휘 분석 단계는 원시 프로그램의 문자들을 어휘 단위로 분류하는 과정이다. 프로그램의 어휘 단위는 식별자, 특수어, 연산자, 구분자 기호 등으로 구성된다. 구문 분석 단계는 어휘 분석기로부터 어휘 단위들을 전달받아, 이들을 사용하여 파스 트리(Parse Tree)라 불리는 계층적 구조를 생성한다. 이러한 파스 트리는 프로그램의 구문 구조를 표현한다. 대부분의 경우에 파스 트리 구조가 실제적으로 구성되지 않고, 이 트리를 생성하는 데 필요한 정보가 생성되고 사용된다.

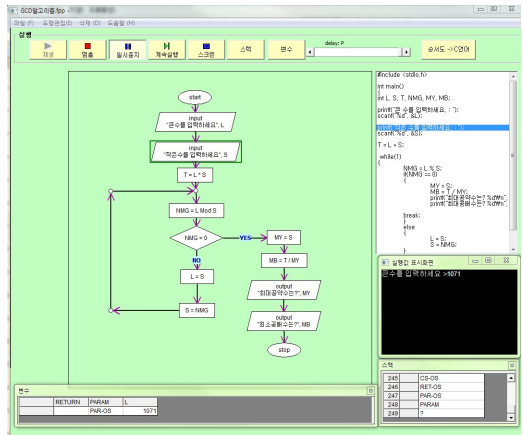
사용자는 작성한 순서도를 진행시켜 보면서 계획한 알고리즘의 진행과정을 확인하게 된다. 이 과정은 알고리즘을 이해하는데 큰 도움을 주며, 계획한 알고리즘에 맞게 순서도가 작성되었는지 확인할 수 있다. 순서도 기호가 부적절하게 배치된 경우 에러 메시지가 출력된다.

### 3.5 소스코드 실행 과정 표현 모듈

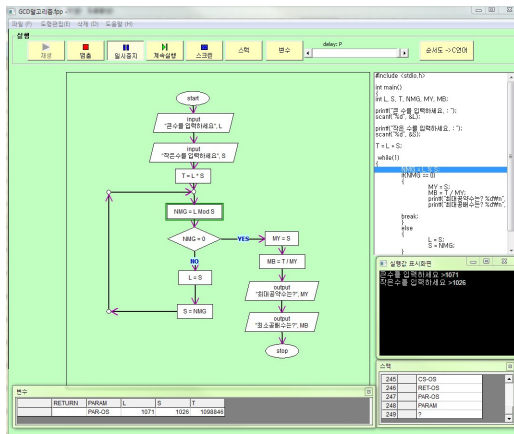
알고리즘의 진행 과정은 순서도 뿐만 아니라 소스코드를 통해 단계별로 진행 과정을 확인할 수 있다. 프로그래밍 기초를 배우고자 하는 공학도들은 순서도를 직접 작성하여 보고, 그 순서도의 실행 결과를 확인할 수 있으며, 소스코드의 단계별 표현과정까지 시각적으로 확인이 가능하다.



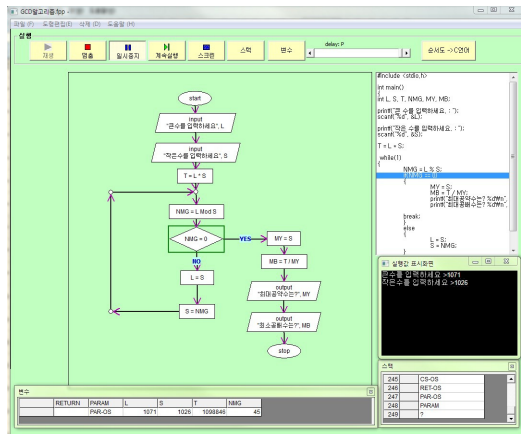
(a)



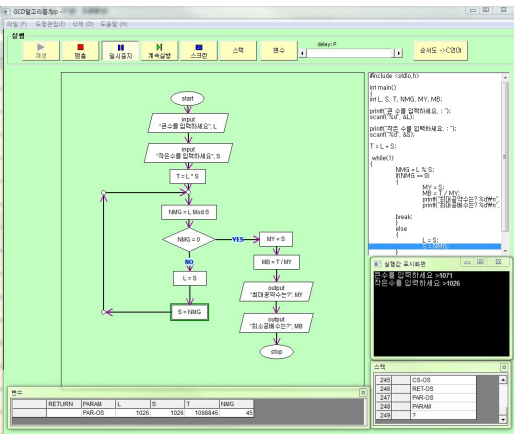
(b)



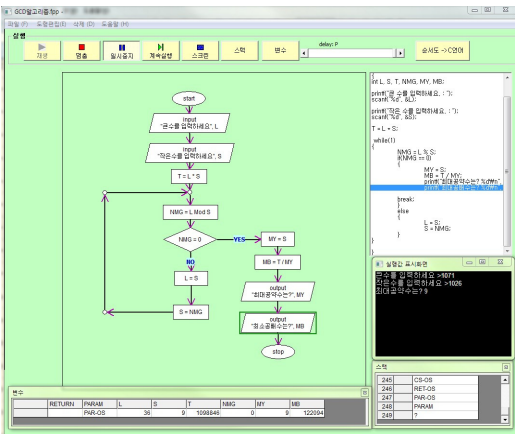
(c)



(d)



(e)



(f)

그림 2. GCD 알고리즘 실행 과정  
Fig. 2 GCD Algorithm Implementation Process

#### IV. 프로그래밍 언어 학습 시스템 구현

대표적인 알고리즘 중 하나인 GCD 알고리즘을 예로 들어 진행과정을 알아본다.

##### 4.1 GCD 알고리즘

GCD 알고리즘의 대표적인 유클리드 호제법(Euclidean algorithm)은 2개의 자연수 또는 정수의 최대공약수를 구하는 알고리즘의 하나이다. 호제법이란 말은 두 수가 서로 상대방 수를 나누어서 원하는 수를 얻는 알고리즘을 나타낸다. 2개의 자연수(또는 정수)  $m, n$ 에 대해서  $a$ 를  $b$ 로 나눈 나머지를  $r$ 이라 하면,  $a$ 와  $b$ 와의 최대공약수는  $b$ 와  $r$ 와의 최대공약수와 같다. 이 성질에 따라,  $b$ 를  $r$ 로 나눈 나머지  $r'$ 를 구하고, 다시  $r$ 을  $r'$ 로 나눈 나머지를 구하는 과정을 반복하여 나머지가 0이 되었을 때 나누는 수가  $a$ 와  $b$ 와의 최대공약수이다. 이는 명시적으로 기술된 가장 오래된 알고리즘으로도 알려져 있다.

##### 4.2 GCD 알고리즘 구현 과정

기초 알고리즘을 활용한 프로그래밍 언어 학습 시스템을 이용하여 알고리즘의 가장 대표적인 GCD 알고리즘을 구현하여 보고자 한다.

이 시스템을 이용하여 사용자가 직접 GCD 알고리즘을 작성하거나 이미 작성된 샘플을 불러온 후, 실행 단추를 이용하여 실행한다. GCD 알고리즘을 구하기 위해 가장 먼저 두 수  $L$ 과  $S$ 를 입력 받는다. 그림 2의 (a)는 GCD 알고리즘이 작성된 화면이며, (b)는 두 수를 입력 받는 화면이다. 알고리즘의 진행과정은 순서도 표시 영역과 C 언어 소스 코드 영역에 표시된다. 두 수는 사용자로부터 입력 받음과 동시에 변수에 기억된다.

사용자가 입력한 두 수  $L$ 과  $S$ 를 곱하여 두 수에서 나올 수 있는 최대값을  $T$ 라는 변수에 기억시킨다. 이때 사용자는 변수의 변화과정을 확인할 수 있다. 최대값  $T$ 는 최소공배수(LCD)를 구할 때  $T$ 를 최대공약수로 나누어 구할 수 있다.

다음은 나머지 NMG를 구하게 된다. 즉 나머지

(NMG)에는 큰 값( $L$ )을 작은 값( $S$ )으로 나누어서 나온 나머지가 입력된다. 그림 2의 (c)는 큰 값을 작은 값으로 나누어서 나온 나머지 NMG가 계산되는 과정이다.

그림 2의 (d)는 계산되어진 나머지(NMG)가 0인가를 확인하여 0일 경우와 0이 아닐 경우에 대하여 조건문을 이용하여 다음 단계로 이동하는 과정으로, 조건식에서 나머지가 0과 같지 않을 경우 계속 반복문을 수행하는 과정이다. 큰 값에 작은 값을 넣고 작은 값에 나머지를 넣어서 다시 계산하여 반복문을 수행한다. 변수 표시영역에는  $L, S, NMG$  값이 반복문을 수행하는 동안 계속적으로 변하고 있음을 확인할 수 있다.

그림 2의 (e)는 반복문을 수행하는 동안 작은 값  $S$ 의 변화과정을 나타내는 화면이다. 계속되는 반복실행으로 인하여 나머지 값이 0이 되면 조건문에 의하여 최대공약수와 최소공배수를 구한다.

그림 2의 (f)는 최대공약수와 최소공배수를 구하여 출력하는 과정이다.

즉 이 시스템의 실행과정에서 큰 수  $L$ 에는 1071이라는 숫자를 넣고, 작은 수  $S$ 에는 1029란 숫자를 입력한다. 변수  $T$ 는  $1029 \times 1071$ 의 값인 1,102,059가 계산되어 입력되며, 변수 NMG에는 큰 수를 작은 수로 나눈 나머지 값이 입력된다. NMG가 0과 다르면 변수  $S$ 의 값을 변수  $L$ 에 넣고, 변수  $S$ 에는 변수 NMG 값을 넣어 루트를 반복한다.

변수 NMG가 0이면 변수  $MY$ 에는 변수  $S$ 의 값이 입력되고, 변수  $MB$ 에는 변수  $T$ 의 값을 변수  $MY$  값으로 나눈다. 최대공약수는  $MY$ 의 값을 출력하고, 최소공배수는 변수  $MB$  값을 출력한다.

프로그램 입문자는 프로그래밍 언어 및 자료구조에 대한 사전지식이 없더라도 프로그램의 진행과정 및 변수의 변화과정을 보면서 자연스럽게 이해할 수 있으며, 샘플로 제공되는 풍부한 기초 알고리즘 및 순서도에 대한 도움말을 보면서 알고리즘 및 프로그래밍 언어에 대한 기초 지식을 습득할 수 있다.

## V. 결 론

프로그래밍 언어의 학습은 수학적 사고와 데이터 구조에 대한 사전지식을 필요로 하므로 프로그래밍 입문자들은 매우 어려워하며, 프로그래밍 입문자들이 쉽게 학습할 수 있는 실기 위주의 보조 프로그램의 필요성이 절실히 대두되고 있다.

본 논문에서는 프로그래밍 입문자들도 알고리즘의 개념 및 프로그래밍 언어에 대한 기본 지식을 쉽게 습득할 수 있도록 순서도를 활용한 프로그래밍 언어 학습 시스템을 구현하였다. 이 시스템은 사용자가 작성하고자 하는 내용의 문제를 인식하고, 실행 순서에 맞게 순서도 기호를 배치한 후 단계별로 실행시켜 볼 수 있는 시스템으로, 프로그래밍 입문자들도 순서도 기호에 대한 자세한 설명을 참조면서 알고리즘을 작성할 수 있도록 하였다. 또한 작성된 알고리즘의 진행과정을 단계별로 진행시켜 볼 수 있으므로 프로그래밍에 대해 가지고 있었던 막연한 두려움을 없앨 수 있으며, 알고리즘에 대한 흥미도 끌어 올릴 수 있다. 그리고 기초 알고리즘의 풍부한 제공으로 알고리즘의 개념을 이해하는데 큰 도움이 되며, 해당 알고리즘에 대한 C 언어 소스의 제공으로 프로그래밍 언어에 대한 기본 흐름도 쉽게 습득할 수 있다.

### 감사의 글

이 논문 또는 저서는 2008년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임  
(KRF-2008-313-D00001)

### 참고 문헌

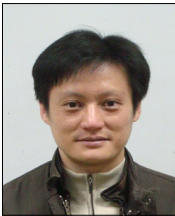
- [1] 홍순정, "지능과 창의성", 양서원 1999.
- [2] 권창미, "프로그래밍 수업이 인지발달 수준과 논리적 사고에 미치는 효과", 안동대학교 석사학위논문, 2007.
- [3] 현혜경, "LOGO 프로그래밍 학습이 논리적 사고에 미치는 영향", 안동대학교 석사학위논문, 2006.
- [4] 오일석, 양재동, 김수형, "국내 컴퓨팅 교육의

- 현황, 문제점, 그리고 개선방향", 정보과학회지, 제25권 제 7호, pp.14-20, 2007.
- [5] 백영균, "컴퓨터 프로그래밍에 대한 심리학적 접근", 교육공학연구, Vol.4, No.1, pp.145-165, 1988.
  - [6] 박관우, 이성근, "웹기반 프로그래밍 교수/학습 시스템", 대구대학원 석사논문, 2001.
  - [7] 박미화, "목표모형에 기반한 중등 일반계 프로그래밍 교육과정 개발", 한국교원대학교 대학원 석사논문, pp.1-2, 2006.
  - [8] 이정모, 이건효, "초보자의 C언어 학습과정에 대한 인지심리학적 분석 연구 : 프로그래밍 학습과정 동안의 은유 사용의 효과", 인지과학, Vol.9, No.4, pp.2-3, 1998.
  - [9] Marc H. Brown, "Techniques for Algorithm Animation," IEEE Computer, Vol.2, No.1, pp.28-39, Jan. 1985.
  - [10] John T. Stasko, "Tango: A Framework and System for Algorithm Animation," IEEE computer, Vol.23, No.9, pp.27-39, Sep. 1990.
  - [11] Marc H. Brown, "Exploring Algorithms Using Balsa-II," IEEE Computer, Vol.21, No.5, pp.14-36, May 1988.
  - [12] Marc H. Brown, "MacBALS A Version Aleph.3," Manuscript, 1989.
  - [13] Marc H. Brown, "Zeus: A System for Algorithm Animation and Multi-View Editing," DEC. SRC research report 75, Feb. 1992.
  - [14] Jone T. Stasko, "The Path-Transition Paradigm: A Practical Methodology for Adding Animation to Program Interfaces," Manuscript, College of Computing, Georgia Institute of Technology, June 1991.
  - [15] Jone T. Stasko, "XTANGO Algorithm Animation Designer's Package," Manuscript, College of Computing, Georgia Institute of Technology, Oct. 1992.
  - [16] Jone T. Stasko, "A Methodology for Building Application-Specific Visualization of Parallel Programs," Technical Report, Graphics, Visualization, and usability Center, Georgia Institute of Technology, 1992.
  - [17] Moreno, A., N. Myller, E. Sutinen and M. Ben-Ari, "Visualizing programs with Jeliot 3," Proceedings of the International Working Conference on Advanced Visual Interfaces AVI 2004, 2004.
  - [18] Rajala, T., M.-J. Laakso, E. Kaila and T.

Salakoski, "VILLE - A language-independent program visualization tool," Proceedings of The Seventh Koli Calling Conference on Computer Science Education, 2007.

- [19] Sajaniemi, J. and M. Kuittinen, Visualizing roles of variables in program animation, Information Visualization, pp.137-153, 2004.
- [20] Virtanen, A. T., E. Lahtinen and H.-M. Järvinen, VIP, "A Visual Interpreter for Learning Introductory Programming with C++", Proceedings of the Fifth Finnish/Baltic Sea Conference on Computer Science Education, pp. 129-134, 2005.

저자 소개



**박경욱(Kyoung-Wook Park)**

1996년 8월 : 순천대학교 전자계산학과 (이학사)  
 1999년 8월 : 전남대학교 전산통계학과 (이학석사)

2004년 8월 : 전남대학교 전산학과 (이학박사)  
 ※ 관심분야 : 병렬 및 분산처리, 그래프 이론, 알고리즘



**오경숙(Kyeong-Sug Oh)**

2007년 8월 : 한국방송통신대학교 컴퓨터과학과 (이학사)  
 2009년 8월 : 순천대학교 컴퓨터과학과 (이학석사)

2009년 9월 ~ : 순천대학교 컴퓨터과학과 박사과정 재학 중  
 ※ 관심분야 : 영상처리, 컴퓨터 그래픽스, HCI



**류남훈(Nam-Hoon Ryu)**

2007년 2월 : 한국방송통신대학교 컴퓨터과학과 (이학사)  
 2009년 2월 : 순천대학교 컴퓨터과학과 (이학석사)

2009년 3월 ~ : 순천대학교 컴퓨터과학과 박사과정 재학 중  
 ※ 관심분야 : 영상처리, 컴퓨터 그래픽스, 알고리즘



**이혜미(Hye-Mi Lee)**

2004년 2월 : 순천대학교 정보통신공학부 (공학사)  
 2008년 9월 : 순천대학교 컴퓨터과학과 석사과정 재학 중

※ 관심분야 : 영상처리, 컴퓨터 그래픽스, HCI



**김응곤(Eung-Kon Kim)**

1980년 2월 : 조선대학교 전자공학과 (공학사)  
 1986년 2월 : 한양대학교 컴퓨터공학과 (공학석사)

1992년 2월 : 조선대학교 컴퓨터공학과 (공학박사)  
 1993년 3월 ~ 현재 : 순천대학교 컴퓨터과학과 교수  
 ※ 관심분야 : 영상처리, 컴퓨터 그래픽스, 멀티미디어, HCI