https://doi.org/10.32431/kace.2018.21.6.004

최소생성사다리를 생성하는 알고리즘 구현 및 컴퓨팅 사고력과의 관련성 탐구

전영국†

요 약

이 연구는 사다리타기 게임에서 등장하는 사다리 모양에 따른 이산구조를 순열과 조합적 사고, 알고리즘적 구현을 통하여 최소생성사다리를 생성하는 방법과 컴퓨팅 사고력과의 관련성을 탐구하는 내용을 다루었다. 먼저 연구자는 사다리 모양의 세로판과 가로판의 조합에 따라서 생성되는 순열 중에서 역순열에 대응하는 사다리(최소생성사다리)를 필터링 기법과 새로 개선한 알고리즘을 고안하여 Mathematica 프로젝트로 진행하였다. 그 결과 최소생성사다리를 생성원(generator)으로 하는 새로운 그래프를 Mathematica로 창출하여 YC그래프라 이름 붙였으며 그에 대한 속성을 조사하였다. YC그래프는 이전 차원의 그래프를 내포하는 재귀적 구조와 다층 구조를 가졌으며 간선대칭의 특징을 보여주었다. 또한 계산복잡도가 증가함에 따라 세로판 5개, 가로판 10개 사다리부터 층별로 최소생성사다리를 생성하도록 탐색 공간을 분할하는 알고리즘을 적용하였다. 이 과정에서 자료의 시각화, 추상화 및 병렬처리 알고리즘 구현을 통한 컴퓨팅 사고력이 새로운 YC그래프의 창출 및 구조 분석에 기여한 것으로 나타났다.

주제어: 최소생성사다리, 그래프 구조, Mathematica 프로그래밍, 컴퓨팅 사고력

Implementation of an Algorithm that Generates Minimal Spanning Ladders and Exploration on its relevance with Computational Thinking

Youngcook Jun†

ABSTRACT

This paper dealt with investigating the number of minimal spanning ladders originated from ladder game and their properties as well as the related computational thinking aspects. The author modified the filtering techniques to enhance Mathematica project where a new type of graph was generated based on the algorithm using a generator of firstly found minimal spanning graph by repeatedly applying independent ladder operator to a subsequence of ladder sequence. The newly produced YC graphs had recursive and hierarchical graph structures and showed the properties of edge-symmetric. As the computational complexity increased the author divided the whole search space into the each floor of the newly generated minimal spanning graphs for the (5, 10) YC graph and the higher (6, 15) YC graph. It turned out that the computational thinking capabilities such as data visualization, abstraction, and parallel computing with Mathematica contributed to enumerating the new YC graphs in order to investigate their structures and properties.

Keywords : Minimal spanning ladder, Graph structure, Computational thinking, Mathematica programming

---

† 종신회원: 순천대학교 컴퓨터교육과 교수(교신저자)
논문접수: 2018년 9월 20일, 심사완료: 2018년 11월 26일, 게재확정: 2018년 11월 29일
* 이 논문은 2016년 순천대학교 학술연구비 공모과제로 연구되었음.

1. 서 론

수학을 공부하는데 추상적 개념을 다루는 것보다 일상생활에서 만날 수 있는 대상을 구체적으로 다루어보는 것은 학생들에게 상당히 매력적일 수 있다. 다루어보고자 하는 대상을 직접 손으로 써보면서 절차대로 작업해 나가다 보면 어려운 수학 내용도 하나씩 차근히 이해할 수 있게 된다. Constructive math로 알려진 수학은 개념적으로 접근하기보다 구체적으로 문제해결의 절차를 구성해 나가는 관점을 반영하여 학생들에게 수학적으로 생각한 바를 절차대로 구현하는 알고리즘에 대한 이해를 제공함으로써 점진적으로 수학적 사고력 향상을 도모할 수 있다.

일례로, 대학생들은 자신이 미적분 문제를 해결하는데 단서가 되는 아이디어를 구상하고 생각하는 대로 절차를 짜보면서 실제 컴퓨터 프로그램을 구현하면서 문제해결에 대한 능력을 향상시킬 수 있다[1]. 이러한 알고리즘적 사고를 컴퓨터 프로그래밍으로 구현하면서 단계별로 이해하는 것을 화이트박스라고 부른다[1]. 그에 비하여 컴퓨터 프로그램 코드를 컴파일하거나 실행한 상태에서 입력값에 대하여 출력 결과를 확인하는 방식을 블랙박스라고 부른다.

이러한 이유로 화이트박스 형태로 수학적 사고력을 신장시키는 학습 방식이 컴퓨터교육에서 전개되는 사례를 탐구할 필요가 증가하고 있다. 이에 이 연구는 고등학생들이 일상생활에서 내기를 하거나 복불복 형태로 어떤 것을 선택하고자 할 때 흔히 사용하는 사다리 게임에 관하여 팀별 프로젝트를 수행하였던 사례를 고찰하고자 한다. 순열과 일대일 대응을 보여주는 사다리 게임의 이산 구조는 수학과 컴퓨터 분야에서 매력적으로 등장한다[2][3][4].

이 연구는 사다리 게임의 순열에 대응하는 함수 개념을 적용한 최소생성사다리의 특성을 탐구하면서 이산수학과 컴퓨팅 사고력의 관련성을 다루고자 하였다. 구체적으로 대학 교수인 연구자는 사다리 게임을 주제로 사사 형태의 장기 프로젝트인 R&E 활동[5]에 참여하였던 고등학생들을 지도하였다. 그 결과 사다리 게임 프로젝트를 심화하는 별도의 후속 연구를 통해 최소생성사다리를 생성하는 새로운 알고리즘을 Mathematica 프로그래밍으로 구현하였고 그 결과 생성된 그래프의 속성을 분석하였다. 연구 문제는 사다리 게임에서 등장하는 최소생성사다리의 개수를 찾는 알고리즘을 향상하는 방법을 통해 등장한 새로운 그래프의 구조를 파악하면서 그 과정에서 최소생성사다리에 관한 자료의 분석 및 시각화, 추상화 작업 등을 통해 나타나는 컴퓨팅 사고력의 양상을 다루는 것이다.

2. 사다리 게임을 활용한 수업 및 컴퓨팅 사고력 소개

수학의 함수 단원에서 사다리 게임을 도입한 수업에 관한 연구를 살펴보면 사다리 게임을 치환 과정의 연속 즉, 순열로 보면서 중고등학생들에게 양질의 학습 자료로 활용한 사례를 보여주었다. 구체적으로 보면 주수아(2003)는 중학생들을 대상으로 사다리타기 게임을 적용한 수업 자료를 개발하고 적용함으로써 학생들의 수학에 대한 흥미 유발과 태도 변화 등을 이끌어 내었다[3]. 한발 더 나아가 한수선(2014)은 중학생들을 대상으로 함수 단원에서 사다리타기 게임을 사용한 함수 지도에 관한 교수-학습 자료를 개발하였다[4].

또한 이광연과 동료들은 고등학교 함수 단원에서 사다리 게임을 도입하기 위하여 사다리 게임의 함수적 속성을 분석하였으며 사다리타기 게임을 통한 함수 지도 방안을 제시하였다[2]. 그들은 사다리 타기 게임에 대하여 수학적으로 접근함으로써 수학 내적인 부분에 대하여 집중할 수 있도록 하였으며 학습 활동의 난이도가 증가할 것을 고려하여 팀별 활동을 제안하였다. 함수의 관점에서 치환을 사용하여 사다리 타기 게임을 수학적으로 분석한 다른 논문은 이산수학적 관점에서 사다리 게임과 관련되는 속성을 탐구한 바 있다[6].

한편, 알고리즘적 사고는 ‘컴퓨팅 사고력’의 개념으로 널리 알려지고 있다. 미국 컴퓨터교사협의회(CSTA Standards Task Force)에서 제시한 기준에 따르면 “컴퓨팅 사고력은 컴퓨터를 활용해 구현할 수 있는 방법으로 문제를 해결해 나가는 접근 방법이다”라고 정의하였다[8]. 국내에서 2015년에 개정된 교육과정은 다음과 같이 컴퓨팅 사고력의 하위 영역을 제시하였다. 즉, 컴퓨팅 사고력은 1) 자료 표현 및 시각화 2) 순열, 조합, 함수 치환 등과 관련된 추상화 (문제분석, 문제분해와 모델링) 및 병렬처리 3) 알고리즘의 설계 및 분석을 통해 컴퓨터 프로그래밍으로 구현하여

문제를 해결하는 능력과 관련되어 있다. 다시 말하면 학생들은 문제해결에 관련되는 논리적이고 계산적인 절차를 알고리즘으로 구상한 후에 컴퓨터 프로그래밍으로 그 문제를 해결하는 역량을 강화시켜야 한다. 이때 컴퓨터 프로그램을 할 때 그들은 배열 등 자료형태를 구분하여 변수와 연산자를 제어구조 속에 배치하고 함수 및 서브루틴을 작성하여 테스트하고 디버깅 하는 절차를 수행하게 된다.

이처럼 알고리즘적 사고는 화이트박스처럼 문제해결에 대한 절차를 단계별로 수행하도록 하는 코딩 작업을 통해 컴퓨팅 사고력을 향상시킬 수 있다는 관점을 분명히 보여준다[7]. 이와 같이 학생들은 컴퓨터 프로그래밍을 해 보면서 테스트해 봄으로써 명료하고 정확한 사고를 하게 되고, 더 나아가 알고리즘에 대한 이해와 디버깅을 통한 논리적 오류를 발견하고 스스로 수정할 수 있게 된다. 이러한 과정에서 그들은 가능한 모든 경우를 고려하는 조합적 사고와 논리적으로 미세한 부분을 검토하는 능력 또한 확대시킬 수 있다. 이러한 컴퓨팅 사고력을 신장시키기 위하여 이유로 컴퓨터 교육과정에서 프로그래밍을 활용한 교육이 매우 중요하게 대두되고 있다[8][9].

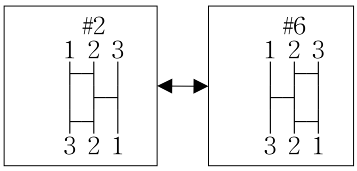
3. 사다리 게임과 관련된 R&E 프로젝트

연구자는 대학부설 과학영재교육원에서 교수 일인당 네 명의 학생들과 함께 장기간 과학탐구를 하는 사사과정을 지도해 본 경험을 갖고 있었다[12]. 이와 유사한 형태로 연구자는 전남 과학고의 요청으로 고등학생들이 사사과정의 일환으로 연구 및 교육활동(Research & Education)을 진행하게 되었다. 2015년 봄 학기에 참가하였던 고등학교 2학년 학생들(남: 3명, 여: 1명)은 사다리의 모양에 따라서 특정 순열이 나타나는지에 대한 관찰을 하면서 치환 연산을 통해 사다리 게임에 대한 다양한 성질을 조사하고자 하였다[10].

3.1 사다리 타기 게임에 대한 수학과 컴퓨터 프로그래밍을 사용한 접근

참여 학생들은 사다리 게임을 통해 나타나는 순열의 결과를 C 프로그래밍으로 조사하면서 사다리 모양에서 나타나는 이산구조를 순열 자료구조로 다루고 순열을 치환하는 일대일 대응 알고리즘을 구현하여 그 결과를 순열과 시각적 모양으로 나타나도록 출력하였다[13]. 이 활동을 통하여 학생들은 사다리의 가로판 개수가 짝수이면 짝순열, 홀수 개이면 홀순열이 나온다는 사실을 알게 되었다. 또한 가로판이 n개, 세로줄이 m개일 때 사다리의 개수는 $(n-1)^m$이다. 일례로 3개의 세로판과 3개의 가로판을 가진 사다리의 경우 모두 8개의 사다리 모양을 얻게 된다. 참여 학생들은 이러한 결과를 직접 손으로 작성하여 알아내는 방법과 C 프로그래밍을 통해 이중으로 검토하였다.

그들은 사다리의 세로판 개수와 가로판의 개수 사이의 관련성을 살펴보았고 C 프로그래밍을 사용하여 사다리 게임의 결과에 해당되는 빈도수를 조사하였다. 빈도수를 분석한 결과 중복되는 원소를 발견하였고 이를 제거하는 방식을 탐구하였다. 이 과정에서 중요한 역할을 하는 것은 <그림 1>의 #2와 #6처럼 서로 대칭되면서 {1,2,3}에 대한 역순열인 {3,2,1}을 대응시켜 주는 사다리이다. 특이한 사다리를 최소생성사다리라고 부른다. 세로판의 개수가 4 이상으로 늘어날 때 주위에 있는 가로판의 영향을 받지 않으면서 독립적으로 존재하는 이러한 ‘변환 대상’을 독립부분사다리라고 부른다[6]. 그리고 독립부분사다리를 <그림 1>처럼 변환시키는 연산자를 독립사다리연산이라고 한다.



<그림 1> 독립부분사다리

기존 연구에서 R&E 프로젝트에 참여하였던 학생들은 C 프로그래밍을 사용하여 초기 순열(예, 1234)을 사다리 타는 게임의 경우를 모두 탐색한 후에 그 결과값이 역순으로 정렬되는 사다리만 나타내고, 그렇지 않은 사다리들은 삭제하였다[10]. 다시 말하면 가로판이 n개, 세로줄이 m개일 때 $(n-1)^m$개의 전체 탐색 공간에서 초기 순열의 역순열에 해당되지 않는 사다리를 제거시키는 필터링 알고리즘을 적용하였다. 그들은 C 프로그램을 테스트한 결과 다음과 같이 최소생성사다리의 수를 구하였다.

<표 1> 최소생성사다리의 개수

세로판	가로판	최소생성사다리
3	3	2
4	6	8
5	10	62
6	15	908

3.2 사다리 게임 프로젝트와 컴퓨팅 사고력과의 관련성

앞서 소개한 R&E 프로젝트 활동은 순열과 조합을 이용하여 단계별로 논리적 사고를 전개하면서 컴퓨터 프로그래밍(예, C 언어)으로 구현하는 화이트박스 기반의 알고리즘 학습의 특징을 보여준다[1]. 부연하면 참여 학생들은 최소생성사다리의 개수를 찾는 과정에서 다음과 같이 알고리즘적 사고를 C 프로그래밍으로 구현하였다[14]. 첫째, 그들은 사다리 타기 게임에서 나타나는 순열과 조합, 사다리 그래프 등 이산적 구조를 사용하여 최소생성사다리의 특징을 포착하였다. 둘째, 최소생성사다리를 찾는데 필요한 조합적 생성과 중복되는 연산의 결과를 제거하는 필터링 알고리즘을 고안하였고 C 프로그래밍으로 구현하여 그들이 예측한 바를 검증하였다. 이 과정에서 그들은 역할 분담을 통해 손으로 계산을 하면서 필터링 알고리즘을 C 언어로 구현하는 등 컴퓨팅 사고력과 관련된 역량을 신장시키는 모습을 보여주었다.

3. 최소생성사다리를 정점으로 하는 새로운 그래프의 생성과 구조 탐색

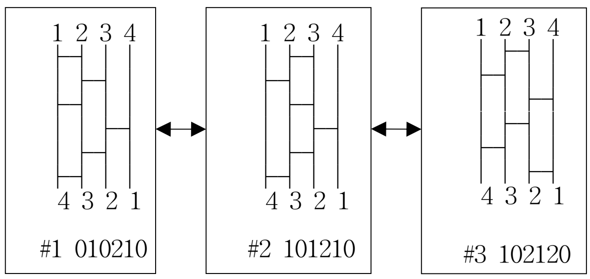
4.1 최소생성사다리의 재귀적 패턴 탐구

연구자는 앞서 소개한 R&E 고등학생들을 지도하면서 별도로 최소생성사다리를 정점으로 하는 새로운 그래프의 생성에 주목하면서 그에 관련된 구조를 탐색하였다. Mathematica를 통하여 탐구한 바에 따르면 5-사다리의 경우에 최소생성사다리에 필요한 가로판의 개수는 10개이며 모두 62개가 생성되었다. <표 1>에 제시된 바와 같이 사다리 게임의 세로판이 하나씩 추가될 때마다 차원이 추가되는 양상을 보여주고 있으며 그에 따라 가로판의 개수가 결정되고 최소생성사다리의 개수도 결정되는 패턴을 보여주었다.

연구자는 62개의 정점을 가진 새로운 그래프의 구조에 주목하면서 (4, 6) 사다리의 최소생성사다리의 집합과 비교하는 작업을 수행하였다. 재귀적 패턴이 나타나는 것에 착안하여 Mathematica 프로그래밍으로 데이터 시각화와 알고리즘에 대한 추상화 작업을 하면서 탐구한 결과 최소생성사다리를 정점으로 하는 새로운 그래프를 발견하여 YC그래프라는 이름을 붙였다. 연구자가 수행하였던 자료분석 및 표현, 추상화, 알고리즘의 설계, Mathematica 프로그래밍으로 구현, 테스트한 과정를 다음에 제시하면서 컴퓨팅 사고력에 관하여 탐구하고자 하였다.

4.2 알고리즘 설계 및 Mathematica를 이용한 사고적 계산 수행

연구자는 <그림 3>에 제시하였던 (3, 3) 사다리의 왼쪽 최소생성사다리를 '010'으로 표시하고 오른쪽을 '101'로 변환하는데 착안하였다. 이 방식을 사용하면 (4, 6) 사다리의 경우에 가로판의 조합에 따라 생성된 모든 사다리 중에서 <그림 4>에서 보듯이 필터링 기법을 사용하여 '4321' 순열을 생성하는 8개의 최소생성사다리를 찾을 수 있다. R&E 고등학생들을 지도하면서 연구자는 다음과 같이 필터링 알고리즘 A를 Mathematica로 구현하였다. 구체적으로, n개의 중복되지 않는 숫자로 이루어진 순열 전체의 집합 [n]에서 역순열을 생성해 주는 사다리 집합을 골라서 같은 모양을 제거하는 필터링 방식을 사용하였다. 이 알고리즘은 순열, 치환, 필터링을 사용하는 비교적 간단한 장점을 가진 반면에 n이 증가할수록 메모리와 연산의 복잡도가 증가하는 O(2^n)의 복잡도를 가지고 있다.



<그림 2> 최소생성사다리에 대한 수열 표기

<그림 2>에 보면 '010210' 사다리에서 '101210'으로 다시 '102120'으로 수열을 치환하는 방법을

사용하여 8개의 최소생성사다리를 모두 생성할 수 있다. 이 방식은 수열 중에서 독립사다리연산을 적용할 부분 수열을 찾아서 1을 더하거나 빼는 연산을 적용하여 부분적으로 치환시켜 그래프를 생성하는 문제로 바뀌었다[9].

다시 말하면, (n, k) 사다리 중에서 최초 발견한 최소생성사다리 중에서 '변환 대상'을 찾아서 독립사다리연산을 계속 적용함으로써 최소생성사다리의 집합을 모두 찾을 수 있었다. 사다리 모양에서 흥미로운 것은 102120과 102102는 서로 동형인 점이다. 즉 이웃하는 숫자와의 차이가 2 이상 나는 것은 서로 자리바꿈이 가능하다는 것을 의미한다. 102102의 경우에 102120로 바꾸고 부분 수열 중에서 212를 취하여 전환사다리 연산을 적용하여 121로 바꾼 뒤에 101210을 얻게 된다. 이와 같은 방식으로 최소생성사다리를 정점으로 하여 전환사다리 연산자를 적용하면서 새로운 최소생성사다리를 만들어 나가며 부모와 자식관의 관계를 간선으로 연결하여 전체 그래프를 생성하게 된다.

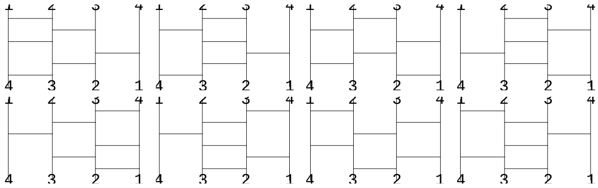
여기서 연구자가 설계 및 구현한 알고리즘은 다음과 같다. 큐브 다면체를 시작 노드로부터 하위 자식(최소생성그래프)을 만들어 나가는 방식을 채택하였다. (4, 6) 사다리의 경우에 '010210' 사다리에서 '101210'으로 수열을 치환하는 방법을 사용하여 8개의 최소생성사다리를 생성하는 그래프를 만들었다. 이를 확장하여 (5, 10) 사다리의 경우에 독립사다리연산을 이용하여 수열의 부분 수열을 치환하는 방식으로 최소생성사다리의 집합을 모두 구하였다.

구체적으로 하이퍼큐브에 트리구조를 임베딩하는 것처럼 다원트리 형태의 그래프 탐색 방법을 적용하여 다음과 같이 알고리즘 B를 설계하였다.

입력: 사다리 게임의 세로판과 가로판의 개수
1) 순열에서 역순열로 가게 하는 최소생성사다리 한 개 (예, <그림2>의 #1) 발견
2) 현재의 최소생성사다리의 부분 사다리 중에서 독립전환사다리 연산자를 적용함
3) 전체 공간에서 너비우선탐색 경로로 이동하면서 더 이상 새로운 자식 노드가 발견되지 않을 때가지 2)의 연산을 반복적으로 적용함

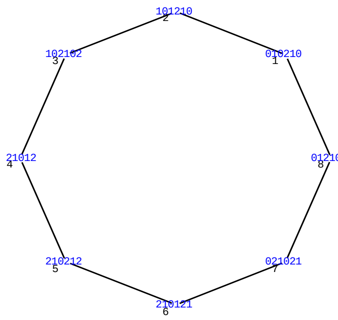
출력: 최소생성사다리의 집합과 대응되는 간선으로 만들어진 새로운 YC그래프

참고로 아래 <그림 3>은 Mathematica 5.2 버전으로 drawAllLadders[3,3,4]를 실행하여 (4, 6) 사다리의 최소생성사다리 모두 출력한 화면을 보여준다.



<그림 3> (4, 6) 사다리의 최소생성사다리

연구자는 알고리즘 A를 Mathematica로 구현하여 (4, 6) 사다리에서 최소생성사다리 8개를 찾아서 출력한 결과 <그림 4>에서 보듯이 팔각형의 모양을 확인하였다.



<그림 4> (4, 6) YC그래프

이 YC그래프는 1층부터 4층까지 2, 2, 2, 2개의 노드를 가지고 있는 계층적 구조를 보여준다. 또한 정점의 분지수가 2인 정규그래프이며 girth 8, 지름 4, 회밀턴 회로를 가지고 있으며 간선 대칭 그래프의 속성을 갖고 있다. 여기서 지름은 임의의 두 정점 간의 최소 거리의 최대값이며 girth는 최소사이클의 경로의 개수이다. 이를 토대로 수정한 알고리즘 C는 다음과 같다.

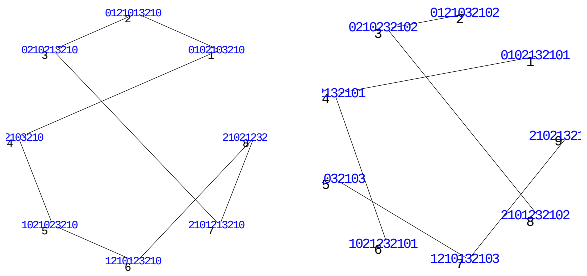
입력: 사다리 게임의 세로판의 개수와 가로판의 개수
1) 이전 차원에서 생성된 YC 사다리를 생성원으로 하여 Map 함수를 사용하여 병렬로 새로운 최소생성사다리를 생성함
2) 동형인 최소생성사다리가 생길 경우에 한 개만 남기고 1)의 단계를 반복하여 이전 차원에서 생성된 YC 사다리의 노드수와 동일한 단계까지 반복함
3) 각층부터 위에 해당되는 층의 노드들끼리 독립사다리연산의 관계가 있으면 간선을 생성함

출력: 새로운 차원의 YC그래프

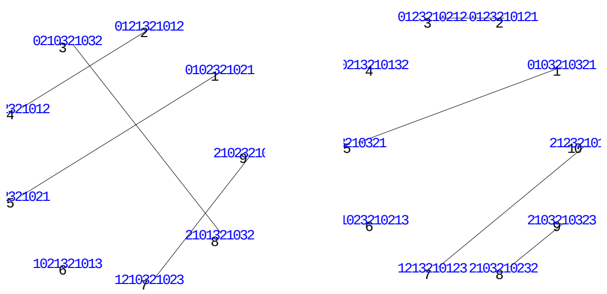
<표 2> (n, k) 사다리에서 생성된 최소생성사다리의 집합을 정점으로 하고 독립사다리연산의 적용에 따른 간선으로 만들어진 YC그래프의 속성

(n, k) YC그래프	V	E	분지수	girth	지름	층 수	간선 대칭
(3, 3) YC그래프	2	1	1	0	1	2	T
(4, 6) YC그래프	8	8	2	8	4	4	T
(5, 10) YC그래프	62	100	3~5	4	10	7	T
(6, 15) YC그래프	908	2128	4~7	4	20	11	?

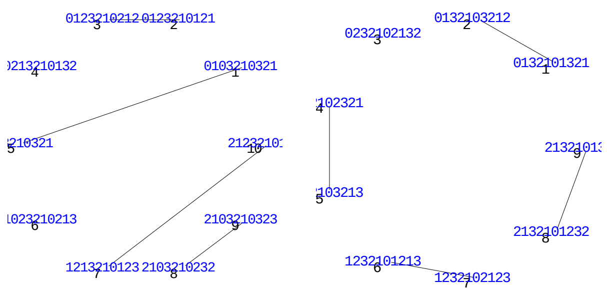
이 같은 방식을 적용하여 최소생성사다리를 정점으로 구성한 새로운 그래프에 대한 주요 속성을 <표 2>에 제시하였다. (5, 10) 사다리에서 생성된 최소생성사다리 62개로 생성된 YC그래프는 100개의 간선을 가지고 있으며 7층의 구조(8, 9, 9, 10, 9, 9, 8)를 보여준다.



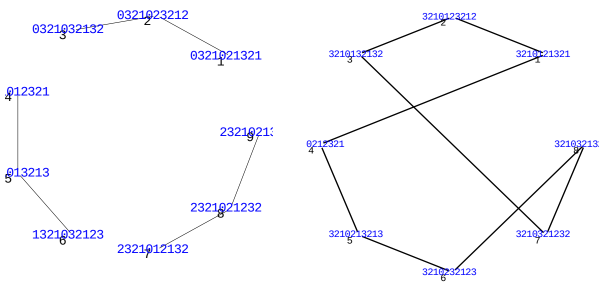
<그림 5> (5, 10) YC그래프의 1층(좌)과 2층(우)



<그림 6> (5, 10) YC그래프의 3층(좌)과 4층(우)



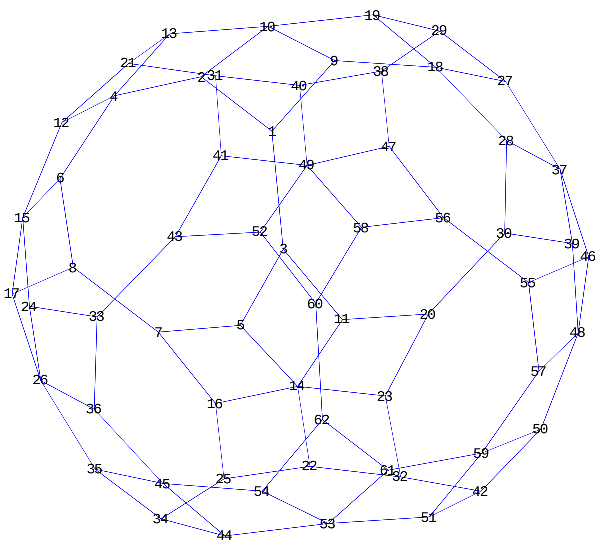
<그림 7> (5, 10) YC그래프의 4층(좌)과 5층(우)



<그림 8> (5, 10) YC그래프의 6층(좌)과 7층(우)

<그림 5>부터 <그림 8>의 출력 결과를 보면 1층부터 7층까지의 노드 개수와 층별로 연결된 간선을 볼 수 있다. 1층과 7층은 팔각형 구조를 2층과 6층, 3층과 5층으로 짝지어서 노드의 개수가 대칭으로 나타나면서 간선의 대응관계가 일대일로 나타나고 있다.

62개의 최소생성사다리와 100개의 간선을 가진 (5, 10) YC그래프는 <그림 9>에서 전모를 드러내고 있다. 이 그래프는 간선 대칭이며 girth 4, 지름 10을 갖고 있는데 정점에서 연결된 간선의 수(분지수)가 3에서 5까지 변화한다.



<그림 9> (5, 10) YC그래프의 7층 구조

한편 (6, 15) YC그래프를 생성하기 위하여 트리 탐색을 이용한 병렬처리 알고리즘을 응용하여 Mathematica로 테스트 하였다. 그 결과 노드 및 간선을 생성하는 과정에서 메모리와 연산이 부하가 너무 많이 걸려서 큐브 다면체의 층별로 나누어서 간선의 개수를 구하는 별도의 서브루틴을 작성하여 결과를 얻게 되었다. (6, 15) YC그래프는 최소생성사다리 908개와 2128개의 간선을 가지고 있다. 흥미로운 것은 이 그래프는 11층의 구조를 가지고 있으며 1층부터 11층까지 노드의 수는 62, 80, 84, 88, 92, 96, 92, 88, 84, 80, 62로 대칭적인 구조를 보여주는 점이다. 또한 이 거대한 그래프는 각 정점의 분지수가 4에서 7까지 변화하며 girth 4, 지름 20, 간선 대칭 그래프의 속성을 보여준다.

4. 컴퓨팅 사고력을 지원하는 학습 사례

이 논문에서 연구자는 R&E 프로젝트에 참가하였던 고등학생들을 지도하면서 함께 사다리 게임에서 등장하는 최소생성사다리의 특징을 탐구하였던 기존 연구[6][10]를 토대로 최소생성사다리를 생성원(generator)으로 하는 새로운 그래프를 창출하였다[11]. 연구자는 Mathematica 언어의 Make 명령어를 사용하여Mathematica로 구현함으로써 그래프의 탐색 알고리즘을 응용하여 새로운 그래프의 속성을 조사하는 프로젝트로 확장하였다[13].

두 가지의 탐구 프로젝트를 수행하면서 R&E 참여 학생들과 연구자는 컴퓨팅 사고력의 함양이 두드러지게 나타나는 공통점을 보여주었다. 그것은 R&E 참여 학생들이 C 프로그래밍을 통해 (5, 10)과 (6, 15)의 사다리에 대한 최소생성사다리의 개수와 그에 대한 속성을 탐구하여 그 결과를 제시한 것으로 입증되었다.

예를 들어, C 프로그래밍을 직접 작성한 학생의 경우에 컴퓨팅 사고력을 통해 사다리 집합에 대한 구조에 대하여 관찰하는 눈을 정확히 보여주었다. 또는 그는 알고리즘적으로 이해를 하고 있었기 때문에 constructive math에서 말하는 것처럼 화이트박스의 관점에서 최소생성사다리의 구조에 대한 이해를 할 수 있었다[1]. 이것은 C 프로그램을 통하여 수학적 구조에 접근할 수 있는 컴퓨팅 사고력의 장점을 두드러지게 보여주는 사례에 해당된다[10].

이 결과는 컴퓨팅 사고력을 수행하는 경우에 일반 학생이라도 컴퓨터 프로그래밍을 통하여 이처럼 어려운 과제에 도전할 가치가 충분히 있음을 시사하였다. 즉, 일반학생들도 이 논문에서 제시한 것처럼 사다리 게임 에 관련되는 순열의 개념에서 출발하여 조합적 사고[7]를 하면서 모든 순열을 생성하는 사다리의 집합을 구하고 그 중에서 역순열만 생성하는 사다리 모양만 걸러내는 필터링 알고리즘을 충분히 고안하고 코딩할 수 있다. R&E 또는 영재반의 사사과정에 참가하지 않는 학생들이라도 그들이 과제에 집중할 수 있는 시간적인 여유와 차분함, 끈기, 수학적 사고력과 결합된 알고리즘적 사고, 즉 컴퓨팅 사고력을 통해 최소생성사다리처럼 이산 구조에 대한 속성을 충분히 탐구할 수 있을 거라고 판단된다.

또한 연구자가 Mathematica로 진행하였던 별도의 탐구 활동에서 컴퓨팅 사고력의 신장이 두드러지게 나타났다. 필터링 알고리즘을 보완하는 과정에서 추상화(문제분석, 문제분해와 상호연결망 그래프의 트리 임베딩) 과정을 거쳐 최소생성사다리를 모두 생성하는 그래프 탐색 알고리즘으로 발전되었다. 이 논문에서 제시하였던 알고리즘은 최초 발견한 최소생성사다리를 시작 정점으로 하여 Mathematica가 가진 Map 함수와 Make 함수를 사용하였다. 최초 생성된 최소생성사다리 중에서 전환사다리 연산을 사용할 수 있는 모든 경우를 따져서 연산자를 적용하여 얻은 YC그래프는 흥미롭게도 재귀적, 위계적인 구조를 보여주었다. 즉, YC그래프는 이전 차원의 그래프를 내포하는 재귀적 구조를 가지고 있었으며 그래프의 속성 중에서 간선대칭적인 특징 외에도 노드, 간선 수가 증가함에 따라 분지수와 지름도 증가하는 패턴을 보여주었다.

5. 결론 및 후속 과제

이 탐구 프로젝트는 계산을 통한 사고의 명료화 (논리적, 구조적, 탐색-추론-검증)를 거쳐 그래프 구조에 대한 이산수학적 탐구 활동으로 확장되면서 수학적 사고와 알고리즘적 사고가 결합하여 컴퓨팅 사고력 함양을 위한 대표적인 학습 콘텐츠로 부각되었다. 또한 Mathematica가 가진 계산적 기능 외에 사다리를 시각적으로 표현하는 기능과 새로운 그래프를 생성하여 제시하는 시각화 기법이 컴퓨팅 사고력을 전개하는데 많은 기여를 하였다. 순열, 치환, 함수를 다루는 이산수

학에 대한 탐구활동이 새로운 그래프의 구조를 생성하는 수준 높은 프로젝트로 발전할 수 있는 것은 연구자가 컴퓨팅 사고력에 부합되는 Mathetmatica의 Map 함수 기능과 Make 명령어 등을 사용하여 패턴 처리 기반의 병렬처리 기능을 구사하였기에 가능하였다.

연구자는 새로이 생성한 그래프의 구조를 탐색하면서 수와 그래프 구조에서 나타나는 수열의 대칭적 아름다움을 느꼈다. 또한 계산적 복잡도가 증가함에 따라 문제를 분해하여 접근하는 추상화 과정을 거쳐 그래프 탐색 알고리즘을 적용하여 응용하면서 수학과 이산구조의 접합점에서 컴퓨팅 사고력을 신장시키는 경험을 할 수 있었다. 이 프로젝트는 컴퓨팅 사고력에 기반하여 과학영재반 학생들의 사사 과정[12]뿐만 아니라 대학교의 이산구조 강좌에서 충분히 활용할 가치가 높은 콘텐츠 개발의 사례를 보여주었다.

상호연결망 그래프의 관점에서 볼 때 이 연구는 최소생성사다리의 개수를 구하는 방법뿐만 아니라 차원이 증가할 때 발생되는 메모리와 연산의 복잡도에 대처하기 위하여 대안적으로 알고리즘 B와 C를 개발하여 YC그래프의 구조를 분석하는 방법을 제시했다는 점에서 기존의 연구[11]와 차별화된다. 최소생성사다리의 생성원을 다원트리로 구성해가는 방식을 차용한 알고리즘 B의 복잡도는 알고리즘 A의 복잡도 O(2^n)보다 작은 것임에 틀림없다.

후속적으로 (6, 15) YC그래프가 노드 대칭인지를 확인하는 작업 외에도 한 차원을 높인 (7, 21) YC그래프의 속성을 탐구할 예정이다. 이 새로운 그래프의 1층과 마지막 층은 908개의 노드가 배치될 것으로 확실하며 노드 대칭적 위계 구조를 가진 층 수는 아마도 약 908/62 보다 큰 자연수인 15 또는 17 등 소수일 가능성이 있다. 만약 15층의 구조를 가질 경우에 이 거대한 그래프는 최소 908*13 즉 13620개 이상의 노드를 가질 것이다.

실제 알고리즘 B를 구현한 Mathematics 프로그래밍으로 파일럿 테스트해 본 결과 16층의 구조를 가지면서 24218개의 노드를 가지는 것으로 나타났다. 그러나 층별 노드 수를 살펴본 결과 1층과 16층은 908개의 노드를 가지고 있으나 2층(1320개)과 15층(1340개)의 노드 수가 일치하지 않아서 간선 대칭 그래프일 것이라는 예측과 빗나갔다. 따라서 향후에 Mathematica 코드를 수정할 예정이며 이에 덧붙여 차원(세로판의 개수)이 증가할수록 YC그래프의 노드 및 간선의 개수를 결정하는 식을 도출하고 알고리즘에 대한 복잡도를 탐구할 예정이다.

이 연구는 R&E 참여 학생들의 사례에서 살펴보았듯이 고등학생들이 팀을 이루어서 장기간에 걸쳐 순열과 조합, 통계적인 관점에서 사다리 그래프에 관한 이산적 구조와 수학적 탐구를 병행하면서 프로그래밍으로 알고리즘을 구현해 보는 화이트박스 기반의 학습의 장점을 보여주었다. 이 연구에서 제시한 학습 콘텐츠는 과학영재교육원의 학생들이 사사과정을 수행하는 프로젝트 기반 학습에도 도입될 수 있으며 더 나아가 대학생들이 수강하는 이산구조 강좌에서 컴퓨팅 사고력을 증진하는데 기여 할 수 있는 프로젝트 기반 학습으로 활용될 수 있을 것이다.

참 고 문 헌

[1] 전영국(2015). 이공계 대학생을 위한 Mathematica 기반의 화이트박스 이러닝 콘텐츠 설계 및 개발. **한국학교수학회논문집, 18**(2), 223-240.

[2] 이광연, 이광상, 유기종(2013). 함수의 도입을 위한 사다리타기 게임의 수학적 분석. **한국수학교육학지 시리즈 E <수학교육 논문집>, 27**(3), 267-281.

[3] 주수아(2003). **수학적 태도 형성을 위한 게임 학습의 사례연구.** 신라대학교 교육대학원 석사논문.

[4] 한수선(2014). **사다리타기 게임을 활용한 수학 교수·학습에 관한 연구.** 부산대학교 석사논문.

[5] 전용주, 홍창의, 김태영, 송기상 (2014). **초등 정보영재의 R&E 사사교육에서 CT기반 프로그래밍 교육 사례분석.** 한국컴퓨터교육학회 학술발표대회논문집, 18(1), 73-80.

[6] 김시주(2013). 최소생성사다리의 특성. **과학영재교육, 5**(1), 15-21.

[7] 장정숙, 전영국, 윤지현(2013). 하노이 탑 프로그래밍 경험에서 나타나는 정보과학적 사고 패턴에 관한 질적 사례 연구. **한국컴퓨터교육학회 논문지, 16**(4), 33-45.

[8] 최숙영(2011). 21st Century Skills와 Computational Thinking 관점에서의 '정보' 교육과정 분석. **컴퓨터교육학회논문지, 14**(6), 19-29.

[9] 전영국(2015). 컴퓨터 프로그래밍과 창의성 발현 활동에 관한 질적 사례 연구: NetLogo 기반의 계산적 사고 중심으로. **한국컴퓨터교육학회 논**

문지, 18(3), 1-14.

[10] 이유섭, 김태정, 임준섭, 강민서, 전영국(2015). **사다리 게임에서 나타나는 최소생성사다리 찾기 알고리즘 탐구.** 한국컴퓨터교육학회 학술발표대회논문집, 19(2), 133-137.

[11] 이형옥, 전영국(2006). 행렬-스타그래프와 팬케익 그래프, RFM그래프 사이의 임베딩 분석. **멀티미디어학회논문지, 9**(9), 1173-1183.

[12] 전영국(2016). 정보 영재반 중학생들의 IT 융합 사사 프로젝트 수행에 관한 질적 분석. **한국컴퓨터교육학회 논문지, 19**(4), 45-58.

[13] 채수환, 유동선(2003). **이산수학원론.** 서울: 교우사.

[14] 황종선, 정영식(1996). **C 언어로 설명한 알고리즘.** 서울: 정익사.

<부록: 사용된 Mathematica 5.2 코드의 일부>

© Programmed by Youngcook Jun 2016

```
<<DiscreteMath`Combinatorica`
<<DiscreteMath`GraphPlot`

swap[n_,{i_,j_}]:=
ReplacePart[ReplacePart[Range[n], j,i], i,j]

compRungPerm[n_,rung_]:=
FoldList[Permute,Range[n],Map[swap[n,#]&,rung]]

(* exhausitive generation *)
generateRungs[n_,sides_]:=
  Module[{rungs,target},
    rungs=Table[{i,i+1},{i,1,n-1}];
    target = Tuples[rungs,sides] ]

compLadderPerm[n_,sides_]:=
   Map[compRungPerm[n,#]&,generateRungs[n,sides] ]

findFinalPerm[result_]:=Map[Last,result]

findMinSpanLadder[n_,result_]:=
  Select[result,(Last[#]   Reverse[Range[n]])&]

countSameNumbers[n_,final_]:=
  Module[{zerolist,inverlist},
    zerolist=Map[Range[n] - #&,final];
    inverlist=Map[Map[If[##   0,1,0]&,#]&, zerolist];
    Apply[Plus,inverlist]]

reconsRung[perm1_,perm2_]:=
   Position[perm1-perm2,_?(#>0||#<0&)]//Flatten

listupRung[perms_]:=
  Table[reconsRung[perms[[i]], perms[[i+1]]],
{i,1,Length[perms]-1}]

minSpanLadderQ[n_,rung_]:=
  Module[{candidate,result, final},
     candidate=Subsets[rung];
     result=Map[compRungPerm[n,#]&, candidate];
  final=Map[Last,result]//Union;
  final==Permutations[Range[n]]]

confirmSpanLadderQ[n_,finalperms_]:=
   Map[minSpanLadderQ[n,#]&,
   Map[listupRung[##]&,finalperms]]

captionText[perm_]:=
  Join[Table[Text[ToString[i],{i,1}],{i,1,Length[perm]}],
  Table[Text[ToString[perm[[i]]],{i,0}],{i,1,Length[perm]}]]

ladderDraw[n_,rung_]:=
  Module[{delta,nr,caption, pole,temp,pdraw,rdraw},
    nr=Length[rung]+1;
    delta =Table[List[1-i/nr,1-i/nr],{i,nr-1}];
    temp=Outer[List,Range[n],{1,0}];
    pdraw=Map[Line[#]&,temp];
    rdraw=MapThread[List,{rung, delta}];
    rdraw=Map[MapThread[List,#]&,rdraw];
Graphics[Join[captionText[compRungPerm[n,rung]//Last],
pdraw, Map[Line,rdraw]]] ]

drawAllLadders[n_,sides_, partition_]:=
  Module[{ladders},
     ladders=Map[ladderDraw[n,#]&,generateRungs[n,sides]];
     Partition[ladders,partition]//GraphicsArray//Show]
```


전 영 국

1986 수원대학교 수학과(이학사)
1986 시카고주립대학교 수학과(이학석사)
1995 일리노이대학교 어바나-샴페인(교육학박사)
1996 ~ 현재 순천대학교 사범대학 컴퓨터교육과 교수
관심분야: 로봇예술, 에이전트 기반 모델링, 컴퓨팅 사고력, 인공지능 등
E-Mail: ycjun@sunchon.ac.kr