



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위논문

스크래치를 이용한 사고력 신장  
알고리즘 학습 프로그램 개발 및 적용 연구

Study on Development and Application  
of Algorithm Learning Program  
for Thinking Ability Improvement Using Scratch

제주대학교 교육대학원

초등컴퓨터교육전공

양 영 훈

2011년 2월

스크래치를 이용한 사고력 신장  
알고리즘 학습 프로그램 개발 및 적용 연구

Study on Development and Application  
of Algorithm Learning Program  
for Thinking Ability Improvement Using Scratch

지도교수 김 종 훈

이 논문을 교육학 석사학위 논문으로 제출함

제주대학교 교육대학원

초등컴퓨터교육전공

양 영 훈

2010 년 11 월

양영훈의

교육학 석사학위 논문을 인준함

심사위원장 김 종 우



심사위원 김 종 훈



심사위원 박 남 제



제주대학교 교육대학원

2010년 12월

## 목 차

국문 초록 .....	i
<b>I. 서론</b> .....	1
1. 연구의 필요성 .....	1
2. 연구내용 및 방법 .....	2
<b>II. 이론적 배경</b> .....	3
1. 교육용 프로그래밍 언어 .....	3
2. 알고리즘 .....	5
3. 논리적 사고력 .....	6
4. GALT (Group Assessment of Logical Thinking) .....	7
<b>III. 교재 개발</b> .....	8
1. 교육과정 분석 및 주제 선정 .....	8
2. 교재의 구성 체제 .....	10
3. 교재의 실제 .....	11
4. 교수학습 과정안과 지도자료 .....	53
<b>IV. 연구 방법 및 현장 적용</b> .....	72
1. 연구 가설 .....	72
2. 연구 대상 .....	72
3. 연구 설계 및 절차 .....	72
4. 연구도구 .....	73
5. 현장 적용 결과 및 해석 .....	74
<b>V. 결론 및 제언</b> .....	78
<b>참고 문헌</b> .....	80
<b>ABSTRACT</b> .....	82

## 표 목 차

<표 II-1> 교육용 프로그래밍 언어의 종류 및 특징 .....	4
<표 II-2> 논리적 사고의 하위요소 .....	6
<표 II-3> 축소본 GALT 문항별 측정내용 .....	7
<표 III-1> 알고리즘 교육 내용 체계표 .....	8
<표 III-2> 차시별 교육 주제 .....	9
<표 IV-1> 연구대상 집단과 사례수 .....	72
<표 IV-2> 연구의 실험설계 .....	73
<표 IV-3> 논리적 사고력 사전검사 결과 .....	74
<표 IV-4> 논리적 사고력 사후검사 결과 .....	75
<표 IV-5> 논리적 사고력 사전·사후검사 차이 검증 .....	76
<표 IV-6> 논리적 사고력 검증 종합 .....	77

## 그림 목 차

[그림 III-1] 안내된 발견식 수업절차와 교재의 구성 체제 .....	10
--	----

국문초록

스크래치를 이용한 사고력 신장  
알고리즘 학습 프로그램 개발 및 적용 연구

양 영 훈

제주대학교 교육대학원 초등컴퓨터교육전공

지도교수 김 종 훈

현대 사회의 흐름에 따라 정보 교육의 가치는 점점 더 높아지고 있다. 선진국들은 창의적이고 논리적인 미래의 인재 육성을 위해 체계적인 교육과정을 마련하여 정보 교육을 실시하고 있으며 우리나라 또한 그 중요성과 필요성을 이해하여 교육과정에 반영시키고 있다. 정보 교육의 가치의 증가와 함께 정보 교육의 내용에서도 기존의 컴퓨터 소양 교육과 컴퓨터 활용 교육에서 벗어나 컴퓨터 원리, 문제 해결 방법 및 절차를 통해 창의적이고 논리적인 사고를 키울 수 있도록 본질적인 학문으로서의 컴퓨터 과학 교육을 강조하고 있다.

이에 따라 본 연구에서는 컴퓨터 과학 교육에서 강조하는 알고리즘 학습을 위해 교육용 프로그래밍 언어인 스크래치를 활용하여 교재를 개발하였다. 개발된 교재는 W초등학교에 재학 중인 학생을 대상으로 투입하여 논리적 사고력 신장에 도움이 되었음을 밝혔다. 앞으로 컴퓨터 과학 교육을 위해 교육용 프로그래밍 언어를 사용한 보다 많은 교육 자료들이 개발되어야 할 것이며, 이를 위한 교육 환경 개선 및 지도교사들의 노력이 필요할 것이다. 또한 정선된 교육과정을 통해 체계적인 교육이 이루어질 수 있도록 교육내용 및 교수방법에 대해서도 많은 연구가 필요하다.

주요어 : 컴퓨터 과학 교육, 알고리즘, 교육용 프로그래밍 언어, 스크래치



# I. 서론

## 1. 연구의 필요성

학교에서 가르치는 것은 생명존중, 사랑, 평화와 같이 시대 불변의 가치도 있지만, 시대의 흐름에 따라 그 시대가 필요로 하는 것을 가르친다. 21세기 사회 전반적으로 '정보'라는 테마가 지배하는 시대에 정보 교육의 중요성은 그 가치가 날로 커져만 가고 있고, 그래야만 미래 사회를 주도하고 바르게 이끌어갈 인재를 양성하는 학교 본연의 목적을 달성할 수 있다.

이런 면에서 정보 기술의 선진국들은 정보 교육의 중요하다는 사실을 미리부터 인지하고, 정부 차원에서 주도하여 취학 전부터 정보 관련 개념들을 습득하고 창의적이고 논리적인 능력을 기를 수 있도록 정보 교육을 장려하고 있다. 이에 발맞추어 우리나라 또한 2007 개정 교육과정의 정보 교육은 제 7차 교육과정의 컴퓨터 소양 교육에서 벗어나 컴퓨터 원리, 문제 해결 방법 및 절차를 통해 창의적이고 논리적인 사고를 바탕으로 미래 사회를 대비할 수 있도록 본질적인 과학으로서의 정보 즉, 컴퓨터 과학 교육을 강조하고 있다.

이러한 컴퓨터 과학 교육의 필요성에 대해 Judith Gal-Ezer와 David Harel (1999)은 컴퓨터 과학은 현대 주요 기술 혁신의 과학적인 토대이며, 알고리즘적인 사고방식은 특별하고 강력한 사고방식으로써 복잡한 현대 사회에 대처하기 위해 필요하며 다른 과학 및 공학 분야보다도 중요하고, 학생들이 다양한 사고방식으로 새로운 형태의 문제에 접근할 수 있는 능력을 갖게 한다고 하였다.

그리고 ACM(Association of Computing Machinery)은 컴퓨터 과학 교육을 다음과 같이 크게 4가지 관점에서 해석하고 있다(ACM, 1991).

- 첫째, 컴퓨터 과학은 컴퓨터에 관한 연구이다.
- 둘째, 컴퓨터 과학은 정보에 관한 연구이다.
- 셋째, 컴퓨터 과학이란 프로그래밍에 관한 연구이다.
- 넷째, 컴퓨터 과학이란 알고리즘에 관한 연구이다.

이런 면에서 스크래치 같은 교육용 프로그래밍 언어를 이용한 프로그래밍 활동은 컴퓨터과학 교육에 적합한 매우 유용한 활동이다.

따라서 본 연구는 초등학교에서 스크래치를 통해 이루어지는 알고리즘 교육의 가치에 대해 연구하고자 한다.

## 2. 연구내용 및 방법

본 논문에서는 스크래치를 통한 알고리즘 교육이 초등학생의 논리적 사고력에 미치는 영향을 알아보기 위하여 연구 순서를 정하였다.

첫째, 국내 및 해외 교육과정을 분석하여 추출한 알고리즘 교육의 내용 중에서 선정 기준에 따라 교육 내용을 선정하였다.

둘째, 학습자의 흥미를 유발하고 알고리즘 원리를 쉽게 이해할 수 있는 수업방법을 채택하고 그것을 수정·보완하여 교재 구성 체계를 만들었다.

셋째, 선정한 교육 내용을 교재 구성 체계에 맞게 구성하여 스크래치를 통한 알고리즘 배우기 교재를 만들었다.

넷째, 교재를 초등학교 4학년 학생들에게 투입하여 교육적 가치가 있는지를 점검하였다.

## II. 이론적 배경

### 1. 교육용 프로그래밍 언어

교육용 프로그래밍 언어란 교육적인 목적을 가지고 개발된 프로그래밍 언어를 말하며, 범용 프로그래밍 언어와 달리 복잡하거나 양이 방대할 필요가 없으며 그 목적이 알고리즘적 사고력, 문제해결력을 키우는 것이므로 쉽고 배우기가 용이하다(이옥화, 1993, p. 88).

ACM에서는 “알고리즘적 사고를 위해서는 LOGO와 같은 교육용 프로그래밍 언어(EPL: Educational Programming Language)를 통한 프로그래밍 교육이 필요하다”고 강조하며 8학년의 컴퓨터 과학기초를 위한 학습단계에서부터 이러한 교육이 필요하다고 주장하고 있다.

#### 가. 교육용 프로그래밍 언어의 종류

East & Hutton(2007)은 교육용 프로그래밍 언어를 그래픽이 활용되는 범위와 방법에 따라 TPL(Typingbased Programming Language)와 VPL(Visual Programming Language)로 구분하였다. TPL은 프로그램의 코딩은 텍스트로 하되, 프로그래밍 환경에서 다양한 방법으로 그래픽을 지원하는 방식이다. 이는 또한 두 부류로 나눌 수 있는데 첫 번째는 텍스트 프로그래밍이 중심이나 그래픽을 지원하는 언어인 C의 GUI 환경이나 Visual Basic 등이 그 예이고, 두 번째는 실행 화면이 기본적으로 그래픽 환경을 쉽게 지원하나 프로그래밍의 코딩은 텍스트로 해야 하는 언어로, Logo, Karel the Robot, KPL 등을 그 예로 들 수 있다.

이와는 다르게 VPL이란 프로그래밍의 코딩 과정 자체가 그래픽을 지원하여 주로 Drag& Drop의 형식으로 쉽고 간단하게 프로그램을 구현할 수 있는 언어로, 실행 결과 또한 다양한 멀티미디어를 지원하는 그래픽 화면을 구성한다. VPL의 대표적인 예로는 Squeak Etoys, Stagecast creator, LegoMindstroms Labview, Scratch, Alice, Game Factory 2 등이 있다(정미연 등, 2008, p. 176-177).

이러한 교육용 프로그래밍 언어 중 초등학생에게 활용 가능한 것을 정리해 보면 다음과 같다(김수환 등, 2008, p. 375-376).

<표 II-1> 교육용 프로그래밍 언어의 종류 및 특징

언어	언어적요소	코드표현	적용요소
Squeak eToys (1997)	object-based	이미지, 타일 스크립트기반	게임, 애니메이션, 프레젠테이션, 초-중
Scratch (2007)	object-based, event-based	이미지, 블록스크립트 기반	게임, 애니메이션, 프레젠테이션, 로봇 프로그래밍, 초-중
Karel J Robot (2001)	object-oriented	텍스트 기반	문제해결, 초-고
AgentSheets (1991)	event-based	이미지, 룰 기반	게임, 시뮬레이션 가능, 상용, 초-고
Stagecast (1995)	event-based	이미지, 룰 기반	게임, 애니메이션 가능, 상용, 초-중
Toontalk (1996)	procedural	애니메이션 기반	게임, 상용, 초-고
Viscuit 2.0 (2007)	event-based	이미지, 룰 기반	애니메이션, 그림책, 초-중
Dolittle (2001)	object-oriented	텍스트 기반	게임, 도형그리기, 로봇 프로그래밍, 초-고

#### 나. 스크래치

스크래치는 MIT 미디어 연구소의 Lifelong Kindergarten Group에서 개발한 교육용 프로그래밍 도구로, 만 8세에서 16세 사이 학생들의 Computational Thinking 능력 향상을 목표로 컴퓨터 프로그래밍 기술의 향상 뿐 아니라 새로운 IT 기술을 스스로 학습하고 이를 활용하여 창의적으로 문제를 해결하거나 프로젝트를 설계할 수 있는 능력 향상을 위해 설계되었다.

또한 스크래치는 시각적인 멀티미디어 개발환경을 제공함으로써 학습하기 어려운 프로그래밍 언어의 단점들을 해결할 수 있다. 이러한 스크래치의 특징은 다음과 같다(이영준, 이은경, 2008, p. 364).

##### 1) 블록 쌓기 프로그래밍 방식

스크래치로 프로그램을 만드는 방법은 매우 간단한 블록 쌓기 방식으로 이루어진다. 즉 프로그램의 각 수행 단계를 직접 코딩하는 것이 아니라, 다양한 색상으

로 구분된 블록들을 원하는 순서대로 가져다가 차곡차곡 쌓기만 하면 된다. 이 블록들은 문법적으로 오류가 없어야만 서로 결합될 수 있도록 설계되어 있어 구문에러가 발생하지 않는다.

또한 블록 내부에 자료 값을 지정할 때 자료형에 따라 서로 다른 형태의 블록 모양에 값이 지정되도록 구성되어 있기 때문에 잘못된 자료 입력에 의한 오류 발생이 방지된다. 따라서 원하는 블록을 마우스로 끌어다 놓기만 하면 누구나 쉽게 원하는 프로그램을 만들 수 있다.

## 2) 다양한 미디어 조작 기능

스크래치는 그래픽, 애니메이션, 소리, 음악과 같은 다양한 미디어들을 서로 혼합하고 제어하기 위한 프로그램을 쉽게 만들 수 있다. 따라서 스크래치를 활용하여 그래픽 프로그램에서 제공하는 필터링 효과나 슬라이드쇼를 위한 화면전환 효과 등을 직접 프로그래밍해 볼 수 있다.

## 3) 공유와 협력 학습 지원 기능

스크래치 공식 웹사이트(<http://scratch.mit.edu>)에서는 스크래치 사용자들을 위한 공유와 협력의 공간을 제공한다. 자신이 직접 만든 프로젝트를 웹사이트에 게시하거나 다른 사람들이 게시한 프로젝트를 실행해보거나 수정할 수 있으며, 의견을 제시할 수도 있다. 이밖에 교사들의 교수 학습을 지원하기 위한 사이트인 SCRATCHED(<http://scratched.media.mit.edu>)가 운영되고 있다.

## 2. 알고리즘

알고리즘이라는 단어는 페르시아 수학자 Abu Ja'far Mohammed ibn Musa al Khwarizimi(780-850)의 이름에서 유래되었다. 알고리즘의 원래 의미는 수학 문제 계산의 절차를 구체화해 놓은 것을 지칭하였다. 그러나 현재에는 성공이 보장되는 문제의 해결책을 통칭하여 광범위하게 알고리즘이라 지칭한다.

알고리즘의 정의에 대해서는 학자에 따라 표현 방식은 조금씩 다르나 개념적으로는 큰 의견 없이 '문제 해결 방법 또는 절차'라고 해석할 수 있으며, 컴퓨터 알고리즘이라는 용어는 '컴퓨터에서 수행이 가능한 문제 해결 방법이나 절차'로 정의할 수 있다. 이런 정의에 따라 주어진 문제에 대한 알고리즘을 찾는다는 것은

주어진 문제를 해결하는 방법을 찾는 것을 말하며, 문제를 해결하는 방법을 찾은 후에는 이를 사람이나 컴퓨터가 이해할 수 있는 명시적인 형태로 표현할 수 있어야 한다. 사람이 이해할 수 있는 형태의 표현 방법은 자연어, Flow Chart 등이 있으며, 컴퓨터를 이해할 수 있는 형태로 표현한 것을 프로그램이라고 한다(김경훈, 2006, p. 43-44).

### 3. 논리적 사고력

논리적 사고력은 비판적 사고의 협의의 개념으로 추리능력을 의미하며, 비판적 사고는 어떤 주장이나 정보에 대한 타당한 가치를 판단하기 위해 객관적으로 분석하는 능력으로 논리적 사고의 포괄적인 개념으로 볼 수 있다. 결국 논리적 사고력은 사상(event)들 간의 관계, 모순 등 일련의 규칙을 타당성(validity)에 그 준거를 두고 사고하는 추리능력이라 말할 수 있다.

논리적 사고의 하위요소는 Piaget 이론에 기반한 계열화논리, 비례논리, 확률논리, 변인통제논리, 조합논리, 명제논리 등을 말한다. 이들 하위요소는 다음과 같은 능력을 의미한다(이좌택, 2004, p. 18-24).

<표 II-2> 논리적 사고의 하위요소

하위요소	내용
계열화논리	어떤 사물의 크기, 무게, 부피의 증감에 따라 일련의 요소들을 규칙에 따라 정신적으로 배열하는 능력이다.
비례논리	두 비에 있어서 그 비의 값이 같다는 논리를 바탕으로 비례와 관련된 규칙 혹은 관계들 간의 관계(relation between relations)를 이해하는 능력이다.
확률논리	우연히 일어날 사건 중에서 어떤 한 사상이 일어날 확률을 계산할 수 있는 논리를 의미한다.
변인통제논리	어떤 문제에 직면했을 때, 상황의 모든 변인들을 인식하고 변인들의 역할에 관한 가설을 설정한 다음, 그 가설을 검증하기 위해 체계적으로 변인들을 통제하여 결론을 도출해 내는 능력이다.
조합논리	문제를 해결해 나가는 과정에서 있을 수 있는 여러 가지 경우를 빠짐없이 중복되지 않게 셀 수 있는 논리를 의미한다.
명제논리	명제는 참인지 거짓인지를 원칙적으로 분명히 판별할 수 있는 문장을 말한다. 명제논리는 명제와 명제의 결합 및 그 결합의 구조를 체계적으로 분석하는 능력이다.

#### 4. GALT (Group Assessment of Logical Thinking)

GALT는 1983년 미국 Georgia 대학의 Vantipa Roadranka, Russell H. Yeany, Michael J. Padilla가 공동으로 개발한 논리적 사고력 검사지이다.

총 문항은 21문항이고 각 문항은 답과 이유를 선택하는 선다형 형식으로 6개의 논리유형을 측정하도록 구성되었다. 보존논리 4문항, 비례논리 6문항, 변인통제논리 4문항, 확률논리 2문항, 상관논리 2문항, 조합논리 3문항이고 대부분의 문항에 문제 상황을 설명하는 짧은 문장과 함께 그림을 제시하였고 초등학교 4학년 이상의 학생들에게 1시간의 수업 시간 동안 집단으로 적용할 수 있다(김경미, 1999, p. 21-23).

또한 GALT 개발자들은 시간 또는 다른 제약이 있을 경우나 논리 유형별로 비교하고자 할 경우, 각 논리 유형별로 2문항씩을 뽑아 총 12문항으로 구성된 축소본 GALT의 사용을 제안하였다. 축소본 GALT의 인지 수준 구분은 0점~4점을 구체적 조작기, 5점~7점을 과도기, 8점~12점을 형식적 조작기로 분류한다(Roadranka 등, 1983).

축소본 GALT의 문항별 측정내용은 다음과 같다(한국교육개발원, 2002, p. 61).

<표 II-3> 축소본 GALT 문항별 측정내용

하위요소	내용	문항번호
보존논리	질량(무게), 부피에 대한 보존 능력	1, 2
비례논리	서로 다른 크기의 물 컵 간의 관계나 균형을 잡기 위한 길이와 무게 간의 관계를 비율로 생각할 수 있는 능력	3, 4
변인통제논리	막대 길이, 추의 무게, 추의 왕복 이동 시간과의 관계 등을 따질 수 있는 능력	5, 6
확률논리	여러 종류의 나무토막 중 제시된 종류의 나무토막을 선택할 수 있는 가능성을 따질 수 있는 능력	7, 8
상관논리	색과 크기 사이의 상관을 추리할 수 있는 능력	9, 10
조합논리	학생들끼리 짝 지을 수 있는 방법 또는 장을 보기 위해 상점을 방문할 수 있는 가능한 방법이나 순서 등을 추리할 수 있는 능력	11, 12

### Ⅲ. 교재 개발

#### 1. 교육과정 분석 및 주제선정

##### 가. 교육과정 분석

컴퓨터 알고리즘 관련 교육 내용을 컴퓨터 관련 교과 교육과정에서 찾고자 하였으나 현행 2007년 개정 교육과정을 살펴보면 초등학교 정보통신기술교육의 내용 체계에는 알고리즘과 관련된 내용이 제시되어 있지 않고 중학교의 정보 교과에 알고리즘 관련 교육 내용이 제시되어 있다.

그러므로 본 연구에서는 해외 및 우리나라 중학교 정보 교과 교육과정을 분석하여 작성한 알고리즘 교육 내용 체계표에서 교육 내용을 추출하였다. 알고리즘 교육 내용 체계표의 내용은 다음과 같다.

<표 Ⅲ-1> 알고리즘 교육 내용 체계표

주제	학습 내용	비고
이진법	실세계의 정보를 컴퓨터에서 표현하는 방법과 원리	자료구조
스택/큐	나열된 자료간의 순서를 제어하는 원리	
트리	복잡한 자료를 계층 구조로 표현하는 방법의 장점과 원리	
그래프	자료간의 복잡한 연결 관계를 도식화하는 방법과 탐색 원리	
정렬	많은 자료 중에서 원하는 것을 기준에 따라 자료를 정렬하는 다양한 원리	알고리즘
탐색	원하는 자료를 찾아가는 방법과 원리	
그리디	선택 순간마다 최적을 취하는 방법의 장·단점	
분할 정복	범위가 큰 문제는 작은 문제로 나누어 해결하는 방법과 동작 원리	
백트래킹	문제 해결 과정에서 해답을 찾지 못하면 이전 과정으로 돌아가면서 해답을 찾아가는 원리	

##### 나. 주제선정

본 연구에서는 알고리즘의 내용을 스크래치 프로그래밍을 통해 익힘으로써 논리적 사고력을 신장시키고자 한다. 따라서 다음과 같은 기준으로 알고리즘 교육 내용을 선정하였다.



첫째, 스크래치 프로그래밍으로 쉽게 구현될 수 있는 내용이어야 한다. 교육용 프로그래밍 교육의 문제점으로 제시되는 것 중 하나가 도구로 제시되는 프로그래밍 언어를 배우는데 많은 시간이 필요하다는 점이다. 본 연구에서도 프로그래밍 교육보다는 알고리즘 교육에 초점을 맞추고 있으므로 학습자들에게 즉각적이고 구체적으로 전달 가능하도록 쉽게 구현되는 내용이어야 한다.

둘째, 실생활에서 쉽게 접할 수 있는 내용이어야 한다. 알고리즘의 원리가 실생활에서 쉽게 찾을 수 있거나 적용할 수 있는 내용이어야 학습자의 내적 동기를 높일 수 있으며 알고리즘의 원리에 대한 인지수준도 높아질 것이다.

셋째, 논리적 사고력의 향상에 효과적이어야 한다. 스크래치 프로그래밍을 통해 학습하게 되는 알고리즘 원리는 학습자의 사고 능력 향상에 도움을 주어야 한다.

넷째, 효과적인 알고리즘 학습을 위해 필요한 자료 구조의 내용도 선정한다. 이상의 선정 기준을 바탕으로 총 8차시의 구성으로 선정한 교육 내용은 다음과 같다.

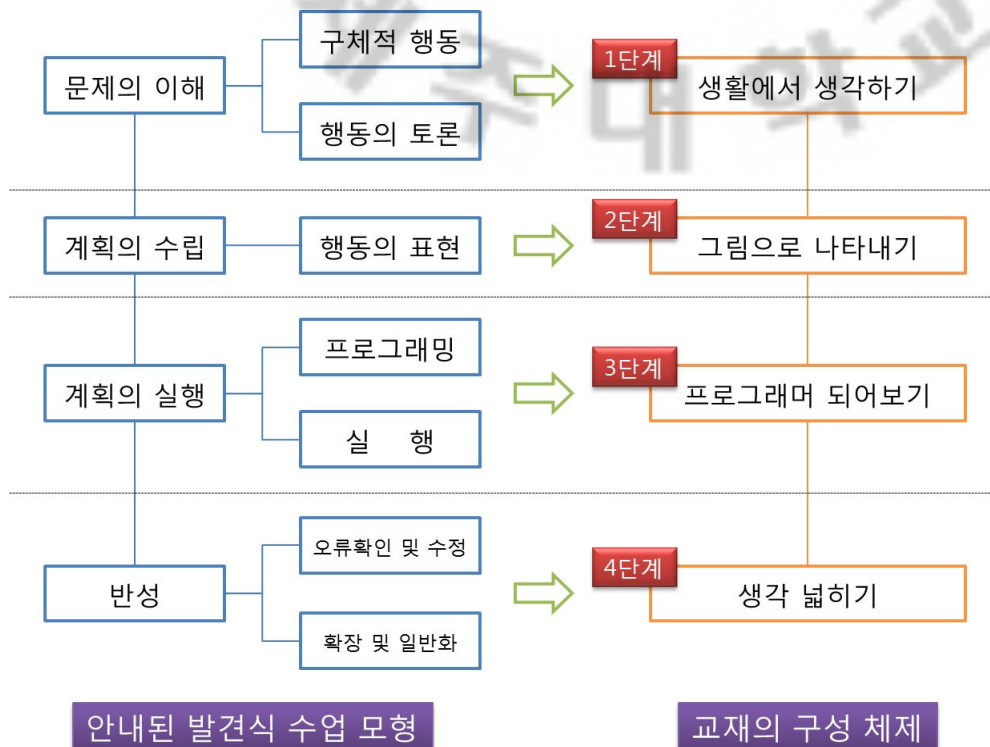
<표 III-2> 차시별 교육 주제

차시	주제	교육 내용
1	스택	스택을 이해하고 스크래치로 스택 표현하기
2	큐	큐를 이해하고 스크래치로 큐 표현하기
3	트리	트리 구조를 이해하고 스크래치로 이진트리 표현하기
4	탐색	탐색의 방법 중 선형탐색의 방법을 이해하고 스크래치로 구현하기
5	정렬	정렬의 방법 중 버블정렬의 방법을 이해하고 스크래치로 구현하기
6	분할 정복	분할 정복의 방법을 사용한 이진탐색의 방법을 이해하고 스크래치 구현하기
7	그리디	그리디를 이해하고 거스름돈 문제 스크래치로 구현하기, 그리디를 사용한 다른 문제 확인하기
8	백트래킹	백트래킹의 방법을 이해하고 4-여왕말 문제 풀이를 스크래치로 구현하기

## 2. 교재의 구성 체제

LOGO 프로그래밍의 창시자인 Papert(1980)는 “교육과정 없는 학습(learning without curriculum)”, “교수 없는 학습(learning without teaching)”을 주장하며 구조화가 적게 된 수업방법일 수록 학습자에게 더 많은 것을 제공한다고 했다. 하지만 실제로 학생들은 구조화가 되지 않은 수업에서 방향을 하는 경우가 많다. 따라서 본 연구에서는 안내된 발견식 수업(Guided Discovery Teaching Method)을 기본으로 교재를 구성하여 교사의 역할을 최소화하고 학생들이 능동적으로 프로그래밍 활동에 참여할 수 있도록 하였다. 이런 안내된 발견식 수업은 학생들이 의식적으로 지식, 정보 또는 학습전략을 습득해 나가도록 돕고, 습득된 지식 및 전략을 다른 새로운 문제 상황에 전이될 수 있도록 해준다(김경미, 2004, p. 75).

안내된 발견식 수업절차(백영균 등, 1994, p. 77)에 따른 교재의 구성 체제를 도식화 하면 다음과 같다.



[그림 III-2] 안내된 발견식 수업절차와 교재의 구성 체제

1단계 생활에서 생각하기는 우리주변에서 이번 시간에 배울 알고리즘(학습주제)의 내용과 관련된 것을 찾아보고 그 알고리즘에 대해 구체적으로 이해하는 단계이다. 실생활의 한 장면을 이용하여 학생들이 쉽게 이해할 수 있도록 한다.

2단계 그림으로 나타내기는 알고리즘의 내용을 직접 프로그래밍 하여 구현하기 전 단계로서 알고리즘의 실행 동작을 그림으로 그려보는 단계이다. 직접 그려봄으로써 알고리즘에 대해 확실히 이해할 수 있으며, 프로그램을 구성하는 명령어에 대해서도 생각해볼 수 있는 기회를 제공한다.

3단계 프로그래머 되어보기는 스크래치의 명령어 블록들을 조합하여 직접 프로그램을 작성해보는 단계이다. 이때 교사는 처음부터 직접 시범을 보이며 지시하는 것이 아니라 학생들과의 질의응답을 통해 올바른 방향으로 프로그래밍 할 수 있도록 지도한다.

4단계 생각 넓히기에서는 다른 방법으로 프로그래밍 한 자료를 자신이 프로그래밍 한 것과 비교해 보며 서로 장단점을 알아보는 과정이다. 이 과정을 통해 더욱 활동에 적극적으로 참여할 수 있는 내적 동기를 유발할 수 있고, 보다 쉬운 프로그래밍 소스에 대해 이해할 수 있다. 또한 이 단계에서는 범위를 더욱 확장하여 실생활 속에서 알고리즘에 대한 내용을 찾아보는 활동을 한다.

### 3. 교재의 실제



# 스크래치를 통한 알고리즘 배우기

## ◆ 첫 번째 알고리즘 ◆

### 스택(stack)

:: 스택을 이해하고 프로그램 짜기 ::

#### 1단계

#### 생활에서 생각하기



수정이는 수납상자에 옷을 잘 정리하여 오른쪽 그림과 같이 차곡차곡 쌓았습니다. 먼저 점퍼를 넣고 그 다음 바지, 재킷, 코트의 순으로 잘 개어서 정리하였습니다. 그러던 어느 날 수정이는 바지가 필요해서 수납상자에서 바지를 꺼내게 되었습니다. 바지를 꺼내기 위해서는 무엇을 먼저 꺼내어야 하나요?

코트
재킷
바지
점퍼

▶ 수납상자에 정리할 때 수납상자에 들어간 순서대로 옷을 차례로 써 봅시다.

▶ 바지를 꺼내기 위해 수납상자에서 꺼내야 할 옷을 순서대로 써 봅시다.

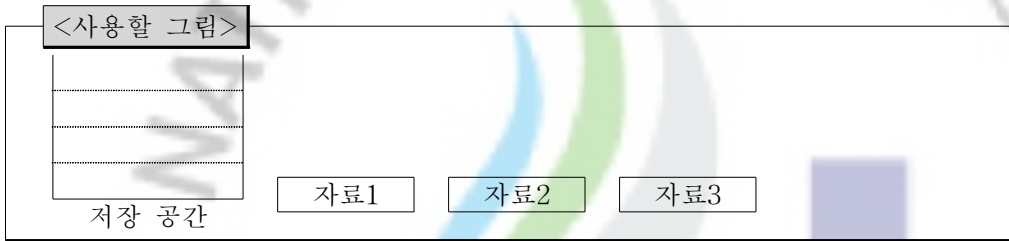
▶ 수납상자에 넣을 때와 꺼낼 때를 서로 비교하여 이야기해 봅시다.

위와 같이 들어가는 곳(삽입)과 나가는 곳(삭제)이 같아서 가장 최근에 들어간 것이 가장 먼저 나오는 구조를 **스택**이라고 합니다. 컴퓨터에서는 자료를 저장할 때 이런 스택구조를 사용하는데 위에서의 수납상자를 컴퓨터의 저장 공간, 옷을 자료라고 생각하면 이해하기 쉽습니다.

#### 2단계

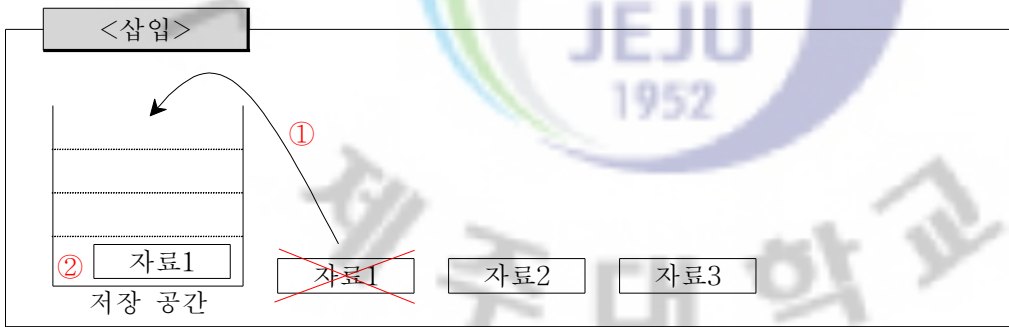
#### 그림으로 나타내기

▶ 스택구조에서 자료를 삽입(입력)하고 삭제(출력)하는 과정을 다음에 제시된 그림으로 나타내어 봅시다.



위와 같이 4개의 자료를 저장할 수 있는 컴퓨터의 저장 공간에 3개의 자료(자료1, 2, 3)를 차례로 삽입하고 삭제하는 과정을 그림으로 나타내어 봅시다.

먼저 삽입과정을 그려보면 자료1을 삽입하면(①) 저장 공간 맨 아래에 자료1이 저장됩니다(②). 자료2, 자료3이 삽입되는 과정도 그림으로 나타내어 봅시다.



다음은 저장 공간에 저장된 자료를 차례대로 삭제하는 과정을 그림으로 나타내어 봅시다.

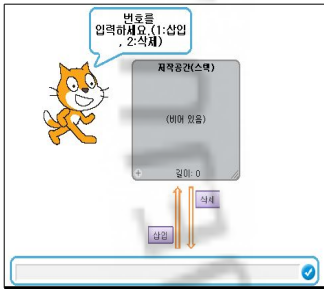


▶ 스크래치를 통해 스택구조를 프로그램으로 만들 때 필요한 명령어를 생각나는 대로 써 봅시다.

3단계

프로그래머 되어보기

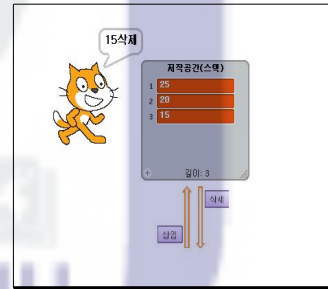
▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<삽입, 삭제 선택>



<삽입 과정>



<삭제 과정>

<명령어 블록>



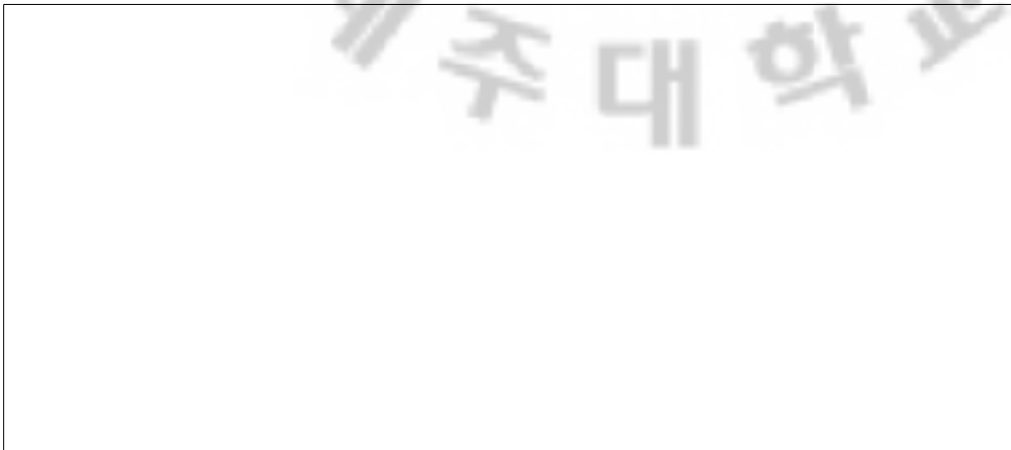
4단계

생각 넓히기

▶ 다른 방법으로 작성한 것을 보고 자신이 작성한 것과 비교하여 생각을 적어 봅시다.



▶ [과제] 주위에서 스택구조와 같은 것을 찾아 적어 봅시다.





# 스크래치를 통한 알고리즘 배우기

## ◆ 두 번째 알고리즘 ◆

### 큐(queue)

:: 큐를 이해하고 프로그램 짜기 ::

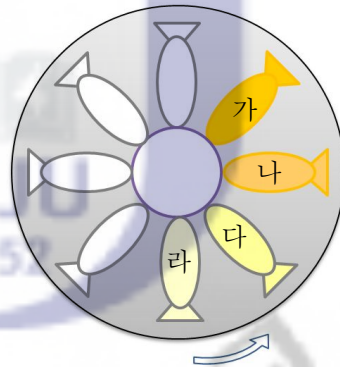
#### 1단계

#### 생활에서 생각하기



붕어빵이 생각나는 계절이 다가오고 있습니다. 기영이는 붕어빵이 먹고 싶어서 붕어빵 가게에 갔더니, 이제야 반죽을 붕어빵 기계에 차례로 넣고 있었습니다. 기영이는 기다리면서 이 과정을 자세히 관찰하였습니다.

기영이가 먼저 받게 될 붕어빵은 어느 것인가요?



▶ 위 그림을 보며 반죽이 붕어빵 기계에 들어간 순서대로 써 봅시다.

▶ 먼저 익어서 나오게 될 붕어빵의 순서를 써 봅시다.

▶ 반죽이 들어간 순서와 붕어빵이 되어 나오는 순서 사이의 관계를 이야기해 봅시다. 그리고 스택과 비교하여 생각을 써 봅시다.

위와 같이 스택과는 다르게 들어가는 곳(삽입)과 나가는 곳(삭제)이 달라서 가장 먼저 들어간 것이 가장 먼저 나오는 구조를 **큐**라고 합니다. 이렇게 들어간 차례대로 처리되는 구조를 컴퓨터에서 사용하는데 위에서의 붕어빵 기계를 컴퓨터의 저장 공간, 반죽을 자료라고 생각하면 이해하기 쉽습니다.



2단계

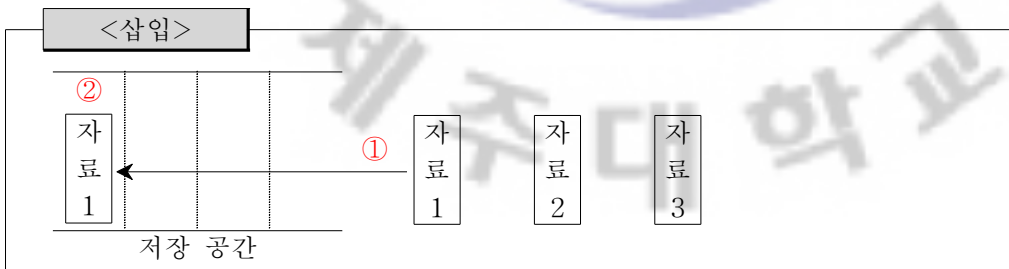
그림으로 나타내기

▶ 큐 구조에서 자료를 삽입(입력)하고 삭제(출력)하는 과정을 다음에 제시된 그림으로 나타내어 봅시다.



위와 같이 4개의 자료를 저장할 수 있는 컴퓨터의 저장 공간에 3개의 자료(자료1, 2, 3)를 차례로 삽입하고 삭제하는 과정을 그림으로 나타내어 봅시다.

먼저 삽입과정을 그려보면 자료1을 삽입하면(①) 저장 공간 맨 처음에 자료1이 저장됩니다(②). 자료2, 자료3이 삽입되는 과정도 그림으로 나타내어 봅시다.



다음은 저장 공간에 저장된 자료를 차례대로 삭제하는 과정을 그림으로 나타내어 봅시다.

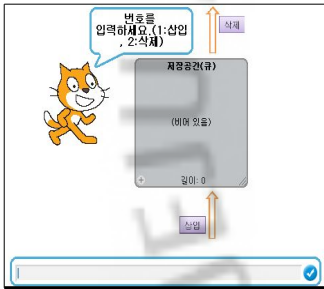


▶ 스크래치를 통해 스택구조를 프로그램으로 만들 때 필요한 명령어를 생각나는 대로 써 봅시다.

3단계

프로그래머 되어보기

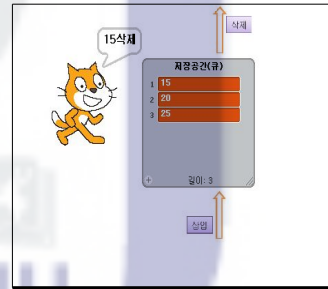
▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<삽입, 삭제 선택>



<삽입 과정>



<삭제 과정>

<명령어 블록>

Blank area for drawing command blocks.

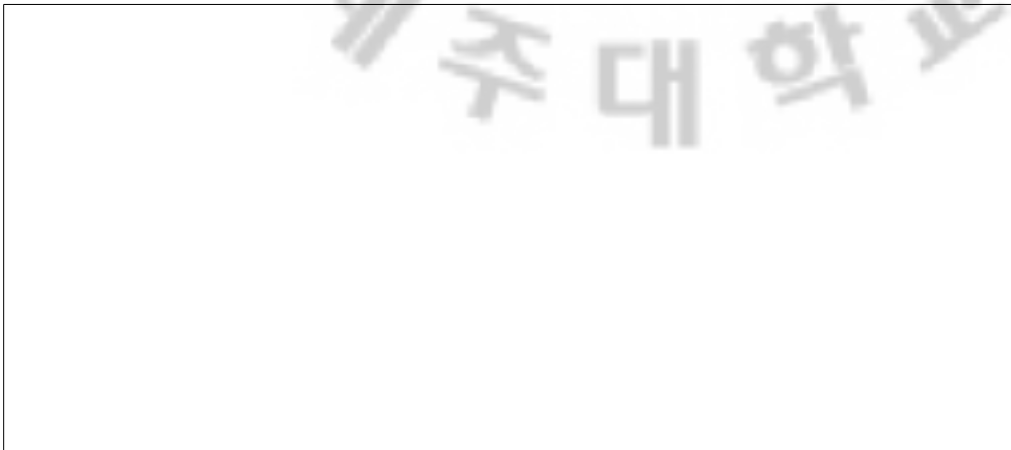
4단계

생각 넓히기

▶ 다른 방법으로 작성한 것을 보고 자신이 작성한 것과 비교하여 생각을 적어 봅시다.



▶ [과제] 주위에서 큐 구조와 같은 것을 찾아 적어 봅시다.





# 스크래치를 통한 알고리즘 배우기

## ◆ 세 번째 알고리즘 ◆

### 트리(tree)

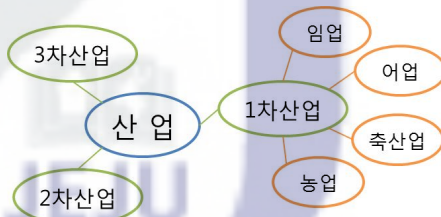
:: 이진트리를 이해하고 표현하기 ::

#### 1단계

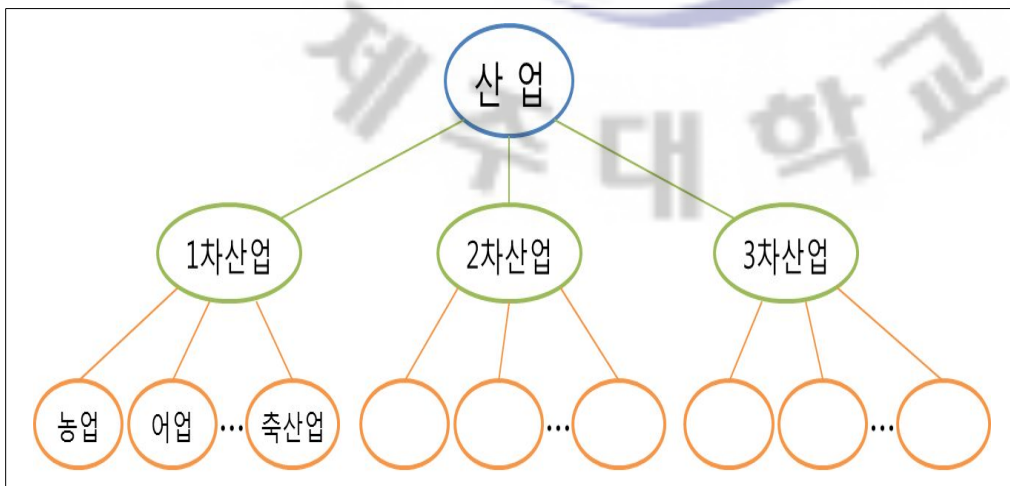
#### 생활에서 생각하기



서현이는 사회시간에 배운 산업의 종류를 오른쪽 그림과 같이 마인드맵 형식으로 나타내 보았습니다.



▶ 위 마인드맵을 아래와 같이 바꾸고 빈 내용을 채워봅시다.

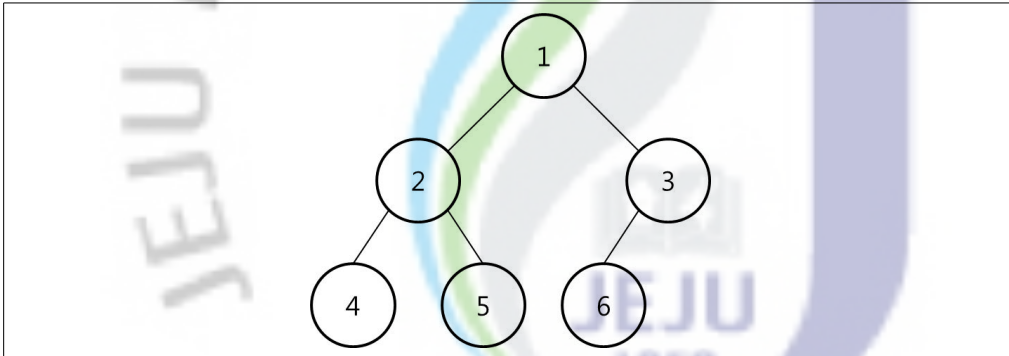


위의 그림은 하나의 줄기에서 여러 가지가 나온 것처럼 표현되어 있습니다. 이처럼 나무를 뒤집어 놓은 모습으로 자료들 사이의 구조를 표현한 것을 **트리(tree)**라고 합니다. 결국 위의 그림은 '산업의 종류'를 트리 구조로 나타낸 것인데, 트리 구조에서 각각의 원을 노드라고 하고, 노드와 노드를 연결한 선을 링크라고 합니다. 특히 가장 위에 하나만 있는 노드를 루트노드라고 하고, 가장 아래 위치한 노드를 단말노드 또는 리프노드라고 합니다. 위의 트리구조에서는 '산업'이 루트노드이고, '농업', '어업' 등이 단말노드가 됩니다. 이번시간에는 트리 중에서 가장 많이 사용되는 **이진트리**에 대해 알아보도록 합시다.

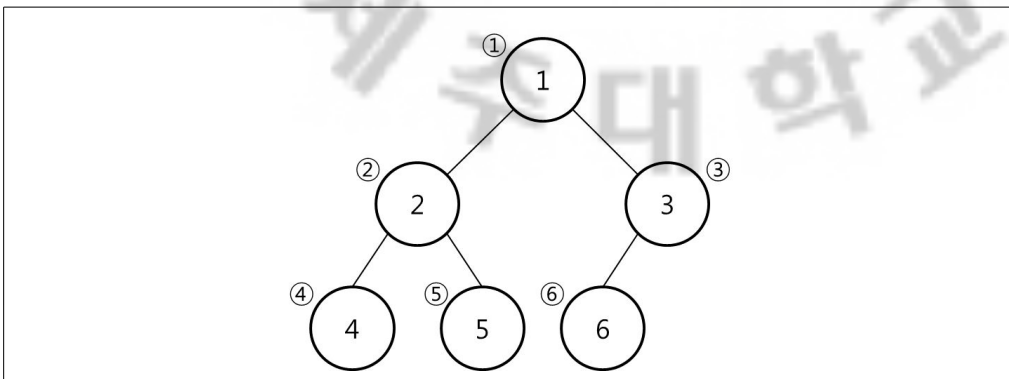
2단계

그림으로 나타내기

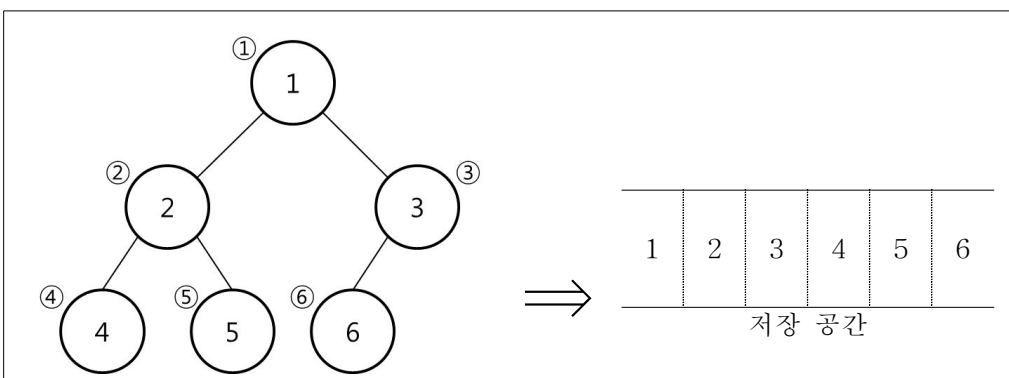
▶ 이진트리는 노드에 연결된 노드가 두 개 이하인 트리를 말하는데, 다음 그림이 이진트리의 예입니다.



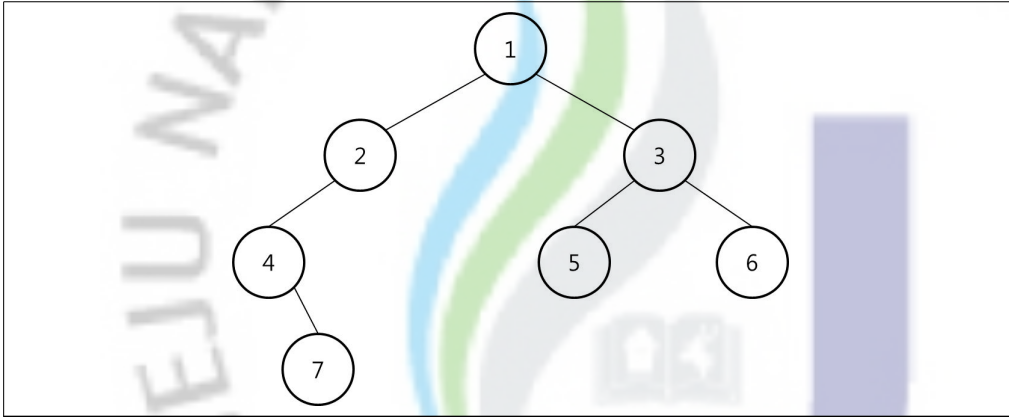
위의 이진트리가 저장 공간에 저장되는 방법을 그림을 통해 알아보시다. 먼저 루트 노드부터 시작해서 아래로 그리고 왼쪽, 오른쪽 순서로 번호를 붙여봅시다.



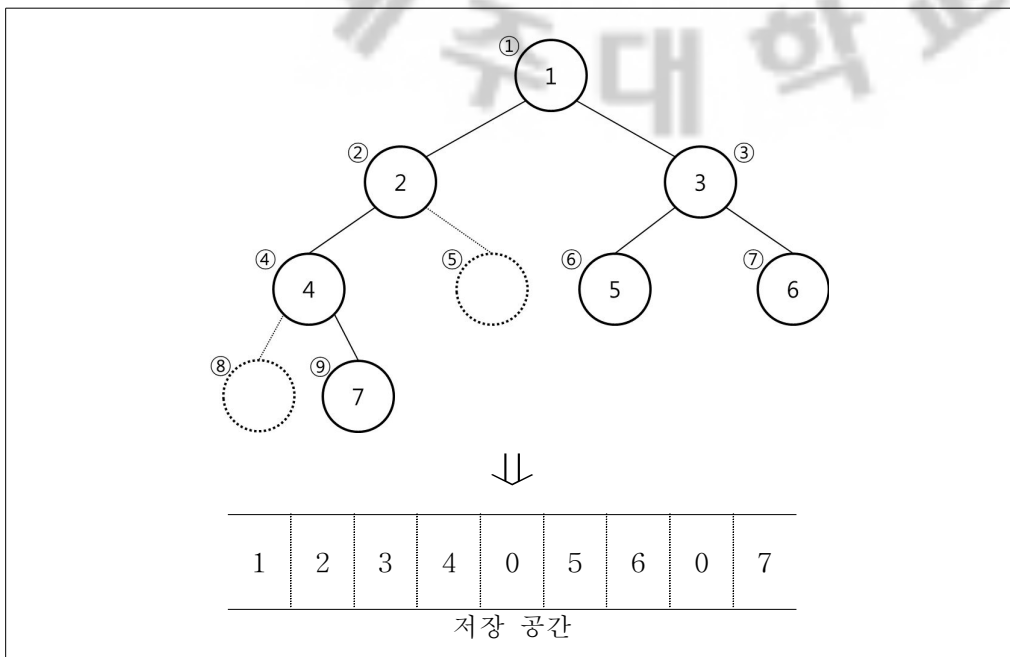
그리고 나서 번호 순서대로 저장 공간에 넣어주면 됩니다.



다른 예를 통해 더 알아보시다. 다음 이진트리가 저장되는 과정을 살펴봅시다.

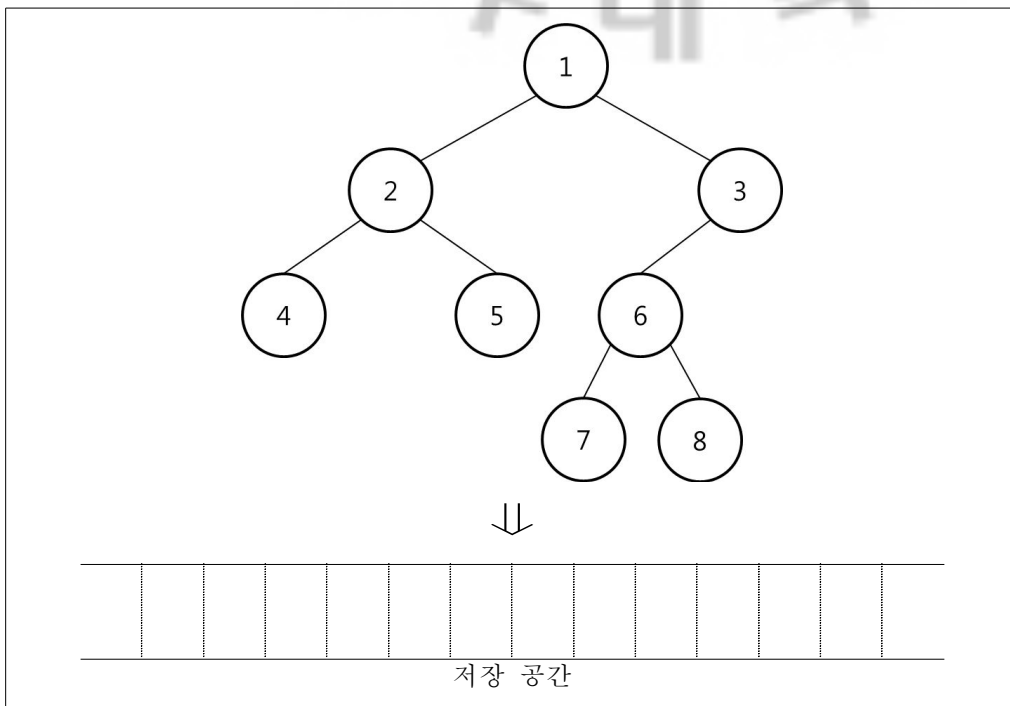
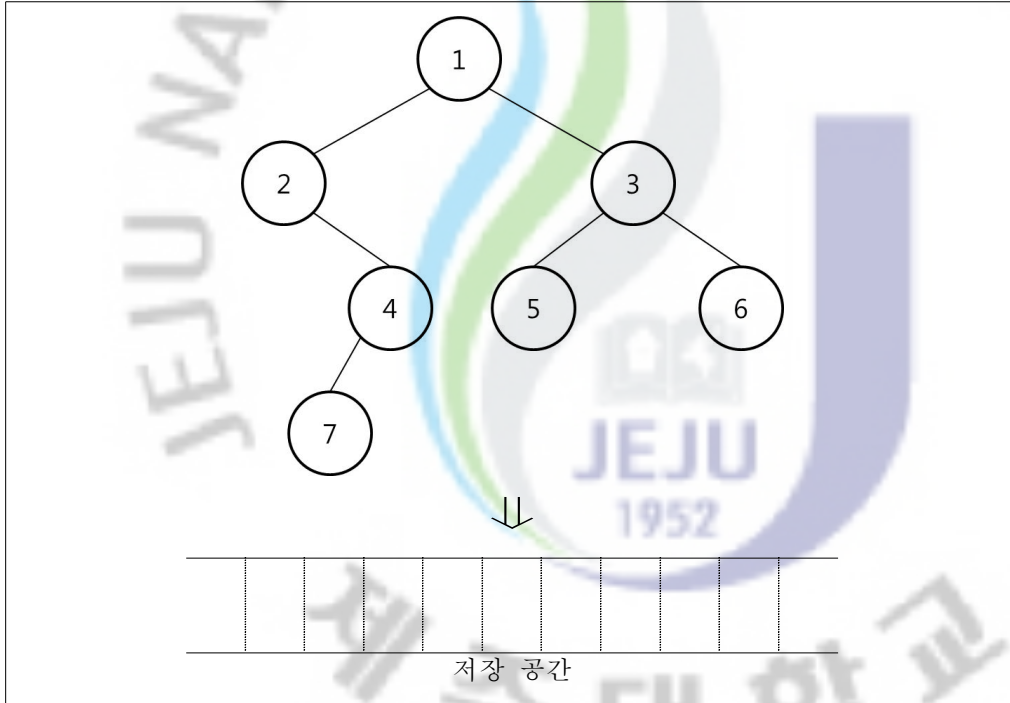


이번에도 순서대로 번호를 붙여줍니다. 단, 이번에는 앞에서 살펴본 이진트리와 차이점이 있습니다. 2에 연결된 노드가 왼쪽에 4밖에 없는데, 오른쪽에도 노드가 있다고 생각하고 번호를 붙여주어야 합니다. 마찬가지로 4의 왼쪽에도 노드가 있다고 생각하고 번호를 붙여줍니다. 그리고 나서 저장 공간에 차례로 넣어주는데, 빈 곳은 0을 넣어줍니다. 그림으로 나타내면 다음과 같습니다.



이렇게 저장하는 이유는 이진트리의 장점 중 하나인 자료의 삽입 때문입니다. 만약 ⑤의 위치에 8을 넣고 싶다면 기존에 저장된 자료의 변화 없이 바로 저장 공간의 5번째 위치에 8을 넣으면 되기 때문입니다. 이렇게 하면 쉽게 자료를 삽입할 수 있습니다.

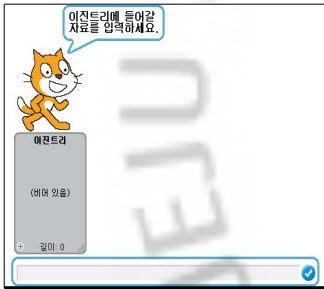
▶ 다음 이진트리가 저장되는 과정을 그림으로 나타내 봅시다.



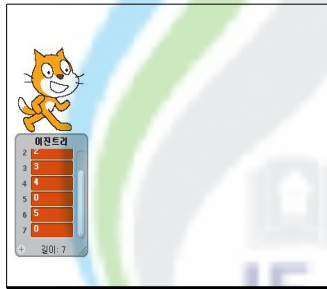
3단계

프로그래머 되어보기

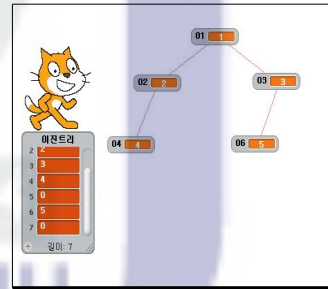
▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<자료 입력>



<저장 완료>



<이진트리 그리기>

<명령어 블록>





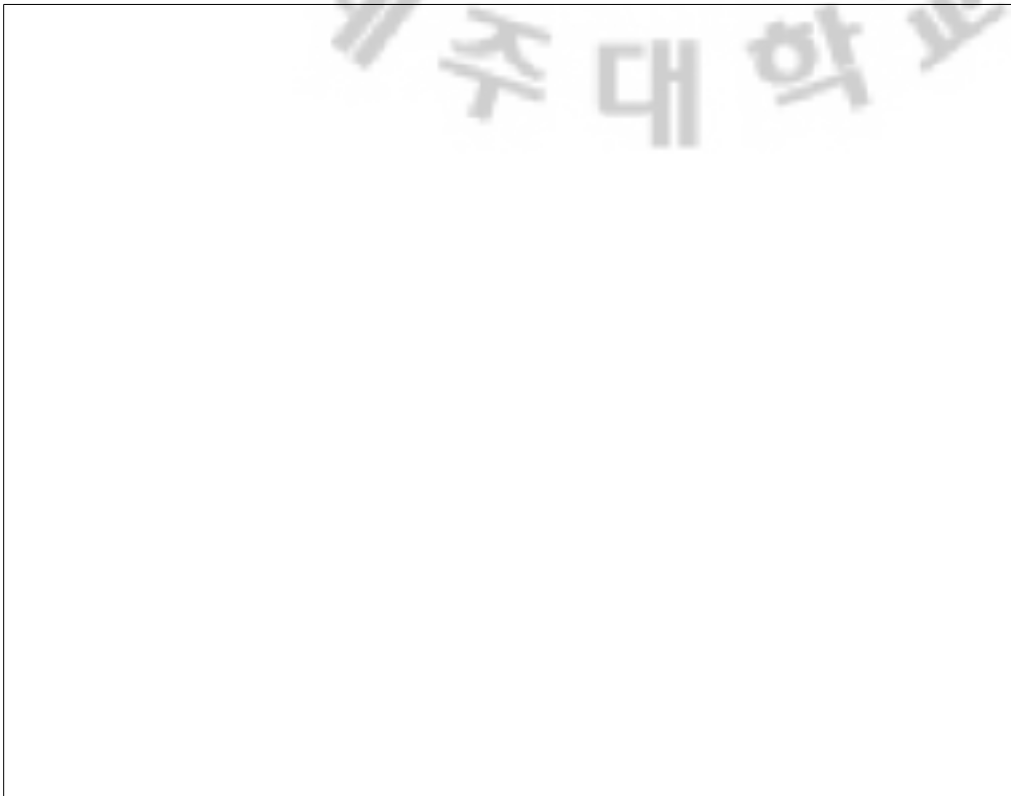
4단계

생각 넓히기

- ▶ 링크를 그리는 명령어 블록을 보고 링크가 그려지는 과정을 생각해 봅시다.



- ▶ [과제] 이진트리 구조를 더 늘려서 프로그램을 작성해 봅시다.





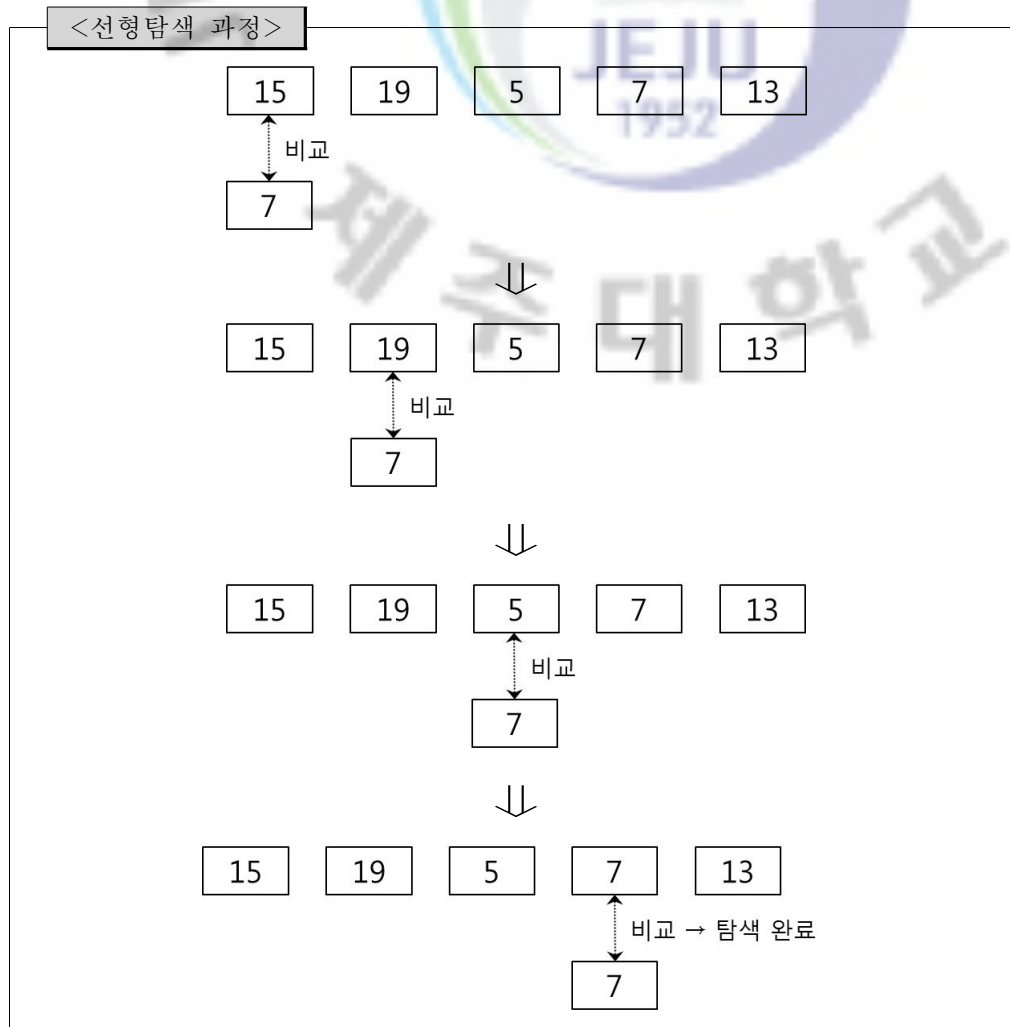
2단계

그림으로 나타내기

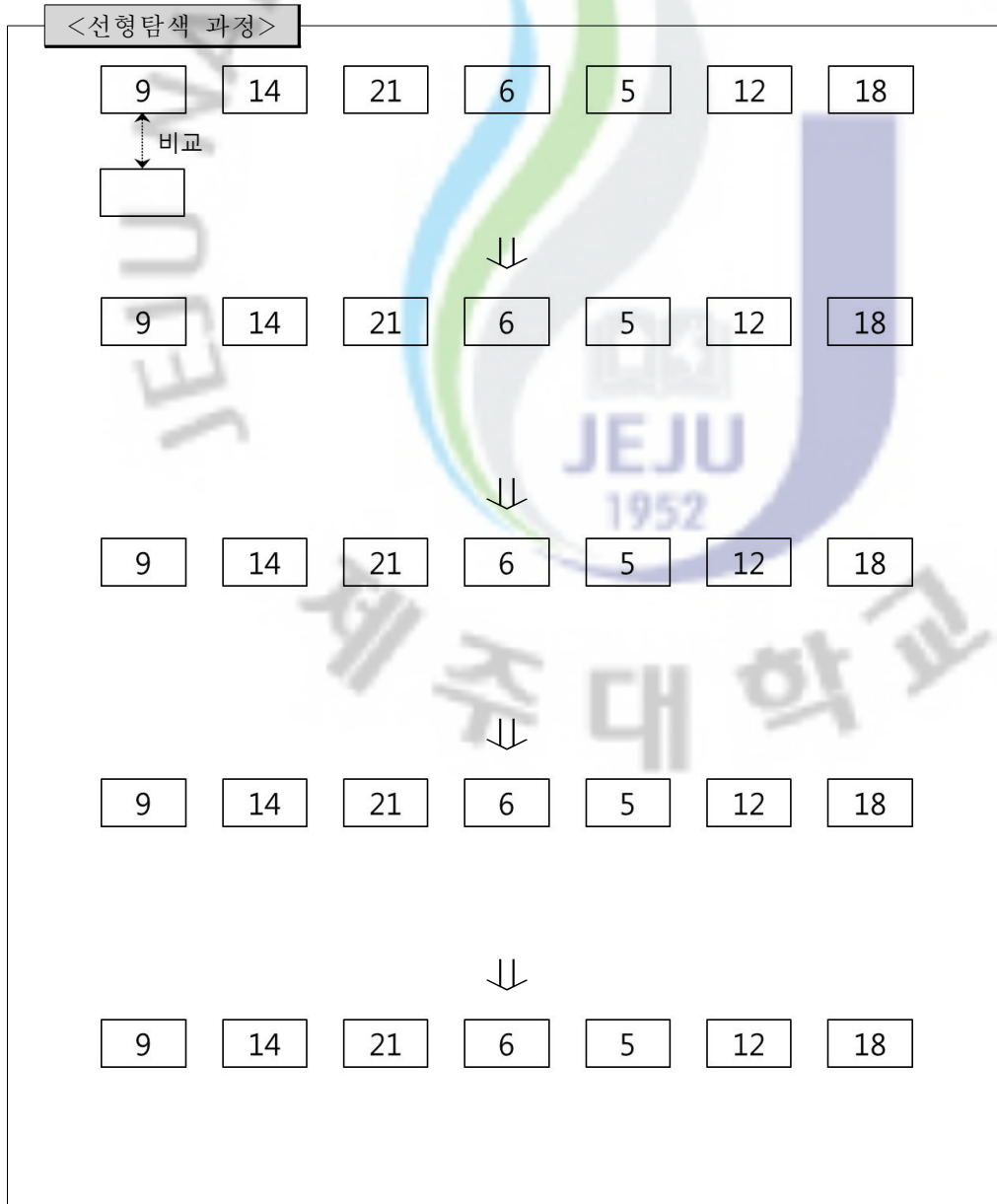
▶ 다음 5개의 자료 중에 7을 선형탐색을 통해 찾는 과정을 알아봅시다.



먼저 첫 번째 자료와 7을 비교하는데, 다르므로 다음으로 이동합니다. 이번에는 두 번째 자료와 7을 비교하는데, 다르므로 다음으로 이동합니다. 이런 방법으로 계속하여 비교하면 네 번째 자료에 있는 7을 찾고, 탐색을 끝낼 수 있습니다. 만약 찾는 자료가 없으면 탐색 실패로 끝냅니다. 이것을 그림으로 나타내면 다음과 같습니다.



▶ 다음 7개의 자료 중에서 선형탐색을 통해 5를 찾는 과정을 그림으로 나타내 봅시다.



이렇게 자료를 처음부터 순서대로 비교하며 찾는 탐색방법이 선형탐색입니다.

3단계

프로그래머 되어보기

▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<자료 입력>



<탐색값 입력>



<탐색 완료>

<명령어 블록>



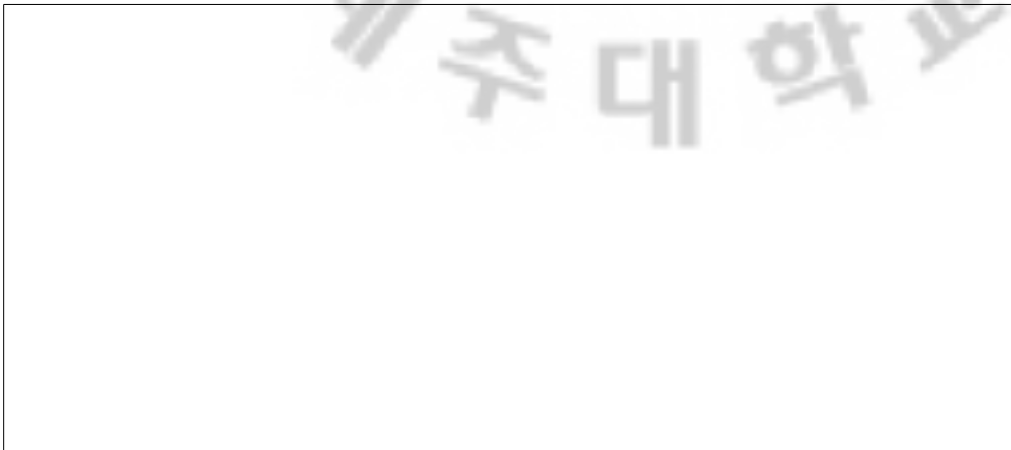
4단계

생각 넓히기

▶ 다른 방법으로 작성한 것을 보고 자신이 작성한 것과 비교하여 생각을 적어 봅시다.



▶ [과제] 다른 탐색 방법을 생각해 봅시다.





# 스크래치를 통한 알고리즘 배우기

## ◆ 다섯 번째 알고리즘 ◆

### 정렬(sort)

:: 버블정렬 프로그램 짜기 ::

#### 1단계

#### 생활에서 생각하기



영수와 친구들은 100m 달리기를 하고 기록을 측정하였습니다. 그 결과는 오른쪽 표와 같습니다. 영수는 친구들과 사이에서 몇 번째로 빠른가요? 기록을 순서대로 나열하면 어떻게 될까요?

이름	기록
수정	22초
기영	18초
영수	19초
상현	20초
서현	17초

▶ 위 표를 보며 기록이 낮은 것부터 높은 것까지 순서대로 나열해 봅시다.

▶ 영수는 몇 번째입니까?

위와 같이 주어진 자료를 일정한 기준으로 순서대로 나열하는 것을 **정렬**이라고 합니다. 위의 예처럼 자료가 적으면 누구나 쉽게 할 수 있지만, 자료의 양이 많으면 많을수록 사람이 처리하기가 어렵습니다. 그래서 처리 속도가 빠른 컴퓨터가 대신 자료를 정렬할 수 있도록 다양한 방법들을 생각하게 되었는데, 대표적인 정렬방법으로는 버블정렬, 선택정렬, 퀵(quick)정렬, 합병정렬 등이 있습니다. 이번 시간에는 이 중에서 **버블정렬**의 방법을 알아보시다.

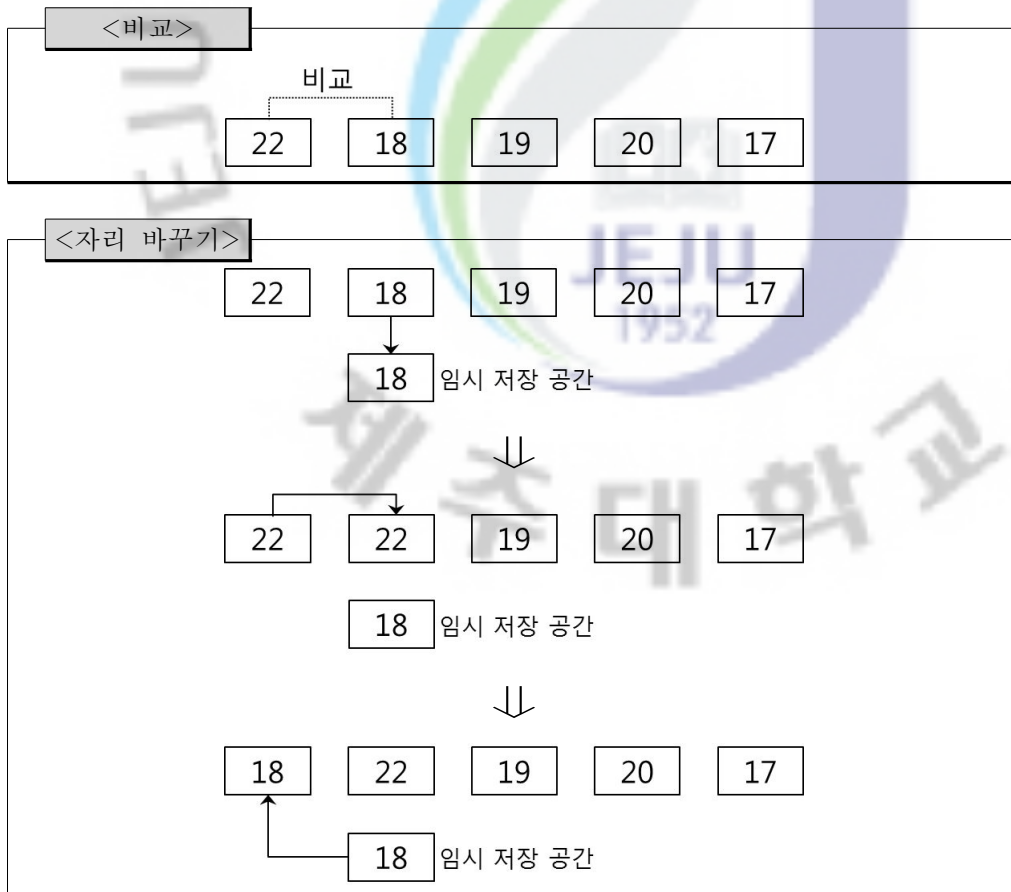
#### 2단계

#### 그림으로 나타내기

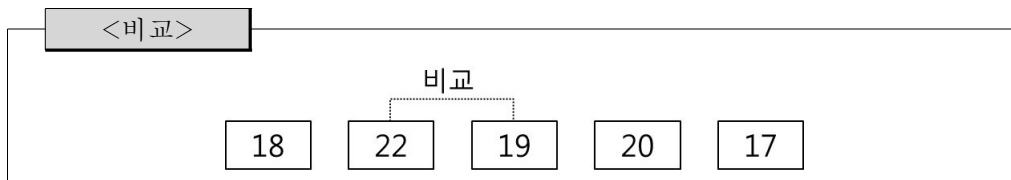
▶ 버블정렬은 자료를 서로 비교하여 가장 큰 자료를 가장 뒤로 보내며 정렬하는 방법입니다. 다음 5개의 자료를 버블정렬의 방법으로 정렬해 봅시다.

22	18	19	20	17
----	----	----	----	----

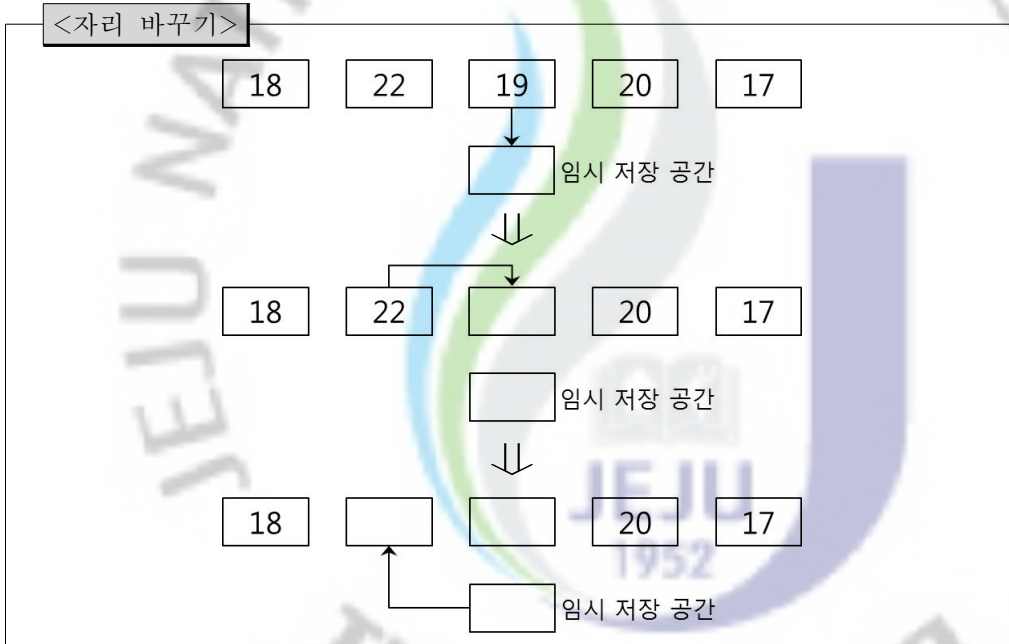
먼저 첫 번째와 두 번째 자료를 비교하고 큰 자료를 뒤로 보냅니다. 22와 18을 비교하면 22가 크기 때문에 22와 18의 자리를 서로 바꿉니다. 자리를 바꾸기 위해서는 임시 저장 공간이 필요합니다. 컴퓨터는 복사를 하면 기존 자료를 지우고 새로운 자료로 바꿉니다. 그렇기 때문에 서로 자리를 바꾸기 위해서는 하나의 자료를 임시 저장 공간에 저장 한 후 자료를 복사해야 합니다. 다음 그림으로 자세히 살펴봅시다.



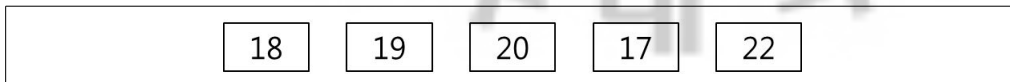
이번에는 두 번째와 세 번째 자료를 비교하여 큰 자료를 뒤로 보냅니다. 22와 19를 비교하면 22가 크기 때문에 22와 19의 자리를 서로 바꿉니다. 그림으로 나타내 봅시다.



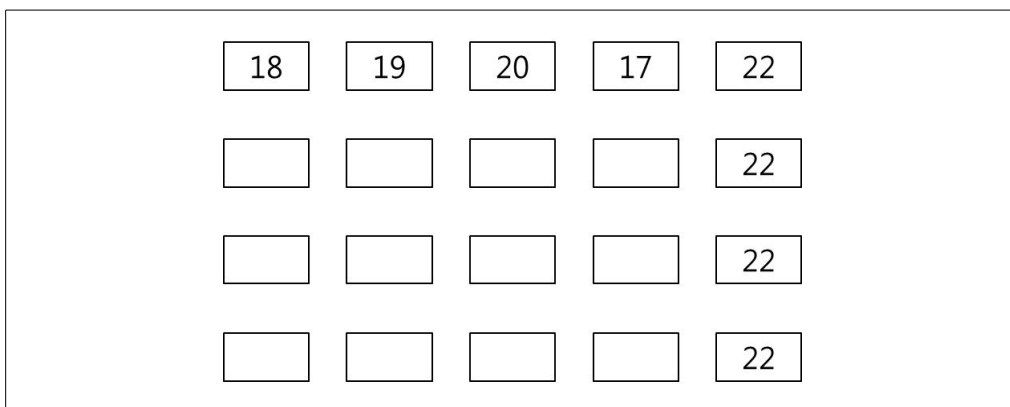




계속해서 세 번째와 네 번째, 네 번째와 다섯 번째를 비교하고 큰 자료를 뒤로 보내면 다음과 같이 5개의 자료 중 가장 큰 자료인 22가 가장 뒤에 위치하게 됩니다.



이제 다시 처음부터 같은 방법으로 비교하며 큰 자료를 뒤로 보냅니다. 이번에는 가장 큰 자료가 다섯 번째에 위치해 있기 때문에 앞의 4개의 자료만 비교합니다. 그림을 통해 과정을 나타내 봅시다.

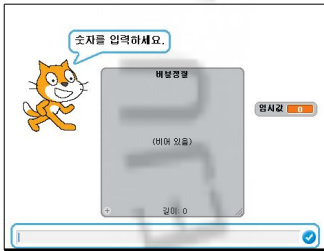


이렇게 큰 자료를 뒤로 보내며 정렬하는 방법이 버블정렬입니다.

3단계

프로그래머 되어보기

▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<자료 입력>



<정렬 과정>



<정렬 완료>

<명령어 블록>



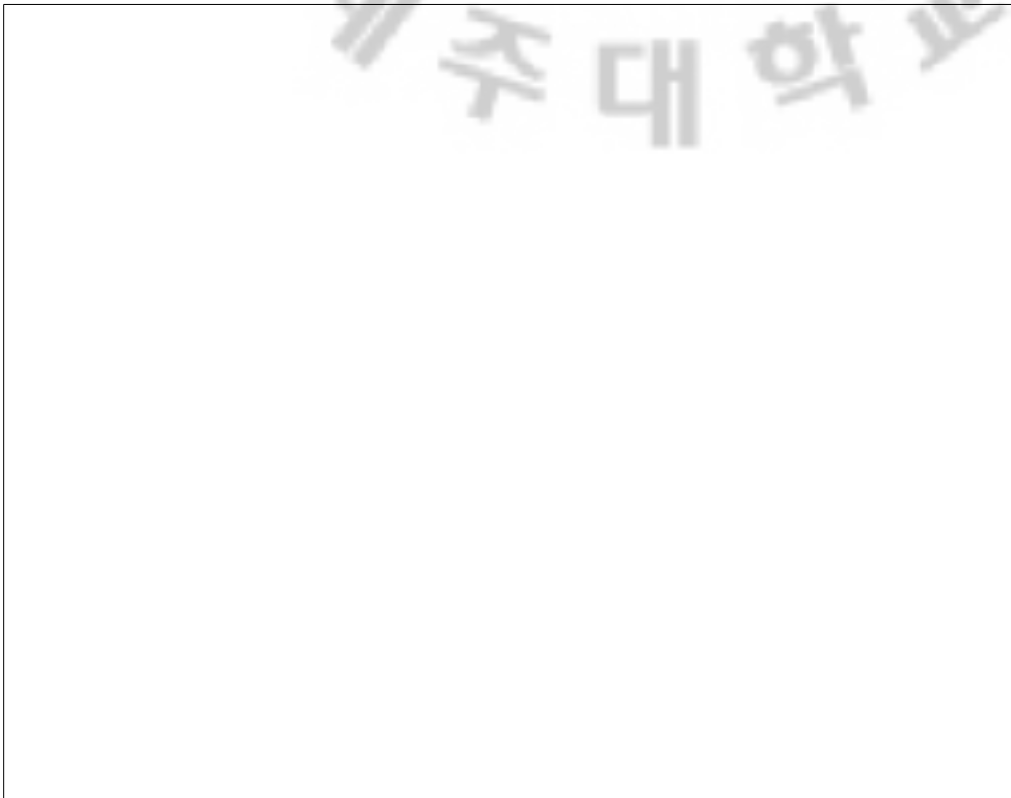
4단계

생각 넓히기

▶ 다른 방법으로 작성한 것을 보고 자신이 작성한 것과 비교하여 생각을 적어 봅시다.



▶ [과제] 다른 정렬 방법을 조사해 봅시다.





# 스크래치를 통한 알고리즘 배우기

## ◆ 여섯 번째 알고리즘 ◆ 분할 정복 [Divide and conquer]

:: 이진탐색 프로그램 짜기 ::

### 1단계

#### 생활에서 생각하기



재현이는 공부시간에 배운 '강아지똥' 책을 찾아 읽기 위해 학교도서관을 찾아갔습니다. 컴퓨터를 통해 검색해보니, 오른쪽 그림과 같이 나왔습니다. 재현이는 어떻게 해야 '강아지똥' 책을 찾을 수 있을까요?

#### 강아지똥

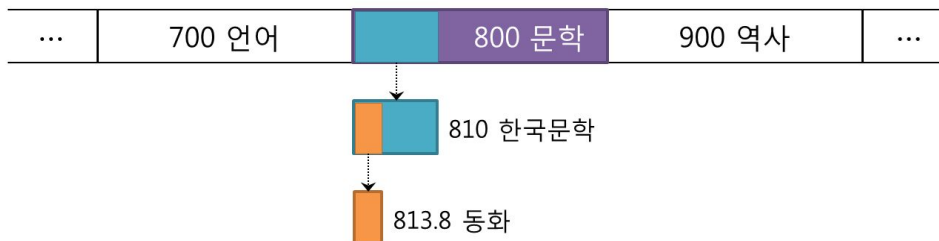
- 권정생 지음; 정승각 그림
- 길벗어린이 (1996)
- 검색번호 : 813.8 권73기

▶ 학교도서관에서 읽고 싶은 책을 찾아본 적이 있나요? 어떻게 찾았나요?

▶ '검색번호'가 뜻하는 것은 무엇인지 조사해 봅시다.

앞에서 우리는 어떤 기준에 따라 순서대로 나열된(정렬된) 자료에서 필요한 자료를 찾는(탐색하는) 방법에는 이진탐색이 있다고 배웠습니다. 위와 같이 도서관에서 책을 찾을 때 검색번호(한국십진분류법)를 통해 찾는 방법이 이진탐색입니다.

위의 예에서 '813.8'이 뜻하는 것을 그림으로 나타내면 다음과 같다.



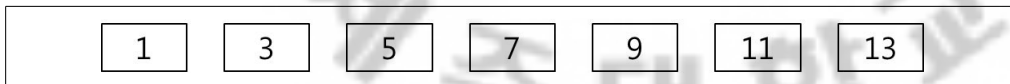
이처럼 많은 자료들(도서관의 책) 중에서 그 범위를 좁혀가면서(800 → 810 → 813.8) 찾는 방법이 **이진탐색**입니다.

이 이진탐색은 분할정복의 한 예입니다. 분할정복은 해결하기 어려운 큰 문제를 작은 문제로 나눠서 풀이하는 방법으로 쉽게 설명하면 계단 5개를 한 번에 오르는 어렵지만, 한 계단씩 나눠서 오르면 쉽게 오를 수 있습니다. 이처럼 어려운 문제를 작은 문제로 나눠서 해결하는 것을 **분할정복**이라고 합니다. 이진탐색 외에도 분할정복의 대표적인 예로는 앞에서 배운 퀵(quick)정렬과 합병정렬이 있습니다. 이번 시간에는 분할정복 중에 이진탐색을 배워봅시다.

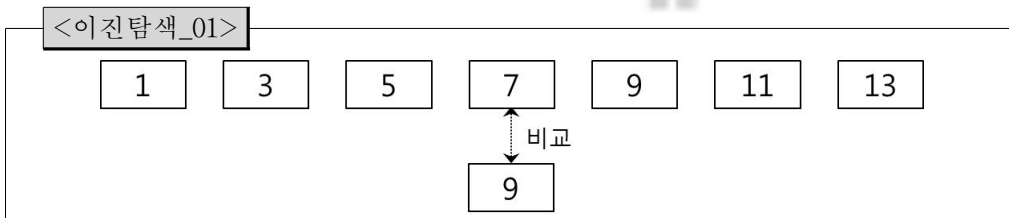
2단계

그림으로 나타내기

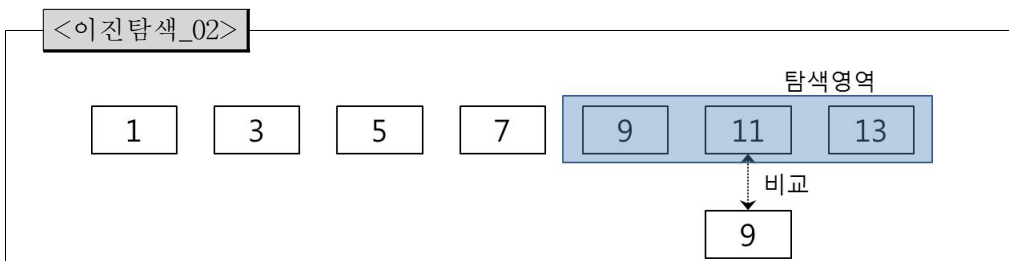
▶ 이진탐색은 앞에서 배운 선형탐색과는 다르게 자료가 정렬된 자료여야 합니다. 다음과 같이 크기 순서로 정렬된 7개의 자료 중에 9를 이진탐색을 통해 찾는 과정을 그림으로 살펴봅시다.



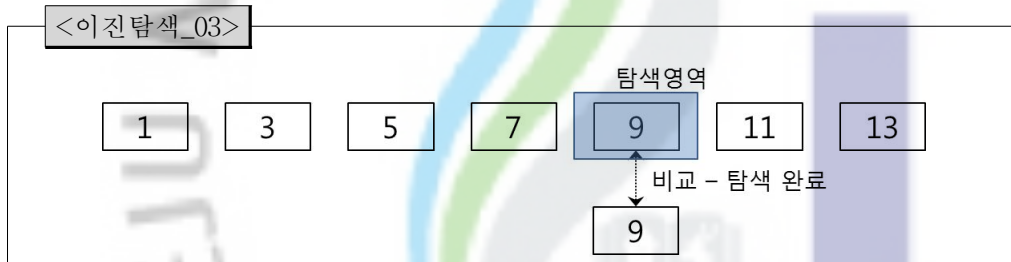
먼저 중간에 위치한 자료인 7과 찾는 자료 9를 비교합니다.



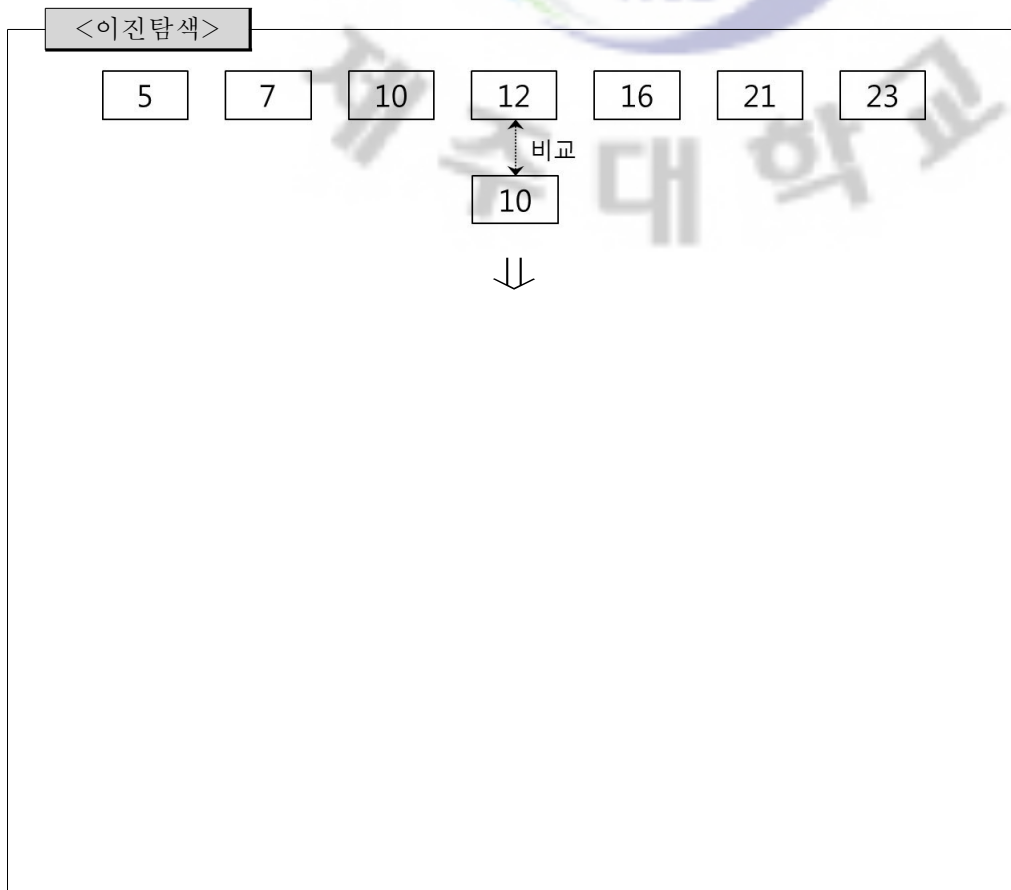
7보다 9가 크므로 우리가 찾고자 하는 9는 7보다 큰 위치인 오른쪽 자료들 중에 있습니다. 그래서 오른쪽에 있는 자료들에 대해 다시 한 번 이진탐색을 합니다. 새로운 탐색영역의 중간에 위치한 11과 9를 비교합니다.



11보다 9가 작으므로 11보다 작은 위치인 왼쪽 자료들에 대해 다시 한 번 이진탐색을 합니다. 새로운 탐색영역의 중간에 위치한 9와 9를 비교하면 같은 자료이므로 탐색을 끝냅니다.



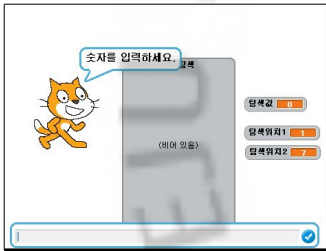
이렇게 정렬된 자료에서 범위를 좁혀가며(자료를 나뉘가며) 탐색하는 방법이 이진탐색입니다. 다음 7개의 자료 중에서 10을 이진탐색으로 찾는 과정을 그림으로 나타내봅시다.



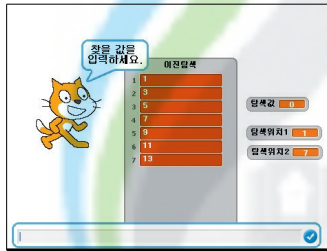
3단계

프로그래머 되어보기

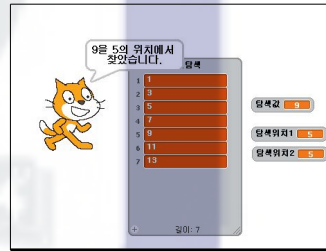
▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<자료 입력>



<탐색값 입력>



<탐색 완료>

<명령어 블록>



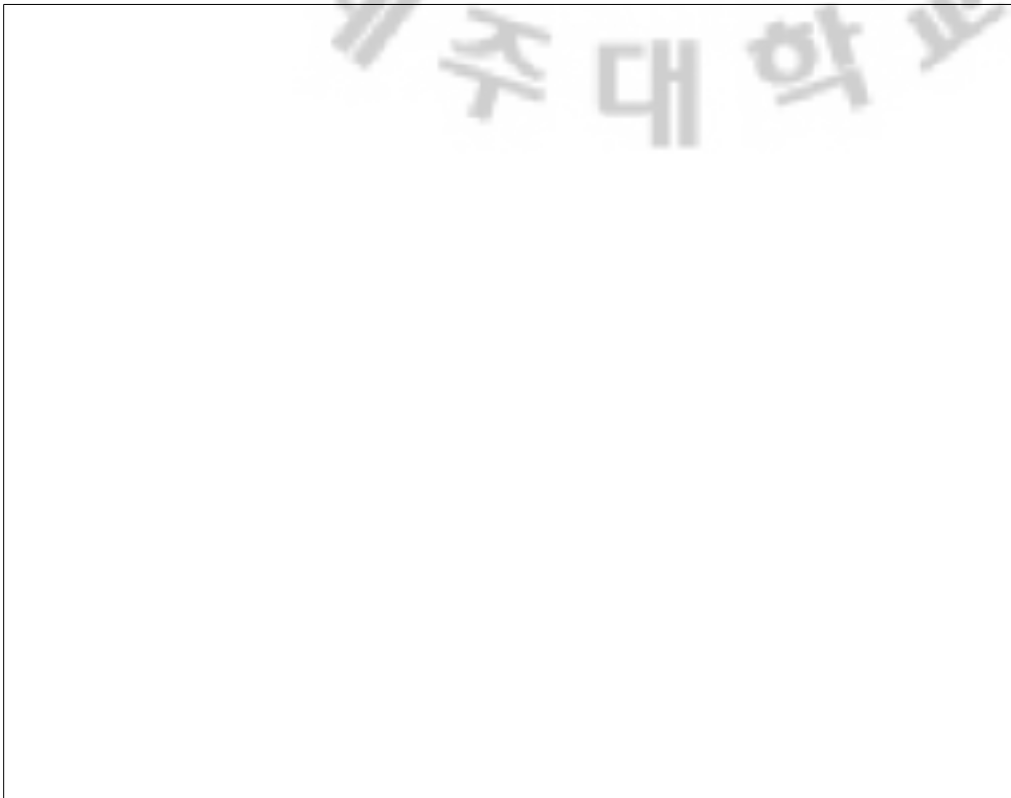
4단계

생각 넓히기

▶ 다른 방법으로 작성한 것을 보고 자신이 작성한 것과 비교하여 생각을 적어 봅시다.



▶ [과제] 정렬과 이진탐색이 같이 동작하는 프로그램을 작성해 봅시다.







# 스크래치를 통한 알고리즘 배우기

◆ 일곱 번째 알고리즘 ◆

## 그리디(greedy)

:: 그리디를 이용한 프로그램 짜기 ::

1단계

### 생활에서 생각하기



상현이는 음료수를 사기 위해 마트에 갔습니다. 음료수 가격을 알아보니 1280원이었습니다. 상현이가 2000원을 내고 거스름돈을 받는다면 얼마를 받아야 하나요? 또, 거스름돈을 받을 때 어떤 동전을 얼마만큼 받아야 하나요?

▶ 상현이가 받아야할 거스름돈은 얼마입니까?

▶ 거스름돈으로 받을 수 있는 동전의 개수를 생각나는 대로 모두 써 봅시다.

▶ 어떤 경우가 동전의 개수가 가장 적습니까?

만약, 570원을 거스름돈으로 받는다고 생각해 봅시다. 받을 수 있는 방법은 다양합니다. 10원짜리 동전을 57개 받는 방법 ( $10 \times 57 = 570$ 원), 50원짜리 동전 11개와 10원짜리 동전 2개 ( $50 \times 11 + 10 \times 2 = 570$ 원)를 받는 방법 등 많은 방법들이 있습니다. 그 중에서 동전의 개수가 가장 적은 방법은 500원짜리 동전 1개와 50원짜리 동전 1개, 그리고 10원짜리 동전 2개를 받는 방법 ( $500 + 50 + 10 \times 2 = 570$ 원)입니다. 이렇게 다양한 방법 중에서 가장 효과적이고 좋은 방법을 찾는 것을 **그리디(greedy)**라고 합니다.

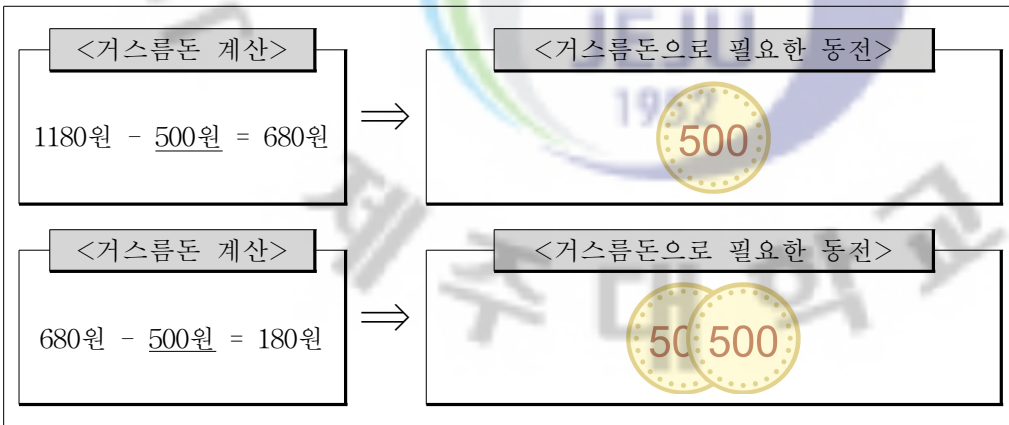
2단계

그림으로 나타내기

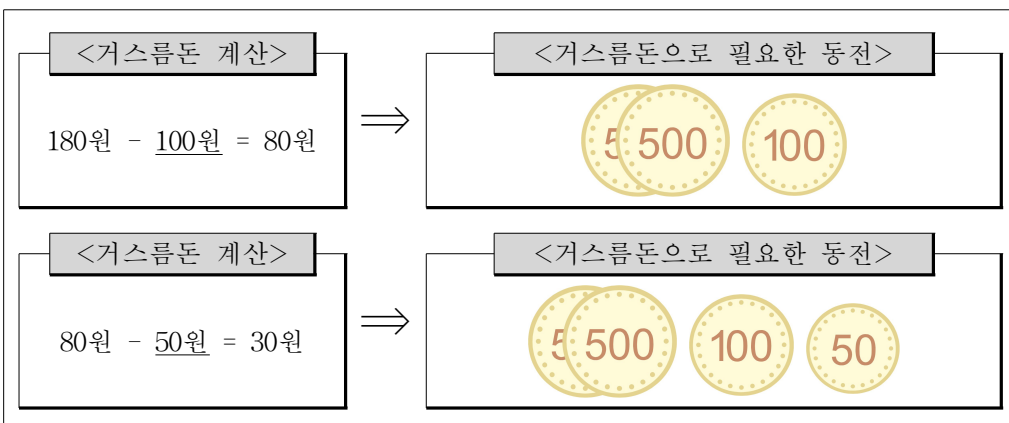
▶ 거스름돈으로 나눠줄 수 있는 동전은 500원, 100원, 50원, 10원이 있습니다.

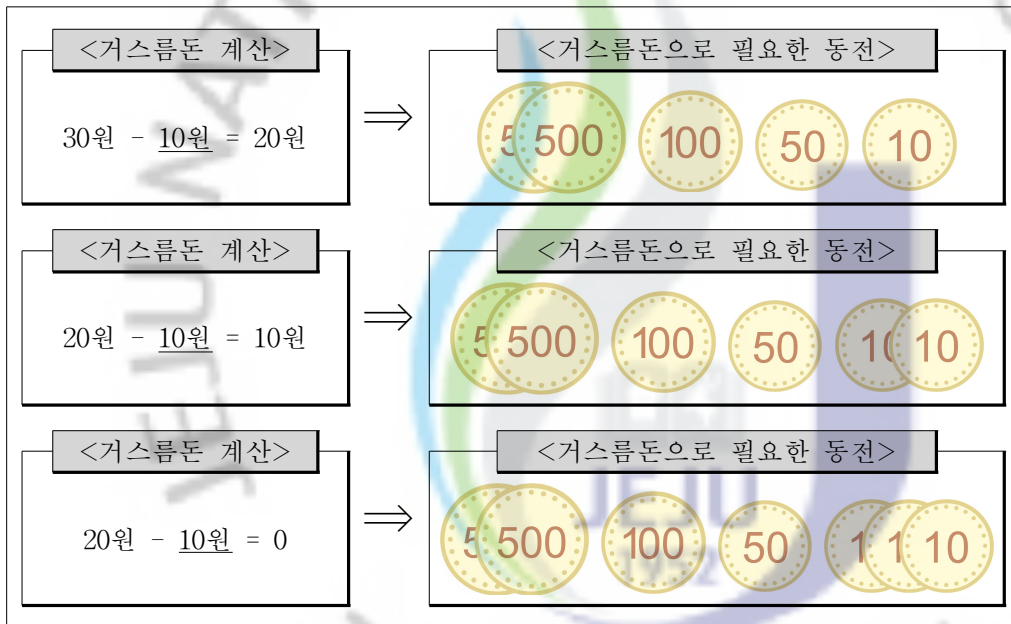


1180원을 거스름돈으로 주기위해 어떤 동전이 몇 개 필요한지 생각해 봅시다. 동전의 개수를 줄이기 위해서는 금액이 큰 동전부터 차례대로 개수를 세어보면 됩니다. 먼저 500원짜리 동전이 몇 개 필요한지 알아보기 위해 거스름돈에서 500원씩을 빼 봅시다.



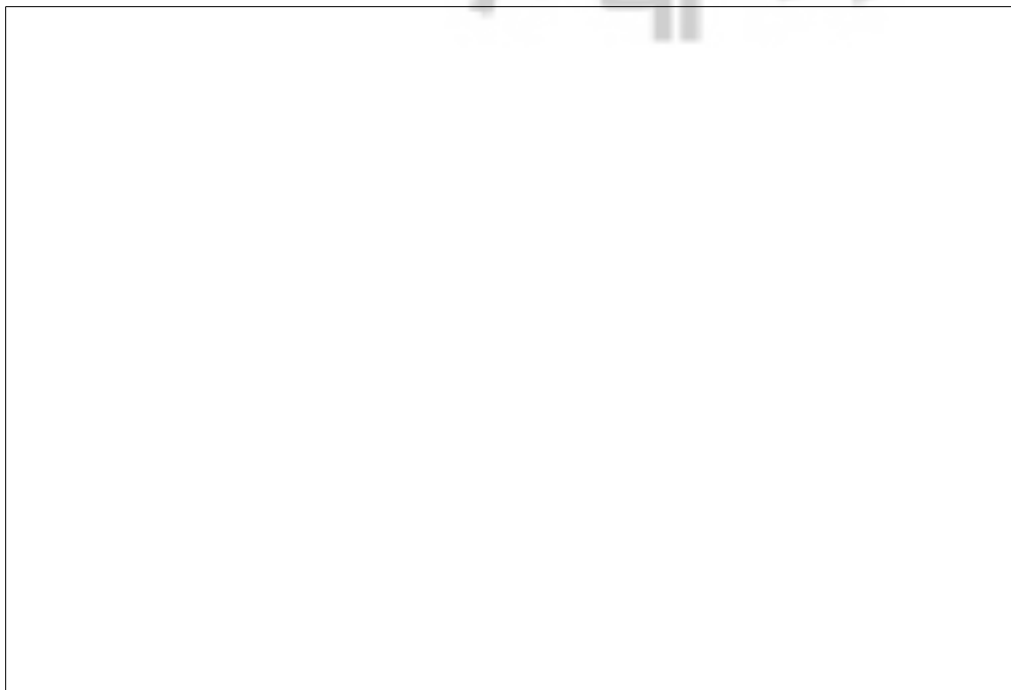
이제 남은 거스름돈이 180원으로 500원보다 작습니다. 이번에는 그 다음 크기의 동전인 100원으로 빼 봅시다. 계속해서 50원, 10원으로 계산하면 필요한 동전을 알 수 있습니다.





결국 1180원을 거슬러 주기위해서 가장 좋은 방법은 500원 2개, 100원 1개, 50원 1개, 그리고 10원 3개입니다.

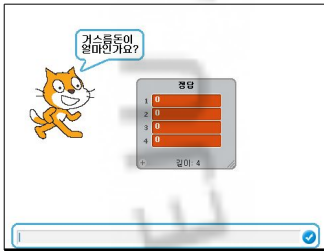
▶ 위와 같은 방법으로 830원을 거슬러 주는 방법을 그림으로 나타내 봅시다.



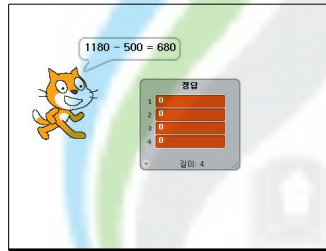
3단계

프로그래머 되어보기

▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<거스름돈 입력>



<답 구하기 과정>



<정답 출력>

<명령어 블록>

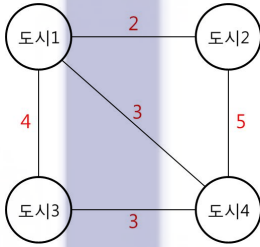


4단계

생각 넓히기

▶ 다음 문제는 그리디를 사용하여 해결할 수 있는 문제입니다. 문제를 살펴봅시다.

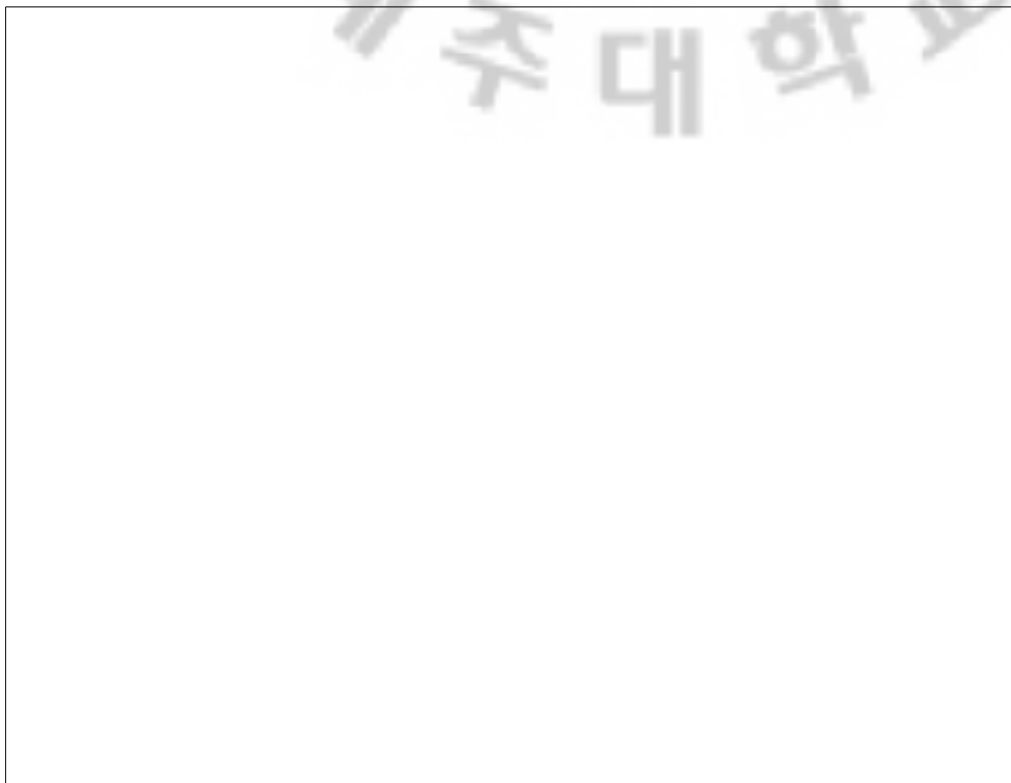
4개의 도시가 있습니다. 이 4개의 도시를 서로 연결하는 도로를 건설하기 위해 비용을 알아보니 오른쪽 그림에 있는 숫자와 같습니다. 그렇다면 최소의 비용으로 모든 도시가 하나의 도로 이상은 연결될 수 있도록 하려면 어떻게 하는 것이 좋을까요?



```
graph TD; D1((도시1)) ---|2| D2((도시2)); D1 ---|4| D3((도시3)); D1 ---|3| D4((도시4)); D2 ---|5| D4; D3 ---|3| D4;
```

이 문제는 프림 알고리즘과 크루스칼 알고리즘이라는 방법으로 해결할 수 있습니다. 이 두가지 방법과 함께 내비게이션에서 최단거리를 구하는 방법인 다익스트라 알고리즘이 그리디의 대표적인 방법입니다.

▶ 위의 문제를 그림으로 해결해 보고, 스크래치로 만든 프로그램을 살펴봅시다.





# 스크래치를 통한 알고리즘 배우기

◆ 여덟 번째 알고리즘 ◆

## 백트래킹(Backtracking)

:: 백트래킹을 이용한 프로그램 짜기 ::

1단계

### 생활에서 생각하기



헤민이는 미로공원에 친구들과 함께 놀러 왔습니다. 미로 찾기를 하면서 놀던 중 헤민이는 미로를 찾는 방법에 대해 생각하게 되었습니다. 그리고 몇 가지 규칙을 발견했습니다. 어떤 규칙일까요?

▶ 미로 길을 찾을 때 처음에는 어떻게 합니까? 그리고 두 갈래의 길이 나오면 어떻게 합니까?

▶ 갈림길에서 한쪽으로 갔더니 막힌 곳이 나왔습니다. 어떻게 합니까?

▶ 갈림길의 다른 쪽 또한 막혀 있습니다. 어떻게 합니까?

위의 내용을 바탕으로 미로 찾기의 규칙을 말해 보면 다음과 같습니다.

규칙1. 처음에는 가능한 곳으로 걸어가 본다.

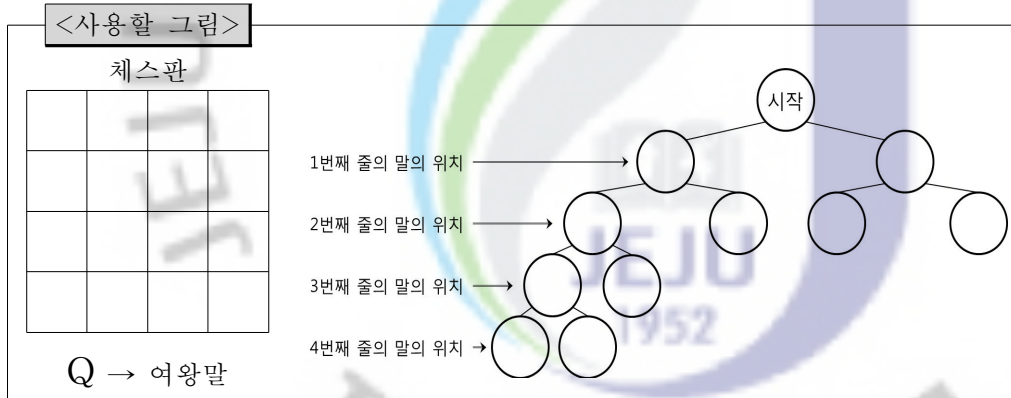
규칙2. 막힌 곳이 나오면 바로전의 갈림길로 되돌아갑니다. 그리고 다시는 막힌 곳으로 가지 않습니다.

이 2가지 규칙을 적용해서 미로를 찾다 보면 어느 순간 도착점에 도달한 것을 알 수 있습니다. 이와 같이 가능한 경우를 모두 생각하여 문제를 해결해 나가다가 불가능한 경우가 생기면 원래자리로 되돌아가서 문제를 다시 풀이하는 방법을 **백트래킹**이라고 합니다. 백트래킹과 관련된 문제 중에 대표적인 것이 **4-여왕말 문제**입니다. 이번 시간에는 이 4-여왕말 문제를 통해 백트래킹의 방법을 살펴보도록 하겠습니다.

2단계

그림으로 나타내기

▶ 서양 장기인 체스에서 여왕말은 가로, 세로, 대각선 모두를 이동할 수 있습니다. 4-여왕말 문제는 가로, 세로 4칸인 체스판에 4개의 여왕말이 서로 공격하지 못하도록 놓는 방법을 말합니다. 그림을 통해 문제를 해결해 봅시다.

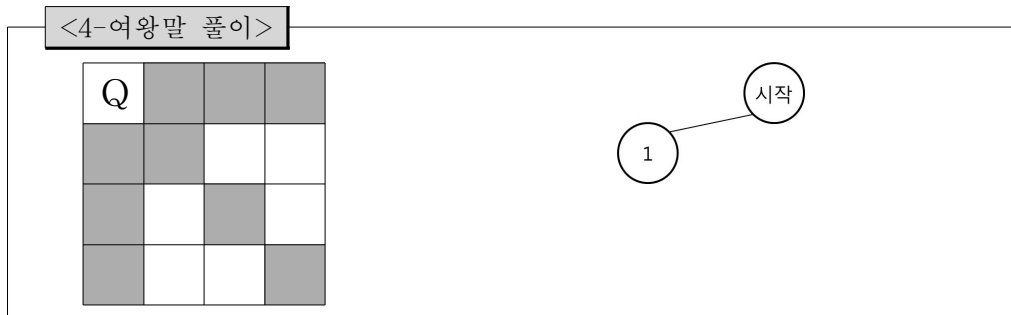


위와 같이 체스판, 여왕말, 그리고 답을 해결해 나가는 과정을 트리 구조로 나타내 보도록 하겠습니다. 미로 찾기에서처럼 백트래킹의 방법으로 해결해 봅시다.

규칙1. 가능한 방법으로 시도해 본다.

규칙2. 불가능한 경우가 생기면 전 단계로 되돌아와서 다른 방법으로 시도해 본다.

규칙1에 따라 1번째 줄의 첫 번째 위치에 여왕말을 놓아봅시다.



이렇게 되면 첫 번째 위치에 놓인 여왕말 때문에 다른 여왕말이 들어갈 수 없는 자리를 알 수 있습니다(회색으로 칠해진 곳에는 다른 여왕말이 들어갈 수 없습니다). 그리고 트리 구조는 루트노드인 '시작'에서 1번째 줄의 말의 위치에 노드 '1'을 넣어 줍니다. 이제 2번째 줄에 여왕말이 들어갈 수 있는 곳은 세 번째, 네 번째 칸입니다. 규

칙1에 의해 먼저 세 번째 칸에 여왕말을 놓아봅시다.

<4-여왕말 풀이>

Q	X	X	X
X	X	Q	X
	X	X	X
X		X	

이제는 첫 번째, 두 번째 여왕말에 의해 다른 여왕말이 들어갈 수 없는 자리를 알 수 있습니다(회색과 'X'표시). 그리고 트리 구조는 2번째 줄의 말의 위치에 노드 '3'이 연결됩니다. 이제 다음 단계로 넘어가 봅시다. 그런데 3번째 줄에 여왕말을 놓을 곳이 없습니다. 즉 지금 방법으로는 문제를 해결할 수 없습니다. 그래서 규칙2에 따라 전 단계로 되돌아가서 2번째 줄에 이번에는 네 번째 칸에 여왕말을 놓아봅시다.

<4-여왕말 풀이>

Q		X	X
X	X	X	Q
		X	X
	X		X

그렇게 하면 위에 그림처럼 됩니다. 트리 구조는 전 단계로 되돌아가서 다시 했기 때문에 노드 '1'에 노드 '4'가 연결됩니다. 이제 3번째 줄에 여왕말을 놓아봅시다.

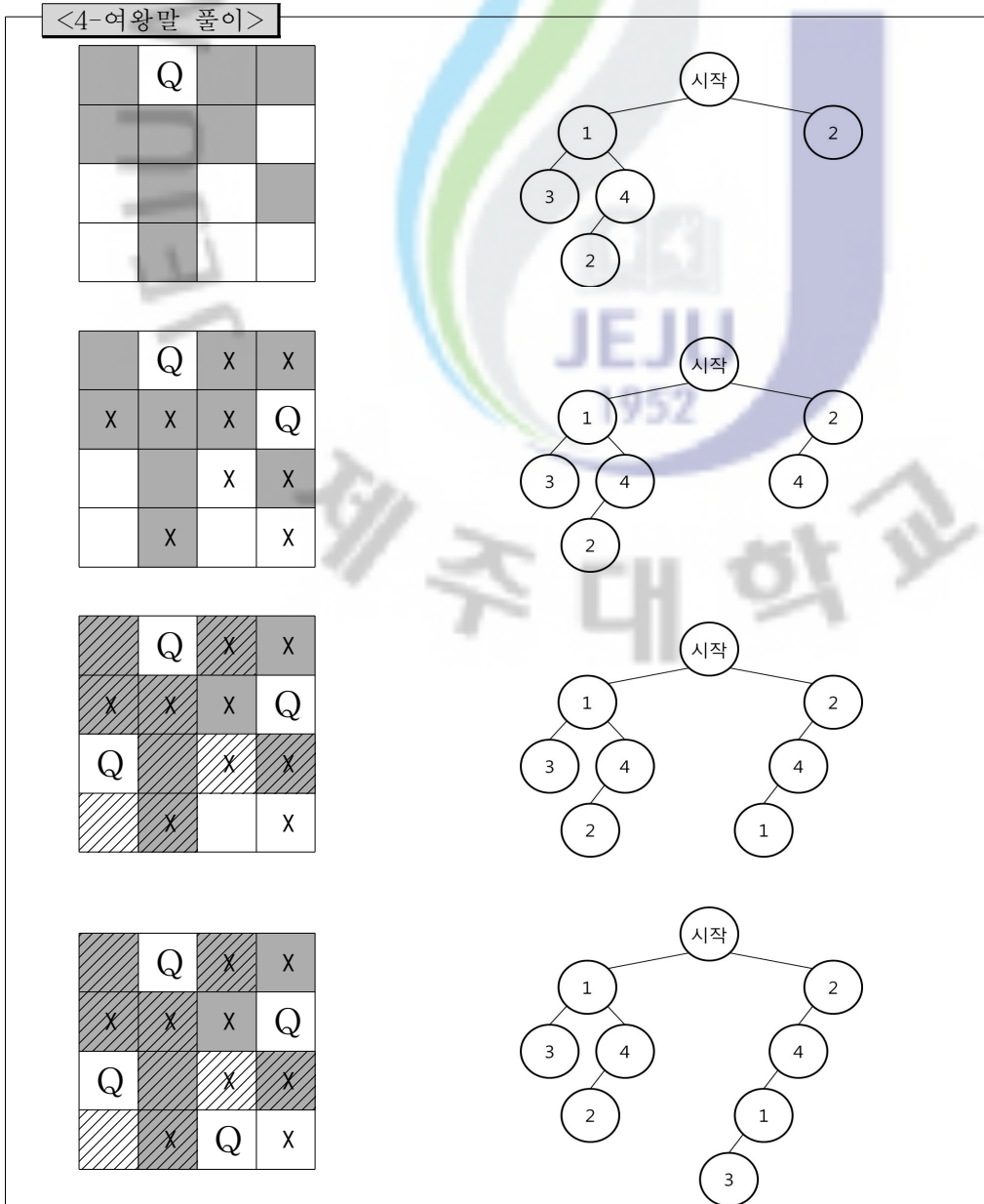
<4-여왕말 풀이>

Q		X	X
X	X	X	Q
	Q		X
	X		X

3번째 줄 두 번째 위치에 여왕말을 놓으면 트리 구조에서는 노드 '4'에 노드 '2'가 연결됩니다. 이제 확인해 보면 4번째 줄에 여왕말을 놓을 수 없습니다. 이 방법 또한 해

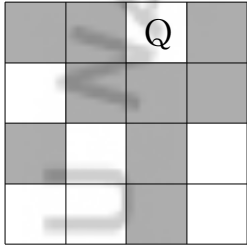


결방법이 아니기 때문에 규칙2에 따라 전 단계로 되돌아갑니다. 계속해서 되돌아가면 1번째 줄에 여왕말을 놓는 곳까지 되돌아가게 됩니다. 이제는 1번째 줄 두 번째 위치에 여왕말을 놓고 다시 문제를 해결해 나갑니다. 그 과정은 아래와 같습니다.



드디어 방법을 찾았습니다. 위 그림과 같이 여왕말을 놓으면 서로 공격하지 않고 놓을 수 있습니다. 계속해서 또 다른 해결방법을 그림을 통해 찾아내 봅시다.

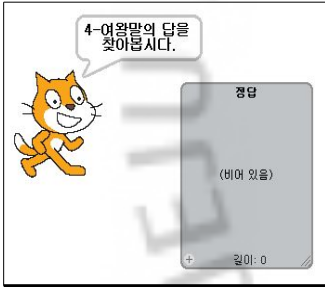
<4-여왕말 풀이>



3단계

프로그래머 되어보기

▶ 아래 그림과 같은 구성으로 프로그램을 작성하려고 합니다. 스크래치로 직접 작성해 보고 명령어 블록을 그려봅시다.



<시작 화면>



<정답 출력>

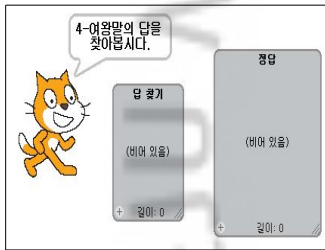
<명령어 블록>



4단계

생각 넓히기

▶ 앞의 프로그램을 수정하여 풀이과정을 볼 수 있는 프로그램을 작성해 봅시다.



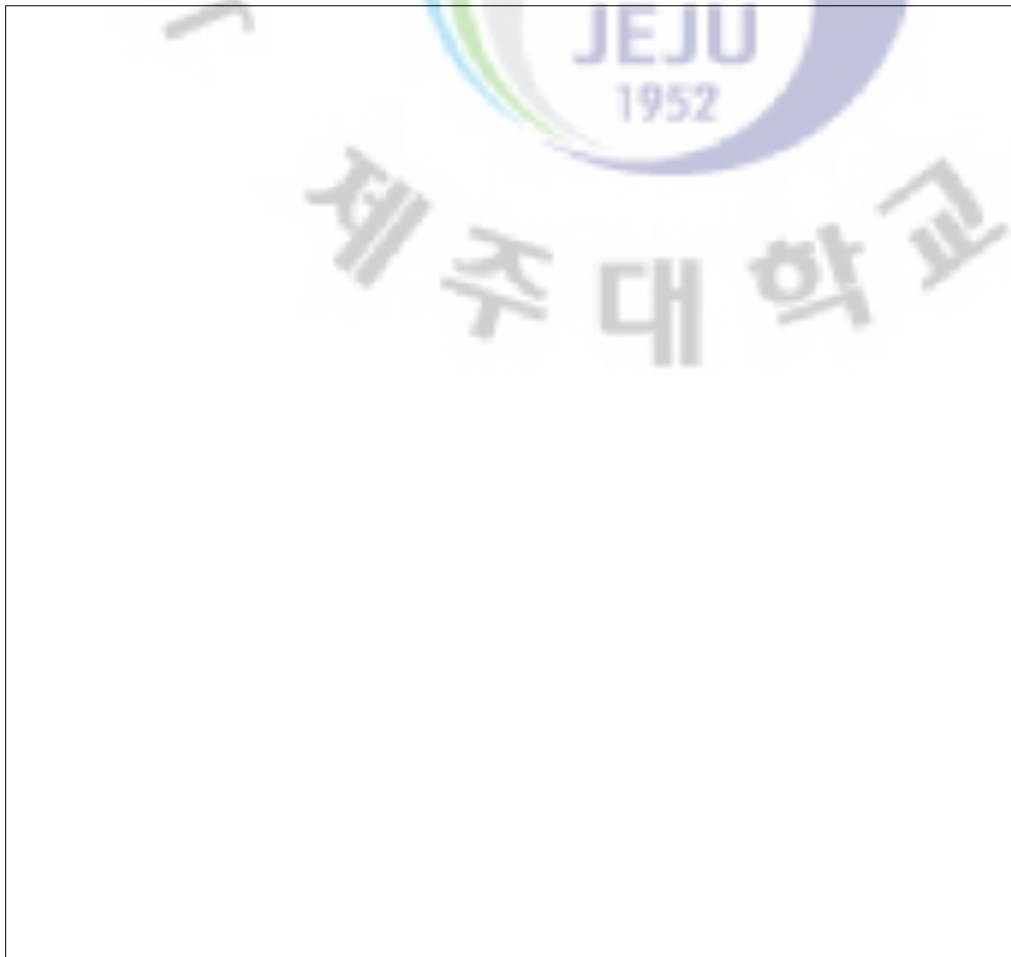
<시작 화면>



<답 찾기>



<정답 출력>



▶ [과제] 5-여왕말 문제를 프로그램으로 작성해 봅시다.


#### 4. 교수학습 과정안과 지도자료

일시	2010. 09.	대상	4학년	차시	1/8
단원	1. 스택			장소	컴퓨터실
학습목표	스택의 구조를 알아보고 스크래치로 표현할 수 있다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 수납상자에 옷을 넣고 꺼내는 순서 알아보기	5	○ PPT, 사진
	학습 문제	★ 스택의 구조를 알아보고 스크래치로 표현해 보자.		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 스택에서 삽입과정 알아보기 - 스택에서 삭제과정 알아보기 - 필요한 명령어 블록 생각해 보기	5	
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 스택 구조에 자료가 삽입되고 삭제되는 과정을 알아볼 수 있는 프로그램 작성하기	20	※ 첫 시간이므로 천천히 차근차근 설명한다.
반성	생각 넓히기	활동3. 생각 넓히기 - 활동2와 다른 방법으로 작성한 것을 보고 생각 적어보고 발표하기 ▷ 과제제시 - 실생활에서 스택 구조 찾아보기 ▷ 차시예고 - 큐	10	※ 자신의 생각을 자유롭게 발표한다.
평가 계획	평가내용		평가방법	평가시기
	스택의 삽입과 삭제 과정을 그림으로 잘 나타내는가?		관찰평가	계획의 수립

《 활동2 프로그램 》

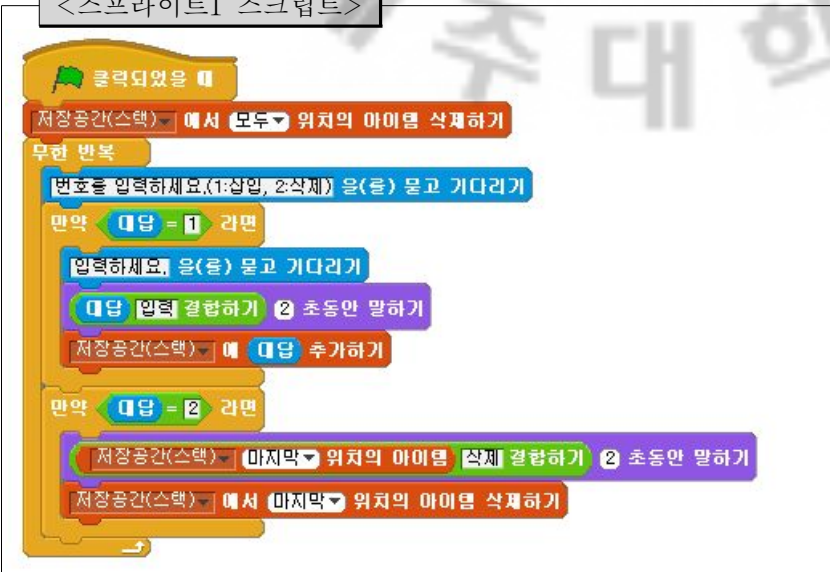
<무대와 스프라이트 구성>



<변수와 리스트 구성>

- ▷ 변수 : 없음.
- ▷ 리스트 : 저장공간(스택)

<스프라이트1 스크립트>



일시	2010. 09.	대상	4학년	차시	2/8
단원	2. 큐			장소	컴퓨터실
학습목표	큐의 구조를 알아보고 스크래치로 표현할 수 있다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 붕어빵이 기계 안에 들어간 순서와 나오는 순서를 생각해 보기	5	○ PPT, 사진
	학습 문제	<b>★ 큐의 구조를 알아보고 스크래치로 표현해 보자.</b>		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 큐에서 삽입과정 알아보기 - 큐에서 삭제과정 알아보기 - 필요한 명령어 블록 생각해 보기	5	○ PPT
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 큐 구조에 자료가 삽입되고 삭제되는 과정을 알아볼 수 있는 프로그램 작성하기	20	※ 아직 익숙치 않으므로 천천히 차근차근 설명한다.
반성	생각 넓히기	활동3. 생각 넓히기 - 활동2와 다른 방법으로 작성한 것을 보고 생각 적어보고 발표하기 ▷ 과제제시 - 실생활에서 큐 구조 찾아보기 ▷ 차시예고 - 트리	10	※ 자신의 생각을 자유롭게 발표한다.
평가 계획	평가내용		평가방법	평가시기
	큐의 삽입과 삭제 과정을 그림으로 잘 나타내는가?		관찰평가	계획의 수립

《 활동2 프로그램 》

<무대와 스프라이트 구성>

<변수와 리스트 구성>

- ▷ 변수 : 없음.
- ▷ 리스트 : 저장공간(큐)

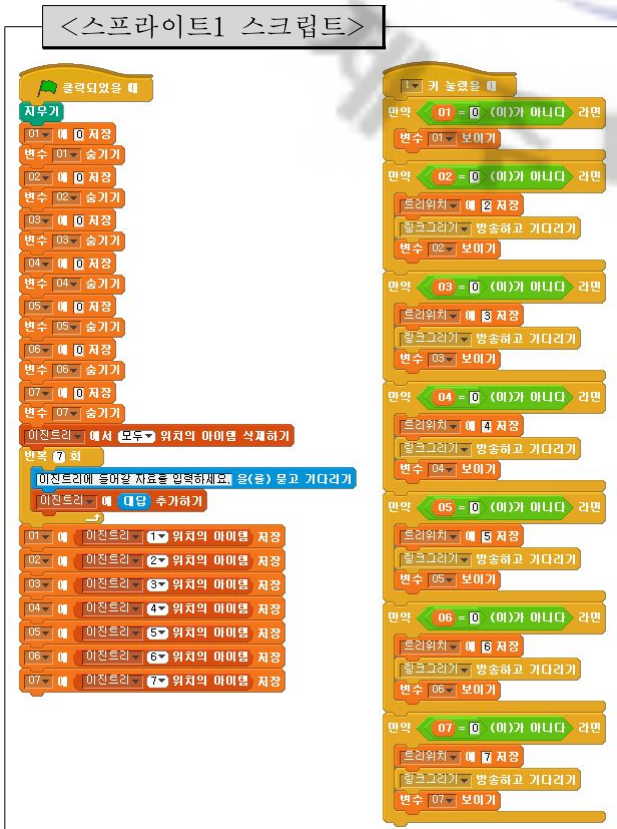
<스프라이트1 스크립트>



일시	2010. 10.	대상	4학년	차시	3/8
단원	3. 트리			장소	컴퓨터실
학습목표	트리의 구조를 알아보고 스크래치로 이진트리를 표현할 수 있다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 빈 공간을 채워 마인드맵을 완성해 보기	5	○ PPT, 사진  ○ PPT
	학습 문제	★ 트리의 구조를 알아보고 스크래치로 이진트리를 표현해 보자.		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 이진트리가 저장 공간에 저장되는 과정 알아보기 - 이진트리의 장점 알아보기 - 직접 이진트리 나타내 보기	5	※ 판서와 설명을 통해 잘 이해할 수 있도록 지도한다.
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 저장 공간에 저장된 자료가 이진트리로 나타내어지는 과정을 보여주는 프로그램 작성하기	20	※ 아직 익숙치 않으므로 천천히 차근차근 설명한다.
반성	생각 넓히기	활동3. 생각 넓히기 - 링크를 그리는 명령어 블록 이해하기 ▷ 과제제시 - 깊이를 더 늘려서 프로그램 작성하기 ▷ 차시예고 - 탐색	10	※ 먼저 스스로 생각해보게 한 후 명령어 블록들을 설명한다.
평가 계획	평가내용		평가방법	평가시기
	이진트리를 그림으로 잘 나타내는가?		관찰평가	계획의 수립

## 《 활동2 프로그램 》



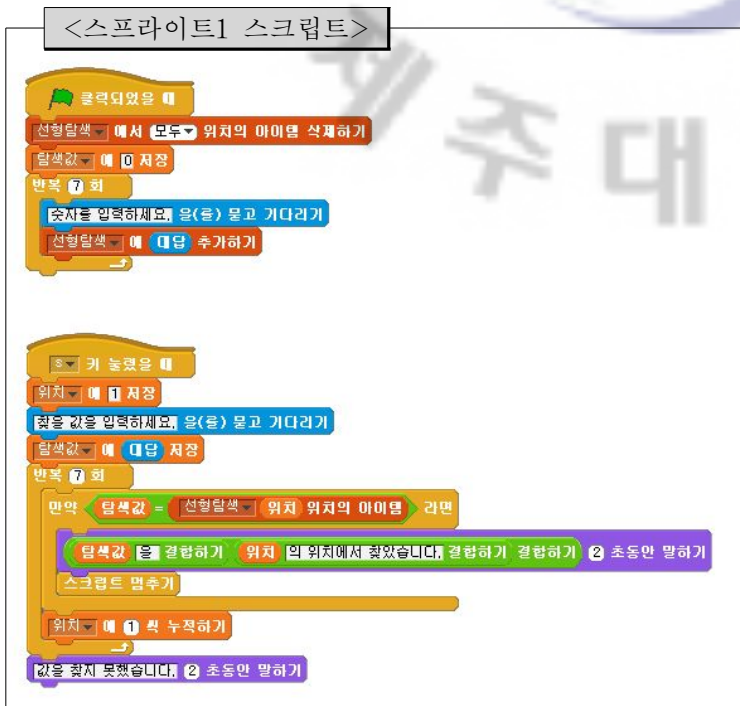
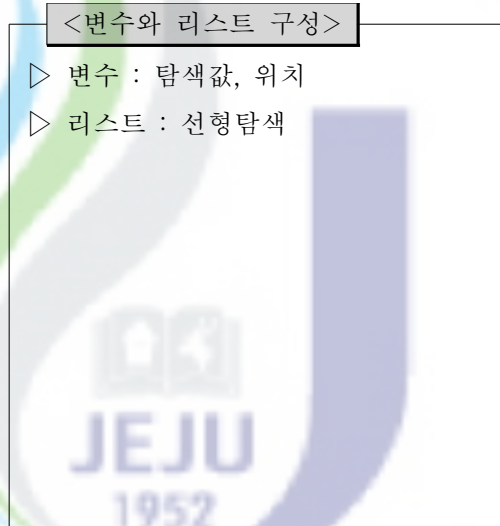
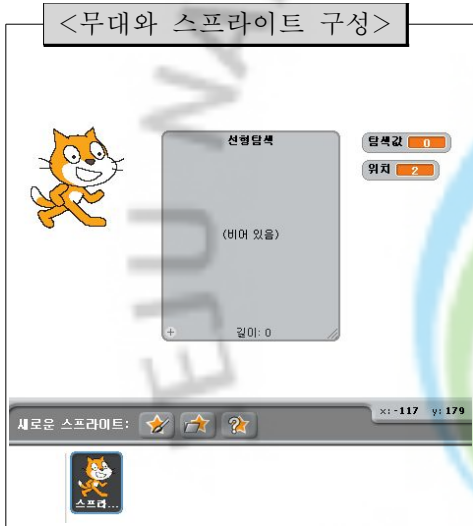
※ 스프라이트2(흰색 점)는 선을 그리기 위한 스프라이트임.

일시	2010. 10.	대상	4학년	차시	4/8
단원	4. 탐색			장소	컴퓨터실
학습목표	선형탐색의 방법을 이해하고 스크래치로 프로그램을 만들어 봅시다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 국어사전에서 필요한 낱말을 찾는 과정 생각해 보기	5	○ 국어사전, PPT  ○ PPT
	학습 문제	★ 선형탐색의 방법을 이해하고 스크래치로 프로그램을 만들어 보자.		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 선형탐색의 탐색과정 알아보기 - 선형탐색의 탐색과정 그림으로 나타내기	5	※ 판서와 설명을 통해 잘 이해할 수 있도록 지도한다.
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 선형탐색을 통해 탐색한 후 결과를 보여주는 프로그램 작성하기	20	
반성	생각 넓히기	활동3. 생각 넓히기 - 활동2와 다른 방법으로 작성한 것을 보고 생각 적어보고 발표하기 ▷ 과제제시 - 다른 탐색방법 생각해 보기 ▷ 차시예고 - 정렬	10	※ 자신의 생각을 자유롭게 발표한다.
평가 계획	평가내용		평가방법	평가시기
	선형탐색의 탐색과정을 그림으로 잘 나타내는가?		관찰평가	계획의 수립



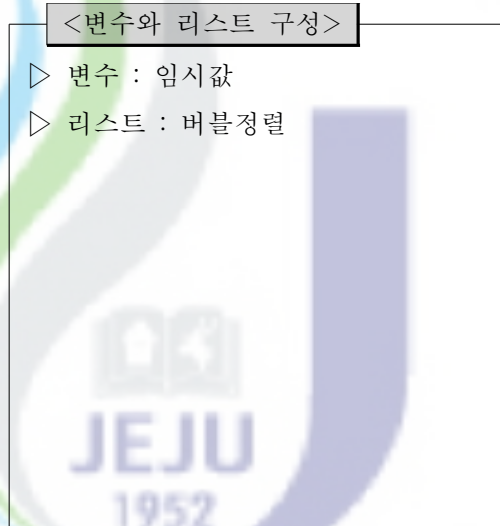
### 《 활동3 프로그램 》



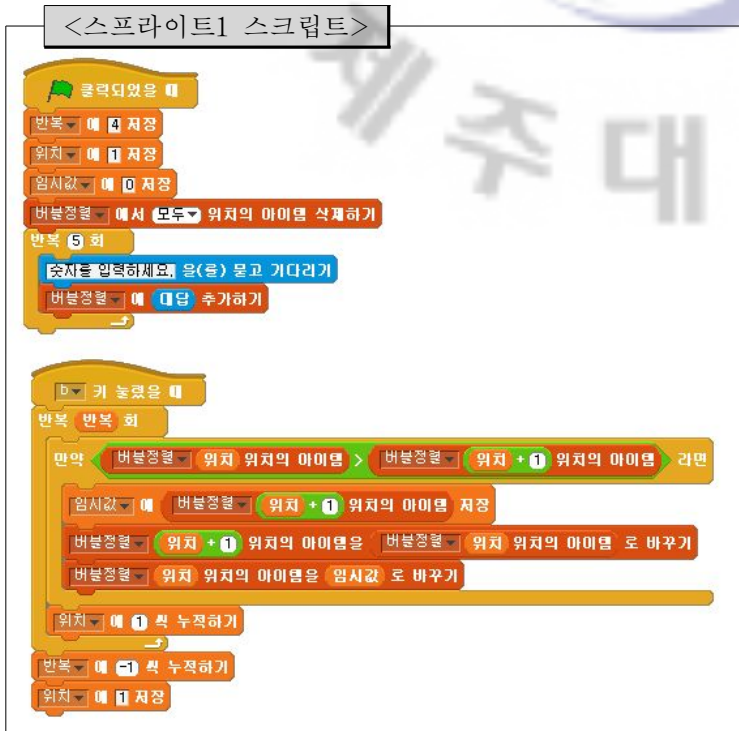
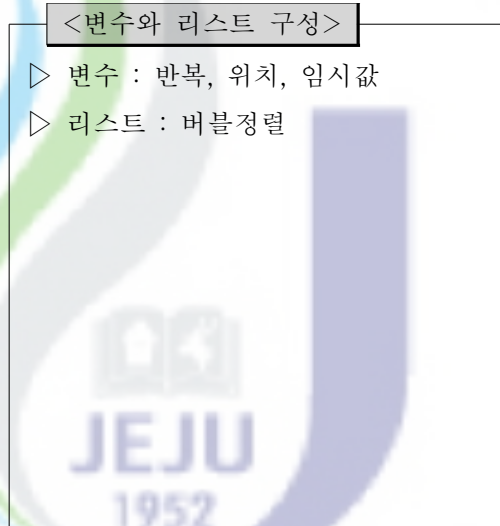
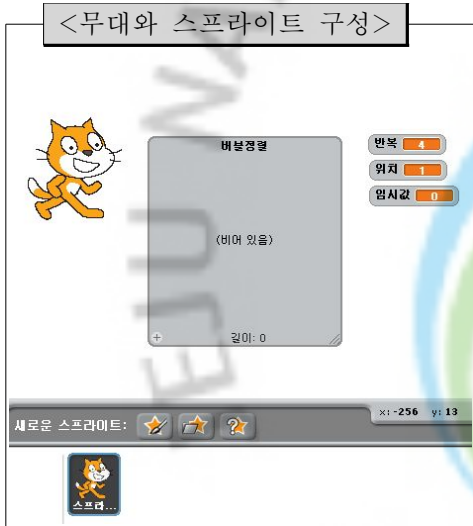
일시	2010. 10.	대상	4학년	차시	5/8
단원	5. 정렬			장소	컴퓨터실
학습목표	버블정렬의 방법을 이해하고 스크래치로 프로그램을 만들어 봅시다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 100m 달리기 기록 순서대로 나열하기	5	○ PPT
	학습 문제	★ 버블정렬의 방법을 이해하고 스크래치로 프로그램을 만들어 보자.		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 버블정렬의 정렬과정 알아보기 - 버블정렬의 정렬과정 그림으로 나타내기	5	※ 판서와 설명을 통해 잘 이해할 수 있도록 지도한다.
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 입력한 자료를 버블정렬의 정렬과정으로 정렬하는 프로그램 작성하기	20	
반성	생각 넓히기	활동3. 생각 넓히기 - 활동2와 다른 방법으로 작성한 것을 보고 생각 적어보고 발표하기 ▷ 과제제시 - 다른 정렬방법 생각해 보기 ▷ 차시예고 - 분할 정복	10	※ 자신의 생각을 자유롭게 발표한다.
평가 계획	평가내용		평가방법	평가시기
	버블정렬의 정렬과정을 그림으로 잘 나타내는가?		관찰평가	계획의 수립

## 《 활동2 프로그램 》



### 《 활동3 프로그램 》

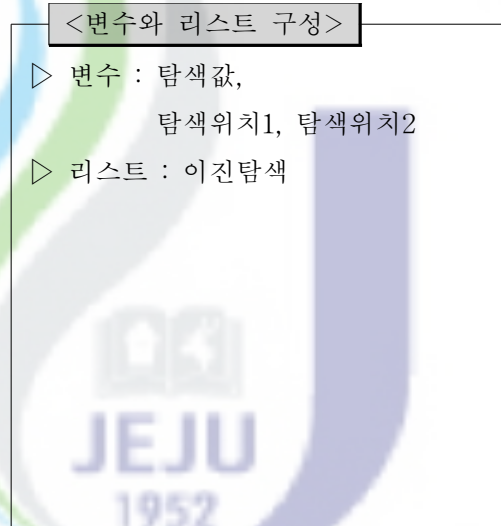




일시	2010. 10.	대상	4학년	차시	6/8
단원	6. 분할 정복			장소	컴퓨터실
학습목표	분할 정복의 방법을 이해하고 이진탐색 프로그램을 만들어 보시다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 도서관에서 검색한 책을 찾는 과정 생각해 보기	5	○ PPT, 도서관에서 빌린 '강아지똥' 책
	학습 문제	★ 분할 정복의 방법을 이해하고 이진탐색 프로그램을 만들어 보자.		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 이진탐색의 탐색과정 알아보기 - 이진탐색의 탐색과정 그림으로 나타내기	5	※ 판서와 설명을 통해 잘 이해할 수 있도록 지도한다.
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 입력한 자료를 이진탐색의 탐색방법으로 탐색하는 프로그램 작성하기	20	
반성	생각 넓히기	활동3. 생각 넓히기 - 활동2와 다른 방법으로 작성한 것을 보고 생각 적어보고 발표하기 ▷ 과제제시 - 정렬도 같이 동작하는 프로그램 작성 ▷ 차시예고 - 그리디	10	※ 자신의 생각을 자유롭게 발표한다.
평가 계획	평가내용		평가방법	평가시기
	이진탐색의 탐색과정을 그림으로 잘 나타내는가?		관찰평가	계획의 수립

## 《 활동2 프로그램 》



일시	2010. 11.	대상	4학년	차시	7/8
단원	7. 그리디			장소	컴퓨터실
학습목표	그리디를 이해하고 거스름돈 문제를 해결하는 프로그램을 만들어 봅시다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 거스름돈으로 받을 수 있는 동전의 개수 생각해 보기	5	○ PPT
	학습 문제	★ 그리디를 이해하고 거스름돈 문제를 해결하는 프로그램을 만들어 보자.		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 거스름돈 문제 해결과정 알아보기 - 직접 거스름돈 문제 해결해 보기	5	
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 거스름돈 문제를 해결하는 프로그램 작성하기	20	
반성	생각 넓히기	활동3. 생각 넓히기 - 최소비용 도로 건설 문제 확인하기 (최소 신장 트리) - 최소 신장 트리 해결과정과 프로그램 확인하기 ▷ 차시예고 - 백트래킹	10	※ 해결과정은 그림으로 설명하고 프로그램은 실행결과만 살펴본다.
평가 계획	평가내용		평가방법	평가시기
	거스름돈 문제를 해결하는 과정을 그림으로 잘 나타내는가?		관찰평가	계획의 수립

## 《 활동2 프로그램 》

### <무대와 스프라이트 구성>



### <변수와 리스트 구성>

- ▷ 변수 : 거스름돈
- ▷ 리스트 : 정답

### <스프라이트1 스크립트>



일시	2010. 11.	대상	4학년	차시	8/8
단원	8. 백트래킹			장소	컴퓨터실
학습목표	백트래킹을 이해하고 4-여왕말을 해결하는 프로그램을 만들어 봅시다.				

단계	학습과정	교수학습활동	시간(분)	자료(○) 및 유의점(※)
문제의 이해	동기 유발	▷ 생활에서 생각하기 - 미로에서 길을 찾는 방법 생각해 보기	5	○ 그림 ※ 그림과 판서를 통해 백트래킹을 설명한다.  ○ PPT
	학습 문제	★ 백트래킹을 이해하고 4-여왕말을 해결하는 프로그램을 만들어 보자.		
	학습 순서 안내	활동1. 그림으로 나타내기 활동2. 프로그래머 되어보기 활동3. 생각 넓히기		
계획의 수립	그림으로 나타내기	활동1. 그림으로 나타내기 - 4-여왕말의 풀이과정 알아보기 - 나머지 답 찾아보기	5	
계획의 실행	프로그래머 되어보기	활동2. 프로그래머 되어보기 - 4-여왕말의 답을 찾아주는 프로그램 작성하기	20	
반성	생각 넓히기	활동3. 생각 넓히기 - 활동2 프로그램에 답을 찾는 과정 (백트래킹)을 보여주는 스크립트 추가하기 ▷ 과제제시 - 5-여왕말 프로그램 작성하기 ▷ 차시예고 - 백트래킹	10	
평가 계획	평가내용		평가방법	평가시기
	4-여왕말의 다른 답을 찾을 수 있는가?		관찰평가	계획의 수립

## 《 활동2 프로그램 》

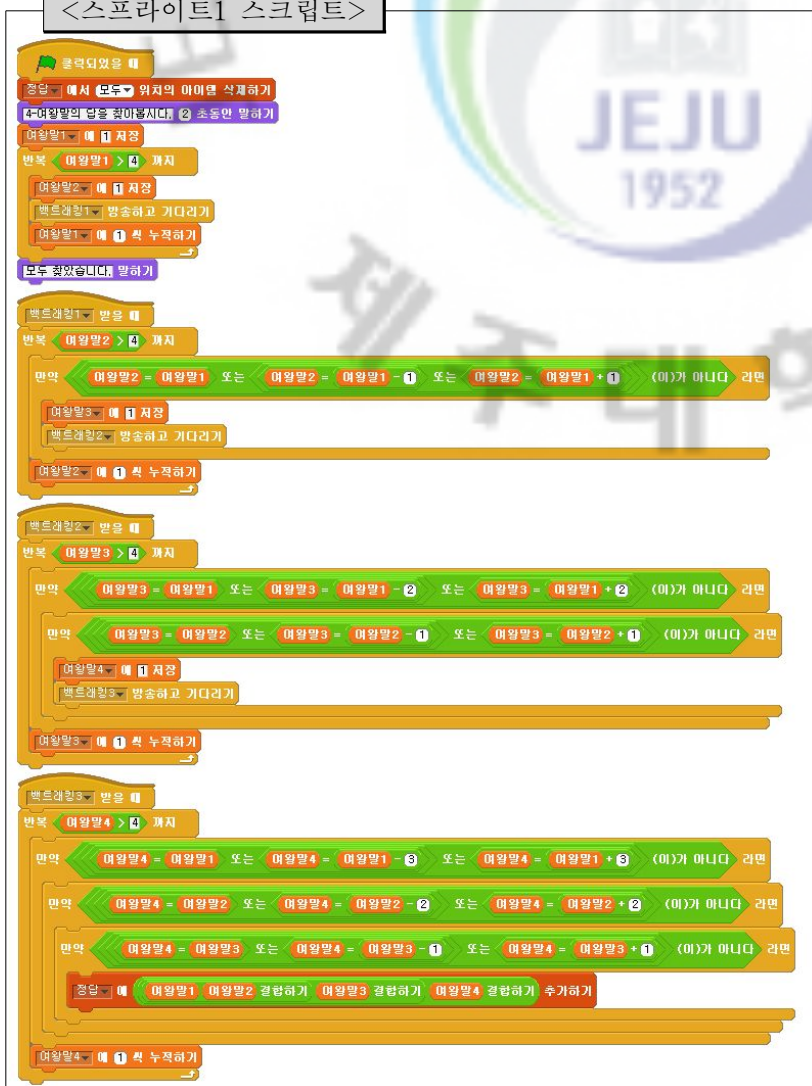
### <무대와 스프라이트 구성>



### <변수와 리스트 구성>

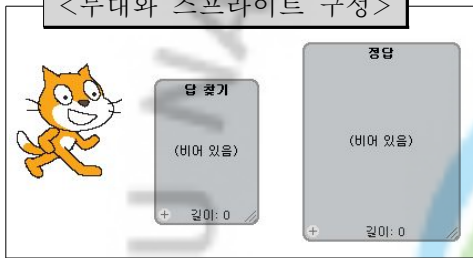
- ▷ 변수 : 여왕말1, 여왕말2, 여왕말3, 여왕말4
- ▷ 리스트 : 정답

### <스프라이트1 스크립트>



### 《 활동3 프로그램 》

#### <무대와 스프라이트 구성>



#### <변수와 리스트 구성>

- ▷ 변수 : 여왕말1, 여왕말2, 여왕말3, 여왕말4
- ▷ 리스트 : 답 찾기, 정답

#### <스프라이트1 스크립트>



## IV. 연구 방법 및 현장 적용

### 1. 연구 가설

본 연구의 목적은 초등학교 알고리즘 수업에서, 스크래치를 활용한 수업이 학습자의 논리적 사고력에 미치는 영향을 분석하고자 하는 것이다. 이러한 연구 목적 달성을 위해 설정한 가설은 다음과 같다.

연구 가설 : 스크래치를 활용한 알고리즘 교육은 초등학교 학습자의 논리적 사고력에 유의한 영향을 미친다.

### 2. 연구 대상

본 연구의 대상은 제주도에 위치한 W초등학교 4학년 2개 반을 각각 실험집단과 비교집단으로 구성하였다. 모든 연구 대상은 스크래치와 알고리즘에 대한 사전지식이나 기초적 소양이 없는 집단이다.

<표 IV-1> 연구대상 집단과 사례 수

실험집단			비교집단			계
남	여	전체	남	여	전체	
15명	15명	30명	15명	15명	30명	60명

### 3. 연구 설계 및 절차

#### 가. 연구 설계

본 연구에서는 실험집단과 비교집단을 임의로 선정하여 실시하는 이질 비교집단 전후검사 설계를 사용하였다. 연구대상으로 실험집단과 비교집단을 먼저 선정하여 사전검사를 통해 실험집단과 비교집단이 동질집단임을 보이고, 실험처치 후



사후검사를 실시하여 실험 효과 여부를 분석하였다. 이러한 연구의 실험설계를 도식화 하면 다음과 같다.

<표 IV-2> 연구의 실험설계

실험집단	O <sub>1</sub>	X <sub>1</sub>	O <sub>2</sub>
비교집단	O <sub>3</sub>	X <sub>2</sub>	O <sub>4</sub>

O<sub>1</sub>, O<sub>3</sub> : 사전검사(논리적 사고력 검사)

O<sub>2</sub>, O<sub>4</sub> : 사후검사(논리적 사고력 검사)

X<sub>1</sub> : 스크래치 활용 알고리즘 수업

X<sub>2</sub> : 컴퓨터소양교육 위주의 ICT활용 수업

#### 나. 연구 절차

실험처치 전 연구대상으로 선정된 집단의 동질성 여부를 검증하기 위해 논리적 사고력을 측정하기 위한 사전검사를 실시하였다. 실험처치는 초등학교 4학년을 대상으로 8차시에 걸쳐 진행되었다. 실험집단에는 스크래치를 활용한 알고리즘 수업을 실시하고, 비교집단은 컴퓨터소양교육 위주의 일반적인 ICT활용 수업을 실시하였다. 실험처치 이후에 논리적 사고력 검사를 실시하여 각 수업이 논리적 사고력에 미치는 영향을 알아보았다.

#### 4. 연구도구

논리적 사고력 검사 도구로 Roadranka 등(1983)이 개발한 GALT검사지 축소본을 사용하였다. 해당 검사지의 각 문항은 답과 이유를 선택하는 선다형 형식으로 6개의 논리적 사고력 하위요소를 측정하도록 구성되었다. 보존논리, 비례논리, 변인통제논리, 확률논리, 상관논리, 조합논리에 대해 각각 2문항씩 총 문항 수는 12 문항이다. 대부분의 문항에 문제 상황을 설명하는 짧은 문장과 함께 그림을 제시하였고 검사 소요시간은 40분이며 답과 이유가 모두 맞을 경우 1점을 부여하였다.

## 5. 현장 적용 결과 및 해석

### 가. 논리적 사고력 사전검사

실험처치 전 실험집단과 비교집단의 논리적 사고력 하위요소 각각의 점수에서 유의한 차이가 존재하는지 알아보기 위해 독립표본 t 검정을 실시하였다. 그 결과는 다음과 같다.

<표 IV-3> 논리적 사고력 사전검사 결과

하위 요소	집단	n	M	SD	t	p
보존논리	실험집단	30	.90	.607	-1.035	.305
	비교집단	30	1.07	.640		
비례논리	실험집단	30	.13	.346	-.356	.723
	비교집단	30	.17	.379		
변인통제 논리	실험집단	30	.27	.521	.553	.583
	비교집단	30	.20	.407		
확률논리	실험집단	30	.10	.305	.460	.647
	비교집단	30	.07	.254		
상관논리	실험집단	30	.10	.305	-.750	.456
	비교집단	30	.17	.379		
조합논리	실험집단	30	.93	.691	.853	.397
	비교집단	30	.77	.817		
논리합계	실험집단	30	2.43	1.165	.000	1.000
	비교집단	30	2.43	1.135		

\*  $p < .05$ , \*\*  $p < .01$

실험집단은 비교집단에 비해 6개의 하위 요소 중 변인통제논리, 확률논리, 조합논리에서 평균점수가 높았고, 보존논리, 비례논리, 상관논리에서 평균점수가 낮게 나타났으나 통계적으로 유의한 차이는 없었다. 또한 전체 논리합계 평균점수가 각각 2.43( $\pm 1.165$ ), 2.43( $\pm 1.135$ )으로 두 집단 간 유의한 차이가 없었다. 따라서 각 항목에 대해서 유의한 차이가 없으므로 두 집단이 동질집단임이 확인되었다.

## 나. 논리적 사고력 사후검사

실험처치 후 실험집단과 비교집단의 논리적 사고력 각각의 하위요소들의 변화가 두 집단 간 통계적으로 유의한 차이를 보이는지 알아보기 위해 독립표본 t 검정을 실시하였다. 그 결과는 다음과 같다.

<표 IV-4> 논리적 사고력 사후검사 결과

하위 요소	집단	<i>n</i>	<i>M</i>	<i>SD</i>	<i>t</i>	<i>p</i>
보존논리	실험집단	30	1.07	.521	.646	.521
	비교집단	30	.97	.669		
비례논리	실험집단	30	.07	.254	-.460	.647
	비교집단	30	.10	.305		
변인통제 논리	실험집단	30	.40	.498	3.265	.002**
	비교집단	30	.07	.254		
확률논리	실험집단	30	.20	.551	.273	.786
	비교집단	30	.17	.379		
상관논리	실험집단	30	.03	.183	-2.047	.047*
	비교집단	30	.20	.407		
조합논리	실험집단	30	1.60	.563	5.006	.000**
	비교집단	30	.87	.571		
논리합계	실험집단	30	3.37	1.098	3.385	.001**
	비교집단	30	2.37	1.189		

\* $p < .05$ , \*\* $p < .01$

실험처치 후 전체 논리합계의 대한 실험집단과 비교집단의 평균점수는 각각 3.37(±1.098)과 2.37(±1.189)로 실험집단의 점수가 높게 나타났으며, 통계적으로 유의한 차이를 보였다. 그리고 변인통제논리의 평균점수는 실험집단 0.40(±0.498), 비교집단 0.07(±0.254)로 실험집단의 평균점수가 높았고, 이는 통계적으로 유의한 차이를 보였다. 조합논리의 평균점수 또한 실험집단 1.60(±0.563), 비교집단 0.87(±0.571)로 실험집단이 높았으며 통계적으로 유의한 차이를 보였다.

그러나 보존논리, 비례논리, 확률논리에 대해서는 유의한 차이를 보이지 않았다. 또한 상관논리에서는 오히려 비교집단의 평균점수 0.20(±0.407)이 실험집단의 평균점수 0.03(±0.183)보다 높게 나타났으며 이는 통계적으로 유의하였다.

따라서 본 연구에서 개발된 스크래치 활용 알고리즘 수업이 기존의 컴퓨터소양 교육 위주의 ICT활용 수업에 비해 변인통제논리와 조합논리 그리고 전체적인 논리적 사고력에 긍정적인 영향을 주는 것으로 확인되었다.

#### 다. 집단별 논리적 사고력 사전·사후검사 차이 검증

실험집단과 비교집단의 각각의 집단 내 실험 처지 전후를 비교하기 위해 대응표본 t 검정을 실시하였다. 검증 결과는 다음과 같다.

<표 IV-5> 논리적 사고력 사전·사후검사 차이 검증

하위 요소	집단	검사 시기	M	SD	t	p
보존논리	실험집단	사전검사	.90	.607	-1.980	.057
		사후검사	1.07	.521		
	비교집단	사전검사	1.07	.640	.769	.448
		사후검사	.97	.669		
비례논리	실험집단	사전검사	.13	.346	1.000	.326
		사후검사	.07	.254		
	비교집단	사전검사	.17	.379	.701	.489
		사후검사	.10	.305		
변인통제 논리	실험집단	사전검사	.27	.521	-1.439	.161
		사후검사	.40	.498		
	비교집단	사전검사	.20	.407	1.682	.103
		사후검사	.07	.254		
확률논리	실험집단	사전검사	.10	.305	-.828	.415
		사후검사	.20	.551		
	비교집단	사전검사	.07	.254	-1.140	.264
		사후검사	.17	.379		
상관논리	실험집단	사전검사	.10	.305	1.000	.326
		사후검사	.03	.183		
	비교집단	사전검사	.17	.379	-.297	.769
		사후검사	.20	.407		
조합논리	실험집단	사전검사	.93	.691	-5.135	.000**
		사후검사	1.60	.563		
	비교집단	사전검사	.77	.817	-.902	.375
		사후검사	.87	.571		
논리합계	실험집단	사전검사	2.43	1.165	-4.474	.000**
		사후검사	3.37	1.098		
	비교집단	사전검사	2.43	1.135	.268	.791
		사후검사	2.37	1.189		

\*  $p < .05$ , \*\*  $p < .01$

실험집단의 조합논리는 사전검사와 사후검사 점수가  $0.93(\pm 0.691)$ 과  $1.60(\pm 0.563)$ 으로 사후검사가 높게 나타났으며, 이는 통계적으로 유의한 차이를 나타냈다. 그러나 비교집단의 경우 사전검사와 사후검사 점수가  $0.77(\pm 0.817)$ 과  $0.87(\pm 0.571)$ 로 점수가 약간 높아졌으나 이는 통계적으로 유의하지 않았다.

실험집단의 전체적인 논리합계는 사전검사와 사후검사 점수가  $2.43(\pm 1.165)$ 과  $3.37(\pm 1.098)$ 로 사후검사가 높게 나타났으며, 이는 통계적으로 유의하였다. 그러나 비교집단의 경우 사전검사와 사후검사 점수가  $2.43(\pm 1.135)$ 과  $2.37(\pm 1.189)$ 로 점수가 다소 낮아졌으며 이는 통계적으로 유의하지 않았다.

보존논리, 비례논리, 변인통제논리, 확률논리, 상관논리에 대해서는 실험집단과 비교집단의 각각의 사전검사와 사후검사의 평균점수 비교 시 통계적으로 유의한 차이를 보이지 않았다.

따라서 스크래치 활용 알고리즘 수업은 조합논리와 전체적인 논리적 사고력을 신장 시키는데 효과적이거나, 기존의 컴퓨터소양교육 위주의 ICT활용 수업은 논리적 사고력에 유의한 변화를 미치지 않는 것으로 분석되었다.

이상에서 살펴본 바와 같이 스크래치 활용 알고리즘 수업이 논리적 사고력을 신장시키는데 효과적임이 확인되었다. 위에서 비교 검증한 항목들 중 전체적인 논리적 사고력 점수에 대한 것을 종합적으로 정리하면 다음과 같다.

<표 IV-6> 논리적 사고력 검증 종합

독립표본 t 검정 대응표본 t 검정	전체적인 논리적 사고력 점수	
	실험집단	비교집단
사전검사 점수	2.43	2.43
사후검사 점수	3.37	2.37

\*  $p < .05$ , \*\*  $p < .01$

## VI 결론 및 제언

누구도 앞날을 예측하지 못한다. 오늘날 IT강국이라 자부하는 우리나라지만, 미래에는 어떻게 될지 알 수가 없다. 하지만 조심스레 예측해보면 그다지 밝지만은 않다. 이렇게 급박하게 변화하는 시대에 다른 나라들은 정보교육의 중요성을 이미 깨닫고 정부차원에서 지원하며 나아가고 있지만, 우리나라의 정보교육은 아직도 제대로 된 발판 하나 마련하지 못하고 있다.

최근 정보교육에서 컴퓨터 과학 교육의 중요성이 부각되고 하나의 학문으로 발전하고 나아가려 하고 있지만, 우리나라에는 그것에 대한 인식과 관심 부족으로 적절한 교재도 부족하며 그에 따라 학교현장에서는 아직도 컴퓨터 소양교육과 활용교육 위주의 수업이 이루어지고 있다.

이에 따라 본 연구에서는 컴퓨터 과학 교육에서 강조하는 알고리즘 학습을 위해 교육용 프로그래밍 언어인 스크래치를 활용하여 교재를 개발하였다. 그리고 그 교재를 학교 현장에 적용하여 그 효용성을 살펴보았다.

본 연구를 통해 기대되는 효과는 다음과 같다.

첫째, 학생들이 스크래치 프로그래밍을 통해 알고리즘적 사고에 익숙해서 논리적 사고력뿐만 아니라 문제해결력이 향상될 것이다.

둘째, 교육용 프로그래밍 언어와 컴퓨터의 원리에 대한 학생들의 이해와 관심이 많아지면서 단순한 놀이 도구로서가 아니라 학습의 도구로서의 컴퓨터의 가치를 확인하게 해 줄 것이다.

셋째, 기능 위주에서 알고리즘 원리 위주로의 프로그래밍 언어 교육에 대한 인식의 변화와 함께 기본 교재로 활용될 것이다.

넷째, 교육용 프로그래밍 언어에 대한 관심과 가치가 높아지고 그에 따라 컴퓨터 과학 교육의 중요성과 필요성이 높아질 것이다.

이상의 연구 결과를 바탕으로 다음과 같이 제언하고자 한다.

첫째, 컴퓨터 과학 교육에서 강조하는 교육용 프로그래밍 언어와 알고리즘 원리를 접목한 교육 자료들이 많이 개발되고 활용되어야 할 것이다.

둘째, 정선된 교육과정을 통해 체계적인 교육이 이루어질 수 있도록 교육내용 및 교수방법에 대해서도 많은 연구가 필요하다.

셋째, 이러한 교육내용과 교수방법이 잘 적용될 수 있도록 교육 환경 개선 및 지도교사들의 연구와 연수를 통한 노력이 필요할 것이다.

하루빨리 이러한 노력들이 한데 어우러져 바람직한 정보교육이 이루어지는 학교 현장을 기대해 본다.

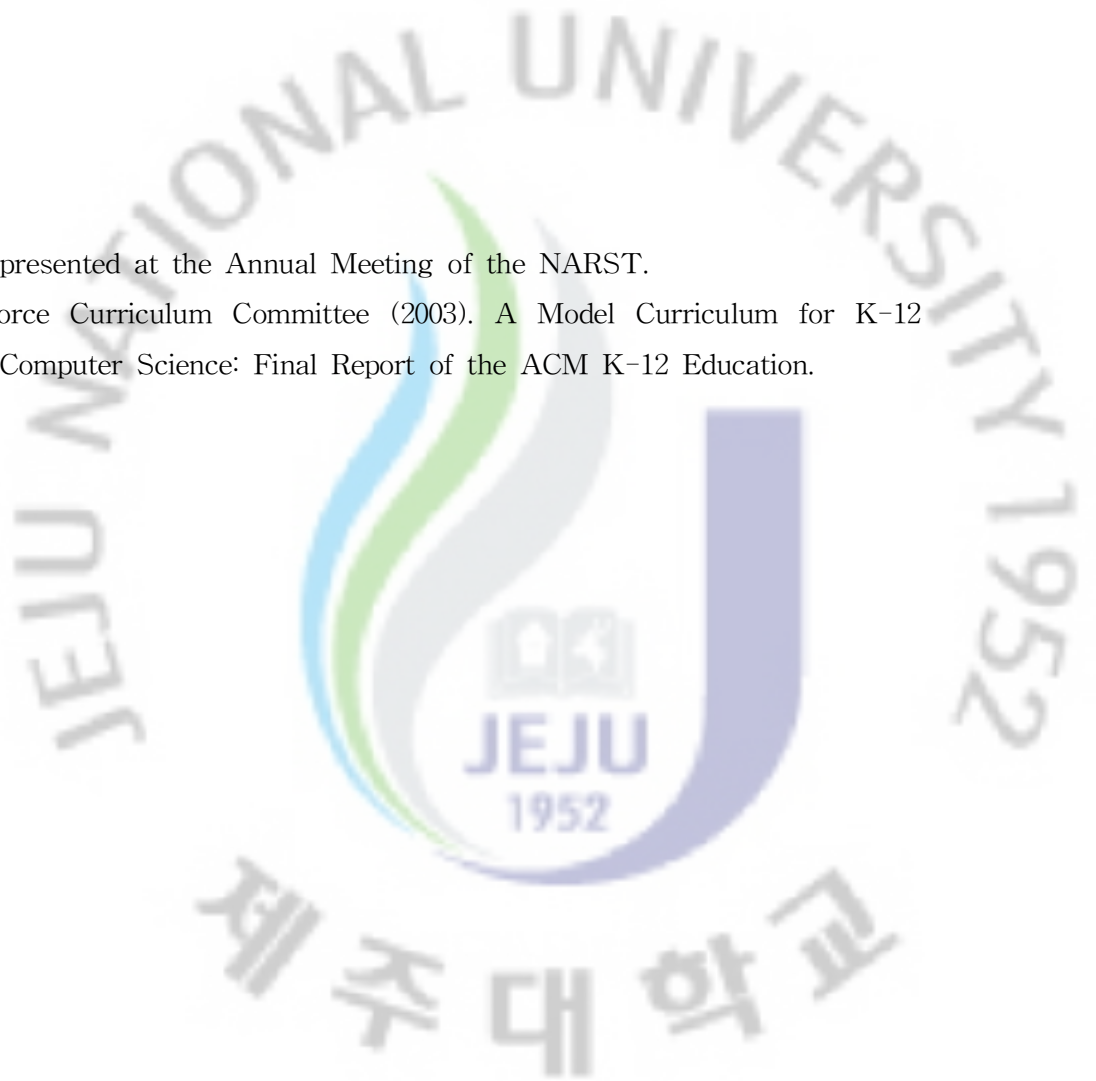
## 참 고 문 헌

- 교육인적자원부. (2008.) **중학교 교육과정 해설(V)**.
- 김경미. (2004). 객체지향형 교육용 프로그래밍 언어 ‘두리틀(Dolittle)’의 수학교육 활용. 고려대학교 석사학위 논문.
- 김경미. (1999). 완본과 축소본 GALT의 비교 연구. 한국교원대학교 석사학위 논문.
- 김정훈. (2006). 중학교 알고리즘 교육 내용의 위계 설정에 관한 연구. **한국컴퓨터 교육학회 논문지**, 9(5). 41-51
- 김수환, 김교준, 김승범, 이원규, 김현철. (2008). 교육용 프로그래밍 언어의 동향과 교육적 적용 방안. **한국정보과학회 학술발표논문집**, 35(1). 375-376.
- 김종훈. (2008). **프로그래밍 비타민**. 서울: 한빛미디어.
- 이영준, 이은경. (2010). **상식의 실현을 위한 스크래치 프로그래밍**. 서울: 교학사.
- 이영준, 이은경. (2008). Scratch 활용 프로그래밍 교육이 중학생의 몰입수준과 프로그래밍 능력에 미치는 영향. **중등교육연구**, 56(2). 359-382
- 이옥화. (1993). 로고 프로그래밍의 교육적 의의와 실천방안 모색. **교육공학 연구**, 8(1). 81-102.
- 이좌택. (2004). 문제기반학습에 터한 로봇 제어 프로그래밍 수업이 중학생의 논리적 사고력에 미치는 효과. 한국교원대학교 박사학위 논문.
- 정미연, 이영준, 이은경. (2008). Squeak Etoys 활용 알고리즘 학습이 중학생의 문제해결력에 미치는 영향. **대한공업교육학회지**, 33(2). 170-191
- 한국교육개발원. (2002). **초·중학생의 지적·정의적 발달수준 분석연구(III)**.
- ACM (1991) 대학의 컴퓨터 과학 커리큘럼.  
<http://www.computer.org/education/cc2001/index.htm>
- East, P., & Hutton, M. (2007). Point/Counter-Point: On Using Graphical Languages to Teach
- Judith Gal-Ezer, David Harel (1999). What (Else) Should CS Educators Know?. *Communications of the ACM* 41(9)
- Roadrangka, V., Yeany, R. H. & Padilla, M. J. (1983). The Construction and validation of group assessment of logical thinking(GALT). Paper



presented at the Annual Meeting of the NARST.

Task Force Curriculum Committee (2003). A Model Curriculum for K-12  
Computer Science: Final Report of the ACM K-12 Education.



## ABSTRACT\*

### Study on Development and Application of Algorithm Learning Program for Thinking Ability Enhancement Using Scratch

Yang, Young Hoon

Major in Elementary Practical Computer Education  
Graduate School of Education  
Jeju National University

Supervised by Professor Kim, Jong Hoon

The value and importance of Information Education are increasing in current society. In advanced countries, the Information Education has been conducted as form of the systematic curriculum to develop creative and reasonable human resources. In Korea, that has been conducted as a part of the regular curriculum on the basis of awareness about its importance and necessity. In content of Information Education, Computer Science Education is more emphasized than the Computer Literacy Education and simple computer use in education. Computer Science Education is focused on development of creative and logical thinking ability through understanding the computer's basic

---

\* A thesis submitted to the committee of Graduate School of Education, Jeju National University in partial fulfillment of the requirements for the degree of Master of Education conferred in February, 2011.

principles and problem solving procedures.

In this study, we developed the teaching material for algorithm learning in Computer Science Education using educational programming language, SCRATCH. That teaching material was applied to the children from W elementary school. And the results showed that was effective on enhancement of logical thinking. In the future, the development of teaching materials using educational programming language is more needed for enhancement of Computer Science Education. And improvement of education Environment and the efforts of teachers will be need. also selection of courses can be made through a systematic curriculum and teaching methods is necessary to study a lot about.