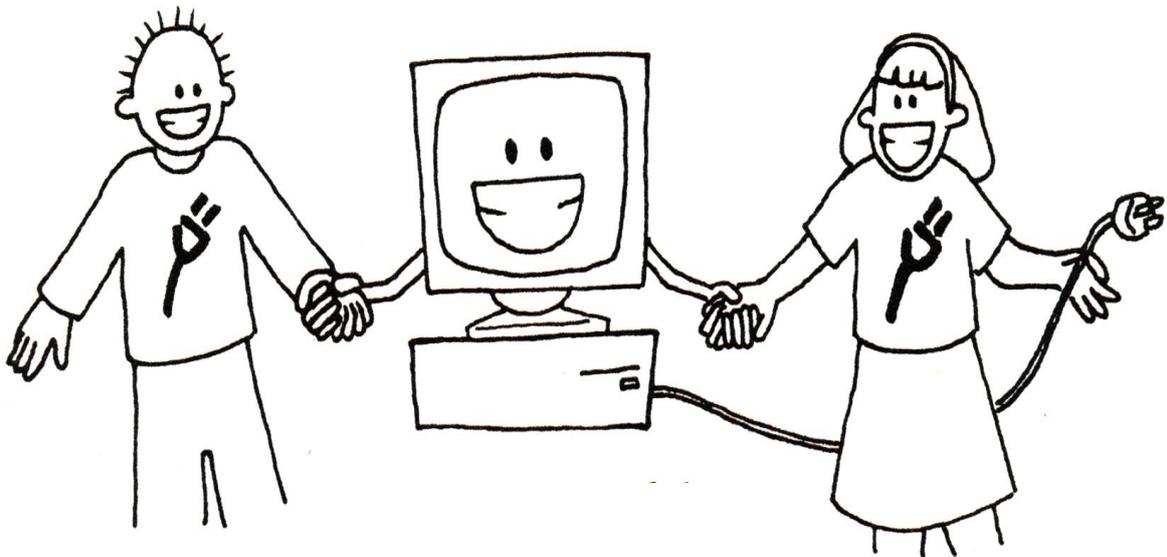


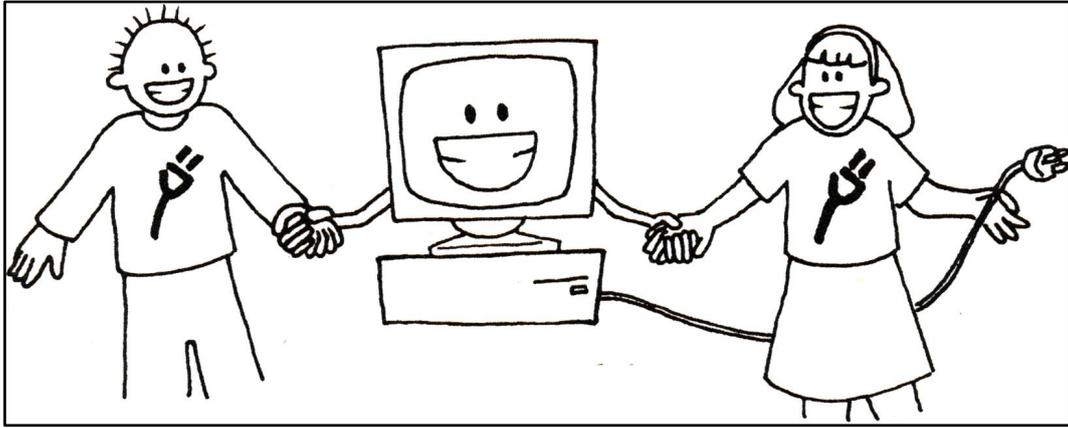
CS

UNPLUGGED

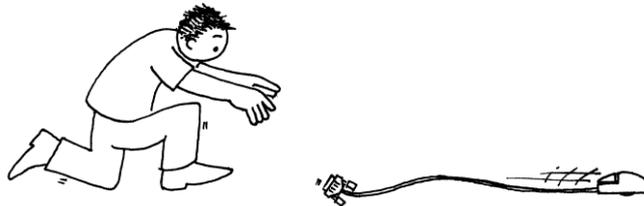
언플러그드 컴퓨팅



언플러그드 컴퓨팅



21 세기 학습자를 위한 심화 프로그램



- 저자 : 팀 벨, 이안 위튼, 마이크 펠로우스
 - 적용 : 로빈 아담스, 제인 맥켄지
 - 삽화 : 맷 포웰
 - 교정 : 샘 자먼
 - 번역 : 경인교육대학교 미래인재연구소
-

소개

컴퓨터는 어디에나 있고 대부분의 사람들이 매일 사용하고 있지만 막상 컴퓨터가 어떻게 작동되는지, 어떻게 생각하는지, 또 사람들이 어떻게 쉽고 편리한 소프트웨어를 만드는지는 잘 모릅니다. 컴퓨터 과학은 이러한 궁금증에 답을 줍니다. 모든 연령의 학생들을 위해 만든 이 책은 쉽고 재미있는 놀이활동을 통해 컴퓨터 없이도 컴퓨터가 어떻게 작동하는지 안내합니다.

이 책은 정보교육 심화 수업 과정, 영재교육 등에서뿐 아니라 일반 정규 수업에서도 활용할 수 있습니다. 학생들과 컴퓨터과학의 원리들을 배우고 즐기기 위해 컴퓨터 전문가가 될 필요는 없습니다. 이 책은 컴퓨터 과학의 간단한 배경지식과 활동방법, 그리고 주어진 문제에 대한 해결방법을 제시하며 실생활과의 관련성을 설명하는 '컴퓨터 과학 이야기' 로 구성되어 있습니다.

많은 활동들이 이진수, 함수와 그래프, 문제의 패턴화 및 정렬, 암호화등 수학을 기반으로 합니다. 또 다른 활동들은 컴퓨터가 어떻게 작동되는지에 대한 지식과 이해, 그리고 정보교육 과정과 잘 연결됩니다. 학생들은 의사소통, 문제해결, 창의력 등 고급사고력을 각 활동에 적용하는 활동들을 통해 컴퓨팅 사고를 탐구하고 신장시킬 수 있습니다.

이 책의 언플러그드 프로젝트 자료는 csunplugged.org 에서 수업활동 전략 그리고 동영상, 사진 등의 온라인 자료와 함께 무료로 제공되고 있습니다. 또한 2015 년 개정 작업을 통해 학교 교육과정에서 컴퓨터과학과 컴퓨팅 사고를 적용할 수 있도록 새로운 커리큘럼과 공개 자료를 제공하는 웹사이트를 새로이 만들었습니다.

이 책은 20 년이 넘는 기간 동안 2 명의 교사와 3 명의 컴퓨터 과학 작가가 수백명의 교육자들에게 피드백을 받아 만들었습니다. 그 과정에서 컴퓨터를 사용하지 않고도 컴퓨터 과학을 가르칠 수 있는 주요 개념을 많이 발견했습니다. 사실 컴퓨터는 교실에서 학습 분위기를 산만하게 하는 요소가 되기도 합니다. 가끔 컴퓨터 과학은 프로그래밍을 가르치는 것으로 시작하는데, 모든 학생들이 여기에 큰 매력을 느끼지는 않습니다. 오히려 컴퓨터 과학의 흥미로운 아이디어에 접근하는데 큰 장벽이 되기도 합니다. 그러니 컴퓨터를 치우고 컴퓨터 과학이 진짜로 무엇인지 배울 준비를 하세요!

이 책은 구글의 도움으로 무료로 다운로드 받을 수 있습니다. 또한 크리에이티브 커먼즈 저작자 표시 -비영리- 를 통해 공유(복제, 배포 및 전송)하거나 재구성할 수 있습니다. 단, 저작자를 표시를 해야하며 상업적 목적으로 사용하지 않아야 합니다. 만약 당신이 이 책을 변형, 대체, 재구성한다면 동일하거나 유사한 라이선스하에 공유해야 합니다. 자세한 내용은 CC BY-NC-SA 3.0 을 검색해 보세요.

이 책은 교육적인 목적으로 사용해주기 바랍니다. 이 책을 복사해 학생들에게 워크시트로 제공할 수 있습니다. 저자들에게 얼마든지 문의하고 제안해보세요! (csunplugged.org)

이 책은 다양한 언어로 번역되었습니다. 웹사이트에서 확인하십시오.

감사의글

여러 학생들과 교사들이 우리의 아이디어를 만드는데 도움을 줬습니다. South Park School (Victoria, BC), Shirley Primary School, Ilam Primary School and Westburn Primary School (Christchurch, New Zealand)의 학생들과 교사들은 언플러그드 활동의 모델이 되어주었습니다. 특히 Linda Picciotto, Karen Able, Bryon Porteous, Paul Cathro, Tracy Harrold, Simone Tanoa, Lorraine Woodfield 그리고 Lynn Atkinson 은 교실을 개방하여 교재 활동을 보완하는데 많은 도움을 주었습니다. Gwenda Bensemman 은 여러 활동을 시범 적용해주었고 수정을 해주었습니다. Richard Lynders and Sumant Murugesh 은 교실 실험을 도와주었습니다.

암호화 활동의 일부는 Ken Noblitz 가 개발했습니다. 활동 중 일부는 Kathy Beveridge 의 도움으로 Victoria “Mathmania” group 에서 운영되었습니다. 초기 일러스트레이션은 Malcolm Robinson 과 Gail Williams 가 만들어주었고 Hans Knutson 의 조언을 받았습니다. Matt Powell 역시 언플러그드 프로젝트 개발에 도움을 주었습니다. 이 책의 개발 초기 단계에 후원을 해준 Brian Mason Scientific Technical Trust 에 감사합니다. 많은 활동을 테스트하고 조언을 준 Paul and Ruth Ellen Howard 에 감사합니다. Peter Henderson, Bruce McKenzie, Joan Mitchell, Nancy Walker-Mitchell, Gwen Stark, Tony Smith, Tim A. H. Bell¹, Mike Hallett, 그리고 Harold Thimbleby 또한 많은 조언을 주었습니다.

도움을 준 분들입니다. : Bruce, Fran, Grant, Judith, Pam, Andrew, Anna, Hannah, Max, Michael, Nikki.

특히 언플러그드 프로젝트를 후원하고 이 책의 무료 다운로드가 가능하도록 도와 준 Google Inc. 에 감사합니다. .

csunplugged.org 를 통해 여러분의 생각을 공유해주시기 바랍니다.

¹ No relation to the first author.

번역과 함께

10 여년전 컴퓨터교육을 대한민국에 확산시키기 위해 깊은 고민을 하던 우리 미래인재연구소 팀원들은 Tim Bell 교수의 언플러그드 컴퓨팅 교육 방법과 사이트(csunplugged.org)를 접하고 컴퓨터교육의 많은 가능성과 희망을 보았습니다. 컴퓨터과학 전공자와 그에 조예가 깊은 전문가를 위한 어려운 컴퓨터 과학이 아니라 초중등 학생들과 교사들도 쉽고 이해하기 편한 보편화된 교육으로 컴퓨터 과학을 안내하는 뛰어난 언플러그드 학습 콘텐츠와 학습방법에 우리는 열광하였습니다.

교육대학교 학부생들을 대상으로 컴퓨터과학 수업을 수차례 실시하였고, 초등학생과 중학생(심지어 유치원생까지)을 대상으로 학습에 적용하기도 하였습니다. 연구소 멤버들 대부분이 초등학교 교사이기 때문에 자신의 반 학생들을 대상으로 수업에 적용하고, 창의컴퓨팅 페스티벌과 스크래치데이(Scratch Day) 행사에서 매번 소개하기도 하였습니다. 이러한 노력의 일환으로 컴퓨터과학의 지식과 개념을 초중등 교사들이 그 내용을 이해하고 수업 현장에서 활용하는 사례가 확산되고 있습니다.

최근 대한민국에서 ‘소프트웨어 중심사회 실현’ 정책을 펼치며 SW 교육을 초중등 교육에 정규교과로 적용하며 미래 인재의 역량으로 창의컴퓨팅 능력을 신장시키는 노력을 하고 있습니다. 코딩과 프로그래밍을 중점적으로 학생들에게 철차적 사고와 창의적으로 문제해결력을 신장하도록 교육과정이 구성되어 있습니다. 이러한 목표에 보다 효과적으로 접근하기 위해서는 알고리즘과 프로그래밍의 기초가 되는 컴퓨터과학의 개념과 원리를 이해해야만 합니다. 컴퓨터 과학의 기초 지식없이 프로그래밍만 하는 것은 결국 단순 코딩을 하는 낮은 수준의 프로그래머만 양성할 뿐 고급 프로그래머로서의 전문성이나 다른 분야와의 융합적 능력 그리고 미래를 설계하는 디자인적 사고의 확산 단계로 성장시키기에는 무리가 따르게 됩니다.

따라서 이러한 미래인재로서의 성장을 위해 컴퓨터 과학의 지식은 초중등학생들에게 많은 도움을 주게 될 것입니다. 언플러그드 컴퓨팅 방법을 통해 초중등교사들이 쉽게 컴퓨터 과학을 이해하고 학생들에게 적용하게 된다면 정규교과로서의 SW 교육에 많은 기여를 할 것으로 기대합니다.

이러한 취지에 의해 미래인재연구소 멤버들이 뜻을 모아 몇 년간 미루어왔던 언플러그드 컴퓨팅 원서를 2015 년에 한글화하였습니다. 이 번역자료가 대한민국 초중등교육에 많은 도움을 주길 기대하며, 팀벨 교수의 철학에 따라 전세계에 이 문서를 공유하고자 합니다.

경인교육대학교 미래인재연구소장 한선관

번역에 참여하신 분

이 워크북의 번역에 참여한 분을 소개합니다.

- 총괄: 경인교육대학교 미래인재연구소장/교수 한선관
- 번역설계: 미래인재연구소 교육연구팀장/교사 류미영
- 세부번역: 미래인재연구소 연구원/교사 민경수, 서정원, 조현룡, 윤종원, 방혜윤

미래인재연구소의 언플러그드 교육 활동

미래인재연구소에서는 언플러그드 활동을 온라인 교육으로 개발하여 보급하고 있습니다. 다음 사이트에서 언플러그드 교육 활동을 볼 수 있습니다.

<http://koreasw.org> 대한민국 주니어 SW 교육사이트(미래창조과학부 주관, 미래인재연구소 개발)

다음 사이트에서 언플러그드 교육 콘텐츠와 교수학습 과정안을 볼 수 있습니다

<http://computing.re.kr> 창의컴퓨팅교육 사이트

팀벨교수의 언플러그드 컴퓨팅 교육 활동의 한국 학교 수업 장면을 볼 수 있습니다.

<http://unpluggd.kr> 언플러그드 교육 동영상 사이트(시범수업 류미영, 촬영 민경수)

한국형 언플러그드 교육자료의 신규 개발 자료를 볼 수 있습니다.

<http://csunplugged.kr> 한국형 언플러그드 교육자료 사이트(설계 한선관, 개발 류미영)

목차

소개	i
감사의 글	iii
번역과 함께	iv
번역에 참여하신 분	v
미래인재연구소의 언플러그드	vi
데이터 이해하기 — 정보의 표현 1	
점의 수-이진수	3
숫자로 칠하기—그림 표현	16
바로 그거야! —문자 압축	27
카드 마술—에러 감지와 수정	37
스무고개—정보 이론	45
컴퓨터 일시키기—알고리즘 53	
전함 놀이—탐색 알고리즘	55
똥보와 흙쭉이—정렬 알고리즘	74
똑딱 시계—정렬 네트워크	82
진흙 도시—최소 확장 트리	90
과일 놀이—네트워크의 경로찾기와 교착상태	96
무거운 벽돌—네트워크 통신 프로토콜	100
컴퓨터가 일하도록 명령하기—절차 표현 107	
보물섬—유한 상태 오토마타	109

<i>행진하라—프로그래밍 언어</i>	125
정말 어려운 문제 풀기—난해성 107	
가난한 지도 제작가— <i>그래픽 문제</i>	133
여행자 도시— <i>지배 집합</i>	146
빙판위의 도로— <i>슈타이너 트리</i>	155
비밀 공유와 해킹 보호 - 암호화 167	
비밀 공유— <i>정보 은닉 프로토콜</i>	172
페루동전 뒤집기— <i>암호화 프로토콜</i>	176
꼬꼬마 암호— <i>공개키 암호화</i>	188
인간과 컴퓨팅 대면하기 - 컴퓨터와 상호작용 200	
초콜릿 공장— <i>인간 인터페이스 설계</i>	204
컴퓨터와의 대화— <i>튜링 검사</i>	219

Part I 1 부

데이터 이해하기—정보 표현

데이터 이해하기: 원시 데이터

어떻게 정보를 컴퓨터에 저장할 수 있을까요?

컴퓨터라는 단어는 계산하고 더한다는 뜻을 함께 가진 라틴어 ‘*컴퓨터어*, *Computare*에서 유래하였지만 오늘날 컴퓨터는 거대한 계산기 이상을 의미합니다. 컴퓨터는 도서관이 될 수도 있고, 글쓰기를 돕고, 정보를 검색하며 음악을 들려주고 영화까지 보여줄 수 있습니다. 컴퓨터는 이러한 다양한 정보를 어떻게 저장하는 걸까요? 믿을지 모르겠지만 컴퓨터는 단지 두가지, 0 과 1 만을 사용한답니다.

자료와 정보의 차이는 무엇입니까?

자료는 원재료 즉, 컴퓨터가 일을 처리하기 위한 숫자들입니다. 컴퓨터는 이것을 우리가 이해할 수 있는 정보 즉, 글, 숫자, 그림 등으로 변환합니다.

숫자, 글자, 단어와 그림들이 어떻게 0 과 1 로 변환될까요?

이번 장에서 우리는 이진수에 대해 배우게 됩니다. 어떻게 컴퓨터가 그림을 그리는지, 팩스가 어떻게 작동하는지, 많은 양의 자료를 저장하기 위한 가장 효율적인 방법은 무엇인지, 갑작스럽게 발생하는 에러를 막기 위한 방법은 무엇인지, 저장하기 위한 정보의 양이 얼마나 되는지 측정하는 방법을 살펴봅니다.



활동 1

점의 수 세기—이진수

활동 개요

컴퓨터의 자료들은 0 과 1 로 나열되어 전송되고 저장됩니다. 단지 두개의 기호를 이용하여 어떻게 단어와 숫자를 표현할 수 있을까요?

교과 연계

- ✓ 수학: 수- 숫자의 크기와 순서 이해하기, 2 의 배수 표현하기.
- ✓ 수학: 대수-순서 패턴 연결하기와 패턴의 규칙 설명하기, 2 의 거듭제곱의 패턴과 관계 이해하기

필수기능

- ✓ 셈하기
- ✓ 대조하기
- ✓ 순서정하기

대상 연령

- ✓ 6 살 이상

수업 자료

교사

- ✓ 활동을 위한 5 개의 이진수 카드 세트 만들기(7 쪽 참고).
A4 카드 스마일 맨 스티커 점이 붙은 A4 카드

학생

- ✓ 5 개의 카드 세트
이진수 카드(활동자료 1)를 복사해서 카드 크기로 잘라서 나눠주기
- ✓ 워크시트 활동: 이진수(6 쪽)

선택 심화 활동

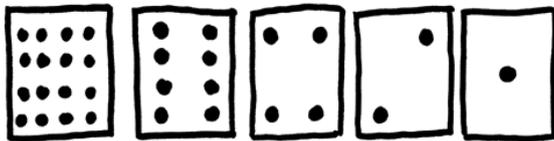
- ✓ 워크시트 활동: 이진수 놀이하기 (8 쪽)
- ✓ 워크시트 활동: 비밀 메시지 전송 (8 쪽)
- ✓ 워크시트 활동: 이메일과 모뎀 (10 쪽)
- ✓ 워크시트 활동: 31 보다 큰수 세기 (11 쪽)
- ✓ 워크시트 활동: 이진수 더 알아보기 (12 쪽)

이진수

소개

6 쪽의 워크시트를 제시하기 전에 전체 그룹에게 원리를 설명하는 게 좋습니다.

이 활동을 위해서는 아래의 그림처럼 한 면은 비어있고 반대쪽 면에는 점들이 그려진 5 장의 카드 한 세트가 필요합니다. 다섯 학생을 뽑아 카드를 아래의 순서대로 교실 앞에 들고 있게 합니다. 들고 있는 순서는 아래와 같습니다.



생각해보기

학생들에게 오른쪽에서 왼쪽 순으로 카드를 보여주면서 다음 카드에 몇개의 점이 있을지 예상하도록 합니다. 점의 수에서 규칙을 발견할 수 있습니까? (각 카드에는 바로 옆 오른쪽 카드의 두배만큼의 점이 있습니다.)

위의 규칙대로 카드를 왼쪽으로 계속 놓는다면 다음 카드에는 몇개의 점이 있을까요?(32 개) 그 다음에는 몇 개의 점이 있을까요?(64 개)

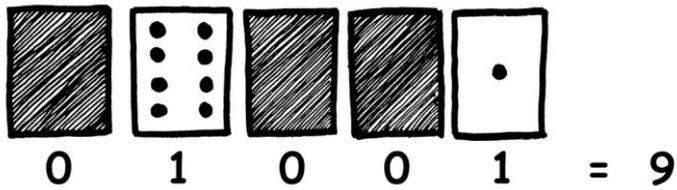
카드의 앞면과 뒷면을 보여주는 활동을 통해 숫자를 표현할 수 있습니다. 학생들에게 카드를 뒤집어서 6 개의 점(4 와 2), 15 개의 점(8, 4, 2, 1), 21 개의 점(16, 4, 1)을 표현하게 해보세요. 단, 각 카드의 점들은 뒤집기를 통해 통째로 보이게 하거나 완전히 뒤집어서 안보이게 해야 합니다. 점들 중 일부만 가리는 것은 안됩니다.

표현할 수 있는 가장 작은 수는 무엇일까요?(학생들은 1 이라고 대답하겠지만 정답은 0 입니다).

0 부터 쪽 숫자를 세면서 카드로 그 숫자를 점으로 나타내보세요..

이제 카드가 뒤집어지는 패턴을 파악해야 합니다(각 카드는 오른쪽 카드에 비해 절반의 횡수로 뒤집힙니다). 두 그룹 이상이 같이 해볼 수 있습니다.

카드가 뒤집혀 점이 보이지 않는 상태는 0 으로 나타냅니다. 반대로 점이 보이는 상태는 1 로 나타냅니다. 이것이 이진수 체계입니다.



학생들에게 카드로 01001 을 만들게 해보세요. 십진수로는 몇입니까? (9) 십진수 17 은 이진수로 어떻게 나타낼까요? (10001)

이 개념을 이해할 때까지 몇번 더 해봅니다.

이해를 더 보충하기 위해 이 강의 뒷부분에 5 가지의 심화 활동이 더 있습니다. 학생들은 얼마든지 더 많은 활동을 해볼 수 있습니다.

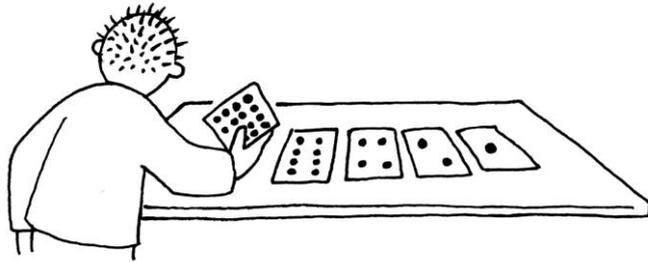
워크시트 활동: 이진수

어떻게 수를 셀까요?

우리는 너무나 당연하게도 수를 세는 법을 알고 있습니다. 여기 또 다른 방법이 있습니다. 컴퓨터는 1 과 0 만을 사용합니다. 컴퓨터에서 보거나 들을 수 있는 글, 그림, 숫자, 소리와 영화는 1 과 0 을 사용하여 저장한 것입니다. 다음의 활동들은 컴퓨터와 같은 방법으로 친구들에게 비밀 메시지를 보내는 법을 안내합니다.

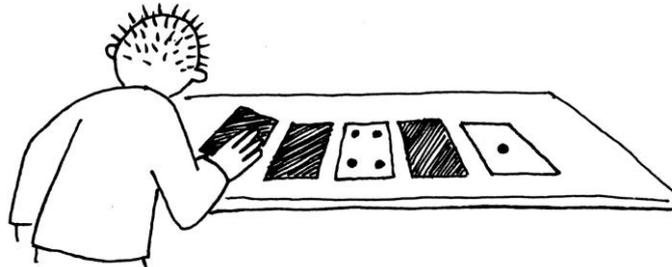
활동 안내

활동자료 1 의 이진수카드 시트에서 카드를 잘라 아래 그림처럼 카드를 배열합니다.



정확한 순서대로 배열되어 있는지 확인하세요.

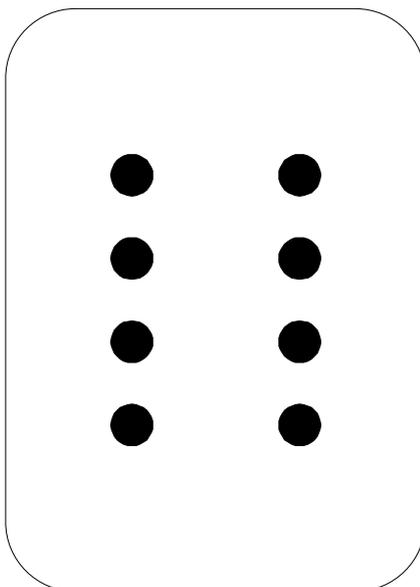
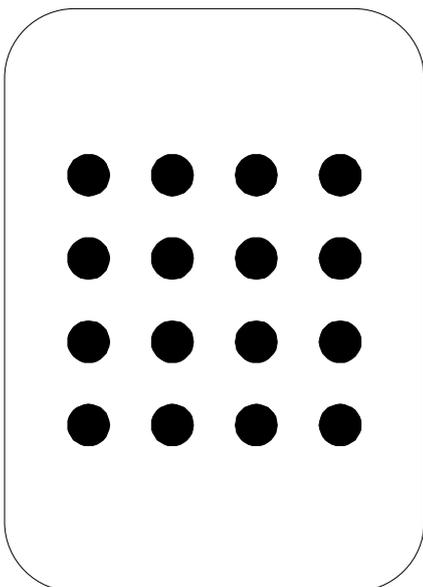
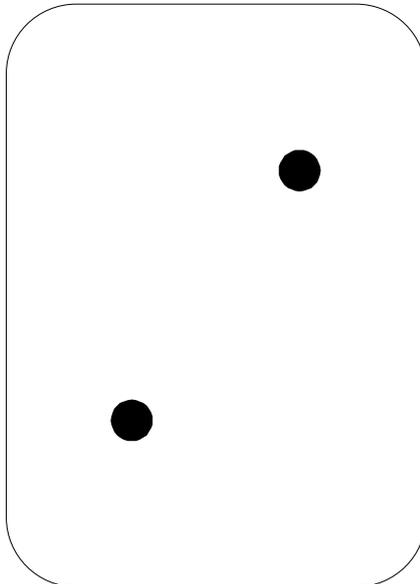
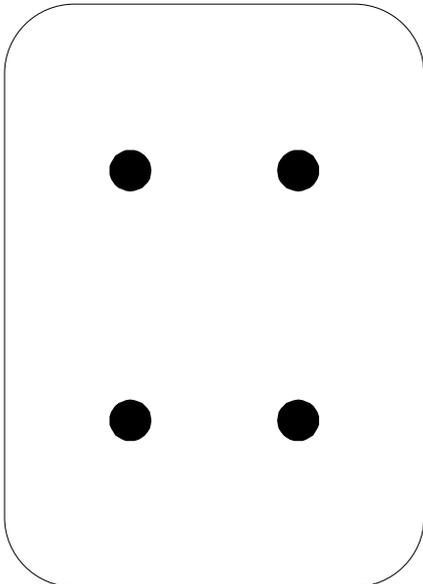
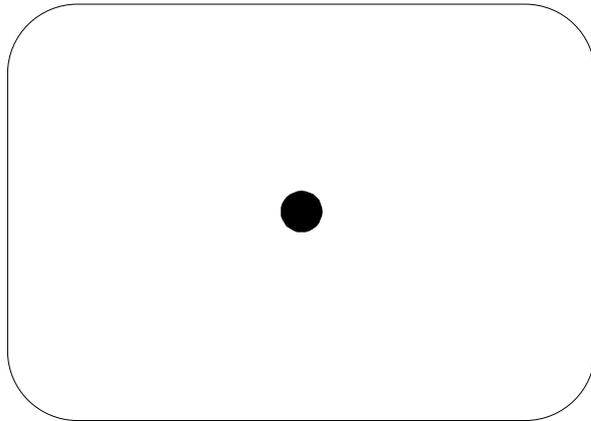
이제 아래의 그림처럼 정확히 5 개의 점이 보이도록 카드들을 뒤집으세요.



3, 12, 19 를 나타낼 수 있는 방법도 찾아보세요. 어떤 수를 표현할 때 그 방법이 여러가지 인가요? 이 카드로 만들 수 있는 가장 큰 수는 무엇인가요? 가장 작은 수는요? 만들 수 있는 가장 큰 수와 가장 작은 수 사이에 만들 수 없는 수가 있나요?

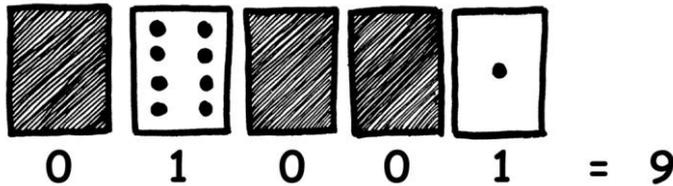
심화: 카드로 1,2,3,4... 순서대로 수를 만들어보세요. 수를 1 씩 늘릴 수 있는, 논리적이고 규칙적인 카드 뒤집기 방법이 있나요?

활동 자료 1 : 이진수 카드



워크시트 활동: 이진수 놀이하기

이진법은 0 과 1 을 이용하여 카드가 앞면인지 뒷면인지를 표시합니다. 0 은 점이 보이지 않는 뒷면이고, 1 은 점이 보이는 앞면입니다.



10101 은 십진수로 몇일까요? 11111 은요?

당신이 태어난 날은 몇일입니까? 이진수로 나타내보세요. 친구들의 생일은 이진수로 어떻게 나타낼 수 있을까요?

아래 기호로된 암호를 이진수로 나타내보세요:

$$\begin{matrix} \boxtimes & \boxcheck & \boxtimes & \boxtimes & \boxcheck & = \\ (\boxcheck=1, \boxtimes=0) \end{matrix}$$

$$\begin{matrix} \uparrow & \downarrow & \uparrow & = \\ (\uparrow=1, \downarrow=0) \end{matrix}$$

$$\begin{matrix} \odot & \odot & \odot & \odot & \odot & = \\ (\odot=1, \circ=0) \end{matrix}$$

$$\begin{matrix} \uparrow & \downarrow & \uparrow & \downarrow & \uparrow & = \\ (\uparrow=1, \downarrow=0) \end{matrix}$$

$$\begin{matrix} \text{☺} & = \\ (\text{☺}=1, \text{☹}=0) \end{matrix}$$

$$\begin{matrix} \text{👍} & \text{👎} & \text{👍} & \text{👎} & = \\ (\text{👍}=1, \text{👎}=0) \end{matrix}$$

$$\begin{matrix} + & + & \times & + & = \\ (+=1, \times=0) \end{matrix}$$

$$\begin{matrix} \cup & \cup & \cup & \cup & \cup & = \\ (\cup=1, \cup=0) \end{matrix}$$

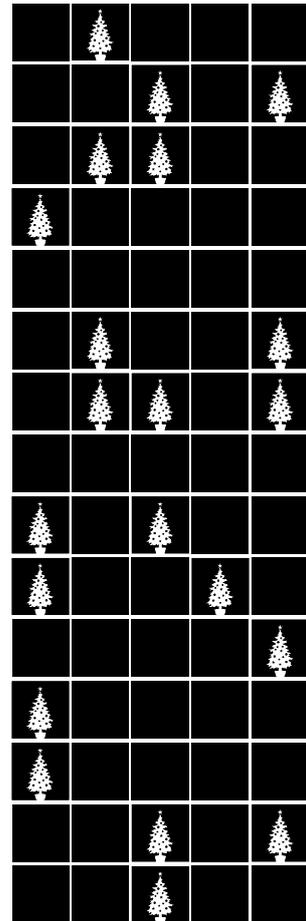
$$\begin{matrix} \blacktriangle & \blacktriangledown & \blacktriangle & \blacktriangledown & \blacktriangledown & = \\ (\blacktriangle=1, \blacktriangledown=0) \end{matrix}$$

$$\begin{matrix} \spadesuit & \spadesuit & \spadesuit & \spadesuit & \spadesuit & = \\ (\spadesuit=1, \clubsuit=0) \end{matrix}$$

심화활동 : 길이 1,2,4,8,16 의 막대 세트를 조합해 31 이상의 길이를 만들 수 있습니다. 혹은 이러한 조합의 무게추와 양팔저울을 이용해 무거운 가방이나 박스의 무게를 잴 수도 있어요!

워크시트 활동: 비밀 메시지 보내기

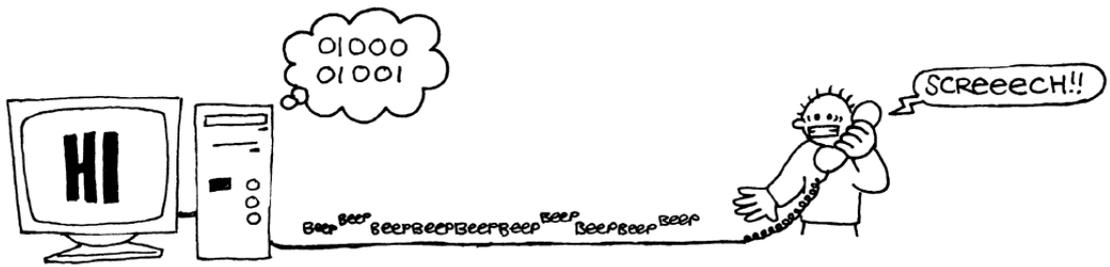
크리스마스 이브날 백화점에서 쇼핑하던 톰은 실수로 꼭대기에 갇혀있습니다. 톰은 선물을 갖고 집에 돌아가고 싶어 소리치며 도움을 구했지만 주변에 아무도 없었습니다. 그런데 길 건너 낮은 건물에서 밤 늦게까지 일을 하고 있는 컴퓨터 업체 직원들을 발견했어요. 톰은 어떻게 그들의 주의를 끌 수 있을까요? 톰은 주변에 쓸만한 것이 있나 찾아봤고 기막힌 아이디어가 떠올랐어요. 바로 크리스마스 트리를 이용해 메시지를 보내는 것이었지요! 톰은 전구들을 켜고 끌 수 있게 전원을 연결했어요. 그 다음 이진수를 사용해 건너편 사람이 확실히 알아볼 수 있도록 메시지를 보냈어요. 그 내용이 무엇인지 알아낼 수 있나요?



1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	e	f	g	h	i	j	k	l	m
14	15	16	17	18	19	20	21	22	23	24	25	26
n	o	p	q	r	s	t	u	v	w	x	y	z

워크시트 활동: 이메일과 모뎀

모뎀으로 인터넷에 연결된 컴퓨터 또한 메시지를 보낼 때 이진법을 사용합니다. 차이가 있다면 컴퓨터는 ‘삐’ 소리를 사용한다는 것입니다. 높은 음의 ‘삐’ 소리는 1 을 나타내고 낮은 음의 ‘부’ 소리는 0 을 나타냅니다. 이 소리들은 너무나 빨리들려 우리 귀에 거슬리는 킁킁 소리로 들립니다. 들어본 적이 없다면 모뎀의 인터넷 연결 소리나 팩스의 전송 소리를 잘 들어보시오(팩스도 모뎀을 사용합니다).



툼이 백화점에서 보냈던 이진법 코드를 ‘삐’와 ‘부’ 의 소리로 전달해보세요. 진짜 모뎀처럼 빠르게 소리 낼 필요는 없습니다.



워크시트 활동: 31 보다 큰 수 세기

다시 이진수 카드를 봅시다. 만약 지금 갖고 있는 이진수 카드의 다음 카드를 만든다면 그 카드에는 몇 개의 점이 있을까요? 그 다음 카드는요? 새 카드를 만들 때에 점의 수는 어떤 규칙이 있나요? 자세히 살펴 보면 적은 수의 카드로도 아주 큰 수를 만들 수 있습니다.

수의 배열을 잘 살펴보면 아주 흥미로운 관계를 발견할 수 있습니다.

1, 2, 4, 8, 16...

더해보세요: $1 + 2 + 4 = ?$ 답은 무엇일까요?

이번에는 $1 + 2 + 4 + 8 = ?$

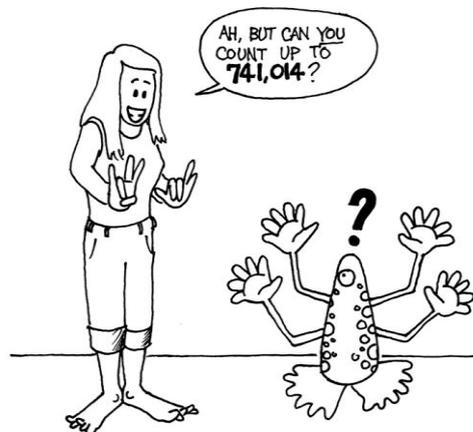
첫번째 수부터 모든 수를 다 더하면 어떻게 되나요?

‘손가락을 걷게 한다’ 라는 말을 들어본 적 있나요? 이제 여러분은 손가락을 구부려 수를 셀 것입니다. 여러분은 손가락의 갯수인 10 이상을 셀 수 있습니다. 이진법을 사용하여 10 이상을 세기 위해서 손가락이 수십개 달린 외계인이 될 필요가 없어요. 손가락 한개를 점이 찍힌 이진수 카드라고 생각하면 한 손의 손가락이 5 개이므로 0 부터 31 까지의 수를 표현할 수 있습니다. 0 을 포함한다면 한 손으로 32 가지 수를 표현하게 되는 것이죠.

편 손가락은 1, 접은 손가락은 0 입니다.

두 손을 모두 사용한다면 10 개의 손가락이므로 여러분들은 0 부터 1023 까지의 수를 나타낼 수 있게 되는 것입니다. 즉 1024 가지의 수를 나타낼 수 있다는 뜻이죠.

만약 발가락까지 활용할 수 있다면(정말 외계인이 되어야겠네요.^^) 더 큰 수를 셀 수 있을 것입니다. 한 손으로 32 가지 수를, 두 손으로 1024($32 \times 32 = 1024$)가지 수를 셀 수 있으니 발가락까지 사용하면 몇가지 수를 셀 수 있을까요?



워크시트 활동: 이진수 더 알아보기

1. 이진수의 맨 오른쪽에 0 을 추가했을 때에 또다른 재미있는 특징이 나타납니다. 십진수의 맨 오른쪽에 0 을 적는다면 그 수는 10 배로 커집니다. 예를 들어 9 의 옆에 0 을 적으면 90 이 되어 열배가 커지죠. 30 옆에 0 을 적으면 300 이 되어 마찬가지로입니다.
2. 이진수의 오른쪽에 0 을 추가하면 어떻게 될까요? 직접 해보세요.

1001 → 10010

(9) (?)

다른 이진수 몇개에 0 을 추가해서 규칙을 찾아보세요. 규칙이 뭘까요? 왜 그렇게 생각하나요?

3. 우리가 지금까지 썼던 각각의 카드는 컴퓨터의 '비트(bit)'를 의미합니다('bit'는 'binary digit'의 줄임말입니다). 우리가 지금까지 써온 알파벳(26 자)은 다섯 장의 이진수 카드(혹은 5 bits)로 전부 나타낼 수 있습니다. 하지만 컴퓨터는 이 알파벳이 대문자인지 소문자인지 구별해야하며 숫자인지 문장부호인지 특수문자(\$, ~)인지도 알아야 합니다.
4. 컴퓨터 키보드에는 몇개의 문자들이 있나 확인해보세요. 모든 문자를 컴퓨터가 표현하기 위해서는 몇 'bit'가 필요할까요?
5. 오늘날 사용되는 대부분의 컴퓨터들은 ASCII(American Standard Code for Information Interchange)라고 불리는 표현 방법을 사용합니다. 이 표현 방법은 문자를 나타낼때 'bit'들을 사용하는데, 영어를 사용하지 않는 몇몇 국가(중국어, 일본어, 한국어 등)에서는 더 긴 코드를 사용합니다.

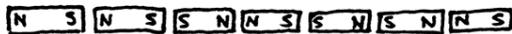


컴퓨터과학 이야기

컴퓨터들은 정보를 나타내기 위해 이진법을 사용합니다. 이진법이라고 하는 이유는 두가지의 수만을 사용하기 때문입니다(인간은 보통 10 가지를 사용하죠). 각각의 0 과 1 은 bit(binary digit)라고 부릅니다. 하나의 bit 는 트랜지스터가 켜지거나 꺼지는것, 콘덴서가 충전되거나 방전되는 것에 따라 컴퓨터의 메인 메모리에 표시됩니다.



정보가 전화선이나 무선으로 전송될 때에는 고음의 소리와 저음의 소리로 1 과 0 을 표시합니다. 자기디스크(하드디스크나 플로피디스크)와 테이프에 bit 를 기록할 때에는 코팅된 표면의 자기장 방향으로 나타냅니다(N-S 극 혹은 S-N 극).



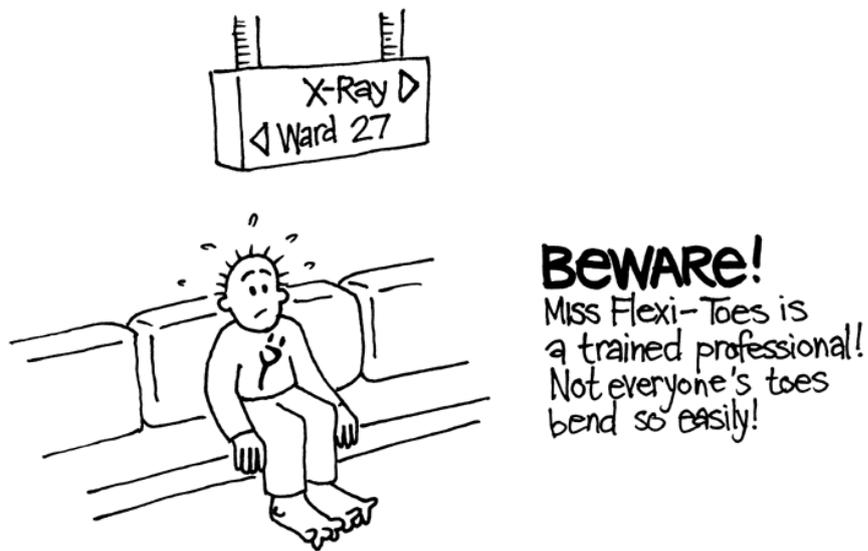
오디오 CD, CD-ROM 과 DVD 저장장치는 빛을 반사하느냐 반사하지 않느냐로 bit 를 나타냅니다.



컴퓨터가 2 개의 값을 사용하는 이유는 이러한 장치가 만들기 쉽기 때문입니다. 10 개의 반사 단계를 지닌 CD(0~9 까지의 수를 표현 가능)를 만들 수도 있었지만 그렇게 만들 경우 복잡하여 아주 비싸고 정밀한 장치가 필요해집니다. 또 다른 이유는 컴퓨터가 고전압과 저전압, N 극과 S 극 등을 통해 0 과 1 만을 저장할 수 있기 때문입니다. ‘빛난다’와 ‘빛나지 않는다’ 라고 쓰는 것보다 ‘1’, ‘0’으로 쓰는 것이 더 빠르기도 하지요. 컴퓨터로 나타내는 모든 정보들-문서, 그림, 노래, 영상, 숫자, 프로그램과 어플리케이션들-은 이진수로 구성됩니다.

하나의 bit 는 많은 정보를 나타낼 수 없습니다. 그래서 흔히 8 개의 bit 를 묶어 0 에서 255 까지의 수를 나타내도록 합니다. 이러한 8 개의 bit 묶음을 byte 라고 부릅니다.

컴퓨터의 속도는 한번에 얼마나 많은 bit 를 처리할 수 있느냐에 따라 결정됩니다. 예를들어 32bit 컴퓨터는 한번의 작업에 32bit 의 수들을 처리할 수 있습니다. 반면에 16bit 컴퓨터는 32bit 의 수들을 작은 조각으로 나눠 처리해 더 느립니다(더 싸긴 하지요!) 이진수를 통해 컴퓨터가 어떤 정보를 나타낼 수 있는지 다음 장의 활동들을 통해 알아보도록 하겠습니다.



해답과 힌트

이진수 (page 6)

3 필요한 카드 2 와 1

12 필요한 카드 8 과 4

19 필요한 카드 16, 2 와 1

어떤 수를 만들 수 있는 방법은 한가지 밖에 없다.

만들 수 있는 가장 큰 수는 31 이며 가장 작은 것은 0 이다. 그 사이의 수는 빠지는 것 없이 전부 만들 수 있다.

심화활동: 어떤 수든 1 만큼 양을 늘리고 싶다면 오른쪽에서부터 왼쪽 순으로, 뒤집어서 카드의 앞면이 나올 때까지 뒤집는다(앞면인 카드는 뒤집어서 뒷면으로 만들고, 뒷면인 카드를 뒤집어 앞면이 되면 더 이상 건드리지 않는다).

이진수 놀이하기(8 쪽)

10101 = 21, 11111 = 31

비밀 메시지 보내기(8 쪽)

메시지: HELP IM TRAPPED(감했습니다. 도와주세요!)

31 보다 큰 수 세기 (11 쪽)

처음부터 지금까지의 수를 차례로 더한다면 그 합은 다음에 나올 수보다 1 부족한 수가 된다.

발가락까지 사용해 수를 센다면 $1024 \times 1024 = 1,048,576$ 가지 수를 셀 수 있습니다.
— 0 부터 1,048,575 까지!

이진수 더 알아보기 (12 쪽)

이진수의 맨 오른쪽에 0 을 추가하면 그 수는 두배가 된다(십진수는 오른쪽에 0 을 추가하면 10 배가 된다).

컴퓨터가 모든 문자를 표시하기 위해서는 7bit 가 필요하다. 이는 128 이상의 문자를 표시하게 해준다. 7bit 는 1bit 만큼 여유를 두고 8bit byte 에 포함된다.

활동 2

수로 칠하기—이미지 표시

활동개요

컴퓨터가 그림, 사진을 어떻게 수로 표현하는지 알아보시다.

교과 연계

- ✓ 수학: 기하학 - 형태와 공간
- ✓ 기술: 다른 종류의 정보를 표현하기 위해 수를 이용하기
- ✓ 기술: 반복되는 데이터가 차지하는 공간 절약하기

필수 기술

- ✓ 수 세기
- ✓ 시각화하기

대상 연령

- ✓ 7 세 이상

수업 자료

- ✓ 프리젠테이션 슬라이드: 수로 칠하기 (19 쪽)

학생들이 필요한 것:

- ✓ 워크시트 활동: 키드 팩스 (21 쪽)
- ✓ 워크시트 활동: 나만의 그림 만들기 (22 쪽)

수로 칠하기

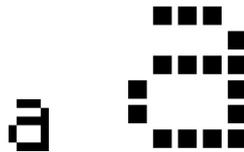
소개

토론용 질문

1. 팩스가 하는 일이 무엇인가요?
2. 컴퓨터는 어떤 상황에서 그림을 저장하나요? (그리기 프로그램, 그래픽이나 멀티미디어 시스템 게임 등)
3. 0,1 로 숫자를 사용하는 컴퓨터가 그림을 어떻게 저장할 수 있을까요?

사전활동으로 학생들이 팩스를 보내고 받는 놀이를 해볼 수 있습니다.

예시용 프리젠테이션



컴퓨터 화면(모니터)은 픽셀-*pixels* (picture elements)라고 불리는 작은 점으로 나누어져 있습니다.

흑백 그림에서 각각의 픽셀은 검은색과 흰색으로 나뉘집니다.

‘a’라는 글자를 확대하면 픽셀들이 보입니다. 컴퓨터가 그림을 저장할 때는 어떤 점이 흑인지 어떤 점이 백인지만 저장하면 됩니다.

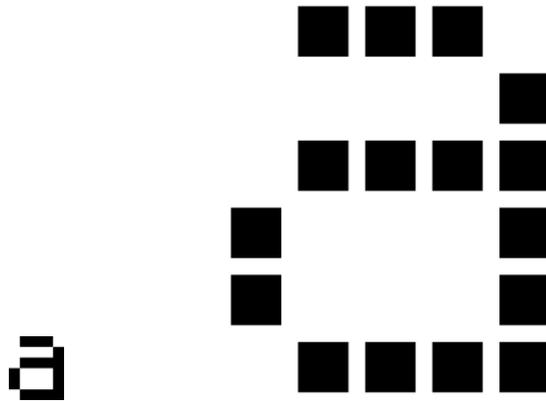
	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

위의 그림은 숫자로 어떻게 그림을 나타낼 수 있는지 보여줍니다. 첫줄은 한칸의 흰 픽셀과 세칸의 검정 픽셀, 또 한칸의 흰 픽셀로 구성되어 있습니다. 그래서 첫줄은 1,3,1로 나타낼 수 있습니다.

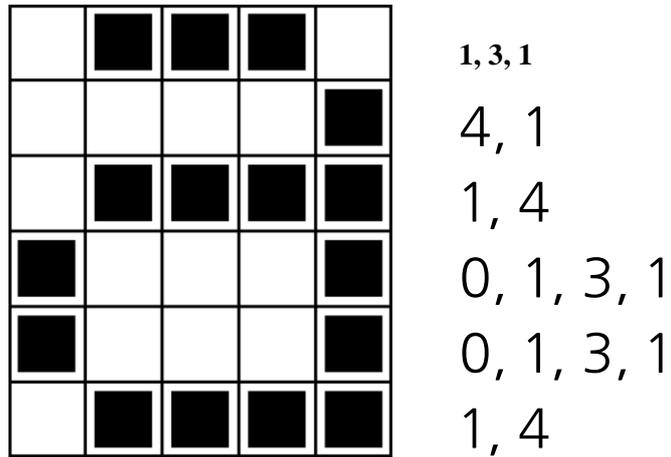
첫번째 수는 항상 흰색 픽셀의 수를 나타냅니다. 만약 첫 픽셀이 검정색이라면 그 줄은 0으로 시작하게 됩니다.

20 쪽의 워크시트에는 예시자료에서 보여준 방법으로 해독을 할 수 있는 그림들이 있습니다.

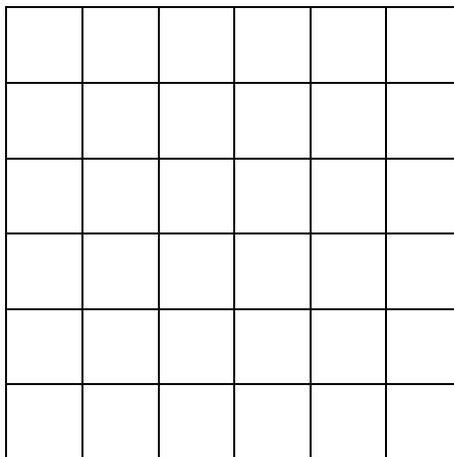
수로 칠하기



▲ 'a'라는 글자를 컴퓨터 화면에서 확대해보면 이를 표현하기 위한 픽셀들을 볼 수 있습니다



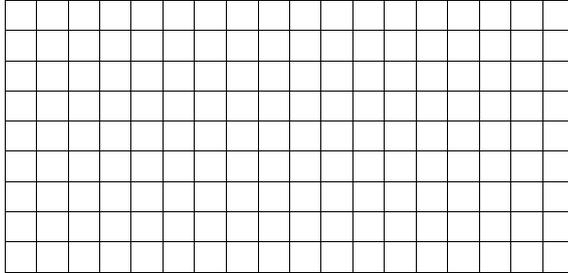
▲ 숫자를 코드로 만들어 나타낸 이미지



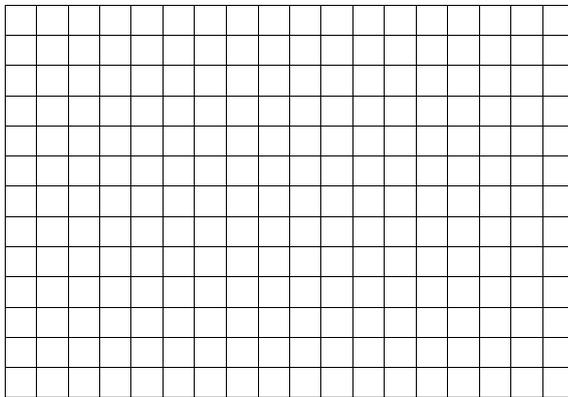
▲ 빈 모눈 칸(교육용)

워크시트 활동: 키즈 팩스

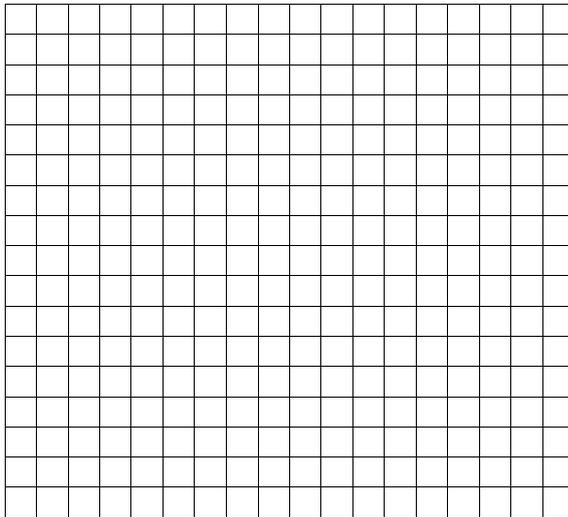
3 가지 활동은 점차로 복잡한 그림을 표현합니다. 실수하기 쉬우니 연필을 사용해
고치면서 즐겁게 활동하세요!



4. 11
4. 9. 2. 1
4. 9. 2. 1
4. 11
4. 9
4. 9
5. 7
0. 17
1. 15



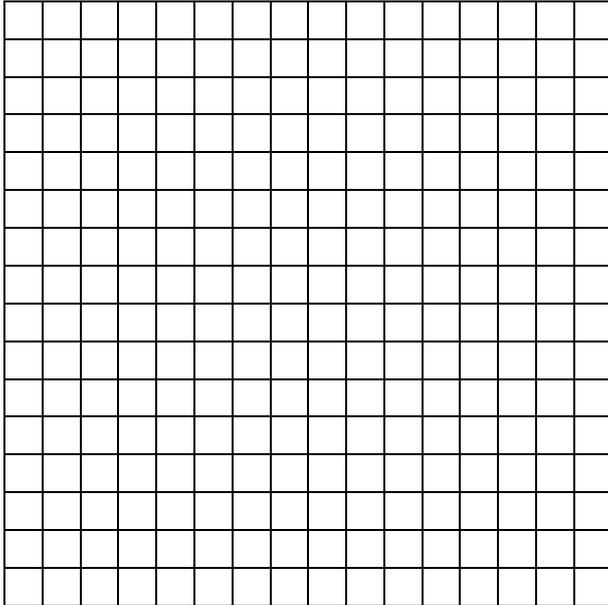
6. 5. 2. 3
4. 2. 5. 2. 3. 1
3. 1. 9. 1. 2. 1
3. 1. 9. 1. 1. 1
2. 1. 11. 1
2. 1. 10. 2
2. 1. 9. 1. 1. 1
2. 1. 8. 1. 2. 1
2. 1. 7. 1. 3. 1
1. 1. 1. 1. 4. 2. 3. 1
0. 1. 2. 1. 2. 2. 5. 1
0. 1. 3. 2. 5. 2
1. 3. 2. 5

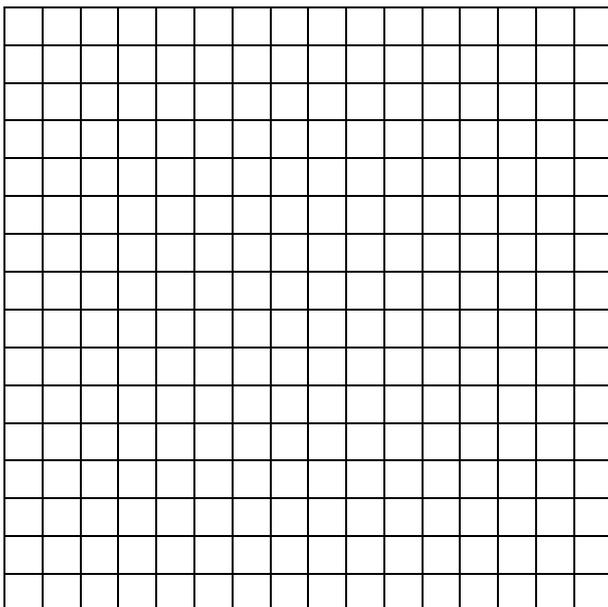


6. 2. 2. 2
5. 1. 2. 2. 2. 1
6. 6
4. 2. 6. 2
3. 1. 10. 1
2. 1. 12. 1
2. 1. 3. 1. 4. 1. 3. 1
1. 2. 12. 2
0. 1. 16. 1
0. 1. 6. 1. 2. 1. 6. 1
0. 1. 7. 2. 7. 1
1. 1. 14. 1
2. 1. 12. 1
2. 1. 5. 2. 5. 1
3. 1. 10. 1
4. 2. 6. 2
6. 6

워크시트 활동: 나만의 그림 만들기

이제 컴퓨터가 어떻게 수를 이용해 그림을 표시하는지 알았나요? 그럼 이제 직접 숫자로 표현된 그림을 친구들에게 보여주세요. 먼저 위에 있는 모눈 칸에 그림을 그린 뒤 그대로 아래에 있는 모눈 칸 옆에 숫자 코드만 적습니다. 그리고 점선을 따라 잘라 수 코드가 적힌 아래 모눈 칸을 친구에게 준 다음, 숫자 코드만을 보고 모눈 칸을 칠해보라고 하세요. (모든 칸을 다 사용할 필요는 없습니다)





한 단계 더!

1. 반투명종이를 모눈 칸 종이 위에 대고 그리도록 하면 다 그린 뒤에 모눈 없이 깔끔하게 그림만 나타나게 할 수 있습니다.
2. 모눈 칸에 칠하는 것 대신 스티커를 붙이거나 큰 격자 칸에 바둑알 같은 물건을 올려놓는 방법으로 활동할 수 있습니다.

생각해보기

이진수로 연속해서 배열되는 픽셀의 길이를 표현할때는 어려울 수 있습니다. 만약 수를 7 까지 밖에 쓸 수 없는 상황이라면 어떻게 연속되는 12 개의 검정 픽셀을 나타낼 수 있을까요? (좋은 방법은 7 개의 검정 픽셀을 배열한 뒤 0 개의 흰 픽셀과 5 개의 검정 픽셀을 배열하는 것입니다.)

컴퓨터과학 이야기

팩스는 흑백 페이지를 1000 × 2000 픽셀로 스캔해 전화선과 모뎀으로 다른 팩스에 전송해 인쇄하는 작은 컴퓨터입니다. 흔히 팩스의 이미지는 큰 흰색 블록이나 검정 픽셀로 되어있습니다. 컬러 사진 또한 무수히 반복되는 패턴을 가지고 있죠. 저장 공간을 아끼기 위해 컴퓨터 프로그래머들은 다양한 압축 기술을 사용합니다. 이 활동에 활용된 방법은 '반복 길이 부호화' 라고 불리는 효과적인 이미지 압축법입니다. 만약 압축하지 않고 이미지를 전송한다면 오랜 시간과 많은 저장공간이 필요할 것입니다. 그렇게 된다면 보낼 정보가 너무 많아 팩스를 보내거나 사진을 웹페이지에 올리기 힘들어집니다. 예를 들어 팩스의 이미지는 보통 원본의 1/7 로 압축됩니다. 압축하지 않는다면 전송하는데 7 배의 시간이 걸립니다.

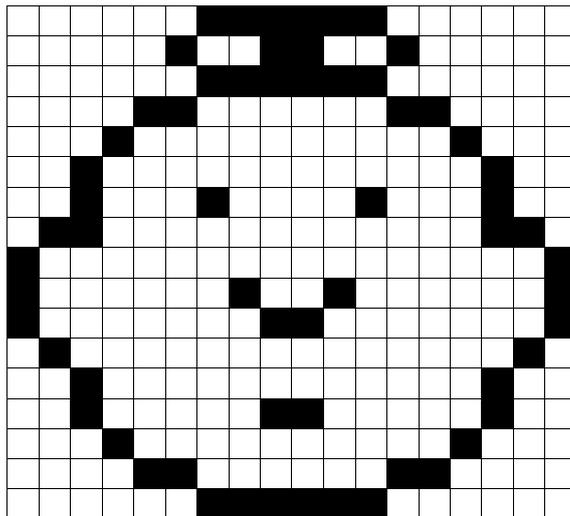
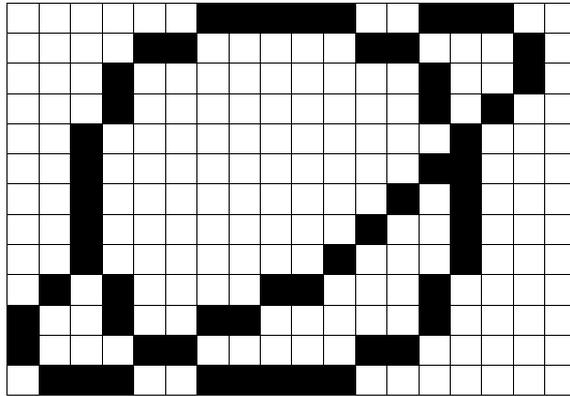
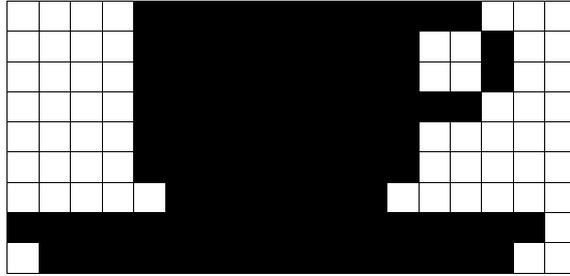
사진과 그림은 보통 1/10 혹은 1/100 으로 압축됩니다. 이는 더 많은 이미지를 저장할 수 있게 해주며 웹에서 더 빠르게 이미지를 볼 수 있게 해줍니다. 컴퓨터 그림 파일 중에 JPEG, GIF 그리고 PNG 라는 확장자를 가진 파일이 있는데 압축기술로 저장된 그림입니다.

프로그래머는 전송할 이미지를 압축하는데 가장 적당한 압축 기술을 선택할 수 있습니다.



해답과 힌트

키즈 팩스의 해답



활동 3

바로 그거야! —문장 압축

활동 개요

컴퓨터의 저장공간은 무한하지 않기 때문에 정보를 효율적으로 표현할 필요가 있습니다. 이러한 방법을 ‘압축’이라고 합니다. 자료를 저장할 때는 코딩(압축)을 하고, 보여주거나 검색할 때에는 디코딩(압축해제)을 하므로써 컴퓨터는 더 많은 정보를 저장하고 인터넷에 전송할 수 있습니다.

교과 연계

- ✓ 국어: 단어와 문장에서 패턴 찾기
- ✓ 기술: 반복되는 정보를 바탕으로 공간 절약하기

필수 기술

- ✓ 글 복제하기

대상 연령

- ✓ 9 세 이상

수업 자료

- ✓ 프리젠테이션 슬라이드 : 바로 그거야! (29 쪽)

학생들에게 필요한 것:

- ✓ 워크시트 활동: 바로 그거야! (30 쪽)
- ✓ 워크시트 활동: 숙련자를 위한 추가 활동 (31 쪽)
- ✓ 워크시트 활동: 짧고 달콤한 (32 쪽)
- ✓ 워크시트 활동: 심화 활동 (34 쪽)

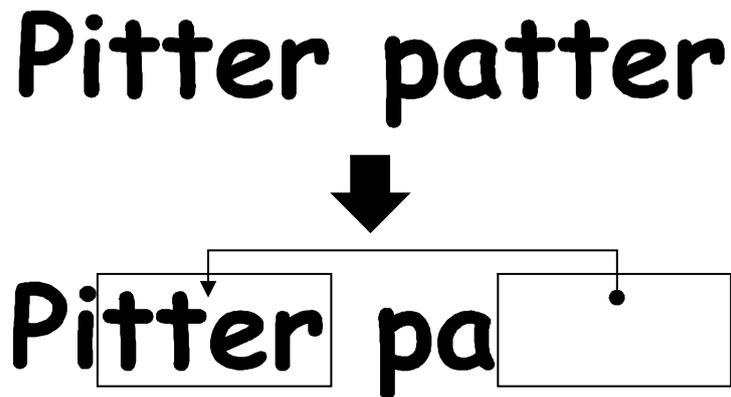
바로 그거야!

소개

컴퓨터는 많은 데이터를 저장하고 전송해야 합니다. 하지만 비용과 경제성 때문에 너무 많은 저장공간과 긴 시간을 쓸 수는 없습니다. 그래서 컴퓨터는 글자를 다음과 같이 압축합니다.

생각해보기

“The Rain” 슬라이드를 보여주세요 (29 쪽). 시에서 나타나는 글자에서 패턴을 찾습니다. 2 개 이상의 글자가 반복되는 문장을 찾을 수 있습니까? 심지어 단어나 문구 전체가 반복되는 것이 있습니까? (아래 그림처럼 상자로 바꿔보세요.)



바로 그거야!

The Rain

Pitter patter

Pitter patter

Listen to the rain

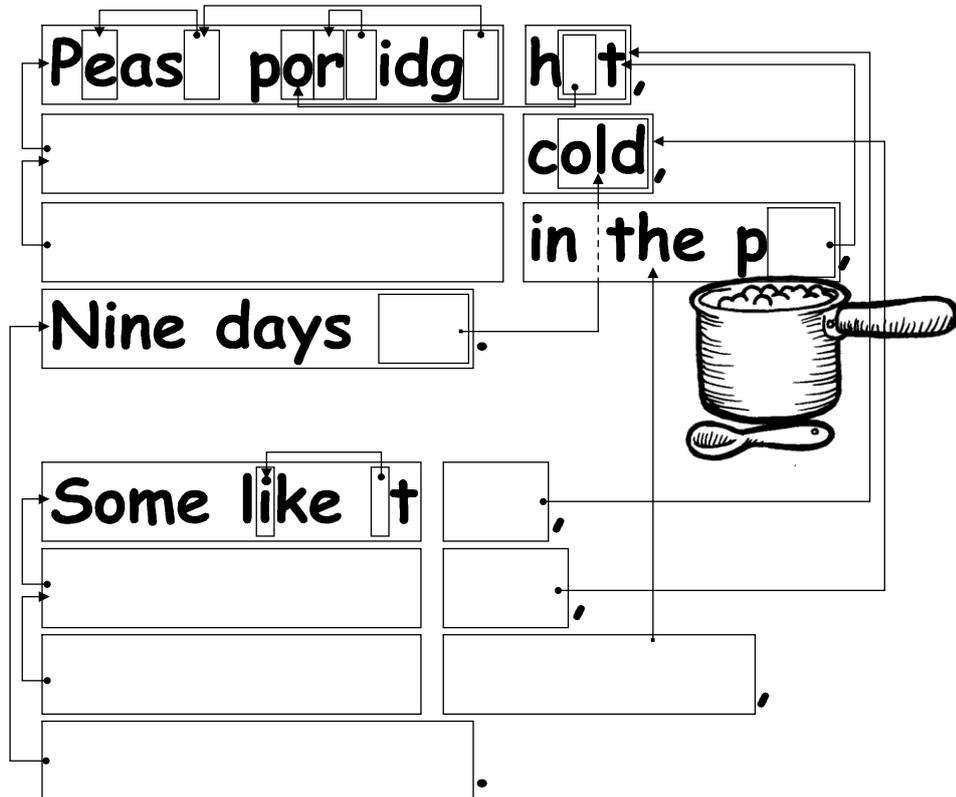
Pitter patter

Pitter patter

On the window pane

워크시트 활동: 바로 그거야!

앞의 시를 압축하기 위해 같은 글자 패턴을 박스로 표현하면 아래 그림처럼 많은 글자와 단어가 보이지 않습니다. 여러분이 사라진 글자와 단어를 채워 올바르게 완성해 보세요? 화살표가 가리키는 박스에서 단서를 얻을 수 있습니다.



여러분도 간단한 시나 전래 동요를 찾아 퍼즐을 만들어 보세요. 화살표는 항상 글의 앞부분을 가리켜야 합니다. 시는 우리가 글을 읽는 순서와 같이 왼쪽에서 오른쪽으로, 위에서 아래로 읽으면서 해석 되어야 합니다.

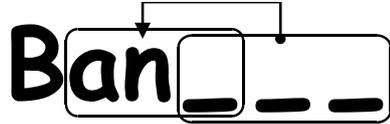
도전: 찾은 시에서 얼마나 적은 글자로 표현이 가능한지 알아보시다!

예시자료: Three Blind Mice, Mary Mary Quite Contrary, Hickory Dickory Dock—or try some Dr Seuss books!

힌트: 화살표가 너무 많아지지 않도록 하세요. 글자 주변에 공간을 많이 두어야 박스를 그리거나 화살표를 연결할 수 있습니다. 퍼즐을 만들 때에는 먼저 시를 적은 후에 박스를 어디에 그릴지 결정하는 것이 더 쉽습니다.

워크시트 활동: 심화 활동

이 퍼즐은 어떻게 풀 수 있을까요?



가끔은 사라진 부분이 또 다른 사라진 부분을 가리키는 경우가 있습니다. 이러한 경우 글자가 왼쪽에서부터 오른쪽으로 복제되어 있다면 해독이 가능합니다. 이것은 특정 글자나 패턴의 반복이 있을때 컴퓨터에서 유용하게 활용됩니다..

여러분도 이와 비슷한 단어를 찾아 이것과 비슷한 활동을 해보세요.

컴퓨터에서는 박스와 화살표가 수로 표현됩니다. 예를들면,

Banana

이것은 Ban(2,3)로 표현될 수 있습니다. 여기서 '2'는 복제가 시작되는 위치를 나타냅니다.

Ban---

그리고 '3'은 연속되는 세 글자를 복제하라는 뜻입니다:

Bana--

Banan-

Banana

이 단어를 압축하기 위해 두 수가 사용된 만큼, 최소 두개 이상의 글자 덩어리들이 압축되어야 효과가 있는 것입니다. 그렇지 않으면 저장 공간을 절약할 수 없습니다. 사실 한 글자를 압축하기 위해 두 수를 사용한다면 파일 크기가 커질 수도 있습니다.



컴퓨터가 압축하는 방식으로 단어를 나타내보세요. 친구들이 해독할 수 있을까요?

워크시트 활동: 짧고 달콤한

여기서 꼭 필요한 단어는 몇개일까요?

여러분의 컴퓨터에 저장공간(하드디스크)의 용량을 최대한 아끼려고 합니다. 이미 나타난 두개 이상의 글자들은 줄로 그어 없애세요. 이것들은 화살표로 바꿀수 있으니 더 이상 필요가 없습니다. 여러분의 목표는 지울 수 있는 글자를 최대한 많이 찾는 것입니다.

I know an old lady who swallowed a bird
How absurd! She swallowed a bird!
She swallowed the bird to catch the spider
That wriggled and jiggled
and tickled inside her
She swallowed the spider to catch the fly
I don't know why she swallowed a fly
Perhaps she'll die...

워크시트 활동: 심화 활동

진짜 어려운 압축을 해볼까요?

컴퓨터 프로그램은 아래의 이야기에서 1,633 글자가 반복되어 지워질 수 있다는 것을 찾아냈습니다. 당신은 얼마나 찾아낼 수 있나요? 최소 2 번 이상 반복되는 글자를 묶어서 표시해야 합니다. 잘 해보세요!

Once upon a time, long, long ago, three little pigs set out to make their fortunes. The first little pig wasn't very clever, and decided to build his house out of straw, because it was cheap. The second little pig wasn't very clever either, and decided to build his house out of sticks, for the "natural" look that was so very much in fashion, even in those days. The third little pig was much smarter than his two brothers, and bought a load of bricks in a nearby town, with which to construct a sturdy but comfortable country home.

Not long after his housewarming party, the first little pig was curled up in a chair reading a book, when there came a knock at the door. It was the big bad wolf, naturally.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the first little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he *did* huff, and he *did* puff, and the house soon collapsed. The first little pig ran as fast as he could to the house of sticks, and was soon safe inside. But it wasn't long before the wolf came calling again.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the second little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he *did* huff, and he *did* puff, and the house was soon so much firewood. The two terrified little pigs ran all the way to their brother's brick house, but the wolf was hot on their heels, and soon he was on the doorstep.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the third little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he huffed, and he puffed, and he huffed some more, but of course, the house was built of brick, and the wolf was soon out of breath. Then he had an idea. The chimney! He clambered up a handy oak tree onto the roof, only to find that there *was* no chimney, because the third little pig, being conscious of the environment, had installed electric heating. In his frustration, the wolf slipped and fell off the roof, breaking his left leg, and severely injuring his pride. As he limped away, the pigs laughed, and remarked how much more sensible it was to live in the city, where the only wolves were in the zoo. And so that is what they did, and of course they all lived happily ever after.

컴퓨터과학 이야기

컴퓨터의 저장능력은 놀라울 정도로 향상되고 있습니다. 지난 25 년동안 일반적인 컴퓨터로 저장할 수 있는 양이 만배로 커졌습니다. 하지만 우리 여전히 컴퓨터에 저장해야 할 대용량의 자료를 계속해서 만들어내고 있습니다. 컴퓨터는 저장할 용량만 있다면 모든 영화, 음악, 도서관의 책의 내용을 모두 기록할 수 있습니다. 용량이 큰 파일은 다운로드 받는 시간이 오래 걸리기 때문에 인터넷에서 문제가 됩니다. 또 우리는 컴퓨터를 더 작게 만들려고 합니다. 심지어 휴대전화나 손목시계, 가전제품에도 많은 정보를 저장하려고 합니다.

이러한 문제는 저장공간의 크기를 키우는 대신 정보를 압축하여 해결할 수 있습니다. 정보를 압축하고 해제하는 과정은 컴퓨터에서 자동으로 이루어지기 때문에 우리 단지 디스크에 더 많은 정보를 저장하거나, 웹페이지 화면을 더 빨리 뜨게 하기 위해서 컴퓨터가 더 많은 작업을 해야 한다는 것만 알 뿐입니다.

저장을 효율적으로 하기 위한 다양한 압축 방법이 개발되어 왔습니다. 우리가 이번 활동을 통해 글자 덩어리들을 화살표로 가리키는 방법에서 살펴보았던 압축 방법은 1970 년대에 두 이스라엘 교수가 개발하여 이들의 이름을 따 'Ziv-Lempel coding' 혹은 'LZ coding'이라고 불립니다. 이 방법은 어떤 언어에서도 활용될 수 있고 데이터를 압축해 크기를 쉽게 줄일 수 있습니다. 때로는 PC 에서 'zip'이라는 파일 확장자를 사용하며 'GIF', 'PNG' 등의 이미지에도 활용됩니다. 또한 고속인터넷에도 활용되어 네트워크 선으로 전송해야 할 데이터의 양을 줄여 더 빨리 전송시킬 수 있게 도와줍니다.

자주 사용되는 글자는 짧은 코드를 지녀야 한다는 생각을 바탕으로 만들어진 압축방법도 있습니다. 모스 부호가 그 예입니다.

해답과 힌트

바로 그거야! (30 페이지)

Pease porridge hot,
Pease porridge cold,
Pease porridge in the pot,
Nine days old.
Some like it hot,
Some like it cold,
Some like it in the pot,
Nine days old.

활동 4

카드마술—에러 감지와 수정

활동 개요

데이터가 디스크에 저장되어 있거나 다른 컴퓨터에서 전송 된 경우, 그 과정에서 데이터는 변경되지 않았을 것이라 생각하게 됩니다. 하지만 가끔 일이 잘못되면 데이터가 실수로 변경되기도 합니다. 이 활동은 데이터가 손상되었을 경우에 그것을 확인하고 수정하는 방법을 마술놀이를 통해 안내합니다.

교과 연계

- ✓ 수학 : 수 - 계산과 어림셈 탐구하기.
- ✓ 수학 : 대수 -패턴과 관계, 누락 된 값 해결 탐구하기.
- ✓ 수학 : 행과 열, 좌표
- ✓ 기술 : 유효성 검사 데이터

필수기능

- ✓ 셈하기
- ✓ 홀수와 짝수에 대한 이해

대상연령

- ✓ 7 살 이상

수업 자료

- ✓ 36 개의 카드 세트(한쪽 면만 색칠된 "냉장고용 자석"카드 세트)
- ✓ 설명을 위한 화이트 보드 (2 명에 하나씩)
- ✓ 36 개의 동일한 카드(한쪽 면만 색칠된 카드)

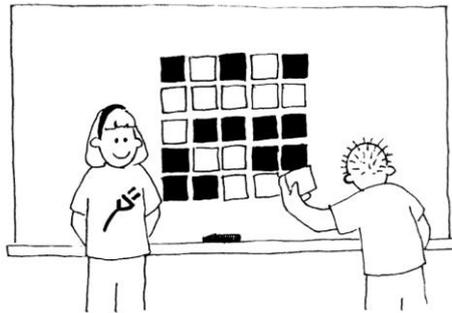
카드마술

설명

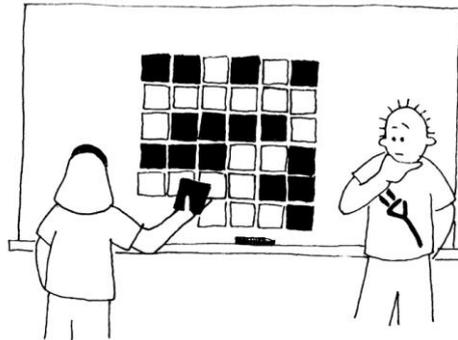
여러분은 이제 마술사입니다!

양쪽 면의 색이 다른 카드를 여러장 준비하세요. 냉장고 자석에 쓰이는 평평한 마그네틱 카드 또는 고무자석을 사용하는 것도 좋습니다. 양쪽면에 자석 있는지 확인하세요. 대부분 한쪽에만 자석이 있는 경우가 많습니다. 이 경우 여러분은 자석끼리 붙이거나, 한쪽 면을 흰 면으로 둘 수 있습니다.

1. 양면이 무작위로 섞여있는 5×5 의 사각 카드를 배치할 학생을 뽑으세요.



배치된 카드를 조금 복잡하게 만들기 위해 "좀 더 어렵게 만들어 보겠다."라고 말하면서 다른 행이나 열을 추가하세요.



새롭게 추가된 카드들은 속임수입니다. 여러분은 각각의 행과 열에서 색 카드가 짝수가 되도록 맞추기 위해 추가할 카드를 잘 선택해야 합니다.

2. 여러분이 눈을 감고있는 동안 다른 학생이 하나의 카드를 뒤집게 합니다. 뒤집힌 카드를 포함한 행과 열은 현재 색 카드의 숫자를 홀수로 만들기 때문에 뒤집힌 카드를 식별할 수 있습니다. 학생들이 이 방법을 추측 할 수 있을까요?

마술의 속임수 알려주기

1. 짝활동으로, 학생들은 자신이 가진 25 개의 카드를 5×5 배열로 배치합니다.
2. 각 행과 열에 색칠된 카드는 몇 개 있습니까? 색칠된 카드의 개수는 홀수입니까, 짝수입니까? 기억하세요. 0 은 짝수입니다.
3. 이제 색칠된 카드의 수가 항상 짝수가 되는지 확인하면서 각 행의 여섯 번째 카드를 추가하세요. 이렇게 추가된 카드를 “홀짝 맞춤”카드라고 부릅니다.
4. 각 열의 색칠된 카드 개수가 짝수가 되도록 확인하면서, 카드 하단의 6 번째 행에 새 카드를 추가합니다. 이렇게 되면 6×6 배열의 카드로 배치됩니다.
5. 이제 카드를 뒤집습니다. 여러분은 행과 열에서 무엇을 알게 되었 있나요?(뒤집힌 카드가 속한 곳 색 카드의 개수는 홀수가 됩니다.) 홀수가 되는 곳을 알 수 있도록 홀짝맞춤 카드를 이용한 겁니다.
6. 이제 순서를 바꿔가며 이 '속임수 마술'을 실제로 해보세요.

심화 활동:

1. 카드 말고 무엇이든 두가지 '상태'가 적합한 다른 물건을 사용하여 시도합니다. 예를 들어, 0 또는 1 이 인쇄된 카드나 동전의 앞면 또는 뒷면을 이용하여 활동할 수 있습니다.
2. 만약 둘 혹은 그 이상의 카드가 뒤집힌다면 어떨까요? (무엇인가 바뀌었다라는 것은 알 수 있지만, 뒤집힌 두 장의 카드가 어느 것인지 정확히 알 수없는 경우가 발생합니다. 보통 카드 두 쌍 중 하나일 것입니다. 4 장이 뒤집힌 경우 모든 홀짝맞춤 카드가 짝수가 될 수도 있고, 그래서 뒤집힌 곳이 발견되지 않을 수 있습니다.)
3. 훨씬 더 큰 카드의 배열로 놀이해보세요. 추가된 행과 열이 10×10 배열로 되게끔, 9×9 배열의 카드로 시작합니다. (어떤 크기 배열로 놀이하든 상관 없으며, 반드시 사각형 일 필요도 없습니다.)
4. 또 다른 흥미로운 활동은 오른쪽 아래 카드를 이용하는 것입니다. 한 행을 짝수가 되게끔 선택했다면, 왼쪽 열은 짝수가 맞습니까? (여러분이 홀짝맞춤 카드를 사용하는 경우 대답은 항상 '예'가 됩니다.)
5. 이 카드 활동에서는 색칠된 카드가 짝수라는 점을 이용하는 홀짝 맞춤 카드를 사용해야만 합니다. 짝수 말고 홀수를 이용하여 홀짝 맞춤을 할 수 있을까요? (이것은 가능하지만, 행과 열의 수가 모두 짝수 또는 홀수인 경우 모두 오른쪽 아래 카드는 그 행과 열에 대해 똑같이 적용됩니다. 예를 들어, 5×9 배열이나 4×6 배열은 상관없지만, 3×4 배열은 이 카드 속임수를 적용할 수 없습니다.)

실생활 사례

이 같은 검사 기술은 책 ISBN 코드와 바코드에 사용됩니다. 출판된 책은 일반적으로 뒷면에 10 자리 또는 13 자리의 코드가 있습니다. 마지막 숫자는 활동에서의 훌쩍 맞춤 카드처럼 확인숫자입니다. 이것은 여러분이 ISBN (국제 표준 도서 번호)가 사용된 책을 주문하는 경우, 웹 사이트에서는 여러분의 주문이 잘못되지 않았는지 확인할 수 있다는 것을 의미합니다. 이렇게 간단하게 확인할 수 있으며 우리가 잘못된 책을 기다리는 일이 생기지 않습니다.

10 자리의 책 코드를 검사하는 방법:

첫 번째 숫자에 10 을 곱하고, 두번째는 9 와, 세번째는 8 과.. 아홉번째 숫자에 이르기까지 역순으로 된 숫자를 곱합니다. 각각 곱해진 값을 함께 더합니다.

예를 들어, ISBN 0-13-911991-4 계산 결과

$$\begin{aligned} & (0 \times 10) + (1 \times 9) + (3 \times 8) + (9 \times 7) + (1 \times 6) \\ & + (1 \times 5) + (9 \times 4) + (9 \times 3) + (1 \times 2) \\ & = 172 \end{aligned}$$

그런 다음 나온 값을 11 로 나눕니다. 나머지는 무엇입니까?

$$172 \div 11 = 15 \dots 7(\text{나머지})$$

만약 나머지가 0 이라면, 검사 합은 0 입니다. 0 이 아닌 경우, 검사 합을 구하기 위해 11 에서 나머지를 뺍니다.

$$11 - 7 = 4$$

다시 보세요. 4 가 ISBN 의 마지막 숫자입니까? 맞습니다!

ISBN 의 마지막 자리가 4 가 아니었다면, 오류가 생겼음을 알 수 있습니다.

그것은 10 번째 값의 검사 합을 찾아낼 수 있게 합니다. 이 때, 하나의 숫자 이상을 필요로 합니다. 이 경우, 문자 X 가 사용됩니다.

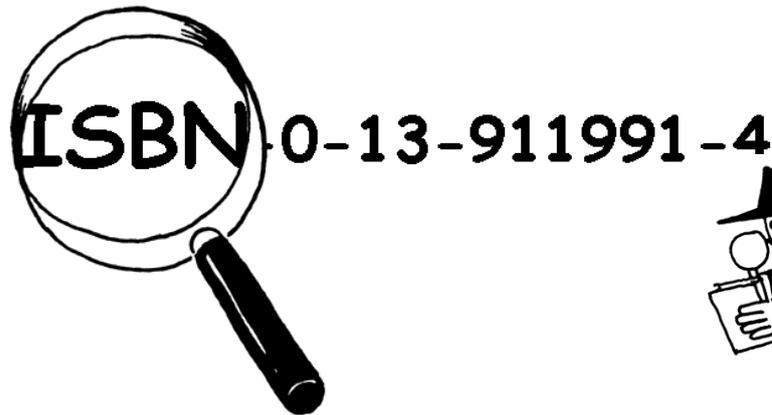


▲ Weet-Bix™상자의 바코드

숫자 검사의 또 다른 예로는 식료품 바코드입니다. 식료품 바코드는 다른 공식을 사용합니다(13 자리의 책 코드에 사용되는 것과 같은 공식입니다). 만약 바코드가 잘못된 경우, 마지막 숫자는 그 계산된 값과 달라야합니다. 이 경우 스캐너에서 신호음이 나고 점원은 코드를 재검색합니다. 숫자 검사는 은행 계좌 번호, 사회 보장 번호, 주민등록번호, 세금 영수증 번호, 기차와 철도 예매권 및 기타 여러 응용 프로그램에 사용됩니다. 사람들이 숫자를 복사하고 제대로 입력되었는지 확인이 필요한 프로그램에 말이지요.

책 검사!

형사 블록 버스터
책 추적 서비스.



저렴한 비용으로 ISBN 검사 합계를 확인해 드립니다.
사이트에 가입하여 여러분의 교실이나 도서관에 있는 실제 ISBN
코드에 대해 검사해 보세요.

검사의 합이 정확합니까?

가끔 오류가 발생합니다.

일반적으로 발생하는 오류 :

- ✕ 숫자가 바뀐 값을 가진 경우;
- ✕ 인접한 두 숫자가 서로 바뀐 경우;
- ✕ 숫자 하나가 삽입된 경우;
- ✕ 숫자 하나가 지워진 경우;

10 자리 숫자 검사 합에 해당하는 문자 X가 적힌 책을 찾을 수 있습니까? 그런 책을 찾는 것은 어렵지 않습니다—11 권 중 1 권 꼴로 있습니다. 발견되지 않는 오류의 종류는 무엇일까요? 숫자를 변경하고, 여전히 정확한 검사 합을 구할 수 있나요? 만약 두 숫자가 일반적인 입력 오류로 뒤바뀐다면 어떨까요?

컴퓨터과학 이야기

자신의 은행 계좌로 10 달러 현금을 입금하고 있다고 상상 해보세요. 출납원은 입금할 비용을 입력하면, 그것은 중앙 컴퓨터로 전송됩니다. 그런데 입금한 비용이 전송되는 동안 약간의 오류가 발생하여 10 달러 코드가 1 천 달러 또는 1 달러로 변경되었다고 가정해보세요. 여러분이 그 고객이라면 여러분에게 문제가 없습니다만 정확히 은행의 문제가 되고 서로 곤란한 상황이 됩니다.

이처럼 송신된 데이터의 오류를 검출하는 것은 중요합니다. 그래서 수신 컴퓨터가 도착한 데이터가 전송되는 중에 전기적 간섭이나 오류에 의해 손상되지 않았는지 확인해야 합니다. 때때로 원본 데이터에 에러가 발생되어 송신되었을 때 다시 보낼 수 있습니다. 하지만, 디스크가 열 또는 물리적으로 손상을 입었을 때는 가능하지 않을 수 있습니다. 만약 데이터가 우주 공간에서 수신되는 경우, 에러가 발생했다면 재전송을 기다리는 것은 매우 번거로운 일입니다! 참고로 지구에서 가까운 태양계 행상인 목성에서 라디오 신호를 받는 데도 반 시간 이상이 걸립니다!)

우리는 데이터가 손상되었을 때를 그 오류를 검출하여 확인 할 수 있어야 하고, 오류를 수정하여 원래의 데이터를 재구성 할 수 있어야 합니다.

"카드 뒤집기"게임에 사용 된 것과 동일한 기술을 컴퓨터에도 사용됩니다. 허수 행과 열로 비트를 추가하고, 각각의 행 및 열로 홀짝 맞춤 비트를 추가함으로써, 우리는 에러가 발생했는지 여부를 감지할 수 있을 뿐 아니라 어디에서 발생했는지도 알 수 있습니다. 잘못된 비트는 다시 변경하고, 오류 보정을 하여 정상적인 원본 데이터를 받을 수 있습니다.

물론 컴퓨터는 다중 오류를 감지하고 정정 할 수 있는 보다 복잡한 오류 제어 시스템을 사용합니다. 컴퓨터의 운영체제는 하드 디스크에 저장된 데이터의 일부에 오류가 발생하는 경우에도 복구할 수 있도록 에러 정정에 많은 양의 공간을 할당합니다. 이번 활동에서 사용하였던 홀짝 맞춤 방식과 유사한 방법을 사용합니다.



해답과 힌트

하나의 숫자가 증가하고 다른 숫자가 감소되어 보상(값이 상쇄)되었을 때, ISBN-10 검사 합에 의해 오류가 감지되지 않아 합계가 여전히 같을 수 있습니다. 그러나 계산이 수행되는 방식 때문에 이런 일이 없을 수도 있습니다. 예를 들어 ISBN-13 과 같은 또 다른 시스템에서, 세 개의 연속 된 숫자가 역전되는 등의 검출되지 않는 다른 유형의 오류가 있습니다. 부정확한 숫자를 입력하거나, 두 개의 인접한 자리가 바뀌는 경우의 일반적인 오류는 대부분은 감지됩니다.

활동 5

스무고개 - 정보 이론

활동개요

1000 페이지에 달하는 책에는 얼마나 많은 정보가 들어 있을까요? 빈 종이 1000 장에도 정보가 들어있을까요? ‘반지의 제왕’ 책과 1000 페이지의 전화번호부 중 어느 것에 더 많은 정보가 들어 있을까요? 이를 측정할 수 있다면, 정보를 저장하는 데 필요한 공간의 양을 추정할 수 있습니다.

예를 들어, 여러분은 다음과 같은 문장을 읽을 수 있습니까?

Ths sntnc hs th vwls mssng.

모음에 충분한 정보가 없기 때문에 읽기거나 이해하기 불가능할 것입니다. 이번 활동에서는 정보의 양을 측정하는 방법을 알아보겠습니다.

교과 연계

- ✓ 수학 : 수 - 수에 대한 탐구 : 이상, 이하, 범위
- ✓ 수학 : 대수 - 패턴과 순서
- ✓ 영어 : 맞춤법, 글의 요소를 인식

필수 기능

- ✓ 수의 범위와 수 비교하기
- ✓ 추론, 연역
- ✓ 질문하기

대상 연령

- ✓ 10 살 이상

수업 자료

- ✓ 첫번째 활동을 위한 준비물은 없습니다.

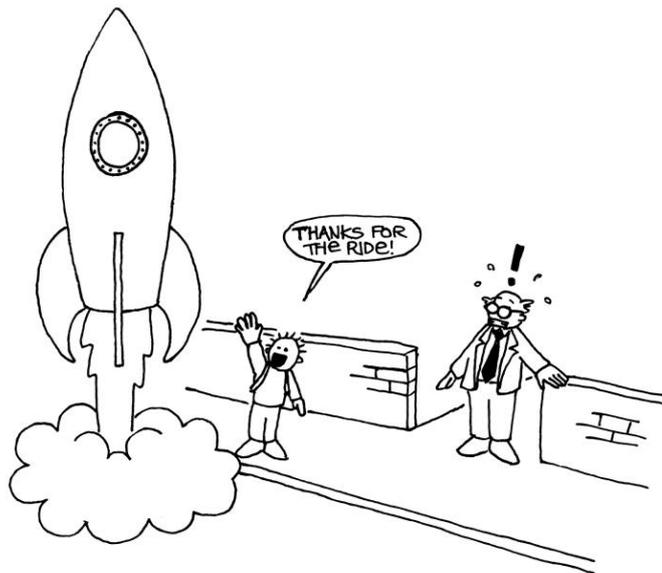
심화활동을 하는 학생 개인 준비물

- ✓ 워크시트 활동 : 의사결정 나무 (49 장)

스무고개

설명

1. 정보가 무엇이라고 생각하는지 학생들과 함께 이야기해 봅시다.
2. 책 한 권에 얼마나 많은 정보가 있는지 어떻게 측정 할 수 있을까요? 측정할때 페이지 수나 단어의 수가 중요한가요? 어떤 한 권의 책이 다른 책보다 더 많은 정보를 가질 수 있을까요? 아주 지루한 책이나 매우 흥미로운 책인 경우는 어떨까요? 이런 저런 내용의 문구가 포함 된 400 페이지의 책은 친구들과 직접 이야기하는 것 그리고 전화로 통화하는 것 보다 더 많은 정보를 담고 있을까요?
3. 컴퓨터 과학자에 의해 메시지 또는 미디어로부터 정보를 측정하는지 색다른 방법이 제기되었습니다. 이미 알고 있는 것을 말하는 것은— 예를 들어, 항상 학교에 걸어오는 친구가 "나는 오늘 학교에 걸어왔다"라고 이야기할 때— 어떠한 정보도 제공하지 않습니다. 놀라운 일이 아니기 때문이지요. 만약 친구가 "나는 오늘 헬리콥터를 타고 학교까지 왔다"라고 이야기 한다면 그것은 놀라운 일이 되기 때문에 우리에게 새로운 정보가 됩니다.
4. 메시지의 놀라운 가치를 어떻게 측정 할 수 있을까요?
5. 한 가지 방법은 이 정보를 추측하는 것이 얼마나 어려운지 보는 것입니다. 친구가 "나는 오늘 학교에 어떻게 왔는지 맞춰봐"라고 이야기 한다면, 아마 처음에는 걸어왔을 것으로 추측할 것입니다. 헬리콥터나 우주선을 타고 왔다는 정보에 도달하기 전까지 더 많은 추측들을 하게 될 것입니다.
6. 메시지에 포함 된 정보의 양은 추측하기에 얼마나 쉬운지 또는 복잡한지에 따라 측정할 수 있습니다. 다음 놀이를 통해 알아봅시다.



스무고개 활동

스무고개 놀이가 적용된 활동입니다. 어떤 수를 생각하고 있는 학생에게 다른 학생들은 정답을 추측할 때까지 질문을 할 수 있습니다. 정답을 맞추기 위해서는 정답을 알고있는 학생에게 반드시 '네' 또는 '아니오'라고 대답할 수 있도록 질문해야 합니다.

제안

- ✓ 1 부터 100 사이의 수
- ✓ 1 부터 1000 사이의 수
- ✓ 1 부터 1,000,000 사이의 수
- ✓ 모든 정수
- ✓ 6 의 배수 (appropriate to the group). 처음부터 끝까지 순서대로 추측한다. (예. 2, 4, 6, 8, 10)

정답을 맞힐 때까지 질문의 개수를 셉니다. 이것을 이용해 "정보"의 값을 측정합니다. 많은 질문을 할수록 정보의 양이 많다는 것을 알 수 있습니다.

생각해보기

어떤 전략을 사용 했습니까? 어느 전략이 최고의 방법입니까?

1 에서 100 사이의 숫자를 찾기 위해 매번 정답이 포함되어 있는 숫자의 범위에서 반이 되는 수를 부른다면 단 7 번의 질문을 하게 됩니다. 예를 들어:

50 보다 작습니까?	네.
25 보다 작습니까?	아니오.
37 보다 작습니까?	아니오.
43 보다 작습니까?	네.
40 보다 작습니까?	아니오.
41 보다 작습니까?	아니오.
42 입니까?	네!

여기서 흥미로운 점은 추측할 숫자의 범위가 1000 으로 증가한다고 해도 10 배의 노력이 들지는 않습니다. 단지 질문을 3 번만 더 하면 될 뿐입니다. 숫자의 범위가 두 배로 될 때마다 당신은 답을 찾기 위해 질문을 한 번씩만 더 하면 됩니다.

이어지는 심화 활동은 학생들의 이해를 돕고 놀이를 좀 더 능숙하게 수행할 수 있게 해줍니다.

심화: 메시지에 얼마만큼의 정보가 들어 있을까요?

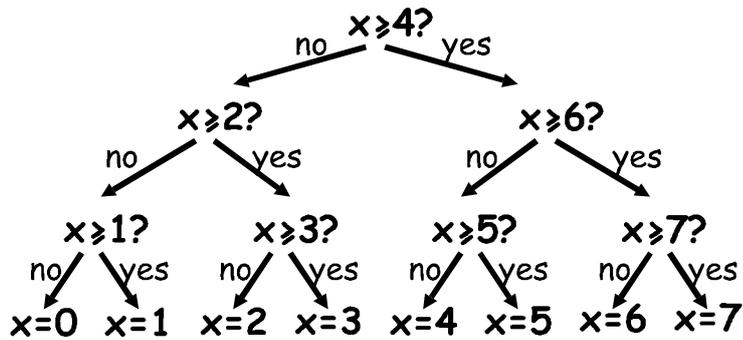
위의 놀이에서 활동한 것과 같이 컴퓨터 과학자들은 숫자와 관련된 정보만을 추측하지 않습니다— 단어나 문장의 다음에 올 가능성이 있는 문자를 추측할 수도 있습니다.

4-6 단어의 짧은 문장으로 추측 게임을 해보세요. 문자는 처음부터 끝까지, 올바른 순서대로 추측할 수 있어야 합니다. 각각의 문자를 찾아내는 데 얼마나 걸리는지 탐색하고, 그 결과를 기록하기 위해 다른 한명의 학생이 역할을 나눕니다. 네/아니오 대답을 이끌어내는 어떤 질문을 사용해도 좋습니다. 예를 들면 "그것은 t 인가요?" "그것은 모음인가요?" "그것은 알파벳 m 앞에 오나요?" 단어 사이의 띄어쓰기 공간도 "문자"로 간주하고 세어야 합니다. 순서를 바꿔가며 메시지의 어느 부분을 가장 쉽게 발견할 수 있는지 확인합니다.

워크시트 활동: 의사결정트리

이미 질문의 정답을 찾기 위한 전략을 알고 있다면, 아무것도 묻지 않고도 메시지를 전송할 수 있습니다.

0 에서 7 사이의 숫자를 추측하기 위한 '의사결정트리' 입니다.



네/아니오 대답만으로 숫자 5 를 '추측'하기 위해서 필요한 질문은 몇 번인가요?

네/아니오 대답만으로 다른 숫자를 '추측'하기 위해서 필요한 질문은 몇 번인가요?

여기서 매우 재미있는 것을 발견할 수 있습니다.

나무의 최종 행 아래에 숫자 0, 1, 2, 3 ...7 이 있지요.

각각의 나무 가지 분기점을 보세요. 무엇이 보이나요?

. no=0 아니면 yes=1 입니다. 이진 숫자로 십진수의 숫자를 확인해 보세요 (이진수 활동 1 을 다시 살펴 보세요).

숫자 추측 게임에서는 응답이 이와 같은 수로 나타날 수 있도록 질문을 잘 선택해야 합니다.

이제 0 에서 15 사이의 숫자를 추측할 수 있는 자신만의 의사결정트리를 만들어보세요.

심화활동 : 특정 인물의 나이를 맞추기 위한 의사결정트리를 만들어 보세요?
문장에서 다음에 올 단어를 맞추기 위한 의사결정 트리는 어떨까요?

컴퓨터과학 이야기

클로드 새너라는 유명한 미국의 수학자는 (꼭예사이기도 함 ^^) 이 게임에 대해 많은 실험을 했습니다. 그는 비트에 담긴 정보의 양을 측정했습니다— 스무고개 게임에서 각각의 네 / 아니오 대답은 1/0 비트에 해당하지 않습니다. 왜냐하면 메시지에 포함된 "정보"의 양은 이미 우리가 알고 있는 전략에 포함되어 있기 때문이죠. 때때로 우리는 많은 질문을 할 필요가 없습니다. 이 경우, 메시지의 정보의 양이 적습니다. 예를 들어, 하나의 동전 던지기의 정보는 일반적으로 1 비트(앞면과 뒷면)에 해당 합니다. 그러나 만약 통계적인 확률이 1/2 이 아닌 상황, 예를 들면 10 번 중 9 번이 앞면이 나오는 일이 발생한다면, 그 정보는 더이상 1 비트가 아닌게 됩니다—믿거나 말거나, 더 적은 정보량을 갖습니다. 동전 던지기에 대해 하나의 질문만으로 어떻게 답을 알아낼 수 있을까요? 간단합니다—단지 "그 다음 두 개의 동전이 모두 앞면으로 던져졌는가?"와 같은 질문을 사용하면 됩니다. 한쪽으로 편향된 동전을 던지는 것을 반복할 경우, 대답은 80%의 비율로 "네"일 것입니다. "아니오"대답이 나온 20 %의 경우에는 두 개의 추가 질문을 해야합니다. 하지만 평균적으로는 동전을 던질 때마다 '앞면이 나올까? 뒷면이 나올까?'라는 하나의 질문만으로도 충분합니다! 앞면이 여러 번 나왔든 말든 다음에 던져 나올 동전의 앞면은 1/2 의 확률로 나오게 될테니까요.



새너는 메시지에 담긴 정보 내용을 "엔트로피"라고 불렀습니다. 엔트로피는 가능한 결과의 수(동전 던지기의 경우 2 가지, 주사위는 6 가지 등) 뿐만 아니라 그 발생 확률에 의해서도 좌지우지됩니다. 하지만 이미 알고 있지 않던 새로운 정보를 추측하기 위해서는 더 많은 질문이 필요합니다. 왜냐하면 우리가 미처 몰랐던 것에 대한 더 많은 정보들이 있기 때문입니다. 헬리콥터를 타고 학교에 왔다는 상황처럼 말이지요.

메시지의 엔트로피는 컴퓨터 과학자들에게 매우 중요한 개념입니다. 이 개념에서 여러분은 엔트로피보다 적은 공간을 차지하도록 메시지를 압축 할 수 없습니다. 엔트로피가 최적의 값으로 수렴되기 때문입니다. 최고의 압축 시스템은 스무고개 놀이와 같은 방식의 알고리즘입니다. 질문 목록은 정답 발견 해결 전략에 의해 나중에 재생될 수 있으므로 컴퓨터 프로그램을 이용하여 그것을 '추측' 할 수 있도록 만들 수 있다면, 답변 길이(비트)만큼 저장공간을 예측하거나 직접 저장할 수 있고 우리는

그것을 복원할 수 있습니다. 이것을 이용한 문자 압축 시스템은 원래 크기의 약 1/4 로 텍스트 파일을 줄일 수 있습니다. 어떤 문자가 다음에 나올지 예측할 수 있으므로 굳이 그 문자를 담은 정보를 보내지 말고 예측할 수 있는 방법을 같이 최소한의 문자와 함께 보내면 수신되는 컴퓨터에서 복원이 가능합니다. 이것을 통해 네트워크 송신 데이터량을 줄이거나 저장 공간을 크게 아낄 수 있죠!

정보를 추측하는 방법은 컴퓨터 사용자가 어떤 글자를 입력하고 다음의 입력을 예측할 수 있는 컴퓨터 인터페이스를 구축하기 위해 사용될 수 있습니다. 이것은 타이핑이 불편한 장애인들에게 매우 유용하고 정상적인 사람들이 잘못 입력한 글자를 자동으로 교정해줄 수 있습니다. 컴퓨터는 그들이 다음에 입력될 가능성이 있는 것을 제시하고, 사용자들은 원하는 것을 선택만 하면 됩니다. 좋은 시스템은 각 질문마다 예 /아니오 대답의 평균횟수만 필요로 합니다. 그리고 마우스나 키보드를 제어하는 데 필요한 작은 움직임 조차도 나타내기 어려운 이들에게 큰 도움이 될 수 있습니다. 이러한 종류의 시스템은 일부 휴대폰에서 문자'입력'을 편리하게 하는데 사용됩니다.

해답과 힌트

네/아니오에 대한 한가지 답변의 정보는 정확히 하나의 비트에 해당하지 않습니다— "그것은 50 개 이상인가?"와 같이 간단한 질문인지, 또는 "20 에서 60 사이인가?"처럼 더 복잡한 질문인지에 따라 그 정보의 양이 다릅니다.

숫자 추측 게임에서, 질문이 특정한 방식으로 선택되는 경우, 답변의 순서는 이진 표현에 불과한 숫자들입니다. 세 개의 이진표현 011 은 0="no", 1="yes", 1="yes"로 의사 결정 트리에 대한 답변 "아니, 그래, 그래"으로 표시됩니다.

당신이 누군가의 나이를 추측할 때 의사결정트리를 사용할 수 있습니다.

문장의 문자에 대한 결정은 이전의 문자가 무엇인지에 달려 있습니다.

Part II

컴퓨터 일 시키기

—알고리즘

컴퓨터 일 시키기

컴퓨터는 설정된 명령어 목록에 따라 작동합니다. 이 명령은 정렬, 검색 및 정보를 전송할 수 있도록 되어있습니다. 가능한 한 일을 빨리 처리하고 네트워크로 정보를 보내고자 한다면, 많은 데이터에서 명령어를 찾기 위한 좋은 방법이 필요합니다.

알고리즘은 주어진 임무를 수행하기 위한 절차적인 명령들의 집합입니다. 알고리즘은 컴퓨터 과학의 핵심개념입니다. 컴퓨터가 문제를 해결하는 방법이 알고리즘입니다. 어떤 알고리즘은 같은 일을 처리하는 다른 알고리즘들보다 빠릅니다. 그리고 좋은 알고리즘은 이전에 시간상 해결이 불가능했던 문제를 해결하기 위해 개발되었습니다 — 예를 들어, 파이의 자리 수백만 자리를 찾는 것, 인터넷 문서 중에서 여러분의 이름이 포함된 모든 페이지를 찾는 것, 여러가지 택배물건을 가장 짧은 경로로 배달하는 방법을 찾는 것, 100 자리가 넘는 매우 큰 숫자에 소수가 있는지 여부를 알아내는 것 등이 그러한 일입니다.

“알고리즘”이라는 말은 Khowarizm 에서 유래되었습니다.

모세의 아들인 Mohammed ibn Musa Al-Khowarizmi—는 기원 후 800 년경 바그다드에 있는 지혜의 집으로 알려진 학술 센터에서 활동했습니다. 그의 일은 힌두 예술을 아랍과 유럽으로 전파시키는 것이었습니다. 라틴어로까지 전파되던 기원 후 1120 년쯤, 이러한 그의 업적을 기려 처음으로 "알고리즘"이라는 말이 불리게 되었습니다.

활동 6

전함놀이—탐색 알고리즘

활동 개요

우리는 종종 컴퓨터로 엄청난 양의 정보 탐색을 하기도 합니다. 그래서 이 일을 신속하고 효율적으로 처리할 방법을 개발해야 합니다. 이번 활동에서는 선형 검색, 이진 검색, 해싱 검색이라 불리는 세 가지 다른 탐색 방법을 알아보겠습니다.

교과 연계

- ✓ 수학 : 수 - 수에 대한 탐구 : ~보다 큰, ~보다 작은, 같은
- ✓ 수학 : 기하 - 모양과 공간에 대한 탐구 : 좌표
- ✓ 컴퓨팅 : 알고리즘

필수 기능

- ✓ 논리적 추론

대상 연령

- ✓ 9 살 이상

수업 자료

학생 개인 준비물:

- ✓ 전함 놀이 학습지
 - 게임 1 을 위한 1A, 1B
 - 게임 2 를 위한 2A, 2B
 - 게임 3 을 위한 3A, 3B
- ✓ 또한 추가 게임 시트의 복사본이 필요합니다.(1A', 1B', 2A', 2B', 3A', 3B')

전함놀이

활동소개

1. 교실의 앞쪽에 15 명의 학생이 줄을 서도록 합니다. 각 학생에게 수가 적힌 카드를 무작위로 나누어 줍니다. 카드를 받은 학생들은 교실에 숫자카드를 숨깁니다.
2. 다른 학생에게 4~5 개의 사탕이 담긴 상자를 줍니다. 사탕을 가진 학생들은 사탕을 친구들에게 주고 카드에 적힌 숫자를 볼 수 있습니다. 자신의 사탕을 모두 사용하기 전에 숫자를 찾는다면, 다른 학생 대신 남은 사탕을 모두 가질 수 있습니다.
3. 원한다면 반복합니다.
4. 이제 카드를 섞고 다시 나누어줍니다. 이번에는 학생들이 오름차순으로 정렬하도록 합니다. 탐색 과정은 반복됩니다.

숫자를 정렬해서 섰을 경우, 가장 좋은 방법은 줄의 가운데 서있는 학생에게 사탕을 주고 카드를 보는 것입니다. 이 과정을 반복함으로써 학생은 단지 세 개의 사탕을 사용하여 해당 번호를 찾을 수 있게 됩니다. 이 방법의 효율성은 이전의 장에서 확인했으니 명백할 것입니다.

활동

학생들은 전함 놀이를 수행하며 컴퓨터가 탐색을 어떻게 하는지에 대한 방법을 알 수 있습니다. 이제는 배를 찾기 위해 사용하는 전략에 대해 생각을 해봅시다.

전함놀이—선형 탐색 게임

학생들에게 아래 설명을 읽어주세요.

1. 짝과 함께 활동하세요. 한 사람은 시트 1A를 갖고, 다른 사람은 시트 1B를 가지세요. 여러분의 짝에게 시트를 보여주지 마세요!
2. 자신이 가진 게임 시트지 맨 윗 줄에 있는 배를 선택하여 동그라미를 그리고, 짝에게 그 숫자를 불러 줍니다.
3. 돌아가면서 짝의 배가 어디에 있는지 추측합니다. (배 아래에 적혀있는 문자를 말하고 짝은 그 문자에 해당하는 배의 숫자를 말합니다.)
4. 짝의 배를 찾는 데 시간이 얼마나 걸리나요? 그것이 이 게임에 대한 점수가 됩니다.

A. (시트 1A'와 시트 1B'는 게임하기를 더 원하거나 실수로 짝의 시트를 본 학생들을 위해 제공되는 여분입니다. 시트 2A', 2B' 그리고 3A', 3B'은 뒤에 이어지는 게임을 위한 것입니다.)

생각해보기

1. 무엇이 점수가 될까요?
2. 최소 그리고 최대 점수는 얼마가 될까요? (학생들이 2번 같은 배를 지목하지 않는 이상 최소 1점과 최대 26점입니다. 이 방법은 모든 위치를 선처럼 나열된 값을 하나씩 연결해가고 있기 때문에 '선형 탐색'이라고 합니다.)

전함놀이—이진수 탐색 게임

소개

게임에 대한 설명은 이전 게임과 동일하지만, 선박에 표기된 숫자는 오름차순입니다. 시작하기 전에 학생들에게 설명합니다.

1. 짝과 함께 활동하세요. 한 사람은 시트 2A, 다른 사람은 시트 2B 를 가집니다.
“자신의 시트를 짝에게 보여주지 마세요!”
2. 자신이 가진 게임 시트지 맨 윗 줄에 있는 배를 선택하여 동그라미를 그리고, 짝에게 그 숫자를 불러 줍니다.
3. 순서대로 여러분 짝의 배가 어디에 있는지 추측합니다. (여러분은 배의 이름을 말합니다. 그리고 짝은 그 이름에 있는 배의 숫자를 말합니다.)
4. 짝의 배를 찾기 위해 얼마큼 걸립니까? 그것이 이 게임에 대한 점수가 됩니다.

생각해보기

1. 무엇이 점수가 될까요?
2. 낮은 점수를 얻기위해 어떤 전략을 사용하나요?
3. 어느 배를 처음으로 선택해야 합니까? (가운데 위치한 배를 선택해야 선택의 폭을 절반으로 줄일 수 있습니다.) 어느 위치를 다음으로 선택해야 할까요? (가장 좋은 방법은 또다시 선택된 구역의 절반을 선택하는 것입니다.)
4. 이 전략이 적용되면 배를 찾기 위해 얼마나 많은 기회를 사용해야 할까요? (대부분 5).
5. 이 방법은 문제 상황에서 항상 두 부분으로 나누기 때문에 ‘이진 탐색’이라고 불립니다.

전함놀이—해상 탐색 게임

소개

1. 이전 게임처럼 시트에 있는 배를 선택하고 그 숫자를 짝에게 알려주세요.
2. 이 게임에서 여러분은 배에 있는 열 (0 ~ 9)을 확인할 수 있습니다. 단순히 배들의 숫자를 함께 더합니다. 합계의 마지막 숫자는 배가 속한 열의 번호입니다. 예를 들어, 배의 위치에 있는 숫자가 2345 라면, 각 숫자를 더합니다.(2+3+4+5), 14 라는 값이 나옵니다. 합계의 마지막 숫자는 4 입니다. 그렇다면 그 배가 속한 곳은 4 열입니다. 당신이 해당 열을 알게되면 해당 열에 있는 선박 중 추측해야 합니다. 이 방법은 숫자가 점차 좁혀지기 때문에 '해상'이라고 부릅니다.
3. 이제 새로운 전략을 사용하는 게임을 해봅시다. 동일한 시트를 사용하여 하나 이상의 게임을 할 수 있습니다—단지 다른 열에서 선택하면 됩니다.
4. (참고로, 다른 게임과 달리 시트 3A'와 3B'도 쌍으로 사용됩니다, 배가 속한 열의 패턴이 일치해야하기 때문입니다.)

생각해보기

1. 이전의 활동처럼 점수를 모으고 결과를 논의합니다.
2. 어느 배가 찾기 쉽나요? (그 열에서 혼자인 것입니다.) 어느 배가 찾기에 어렵나요? (그 열에 다른 배가 많은 것입니다.)
3. 세 가지 탐색 과정 중 어느 전략이 가장 빠릅니까? 이유가 뭔가요?

세 가지 탐색 전략 각각의 장점으로 무엇이 있습니까? (두번째 전략은 첫번째 방법보다 빠르지만, 첫번째 전략은 배들을 순서대로 정렬할 필요가 없습니다. 세번째 전략은 보통 두 가지 전략보다 빠르지만, 어떤 때는 매우 느리기도 합니다. 가장 좋지 못한 전략은 모든 배가 같은 열에서 끝나는 경우입니다. 첫번째 방법과 같이 느리게 찾을 수 있게 됩니다)

심화 활동

1. 학생들이 세 가지 전략을 사용하여 자신의 게임을 만들어 보도록 합니다. 두 번째 게임을 위해 오름차순으로 숫자를 정렬해야 합니다. 어려운 해싱 게임을 어떻게 하면 효율적으로 이해 할 수 있는지 질문합니다. (가장 어려운 게임은 모든 배가 같은 열에 있을 때입니다.) 가능한 가장 쉽게 만들려면 어떻게 합니까? (각 열에 같은 수가 있는 배가 있어야 합니다.)
2. 배가 없는 것으로 보여질 때는 어떻게 할까요?(선형 탐색 게임에서 26 번 가능합니다. 이진수 탐색 게임에서는 둘로 나누기 위해 5 번이 쓰입니다. 해싱 탐색 게임에서는 많은 배가 있는 열과 관련됩니다.)
3. 백개의 배의 위치 (약 6 번), 천개의 위치 (약 9 번), 또는 만개의 위치 (약 19 번)가 있다면 이진수 탐색 전략을 사용 할 때 몇 번의 기회가 필요할까요? (물어보야 하는 기회가 늘어나며 배의 숫자를 비교하는 데 오래 걸립니다. 크기가 두 배가 될 때마다 배의 숫자도 비례하여 늘어납니다.)

My Ships

Number of Shots Used:

9058	7169	3214	5891	4917	2767	4715	674	8088	1790	8949	13	3014
A	B	C	D	E	F	G	H	I	J	K	L	M
8311	7621	3542	9264	450	8562	4191	4932	9462	8423	5063	6221	2244
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1A

My Ships

Number of Shots Used:

1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625
A	B	C	D	E	F	G	H	I	J	K	L	M
2587	7187	5258	8020	1919	141	4414	3056	9118	717	7021	3076	3336
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1B

My Ships

Number of Shots Used:

163	445	622	1410	1704	2169	2680	2713	2734	3972	4208	4871	5031
A	B	C	D	E	F	G	H	I	J	K	L	M
5283	5704	6025	6801	7440	7542	7956	8094	8672	9137	9224	9508	9663
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2A

My Ships

Number of Shots Used:

33	183	730	911	1927	1943	2200	2215	3451	3519	4055	5548	5655
A	B	C	D	E	F	G	H	I	J	K	L	M
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2B

My Ships

Number of Shots Used:

0	A 9047 B 1829	1	C 3080 D 9994	2		3	E 5125 F 1480 G 8212	4	H 8051 I 1481 J 4712 K 6422	5	L 7116 M 8944 N 4128	6	O 6000 P 7432 Q 4110	7	R 9891 S 1989 T 2050 U 8199	8	V 4392	9	W 1062 X 2106 Y 5842 Z 7057
---	------------------	---	------------------	---	--	---	----------------------------	---	--------------------------------------	---	----------------------------	---	----------------------------	---	--------------------------------------	---	--------	---	--------------------------------------

Your Ships

Number of Shots Used:

0	A B C D	1	E F G	2	H I J	3	K	4	L M N	5		6	O P Q	7	R S T U	8	V W X	9	Y Z
---	------------------	---	-------------	---	-------------	---	---	---	-------------	---	--	---	-------------	---	------------------	---	-------------	---	--------

3A

My Ships

Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9
A 9308 B 1478 C 8417 D 9434	E 6519 F 2469 G 5105	H 1524 I 8112 J 2000	K 4135	L 9050 M 1265 N 5711		O 4200 P 7153 Q 6028	R 3121 S 9503 T 1114 U 7019	V 2385 W 5832 X 1917	Y 1990 Z 2502

Your Ships

Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9
A B	C D		E F G	H I J K	L M N	O P Q	R S T U	V	W X Y Z

3B

My Ships

Number of Shots Used:

6123	1519	9024	5164	2038	2142	7156	9974	9375	7104	1004	1023	5108
A	B	C	D	E	F	G	H	I	J	K	L	M
1884	3541	5251	4840	3289	3654	2480	5602	8965	4053	2405	2304	1959
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1A'

My Ships

Number of Shots Used:

2387	9003	3951	5695	1284	4761	7118	1196	1741	3791	3405	3132	6682
A	B	C	D	E	F	G	H	I	J	K	L	M
9493	9864	7359	1250	7036	2916	7562	9299	8910	6713	5173	8617	4222
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1B'

My Ships

Number of Shots Used:

28	326	943	1321	1896	2346	2430	2929	3106	3417	4128	4717	4915
A	B	C	D	E	F	G	H	I	J	K	L	M
5123	5615	6100	7015	7120	7695	7812	8103	8719	9020	9608	9713	9911
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2A'

My Ships

Number of Shots Used:

56	194	306	1024	1510	1807	2500	2812	3011	3902	4178	5902	5915
A	B	C	D	E	F	G	H	I	J	K	L	M
6102	6526	6818	7020	7155	7913	8016	8230	8599	8902	9090	9526	9812
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships

Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2B'

My Ships

Number of Shots Used:

0	A 1982 B 7841	1	C 6113 D 1055	2		3	E 9121 F 1011 G 2984	4	H 5009 I 2651 J 1751 K 4848	5	L 1248 M 1716 N 2148	6	O 2004 P 5173 Q 2806	7	R 9369 S 1321 T 3004 U 7190	8	V 3285	9	W 9172 X 2052 Y 6012 Z 7525
---	------------------	---	------------------	---	--	---	----------------------------	---	--------------------------------------	---	----------------------------	---	----------------------------	---	--------------------------------------	---	--------	---	--------------------------------------

Your Ships

Number of Shots Used:

0	A B C D	1	E F G	2	H I J	3	K	4	L M N	5		6	O P Q	7	R S T U	8	V W X	9	Y Z
---	------------------	---	-------------	---	-------------	---	---	---	-------------	---	--	---	-------------	---	------------------	---	-------------	---	--------

3A'

My Ships

Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9
A 8615 B 7003 C 1991 D 6211	E 1361 F 7644 G 5600	H 7726 I 9003 J 5557	K 3000	L 1814 M 2002 N 8844		O 9656 P 4002 Q 1221	R 6993 S 3121 T 4300 U 1907	V 8208 W 9423 X 4176	Y 2917 Z 4122

Your Ships

Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9
A B	C D		E F G	H I J K	L M N	O P Q	R S T U	V	W X Y Z

3B'

컴퓨터과학 이야기

컴퓨터의 역할은 많은 정보를 저장하고 신속하게 옮기고 탐색할 수 있어야 합니다. 세계 최대 검색 문제 중 하나는 순식간에 수십억 개의 웹 페이지를 검색할 수 있는 인터넷 검색 엔진과 관련된 것입니다. 컴퓨터를 통해 인터넷 문서를 검색하도록 요청하는 단어, 바코드 번호 또는 저자의 이름 등과 같은 데이터를 검색 키라고 합니다.

컴퓨터는 빠른 처리 속도를 이용하여 찾고자 하는 정보를 검색할 때 저장 공간의 시작 부분에서부터 시작하여 원하는 정보를 찾을 때까지 계속 반복합니다. 이것은 우리가 선형 검색 게임에서 했던 것과 같은 방식입니다. 그러나 이 방법은 찾고자 하는 정보의 양이 많을 때 아무리 빠른 초대형 컴퓨터조차도 매우 오래 걸립니다. 예를 들어, 슈퍼마켓의 선반에 10,000 가지의 다양한 제품을 가지고 있다고 가정합니다. 바코드가 계산대에서 스캔 될 때, 컴퓨터는 제품명, 가격을 찾기 위해 10,000 가지 번호를 살펴봅니다. 만약 1 개의 코드를 확인할 때 1 초가 걸린다면 1000 개의 코드를 하나씩 순서대로 확인한다고 가정한다면, 전체 목록을 살펴는 데에 10 초가 걸립니다. 그럼 이 슈퍼마켓에서 가족을 위한 식료품들을 탐색하기 위해 얼마나 걸릴지 상상해보세요!

보다 좋은 방법은 이진 검색 방법입니다. 이 방법에서 숫자는 순서대로 정렬합니다. 목록의 중간 항목을 확인하는 것은 검색 열쇠의 반절을 이미 검색하게 될 것입니다. 찾고자 하는 품목이 발견될 때까지 반복하면 됩니다. 슈퍼마켓의 예로 돌아 가면 10,000 항목은 1 초에 200 번째에 도달하므로 14 가지 조사로 원하는 물건을 검색할 수 있습니다.

데이터를 찾기 위한 세 번째 전략은 해싱검색이라고 불립니다. 여기서 검색의 열쇠는 어디에서 정보를 찾을 수 있는지 정확히 나타내기 위해 조작해야 합니다. 예를 들어, 검색 키가 전화 번호인 경우에는 11로 나누었을 때 때 나머지를 더할 수 있습니다. 이러한 관점에서, 활동 4에서 살짝 논의된 숫자 확인과 같은 해시 열쇠를 사용하는 방법과 같습니다— 데이터 값의 작은 부분은 다른 데이터가 처리되는에 따라 달라집니다. 이것을 이용하면 일반적으로 컴퓨터는 바로 원하는 것을 찾을 수 있습니다. 몇 개의 키가 동일한 위치에 끝날 가능성이 약간 있기 때문에 이 경우에 컴퓨터는 그것이 찾고 있는 것을 찾을 때까지 검색하면 됩니다.

검색시 데이터를 순서대로 유지하는 것이 중요하지 않다면 컴퓨터 프로그래머는 일반적으로, 느린 응답을 무시해도 되는 경우를 제외하고는 검색을 위한 해싱 전략의 일부분을 사용합니다.

활동 7

동보와 흠쪽이—정렬 알고리즘

활동 개요

컴퓨터로 일을 처리하기 위해서는 여러 가지 목록들을 정렬해야 합니다. 예를 들어 이름 목록을 알파벳 순으로 정렬하기도 하고, 일정이나 메일 목록을 날짜별로 정렬하기도 하며 물건들을 번호순으로 정렬하기도 합니다. 정렬하는 것의 잇점은 우리가 어떤 것을 좀 더 빠르게 찾을 수 있도록 해줄 뿐만 아니라 양쪽 극단의 값을 쉽게 볼 수 있게 도와줍니다. 만약 시험 결과를 성적 순으로 정렬하게 된다면 여러분은 가장 낮은 점수와 가장 높은 점수를 확인할 수 있을 것입니다.

그런데 잘못된 방법을 사용한다면 아무리 빠른 컴퓨터라도 큰 목록을 정렬하는 데에 매우 오랜 시간이 걸릴 수 있습니다. 다행히 컴퓨터 과학자에 의해 빠르고 효과적인 정렬 방법이 개발되었습니다. 이번 활동에서는 정렬을 위한 여러 방법을 살펴보고, 단순한 정렬 방법 대신 더욱 더 일을 빠르게 처리할 수 효과적인 정렬 방법을 살펴보겠습니다.

교과 연계

- ✓ 수학: 측정 - 실제로 무게 재보기
- ✓ 컴퓨터: 알고리즘

필수기능

- ✓ 균형 저울 사용하기
- ✓ 순서정하기
- ✓ 비교하기

대상 연령

- ✓ 8 살 이상

수업 자료

학생

- ✓ 크기가 같고 무게가 다른 8 개의 상자 (예를 들어 모래를 채운 우유 팩이나 필름 통)
- ✓ 균형저울
- ✓ 워크시트 활동: 무게 정렬하기 (76 쪽)
- ✓ 워크시트 활동: 분할하고 획득하기 (77 쪽)

뚱보와 흘죽이

생각해보기

컴퓨터는 종종 목록을 정렬합니다. 일상 생활에서 정렬이 필요하거나 적용하여 효과적으로 일을 처리하는 경우에 대해 브레인스토밍해봅시다. 그것들이 순서대로 정렬되지 않는다면 어떠한 일이 일어나게 될까요?

컴퓨터는 한 번에 두개의 값만을 비교합니다. 여러 개를 동시에 비교하지 못합니다. 다음 페이지의 활동에서는 학생들이 정렬의 개념을 먼저 익힐 수 있도록 두개의 값만 비교하도록 제한합니다.

활동 안내

1. 학생들을 모둠으로 나눕니다.
2. 각 모둠에 74 쪽 활동 워크시트와 준비물(무게를 잴 것들, 저울)을 나누어 줍니다.
3. 학생들은 활동을 한 후 결과에 대해 논의해봅니다.

워크시트 활동: 무게 정렬하기

목표: 무게를 일정한 순서로 정렬하는 가장 좋은 방법 알아내기.

준비물: 모래나 물, 8 개의 똑 같은 상자, 균형 저울

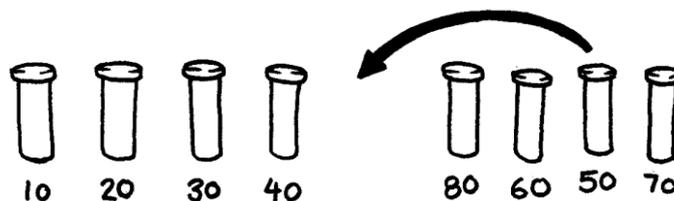
활동 안내:

1. 각각의 상자에 다른 양의 모래 혹은 물로 채워준 후 잘 막습니다.
2. 속이 보이지 않는 상자들을 잘 섞어서 무게를 더 이상 알 수 없도록 합니다.
3. 가장 가벼운 상자를 찾아봅니다. 가장 쉽게 찾을 수 있는 방법은 무엇일까요?
주의: 여러분은 각각의 상자가 얼마나 무거운지 알아내기 위해 저울만 사용할 수 있습니다. 또한 한 번에 두개의 상자만 무게를 비교할 수 있습니다.
4. 저울만 사용하여 무작위로 3 개의 상자를 선택한 후 가장 가벼운 것부터 가장 무거운 것 순으로 정렬해봅시다. 어떻게 하였나요? 여러분이 비교하는 횟수를 가장 적게 한다고 할 때 그 횟수는 얼마인가요? 왜 그럴까요?
이제 모든 상자를 가장 가벼운 것부터 가장 무거운 것 순으로 정렬해봅시다.

위의 활동을 다 마친 후, 각각의 상자를 두 개씩 무게를 다시 재어서 여러분의 정렬이 잘 맞았는지 확인해봅시다.

선택 정렬

컴퓨터가 사용할 수 있는 한 가지 정렬 방법은 선택 정렬입니다. 선택 정렬 활동 방법에 대해서 살펴봅시다. 우선 가장 가벼운 것을 찾아 한쪽에 둡니다. 다음으로 남아 있는 것들 중 가장 가벼운 것을 찾아 한쪽으로 옮겨 둡니다. 모든 상자의 무게를 재서 모두 옮겨질 때까지 이 과정을 반복합니다.



이 활동에서 여러분이 얼마나 많은 비교를 했는지 세어봅시다.

심화 활동: 8 개를 정렬할 때 얼마나 많은 비교를 했는지 수학적으로 계산하는 방법을 말해봅시다. 그렇다면 9 개일 때는 어떻게 해야할까요? 20 개 일때는 어떨까요?

워크시트 활동: 분할하고 획득하기

퀵 정렬

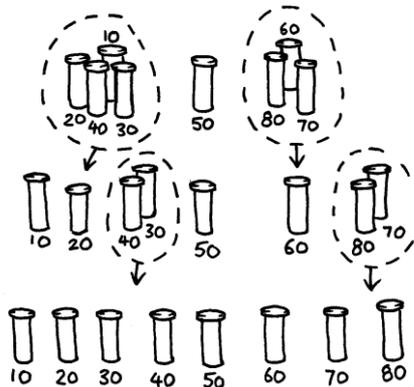
퀵 정렬은 선택 정렬보다 훨씬 빠릅니다. 특히 비교할 목록이 많을 경우 더더욱 빠릅니다. 실제로 퀵 정렬은 알려져 있는 정렬방법 중에 가장 좋은 방법의 하나입니다. 다음은 퀵 정렬의 절차를 보여줍니다.

임의로 하나의 상자를 선택한 후 그것을 균형 저울의 한쪽에 놓습니다.

이제 남아있는 각 상자들을 하나씩 비교합니다. 선택된 상자를 가운데에 두고 더 가벼운 것들은 왼쪽에, 더 무거운 것들은 오른쪽에 둡니다. 이때 선택 상자의 좌우에 상자들이 똑같이 나뉘지 않고 한쪽에 더 많은 상자가 있을 수도 있습니다.

한 그룹을 선택하고 이 과정을 다시 한 번 반복합니다. 다른 그룹에 대해서도 동일한 작업을 수행합니다. 여러분이 선택하여 알고있는 상자를 가운데에 두는 것을 유지하며 과정을 반복합니다.

모든 그룹에서 한 개체만 남을 때까지 나머지 그룹도 이 과정을 계속 반복해서 수행합니다. 모든 그룹이 단일한 상자로 분리되고 나면, 그 개체는 가장 가벼운 것부터 가장 무거운 것 순서로 정렬되어 있을 것입니다.



이 과정에서는 몇번의 비교 횟수로 진행되었나요?

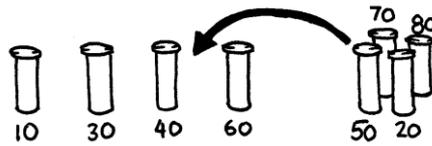
활동을 시작하자마자 가장 가벼운 것 혹은 가장 무거운 것을 선택하는 일이 일어나지 않는 한, 퀵 정렬은 선택 정렬보다 훨씬 더 효율적이라는 것을 알 수 있습니다. 여러분이 전체 상자 중에서 운이 좋게도 가운데에 속한 상자의 무게를 선택한 경우, 선택 정렬은 28 번의 비교를 해야하는 것에 비해 퀵 정렬은 14 번의 비교만 하면 됩니다. 퀵 정렬은 선택 정렬보다 더 나으면 나았지, 더 비효율적인 일은 절대 없습니다!

심화 활동: 만약 퀵정렬이 우연히도 항상 가장 가벼운 개체를 선택하게 된다면, 얼마나 많은 비교를 하게 될까요?

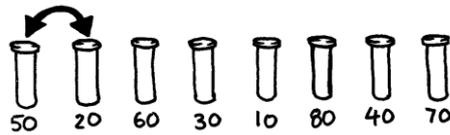
한 단계 더!

이 외에도 정렬을 위한 또 다른 방법들이 있습니다. 다음과 같은 방법들을 사용하여 무게를 정렬해 보세요.

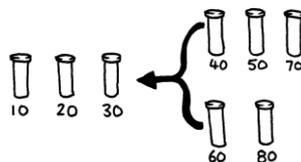
삽입 정렬은 정렬되지 않은 그룹에서 각각의 개체를 하나씩 다른 곳으로 옮기는 과정에서 순서에 맞는 개체들의 사이에 삽입하는 방법을 사용합니다. (아래 그림 참조). 각각의 삽입을 통해 정렬되지 않은 그룹은 축소되고 정렬된 목록은 늘어나는데, 이러한 과정을 통해 결국 전체 목록이 정렬됩니다. 카드를 전문적으로 하는 겜블러들은 종종 손으로 이 방법을 사용하여 카드를 순서대로 정렬을 합니다.



버블 정렬은 정렬되지 않은 개체들 중 이웃하고 있는 두개 쌍의 개체씩 비교하여 작은 수를 앞으로 보내고 큰수를 뒤로 보내는 교체 과정을 반복합니다. 전체 목록을 두쌍씩 비교하는 동안 일어나는 교체의 과정이 더 이상 일어나지 않을 때, 그 목록은 정렬이 됩니다. 이 방법은 매우 효율적인 것은 아니지만 일부 사람들은 다른 정렬 방법보다 이 방법을 더 쉽게 이해합니다.



병합정렬은 전체 목록을 정렬하기 위해 분해하기와 획득하기 방법을 사용합니다. 우선 전체 목록을 임의로 절반을 잘라 두개의 목록으로 분할합니다. 홀수개의 목록이라면 양쪽 모두 같은 수의 목록으로 분할됩니다. 각각의 절반크기로 나뉜 목록을 앞서 설명한 정렬방법으로 정렬한 후 두개의 목록을 다시 병합합니다. 두 개의 정렬된 목록을 병합하는 것은 쉬운 일입니다. —두 목록에서 가장 앞에 있는 두 개의 항목들을 비교하여 그 중 작은 것을 선택하여 다른 곳으로 이동시키는 일을 반복합니다. 아래의 그림을 보면, 2 개의 목록에 가장 앞에 있는 40 그램과 60 그램을 비교하여 무게가 작은 40 그램을 정렬할 곳으로 이동시키는 것입니다. 더 작은 목록은 어떻게 정렬하면 될까요? 간단합니다—병합정렬을 사용하면 됩니다. 결국에는 모든 목록들이 개개의 항목으로 줄어들 것입니다. 그러니 언제까지 반복 해야할지 걱정할 필요가 없습니다.



컴퓨터과학 이야기

정보는 정렬된 목록에서 찾는 것이 훨씬 더 쉽고 빠릅니다. 전화번호부, 사전, 책, 주소록의 색인은 모두 알파벳 정렬을 사용하고 있는데, 이러한 것들이 이렇게 정렬되어 있지 않다면 우리의 생활은 훨씬 어려워질 것입니다. 비용을 적은 목록과 같은 여러 개의 목록이 순서대로 정렬됨으로써 가장 많거나 가장 적은 경우들을 쉽게 알아 볼 수 있습니다. 왜냐하면 어떤 것이든 순서가 있다면 목록의 시작 부분이나 끝 부분에 있을 테니까요. 또한 중복되는 것도 쉽게 찾을 수 있습니다. 왜냐하면 중복되는 것들은 정렬할 때 함께 합쳐지거나 한부분은 삭제되기 때문이죠.

컴퓨터는 항목들을 순서대로 정렬하는 데 많은 시간을 사용합니다. 그래서 컴퓨터 과학자들은 이 과정을 빠르고 효율적으로 할 수 있는 방법을 찾으려고 노력 합니다. 삽입 정렬, 선택 정렬, 버블 정렬과 같은 느린 방법들은 특별한 상황에서 유용하게 사용되기도 하지만, 많은 가지수를 가진 목록은 더 빠르게 정렬해주는 퀵 정렬이나 병합 정렬 등을 보통 사용합니다. 예를 들어, 100,000 개의 항목에 대해 퀵 정렬은 선택정렬보다 약 2,000 배 빠르고 1,000,000 개의 항목에 대해서는 20,000 배 빠릅니다. 컴퓨터는 종종 만 개 이상의 항목을 처리해야 합니다. 예를 들면, 대부분의 웹사이트들은 수 만명의 고객들을 보유하고 있습니다. 심지어 저렴한 카메라로 찍은 사진 한 장에도 만개 이상의 픽셀을 가지고 있습니다. 앞의 두 알고리즘의 차이는 같은 일을 하는 데 하나는 1 초에 처리하고 다른 하나는 5 시간만에 처리를 한다는 것입니다. 느리게 정렬되는 것은 참을 수 있을지도 모르겠지만, 20,000 배나 많은 전력을 사용하는 것은 매우 비효율적인 일입니다. 환경문제에도 영향을 끼칠 뿐만 아니라 휴대용 장치의 배터리 수명까지 감소시킬 것입니다. 그러니 적절한 알고리즘을 선택하는 것만으로도 보다 심각한 문제를 사전에 예방할 수 있는 방안이 될 수 있습니다.

퀵정렬은 분할과 획득이라는 접근 방식을 사용합니다. 퀵정렬에서는 계속해서 목록을 작은 부분으로 분할한 후 분할된 각 부분에서 같은 과정을 진행합니다. 목록들은 항목을 획득할 만큼 충분히 작아질 때까지 반복적으로 분해됩니다. 퀵정렬은 목록들이 오직 하나의 항목만 남길 때까지 분해하는 것입니다. 분할된 한 항목들을 순서대로 정렬하는 것은 매우 간단합니다! 이것은 매우 복잡해 보이지만, 실제로는 다른 방법보다 훨씬 빠르게 처리가 됩니다. 이것은 문제를 해결하기 위해 원래 처리하려고 했던 알고리즘을 다시 반복하여 사용하는 ‘재귀’라는 컴퓨터과학의 원리를 이용합니다. 이러한 개념이 이상하게 들릴 수 있겠지만 이러한 방법은 수학적으로도, 컴퓨터에서도 잘 작동되는 방법입니다.

해답과 힌트

1. 가장 가벼운 물체를 찾을 수 있는 가장 좋은 방법은 가장 작은 것을 찾을 때까지 목록을 건드리지 않고 개체 하나하나씩을 통과하는 것입니다. 즉, 두 개체를 비교하고 더 가벼운 것을 유지하는 것입니다. 이제 어떠한 개체를 다른 것과 비교한 후 더 가벼운 것을 유지해봅시다. 그리고 모든 개체가 사용될 때까지 그 과정을 반복해봅시다.
2. 균형저울로 무게를 비교해봅시다. 이것은 쉽게 세 개를 비교할 수도 있고 때때로 두 개로도 충분할 것입니다. —만약 비교 연산이 쉽게 되는 경우라면 말입니다. (즉, 만약 A가 B보다 가볍고 B가 C보다 가볍다면 A는 C보다 더 가벼울 것입니다.)

심화 활동:

선택 정렬을 할 때 사용되는 비교의 수를 추가할 수 있는 쉬운 방법은 다음과 같습니다.

두 개체를 비교할 때는 최소 한 번의 비교가 필요하고, 세 개를 비교할 때는 두 번, 네 개를 비교할 때는 세 번의 비교가 필요합니다. 8개의 개체를 선택 정렬을 사용하여 비교할 때는, 첫 번째 개체를 찾기 위해 7번의 비교가, 그 다음 개체를 찾기 위해서는 6번의 비교가, 그 다음 개체를 찾기 위해서는 5번의 비교가 진행되어야 합니다. 아래의 식처럼 말이죠.

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ 번의 비교}$$

n 개의 개체를 정렬하기 위해서는 $1 + 2 + 3 + 4 + \dots + n - 1$ 번의 비교가 필요합니다.

우리가 이 수들을 재편성할 경우 새로 수를 추가하는 것은 쉽습니다.

예컨대, 수를 추가하기 위해서 $1 + 2 + 3 + \dots + 20$ 을 다음과 같이 재편성할 수 있습니다.

$$(1 + 20) + (2 + 19) + (3 + 18) + (4 + 17) + (5 + 16) +$$

$$(6 + 15) + (7 + 14) + (8 + 13) + (9 + 12) + (10 + 11)$$

$$= 21 \times 10$$

$$= 210$$

일반적으로 합은 $1 + 2 + 3 + 4 \cdots + n - 1 = n(n - 1)/2$ 입니다.

활동 8

똑딱 시계—정렬 네트워크

활동 개요

비록 컴퓨터가 빠를지라도 여러가지 문제를 해결하는 데 있어 한계가 있습니다. 문제 해결을 단축하는 방법 중 하나는 여러 가지 문제를 해결하기 위해 여러 대의 컴퓨터를 사용하는 것입니다. 이 활동에서는 동시에 여러 정렬 비교를 시행하는 정렬 네트워크에 대해 알아보겠습니다.

교과 연계

- ✓ 수학: 수 - 수 탐색하기: ~보다 큰, ~보다 작은

필수기능

- ✓ 비교하기
- ✓ 순서정하기
- ✓ 알고리즘 개발하기
- ✓ 협력적으로 문제 해결하기

대상 연령

- ✓ 7 살 이상

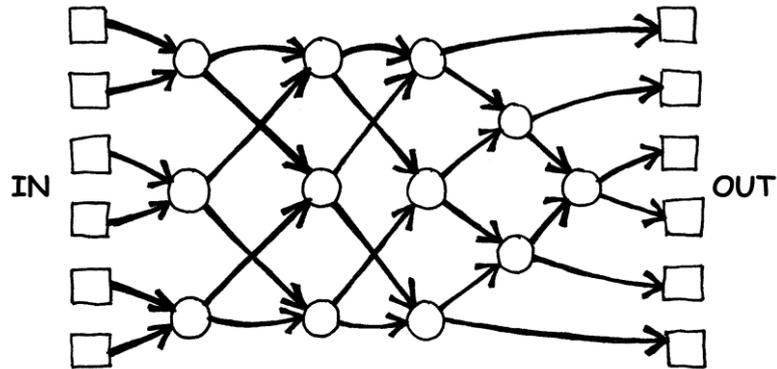
수업 자료

이번 활동은 야외 모둠 활동입니다.

- ✓ 분필
- ✓ 6 개의 카드 세트(2 세트)
정렬 네트워크 학습자료(83 쪽)를 복사 해서 오려 쓰세요.
- ✓ 스톱워치

정렬 네트워크

활동에 앞서 운동장이나 체육관에서 끈 또는 분필을 이용하여 아래의 네트워크를 구현합니다.

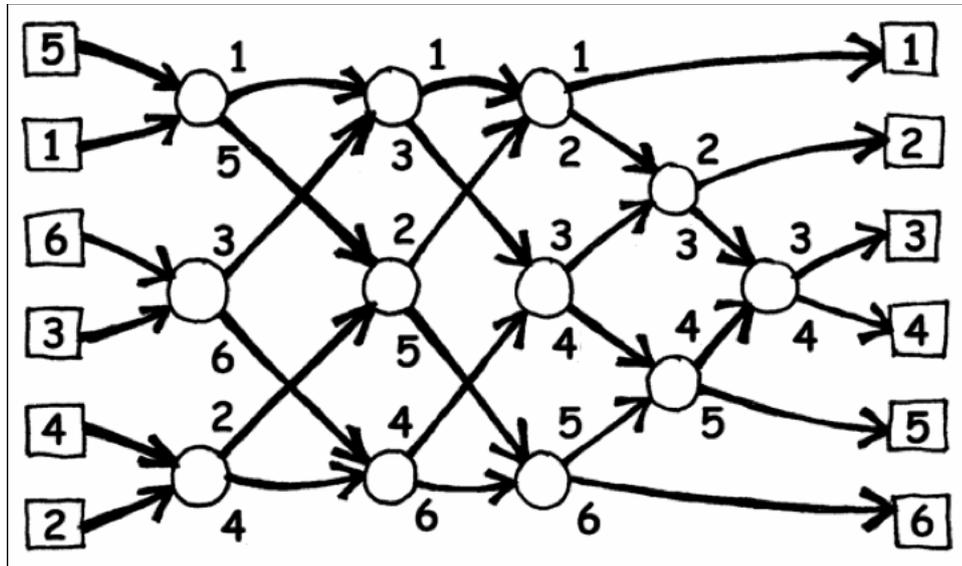


활동 안내

이 활동은 어떻게 컴퓨터가 정렬 네트워크를 사용하여 임의의 수들을 정렬하는지 보여줍니다.

1. 여섯명이 한 그룹을 만드세요. 그리고 한 번에 한 그룹만이 이 네트워크를 사용합니다.
2. 각 팀의 구성원들은 숫자가 쓰여진 카드를 가져갑니다.
3. 구성원들은 네트워크 왼쪽에 있는 네모 칸에 서 있습니다. 여러분이 가지고 있는 숫자들은 뒤죽박죽 순서가 될 것입니다.
4. 표시된 선을 따라 이동한 후 원에 도착했을 때 여러분은 다른 곳에서 오는 누군가를 반드시 기다려야 합니다.
5. 다른 멤버가 여러분의 원에 도착했을 때 두명의 카드에 적힌 숫자를 비교해봅니다. 더 작은 수의 카드를 가진 사람이 왼쪽에 있는 선으로 이동합니다. 만약 여러분이 더 높은 수를 가지고 있다면 오른쪽 선으로 이동합니다.
6. 네트워크의 오른쪽에 모든 멤버가 도착했을 때, 여러분이 든 카드는 크기 순으로 정렬이 되었나요?

만약 한 명에서 오류가 생겼다면, 그 학생은 다시 시작해야 합니다. 작은 값은 왼쪽으로 가고 큰 값은 오른쪽으로 가는 네트워크 마디(선)와 노드(원)의 작동을 이해했는지 확인해봅시다. 다음은 그 예입니다.



학습자료: 정렬 네트워크

1

2

3

4

5

6

156

221

289

314

422

499

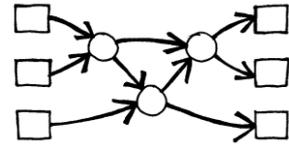
변형

1. 학생들이 활동에 익숙해지면, 스톱워치를 이용하여 각 팀이 네트워크를 통과하는 데 얼마나 오랜 시간을 걸렸는지 재어봅시다.
2. 더 큰 수가 적힌 카드를 사용해봅시다. (예를 들어 세 자리 수가 적힌 카드를 만듭니다.)
3. 비교하는 데 노력이 좀 더 드는 훨씬 더 큰 수로 구성해보거나, 길이가 다른 끈을 이용하거나, 단어를 사용하여 알파벳 순으로 비교하는 활동을 해봅시다.
4. 이 활동은 다른 과목 수업 시작 전에 연습게임으로도 활용될 수 있습니다. 예를 들어 음악 수업에서 카드에 그려진 음들을 비교해 본 다음 음들을 낮은 것부터 높은 것으로, 혹은 짧은 것부터 긴 것으로 정렬해볼 수 있습니다.

추가 활동

1. 만약 더 작은 것이 왼쪽 대신 오른쪽에 가게 된다면 어떻게 될까요? (반대의 순서로 수들이 정렬될 것입니다.)

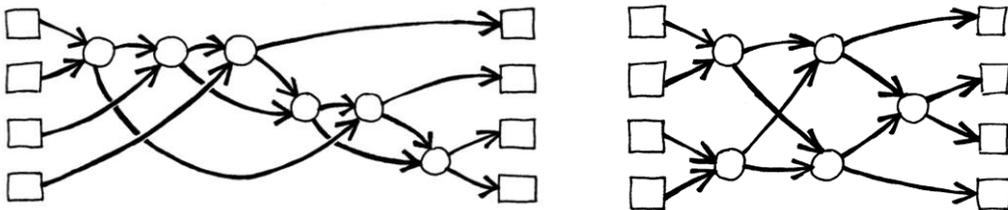
네트워크가 역방향으로 사용된다면 바르게 작동할까요? (제대로 작동하지 않을 것입니다. 여기서 학생들에게 잘못된 순서로 나오게 되는 입력의 예를 찾도록 안내합니다.)



2. 더 작거나 더 큰 네트워크를 디자인해봅시다. 예를 들어 오른쪽 그림은 세 수를 정렬하는 네트워크입니다.

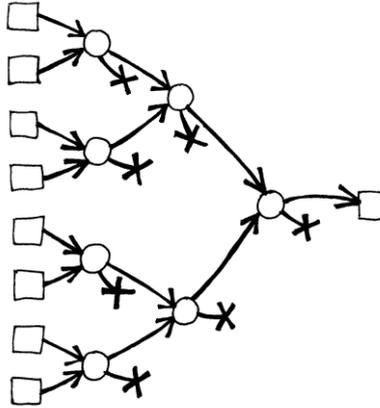
학생들은 이렇게 여러 개의 카드를 갖는 네트워크를 그들 스스로 구성할 수 있도록 노력해야 합니다.

3. 아래의 그림은 4 가지의 입력을 정렬하는 두 가지 형태의 네트워크입니다. 무엇이 더 빠를까요? (두번째가 더 빠릅니다. 첫번째는 모든 비교들이 순차적으로 이루어져 비교할 동안 다른 친구들이 기다려야 하는 반면, 두번째는 동시에 몇몇의 비교가 진행됩니다. 첫번째 네트워크는 직렬처리의 한 예이고, 두번째는 병렬처리의 한 예로서 더 빠르게 작동되는 네트워크입니다.)



4. 더 많은 비교를 하는 정렬 네트워크를 만들어 봅시다.
5. 네트워크는 입력 값 중 가장 큰 값이나 가장 작은 값을 찾는 데도 사용될 수 있습니다. 예를 들어, 아래의 그림은 8 개의 입력값으로 구성된 네트워크입니다. 마지막의

출력값은 입력된 값 중에서 가장 작은 값을 나타내게 될 것입니다. (나머지 다른 값들은 네트워크에서 비교한 막다른 끝부분에 하나씩 남아있을 것입니다.)



- 일상의 어떤 처리 과정들이 병렬을 사용했을 때 속도가 더 붙게 될까요 아니면 더 느려질까요? 예를 들어, 요리를 할 때 한가지 음식을 하는 것과 여러가지 음식을 동시에 하는 것 중 어떤 것이 더 빠르게 식탁을 차릴 수 있을까요? 어떤 일들을 하려고 할때 과연 많은 사람을 고용할수록 빠르게 완료될까요? 어떤 일들이 그 반대의 경우가 될까요?

컴퓨터 과학 이야기

우리는 컴퓨터를 더욱 더 많이 사용할수록 각각의 컴퓨터가 더 빠르게 정보를 처리하기를 원합니다.

컴퓨터의 속도를 증가시킬 수 있는 방법 중의 하나는 좀 더 적은 계산 단계를 사용하는 프로그램을 쓰는 것입니다. (활동 6 과 7 에서 본 것과 같이 말이죠.)

문제를 빠르게 해결하는 또 다른 방법은 동시에 여러 대의 컴퓨터를 사용하여 여러가지 일을 분담하여 처리하는 것입니다. 예를 들어 여섯 개의 정렬 네트워크는 총 12 번의 비교를 필요로 할지라도 세 번의 비교까지는 동시에 수행합니다. 즉 5 번의 비교를 하는 정도의 시간이 필요할 것입니다. 이러한 병렬 네트워크 정렬은 한 번에 하나의 비교만 수행하는 시스템보다 두배는 빠르게 목록을 정렬해 줍니다.

하지만 병렬 계산 방법을 사용한다고 해서 모든 작업이 더 빠르게 진행되는 것은 아닙니다. 다음의 예를 생각해봅시다. 10m 길이의 도랑을 한 사람이 파는 것보다는 1m 길이의 도랑을 열 명의 사람이 각각 파는 것이 훨씬 빠릅니다. 하지만 이러한 전략은 10m 깊이의 도랑을 아래로 파들어가는 데에는 적용할 수 없습니다. 2m의 깊이에 도달하기 위해서는 반드시 1m 를 판 뒤에 가능하기 때문입니다.

컴퓨터 과학자들은 병렬로 작동하는 컴퓨터에 의해 문제를 해결 할 때 더 빨리 문제를 해결할 수 있는 좋은 방법을 찾기 위해 여전히 노력 중입니다.

활동 9

진흙 도시—최소 확장 트리

활동 개요

우리가 사는 사회는 많은 네트워크들로 연결되어 있습니다. 전화 네트워크, 상하수도과 전기 공급 네트워크, 컴퓨터 네트워크, 도로 네트워크처럼 말이죠. 우리는 특정한 네트워크에 대해 도로, 케이블 혹은 무선 링크를 배치할 수 있는 장소를 선택할 수 있습니다. 우리는 네트워크 내에서 개체들을 효율적으로 연결할 수 있는 방법을 생각할 필요가 있습니다.

교과 연계

- ✓ 수학: 기하학-모양과 공간 탐색
- ✓ 사회: 지도에서 최단 경로 찾기

대상 연령

- ✓ 9 살 이상

필수 기능

- ✓ 문제 해결력

수업 자료

학생

- ✓ 워크숍 활동: 진흙 도시 문제 (89 쪽)
- ✓ 빗줄이나 사각형 모양의 판지 (한 학생 당 대략 40 개)

진흙 도시

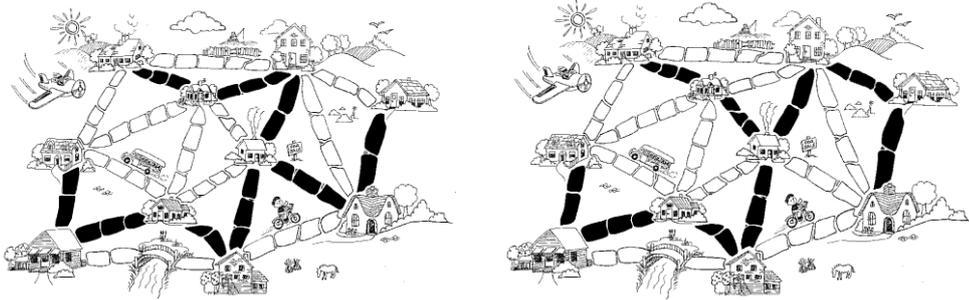
소개

이 활동에서는 실제 생활의 문제(예를 들어 여러 주택에 전기선을 이을지나 택배 문제 등)에 대한 가장 좋은 해결책을 찾기 위해 어떻게 컴퓨터를 이용하는지 알아봅니다. 학생들은 진흙 도시 문제에 대해 설명하고 있는 89 쪽 워크시트를 활용합니다.

생각해보기

학생들이 찾아낸 해결책을 공유해봅시다. 학생들은 어떠한 전략을 사용하였나요?

가장 좋은 해결책을 찾는 전략 중 한가지는 연결선이 없는 빈지도로 시작하여 모든 집이 연결될 때까지 점차 밧줄을 추가하는 것입니다. 이때 이미 연결되어 있는 주택들은 연결하지 않으면서 실의 길이를 증가시켜 길을 추가해나갑니다. 만약 같은 길이의 경로가 추가되는 순서를 바꾸어본다면 다른 해법을 찾을 수도 있게 됩니다. 두 가지의 가능한 해법은 다음과 같습니다.



또 다른 전략은 모두 포장된 도로로 시작하여 필요없는 경로를 지워나가는 것입니다. 하지만 이러한 전략은 훨씬 더 많은 노력을 필요로 합니다.

여러분은 여러분의 일상 생활 중 어디에서 네트워크를 찾을 수 있나요?

컴퓨터 과학자들은 이러한 네트워크들의 구조를 ‘그래프’라고 부릅니다. 실제의 네트워크는 시작과 끝이 연결된 그래프로 표현될 수 있습니다. 예를 들어 도시 지역들 사이의 도로를 건설하거나 세계의 항공 노선을 설계할 때 그래프를 활용할 수 있습니다.

그래프를 활용할 수 있는 다른 알고리즘들이 많이 있습니다. 예를 들어 두 지점 간의 가장 짧은 거리를 구할 때나 모든 지점을 방문하는 가장 짧은 경로를 찾았을 때 그래프를 활용할 수 있습니다.

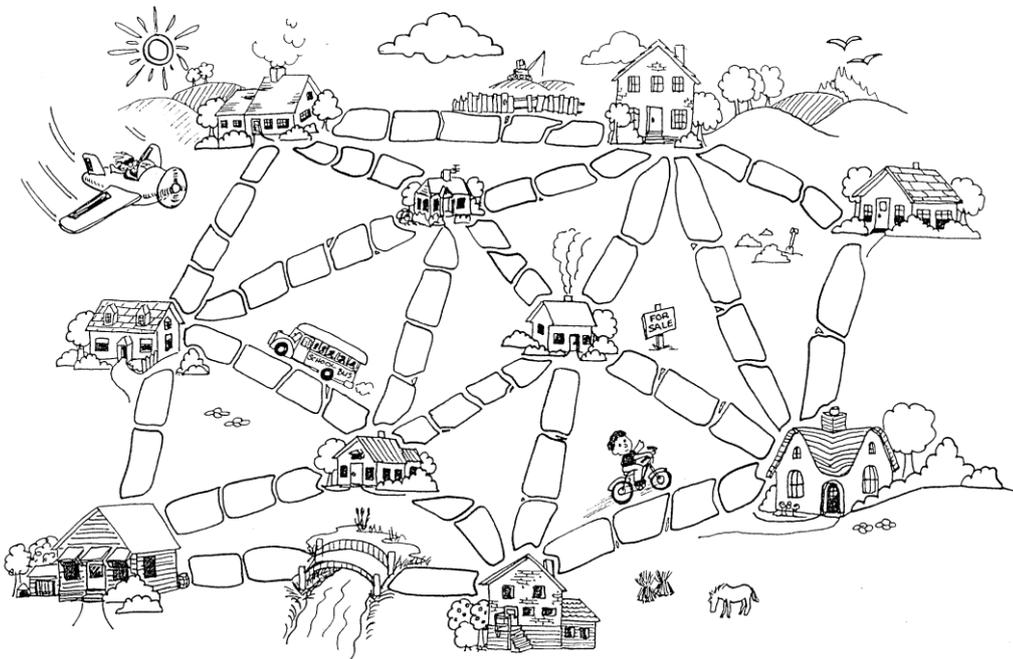
워크시트 활동: 진흙 마을 문제

옛날에 도로가 없는 마을이 있었습니다. 비바람이 온 후에는 그 마을을 돌아다니는 것이 어려웠습니다. 땅이 진흙으로 변했기 때문이죠. 자동차들은 진흙에 박혀버렸고 사람들의 장화는 더러워지기 일쑤였습니다. 그 마을의 촌장은 일부 도로들을 포장해야겠다고 결심하였지만 필요 이상으로 많은 돈을 사용하고 싶지는 않았습니다. 왜냐하면 도시에 수영장도 지어야 했기 때문이죠. 이러한 이유로 촌장은 두 개의 조건을 지정했습니다.

1. 모두가 그들의 집에서 다른 사람의 집으로 이동할 때 포장된 도로만을 이용할 수 있도록 길이 충분히 포장되어야 한다.
2. 포장에 드는 비용은 최대한 적게 들어야 한다.

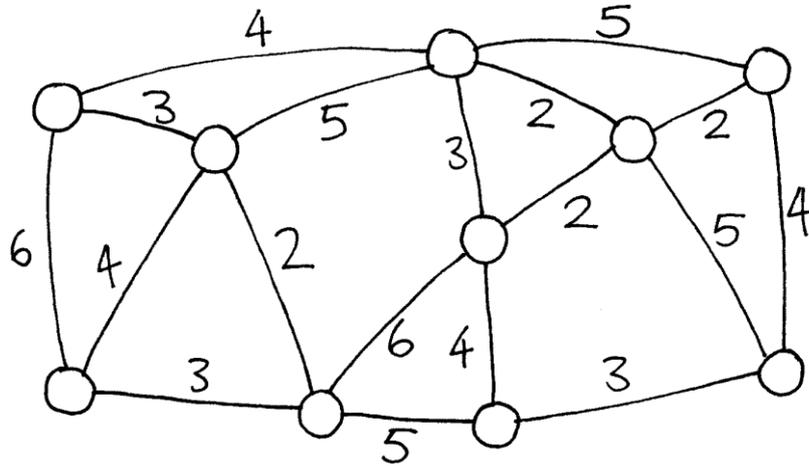
아래는 마을 전경을 나타낸 것입니다. 각각의 집 사이의 포장된 돌들의 수는 그 길을 포장할 때 드는 비용을 표현한 것입니다. 모든 집들을 연결할 수 있는 가장 좋은 경로를 찾아봅시다. 이 때 가능한 가장 적은 밧줄 혹은 포장 돌을 사용하여야 합니다.

문제를 해결하기 위해 어떠한 전략을 사용하였나요?



한 단계 더!

마을의 길을 나타내는 간단한 표현 방법은 다음과 같습니다



집은 원으로 표현되고 진흙 도로는 선으로 표현되며 도로의 길이는 선 옆에 쓰여진 수로 표현됩니다.

컴퓨터 과학자들과 수학자들은 실세계의 문제를 종종 이러한 종류의 그림을 사용하여 문제를 표현합니다. 그들은 이것을 *그래프*라고 부릅니다. 사실 '그래프'는 때때로 통계에서 수치데이터를 보여주기 위한 차트로서 사용되기 때문에 처음에는 혼동될 수 있습니다. 예를 들어 막대그래프가 쓰이죠. 그러나 컴퓨터 과학자들이 사용하는 그래프는 이것과 용도가 다릅니다. 그래프 상의 길이가 실제 범위를 나타낼 필요가 없기 때문입니다.

여러분만의 진흙 마을 문제를 만들어보고 친구들과 함께 해결해봅시다.

가장 좋은 해결책을 위해 얼마나 많은 도로들 혹은 연결 경로들이 필요한지 나타내는 방법을 찾아내었나요? 마을에 얼마나 많은 집들이 있는지에 따라 결과가 달라지나요?

컴퓨터 과학 이야기

전기, 가스, 물 등의 생활 기반 시설을 새로운 마을에 설치하기 위해 여러분이 설계하고 있다고 생각해봅시다. 전기선이나 수도관 등으로 이루어진 네트워크는 모든 집들과 관련 회사를 연결해주기 위해 꼭 필요합니다. 모든 집들은 어떠한 지점에서든 연결될 필요가 있지만 기반 시설이 집으로 이동되는 경로는 매우 중요합니다. 효율적인 설계를 고려하지 않고 단지 연결만 한다면 길든 짧은 어떠한 경로든 존재합니다.

이때 총 길이가 최소인 네트워크를 설계하는 일을 *최소 신장 트리* 문제라고 부릅니다.

최소 신장 트리는 가스나 전력 네트워크에서만 유용한 것이 아닙니다. 컴퓨터 네트워크, 전화 네트워크, 기름 파이프라인, 공항 경로 등에서의 문제를 해결할 때도 도움이 됩니다. 또한 사람들이 여행을 위한 가장 최적의 경로를 결정할 때 얼마의 비용이 드는지의 문제뿐만 아니라 얼마나 그 여행이 편리한지를 고려 합니다. 단지 저렴하다는 이유로 다른 나라로 돌아가는 항공편을 통해 오랜 시간에 걸쳐 여행하길 원하는 사람은 없습니다. 진흙 마을 알고리즘은 이러한 네트워크들에는 그리 많이 사용되지 않습니다. 왜냐하면 그것은 단순히 도로의 길이나 비행 경로의 총길이를 최소화 시키는 것일 뿐이니까요.

최소 신장 트리는 그래프에 대한 다른 문제를 해결하는 단계 중 하나로서도 유용하게 사용됩니다. 네트워크 상의 모든 지점을 가장 짧은 경로로 방문할 방법을 찾아보는 '세일즈맨 문제'가 그 일례입니다.

최소 신장 트리 문제를 해결하는 효율적인 알고리즘이 있습니다. 최적의 해결책을 가져올 단순한 방법은 아무런 연결이 없는 상태에서 시작한 후 크기가 점차 늘어날 때 각 노드에서 짧은 연결선을 추가하는 것입니다. 이 때는 이전에 연결되지 않은 네트워크들의 일부만 연결에 추가해주면 됩니다. 이것은 J.B. Kruskal 이 1965 년에 출판한 저서 이후에 크루스 칼 알고리즘이라고 명명됩니다.

'세일즈맨 문제'를 포함한 많은 그래프상의 문제들에 대해 컴퓨터 과학자들은 충분히 빠른 방법을 찾기 위한 가장 좋은 해결책을 아직 찾지 못하였습니다.

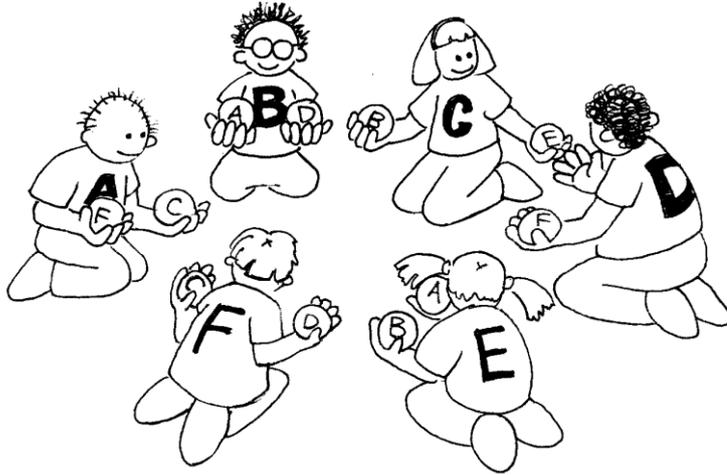
해답과 힌트

한 단계 더! (89 쪽)

n 개의 주택이 있을 때 얼마나 많은 도로와 연결이 필요할까요? 최적의 해결책은 정확히 $n-1$ 개의 연결만 있으면 충분하다고 밝혀졌습니다. 그리고 하나를 더 추가하는 것은 주택 간의 불필요한 경로를 만들어 내는 것이 됩니다.

활동 10

과일 놀이—네트워크의 경로찾기와 교착상태



활동 개요

여러분은 하나의 자원을 사용하는 사람이 많은 경우에(예를 들어, 좁은 교차로, 도로를 사용하는 자동차나 인터넷을 통해 얻은 메시지), “교착상태”가 될 가능성이 있다고 말합니다. 협력 활동은 이런 현상을 방지하기 위해 필요합니다.

교과 연계

- ✓ 수학: 논리적 사고력과 추론능력 기르기

필수 기능

- ✓ 협력 문제 해결 능력
- ✓ 논리적 추론 능력

대상 연령

- ✓ 9살 이상

수업자료

학생:

- ✓ 같은 문자로 표시되는 두 개의 과일이나 테니스 공, 또는 종이컵
- ✓ 과일들의 철자를 나타내는 이름표나 스티커, 또는 색깔 모자, 배지 또는 과일의 첫글자와 일치하는 문자

과일놀이

소개

이 놀이는 협력을 통해 문제를 해결하는 놀이입니다. 놀이의 목표는 각자의 등 뒤에 적혀있는 알파벳과 같은 알파벳이 적혀있는 과일을 가지는 것입니다.

1. 5명 이상의 학생들이 원으로 둘러 앉습니다.
2. 학생들은 알파벳이 새겨진 이름표나 스티커를 받고 색깔을 정합니다(아마 그들의 옷이나 모자의 색깔로 정하면 좋습니다). 알파벳이 사용될 경우, 학생의 알파벳과 과일의 알파벳이 일치하는지 확인하기 위해 한 명의 학생만 하나의 과일만 가지고 나머지 학생들은 알파벳이 적혀 있는 두 개의 과일을 가집니다. 예를 들면, 하나의 과일만 가지고 있는 학생을 제외하고, 노란색 모자를 쓴 학생은 두 개의 바나나를 가질 수도 있고, 초록색 모자를 쓴 학생은 두 개의 초록색 사과를 가질 수 있습니다.
3. 원을 만들어 앉아 있는 학생들에게 무작위로 과일을 나누어 줍니다. 한 명의 학생만 하나의 과일을 가지고, 나머지 학생들은 두 개의 과일을 가집니다. (단, 자신이 가지고 있는 과일이 색깔이 같아서는 안 됩니다.)
4. 각 학생은 자신의 알파벳으로 표시된 과일(또는 자신과 같은 색깔의 과일)을 얻을 때까지 과일을 다른 학생에게 전달합니다. 학생들은 2 가지 규칙을 지켜야 합니다.
 - a) 한 손에는 하나의 과일만 가지고 있어야 합니다.
 - b) 바로 옆에 있는 친구의 한 손이 비어 있을 경우에만 과일을 전달할 수 있습니다.(옆의 학생에게 두 개 중 한 개의 과일만 전달할 수 있습니다.)

욕심이 많은 학생의 경우 빨리 자신의 과일을 찾으려 할 것입니다 (그들은 자신의 과일을 얻자마자 그 과일을 소유하고 다른 친구들에게 전달하지 않으려 합니다.) 이렇게 될 경우 놀이의 목표에 도달하지 못할 것입니다. 이기적으로 개인의 과일이나 알파벳을 찾는 것이 과일놀이의 목표가 아니라는 것을 강조해야 할 것입니다. 학생들이 협력하여 자신 뿐만 아니라 다른 친구들도 함께 과일을 모두 찾았을 때 이 놀이의 목표를 달성하게 됩니다.

토론거리

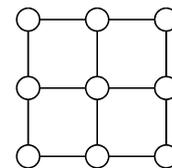
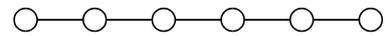
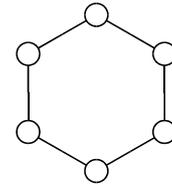
학생들은 문제해결을 위해 어떤 방법들을 사용했습니까?

현실에서 여러분들은 어떤 교착상태를 경험했습니까? (교통체증, 야구경기에서 베이스 돌기, 한 개의 출입구에 다수의 사람들이 들어가는 상황들을 예로 들 수 있습니다.)

심화활동

더 작은 원이나 큰 원을 구성하여 여러 학생들과 활동을 해보십시오.

- 학생들이 새로운 규칙을 만들게 해보십시오.
- 말을 하지 않고 활동하게 해보십시오.
- 한 줄로 앉거나 자신 옆에 두 명 이상의 학생들이 앉게 하는 등 다양한 구성을 해보십시오. 몇 가지 예시들이 옆에 있습니다.



컴퓨터과학 이야기

라우팅과 교착 상태는 도로 시스템, 전화, 컴퓨터 시스템에서 많이 일어나는 문제입니다. 기술자들은 이러한 문제들을 해결하기 위해서, 이러한 문제들을 더 쉽게 해결할 수 있는 네트워크 구축을 위해서 많은 노력을 합니다.

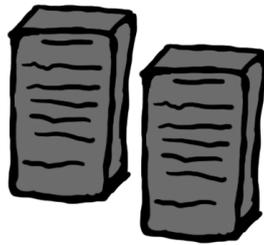
라우팅, 혼잡 및 교착 상태가 네트워크를 방해하는 문제를 일으킬 수 있습니다. 여러분이 자주 접하게되는 출퇴근시간 교통을 생각해보십시오! 때때로 뉴욕의 도로는 혼잡과 교착상태에 빠지게 됩니다. 도로에서 차가 움직이지 않습니다! 간혹 은행과 같은 회사에서 컴퓨터가 "다운"되었을 경우, 원인은 통신 네트워크 교착상태에 있습니다. 라우팅이 쉽고 효율적이고 혼잡을 최소화할 수 있도록 네트워크를 만드는 것은 많은 기술자들이 직면한 어려운 문제입니다.

때때로 두 명 이상의 사용자는 동시에 동일한 데이터를 원합니다. 은행 고객의 잔고와 같은 데이터의 일부분이 갱신되는 경우, 갱신되는 동안 "잠금"을 유지하는 것이 중요합니다. 잠금이 되어있지 않으면, 누군가가 동시에 그 데이터를 업데이트 할 수 있고 은행 고객의 잔고가 잘못 기록될 수 있습니다. 그러나 이 잠금이 다른 항목의 잠금에 의해 방해되는 경우, 교착 상태가 발생할 수 있습니다.

가장 흥미로운 컴퓨터 설계의 발전들 중 하나는 PC 와 같은 프로세서의 수백개 또는 수천개가 하나의 강력한 컴퓨터를 형성하기 위해 (네트워크에서) 결합되는 병렬 컴퓨팅의 등장입니다. 과일놀이에서 발견할 수 있는 것과 같은 같은 문제들은 네트워크상 이 병렬구조 컴퓨터들의 작업에서 지속적으로 매우 빠르게 일어나고 있습니다.

활동 11

무거운 태브릿 벽돌—네트워크 통신 프로토콜



활동개요

컴퓨터는 인터넷에서 메시지를 통해 서로 대화 나눕니다. 그러나, 인터넷은 신뢰성이 떨어지며 가끔 전달되는 메시지들이 손실되기도 합니다. 정보의 특정 비트에는 그들이 전송하는 메시지에 우리가 확인할 수 있는 추가 메시지가 있습니다. 이 정보를 이용해 전달 방법을 약속하는 프로토콜을 구성합니다.

교과연계

- ✓ 수학: 논리적사고력과 추론능력
- ✓ 영어: 의사소통, 듣기

필수기능

- ✓ 협력문제해결능력
- ✓ 논리적 추론능력

대상연령

- ✓ 9 세이상

수업자료

학생:

- ✓ 여러 개의 공백 카드

메신저:

- ✓ 활동 메시지 카드

교사:

- ✓ 시계

무거운 태블릿 벽돌

소개

이 활동에서 학생들은 각각의 의사소통 방법이 어떻게 성공적으로 이루어질 수 있는지를 생각하게 됩니다. 의사소통에서의 규칙과 절차를 살펴보면서, 학생들은 통신 프로토콜 학습에 입문하게 됩니다. 역할놀이 활동을 하면서 학생들은 인터넷 즉, TCP/IP 을 이용한 패킷 교환에서 발견한 것과 유사한 비신뢰 환경에서 동작하는 자신의 프로토콜을 테스트합니다.

준비

1. 먼저 카드들을 준비합니다. (아래) 활동카드를 복사하여 자릅니다.
2. 다음으로 학생이 보내는 몇 가지 메시지를 결정합니다. 그것들이 영어문장이 아닌지 또는 그것들의 구조에 의해 다시 함께 놓여질 수 있는지의 여부가 중요합니다. 때때로 “1LHC255HD2RLLS”와 같은 메시지나 전화번호 등이 적합할 수 있습니다.
3. 여러 개의 빈 카드를 출력합니다. 각각의 카드에는 6 개의 알파벳이나 숫자가 적혀있고, 학생들의 카드 안에 있는 전체 메시지가 맞지 않을 수도 있습니다. 학생 한 명당 대략 30 개 정도의 카드가 필요하며, 이것은 활동시간에 따라 달라질 수 있습니다.

참고: 활동카드의 종류는 3 가지 입니다; 지연, 제공, 미제공. 이 카드들의 비율을 조정하는 것은 학생들의 메시지의 품질을 좌우할 것입니다. 더 많은 "제공"카드는 보다 안정적인 네트워크를 의미합니다. “지연”과 “미제공”카드는 보다 덜 신뢰적인 네트워크를 의미합니다. 이러한 카드들은 컴퓨터 네트워크/통신 채널과 유사합니다.

활동하기

1. 교실을 반으로 나눕니다. 짝을 이룬 학생들이 서로를 볼 수 없고 의사소통할 수 없도록 나누는 것이 중요합니다. 교실의 양측 반대편에 있는 두개의 공간으로 나누고 학생들이 앉을 수 있도록 하면 좋습니다.
2. 한쪽 편에 앉은 짝의 메시지를 그들의 짝꿍 학생에게 전달합니다.
3. 활동카드를 섞은 후 메시지를 선택합니다. 여러분이 메시지가 될 수 있고, 인원이 홀수인 경우에도 메시지가 될 수 있습니다. 규모가 큰 교실일 경우에는 하나 이상의 메시지가 필요할 것입니다.
4. 학생은 자신의 태블릿카드에 작성하고, 그것을 메시저에게 줍니다. 태블릿카드에는 적어도 다른 사람의 이름이 있어야 합니다.

5. 이제 메신저는 가장 위에 있는 카드를 선택하여, 그것을 뒤집고, 그것을 읽고, 테블릿이 수행할 작업을 결정하는데 사용합니다.
6. 각각의 테블릿에 4 단계와 5 단계를 반복합니다.

처음 시작후 5분에서 몇분 후에, 학생들은 하나의 이름이 하나의 프로토콜을 구성하기에 충분하지 않다는 것을 깨닫게 됩니다. 활동을 멈추고 이것에 대해 토론합니다. 첫번째 문제는 무엇인가? 그것은 순서가 있나? 테블릿 번호를 넣어 그 6 개의 슬롯 중 하나를 사용하는 것이 가장 좋은가? 이것은 실제 데이터에 대한 공간이 부족하다는 것을 의미합니다 - 이것은 우리가 지금 사용해야하는 테블릿 갯수의 관점에서 무엇을 의미하는 것일까요?

시간이 더 흐르면, 다른 문제를 발견하게 될 것이고 이에 관해 토론을 하여야 합니다. 일어날 수 있는 문제들은 테블릿에서 누락되는 것, 테블릿이 전송된 것을 모르는 것, 테블릿의 재전송 여부를 모르는 것이 될 수 있습니다. 여러분이 제안할 수 있는 해결방법은 승인한 것을 다시 보내는 것, 다시 보내기 전에 이것들을 다시 받기 위하여 기다리는 것- 이것은 메시지를 수신한 학생도 메시지를 보낼 수 있도록 공백의 테블릿이 필요하다는 것을 의미하고, 그들은 다시 활동을 하기 전에 6 개의 알파벳의 응답이 무엇을 의미하는지에 동의해야 합니다.

이 활동에는 적어도 두 명의 학생이 필요하지만 가능한 많은 학생이 활동에 참여하는 것이 좋습니다. 교실이 클 경우 몇 가지의 메신저를 생각해보십시오. 다시 한 번 교실에서 이것을 의논해 보세요. 메신저가 많을 경우 어떤 일들이 생길까요? 하나의 메신저만 있을 경우에는 무슨 일들이 발생할까요?

지금 이 테블릿을 전달하시오	다음번에 메시지를 전달하시오
지금 이 테블릿을 전달하시오	다음번에 메시지를 전달하시오
지금 이 테블릿을 전달하시오	다음번에 메시지를 전달하시오
지금 이 테블릿을 전달하시오	이 메시지를 전달하지 마시오
지금 이 테블릿을 전달하시오	이 메시지를 전달하지 마시오

<p>받는 곳:</p> <table border="1" data-bbox="256 338 743 434"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>							<p>받는 곳:</p> <table border="1" data-bbox="788 338 1291 434"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>						
<p>받는 곳:</p> <table border="1" data-bbox="256 757 743 853"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>							<p>받는 곳:</p> <table border="1" data-bbox="788 757 1291 853"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>						
<p>받는 곳:</p> <table border="1" data-bbox="256 1176 743 1272"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>							<p>받는 곳:</p> <table border="1" data-bbox="788 1176 1291 1272"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>						
<p>받는 곳:</p> <table border="1" data-bbox="256 1594 743 1691"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>							<p>받는 곳:</p> <table border="1" data-bbox="788 1594 1291 1691"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>보내는 곳:</p>						

무거운 벽돌

고대도시에는 다수의 중요한 통치자가 있습니다. 이 통치자들은 도시의 운영방향을 결정하고 중요한 결정들을 합니다. 그들은 도시 전역에서 각각의 다른 집에 살고 있습니다.

통치자들은 종종 의사소통이 필요해서 메시지들을 보내고 도시 전역의 메시지들을 수신합니다. 통치자들은 그들의 집번호로 구별되며 그들 모두는 작업을 메시지로 전달하는 메신저(배달부)의 그룹에 대한 권한을 갖습니다.

메시지를 보내는 유일한 방법은 벽돌에 메시지를 작성하는 것인데, 그것을 메신저가 그들의 목적지로 운반합니다. 벽돌의 크기는 고정되어 있으며, 단지 6 개의 정보만 기록할 수 있습니다. 정보의 한 조각은 하나의 문자 또는 하나의 숫자가 될 수 있습니다. 메시지들은 종종 몇 개의 테블릿 벽돌로 분할되며, 이 테블릿 벽돌은 매우 무거워서 한번에 하나만 운반할 수 있습니다.

건망증과 게으른 메신저들 때문에 항상 정확하게 메시지를 전달되는지 신뢰할 수 없습니다. 그들은 종종 근무 시간 동안 휴식을 위해 잠적하기도 하고 심지어 도시에서 탈출하려고도 하기 때문입니다.

통치자들은 자신의 명령과 데이터를 확실하게 전송하여 모든 통치자들이 정보를 주고 받는 구체적인 방법을 찾고 있습니다. 이렇게 함으로써 그들은 자신들의 메시지가 전달되었는지 여부를 확인할 수 있어야 합니다. 통치자들은 테블릿 벽돌에 수신할 다른 통치자의 이름을 넣은 뒤 메시지를 작성해야 한다고 우선 결정합니다.

여러분의 그룹에서 여러분의 임무는 통치자가 통신에 사용할 규칙을 개발하는 것입니다.

컴퓨터과학 이야기

인터넷에서 데이터는 통신 패킷 조각으로 나뉩니다. 그러나 이들 패킷이 이동하는 채널은 항상 신뢰할 수 없습니다. 개별 패킷은 때때로 분실되거나 손상 또는 순서가 뒤바뀌는 경우가 있습니다.

무거운 벽돌놀이에서 벽돌은 패킷이며 그 내용은 데이터입니다. 패킷은 데이터 및 헤더 정보를 모두 포함합니다. 헤더 정보의 크기가 데이터를 많이 전송할 수 있는지에 영향을 줍니다. 그래서 패킷은 유한하고 크기가 같게 균형을 잡아야 합니다.

학생들은 이러한 패킷 번호 및 총 패킷과 같은 정보를 박스 데이터의 일부로 교환할 필요가 있는지, 패킷이 응답 확인 패킷인지 아닌지 알아냅니다. 정보 데이터 상자를 얻기 위해서는 더 많은 패킷을 필요로 하게 됩니다.

이렇게 균형이 맞춰진 TCP 와 UDP 같은 인터넷 프로토콜은 데이터 전송을 신뢰할 수 있고 효율적으로 이루어지게 합니다.

이 활동은 "컴퓨팅 과학 인사이드"의 프로젝트 (csi.dcs.gla.ac.uk)를 통해 구체적으로 적용한 사례를 보여 줍니다.

Part III

컴퓨터가 일하도록 명령하기—절차표현

컴퓨터가 일하도록 명령하기

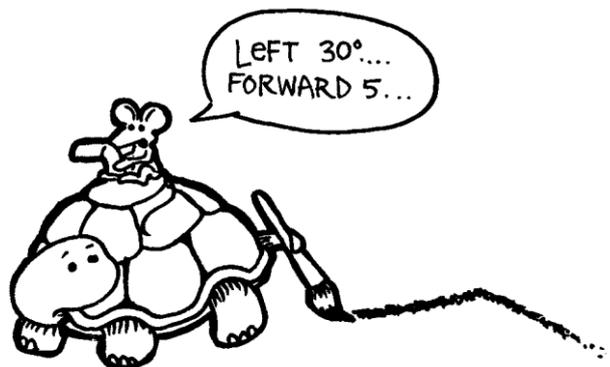
컴퓨터는 명령을 따릅니다 — 매초마다 수백만개의 명령을 수행합니다. 무엇을해야 하는지를 컴퓨터에게 명령하기 위해 컴퓨터에게 올바른 명령을 내려야 합니다. 하지만 이것은 그리 쉬운 일은 아닙니다!

우리가 누군가에게 명령을 받아 그것이 무엇을 무엇을 의미하는지 해석할 때는 우리가 가지고 있는 상식을 사용합니다. 누군가가 "그 문을 통과하라"고 말했을 경우, 그것은 실제로 문을 부수고 통과하는 것을 의미하지 않습니다. 그것은 먼저 손잡이를 돌려 문을 열고 필요한 경우, 출입구를 통과하는 것을 의미합니다. 컴퓨터들은 명령한 것에 따라 각기 다르게 수행합니다. 이것을 이동 로봇에 장착하면 명령을 말 그대로 해석하여 사람을 다치게 하는 위험을 유발할 수도 있기 때문에 이것을 예방하기 위한 명령까지 내려 주어야 할 필요가 있습니다. 문을 통과하려고 하는 것과 같이 아무 생각없이도 그런 명령에 정확하게 지시를 따르도록 다루는 것에 익숙해지는 것은 노력과 시간이 걸립니다.

이 장에서의 두가지 활동은 몇가지 고정된 명령을 사용하여 기계가 문자를 인식하는 활동을 안내합니다.

첫번째 활동은 컴퓨터가 단어, 숫자, 기호 문자열을 인식하는 데 사용하는 "장치"에 대해 알려줍니다. 이러한 장치들을 '유한상태 오토마타'라고합니다.

두 번째 활동은 우리가 컴퓨터와 통신할 수있는 방법을 소개합니다. 훌륭한 프로그래머는 문자 그대로 명령을 사용하여 무엇을 해야하는지를 컴퓨터에게 전달할 수 있습니다. 절차적인 명령들의 집합체가 프로그램입니다. 프로그래머가 명령을 작성하기 위해 고를 수 있는 프로그래밍 언어가 매우 다양하게 많이 있으나, 우리는 컴퓨터없이 사용할 수있는 간단한 언어를 사용합니다.



활동 12

보물섬—유한 상태 오토마타

활동개요

컴퓨터 프로그램들은 문서의 문자나 기호, 심지어 다른 컴퓨터 프로그램의 문자도 처리 가능합니다. 컴퓨터 과학자들은 이 작업을 위해 유한 상태 오토마타를 사용합니다. 유한 상태 오토마타(FSA)는 컴퓨터가 단어나 기호 문자열을 인식할 수 있도록 도와주는 개념입니다. 보물섬 놀이를 통해 이와 같은 활동을 해봅시다.

교과연계

- ✓ 수학: 논리적 사고력과 추론능력의 발달—문자들과 묘사하기 위한 기호들 그리고 계속되는 패턴 이해
- ✓ 사회적 연구
- ✓ 영어

필수기능

- ✓ 간단한 지도 해석능력
- ✓ 패턴 이해능력
- ✓ 논리적 사고력
- ✓ 명령의 이해

대상연령

- ✓ 9 세이상

수업자료

교사:

- ✓ 섬 카드 한세트(명령은 지도를 그리기 위해서 숨겨야합니다!)
섬 카드 복사본: 섬 카드들(114 쪽) 을 자릅니다.
점선을 따라 접고 플로 붙이면 카드의 앞면은 섬의 이름이, 뒷면은 명령이 쓰여져 있습니다.

학생:

- ✓ 워크시트 활동: 보물섬에 가기 위한 여러분의 방법을 찾아보십시오(113 쪽)
 - ✓ 필기 도구
- 학생의 확장활동: 워크시트 활동: 보물섬 (119 쪽)
- ✓ 워크시트 활동: 신기한 동전게임(120 쪽)

보물섬

소개

여러분의 목표는 보물섬을 찾는 것입니다. 흥미진진한 해적선은 섬 사이의 고정된 노선을 따라 항해하며 여행자들에게 이동할 수단을 제공합니다. 각 섬에는 여러분이 여행목적지를 선택할 수 있는 두 개의 항로 A와 B가 있습니다. 여러분은 보물섬의 최적의 경로를 찾아야합니다. 도착한 섬에서는 해적선 A와 해적선 B 중 하나를 요청할 수 있으며 동시에 두개를 탈 수 없습니다. 섬에 있는 사람은 여러분에게 배가 어디로 가지만 설명할 뿐 전체 항로에 대해서는 알려주지 않습니다. 여러분이 가고 있는 경로와 배의 방향을 알기 위해 여러분의 지도를 사용하세요.

모의활동

(참고: 이것은 실제활동과는 다른 지도입니다.)

칠판을 사용하여, 다음과 같이 세 개의 섬을 그립니다.

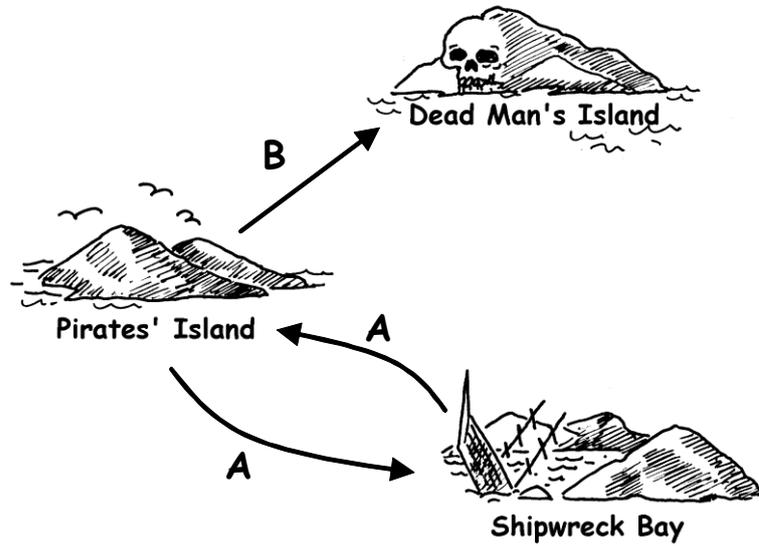


다음 두 페이지에 있는 세 개의 카드를 복사하고, 학생 한명씩 각각의 카드를 가집니다. 이 카드의 경로는 주요활동과는 다릅니다.

해적섬에서 시작하며 난파만에 있는 사람에게 선박 A 를 요청합니다. 그리고 지도에 그 경로를 표시하세요. 난파만에서 다시 선박 A 를 요청하세요. 그러면 다시 되돌아오게 됩니다. 이것을 지도에 표시하세요. 이번에는 선박 B 를 요청합니다. 이것을 지도에

표시하세요. 이 경로는 죽은 사람의 섬으로 가게 되는데 바로 여러분이 머무를 곳입니다.

여러분의 마지막 지도는 아래와 같아야 합니다.

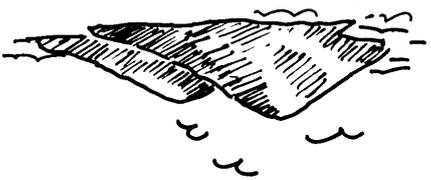


모의활동을 위한 카드


Pirates' Island

A → 
 Shipwreck Bay

B → 
 Dead Man's Island

Pirates' Island


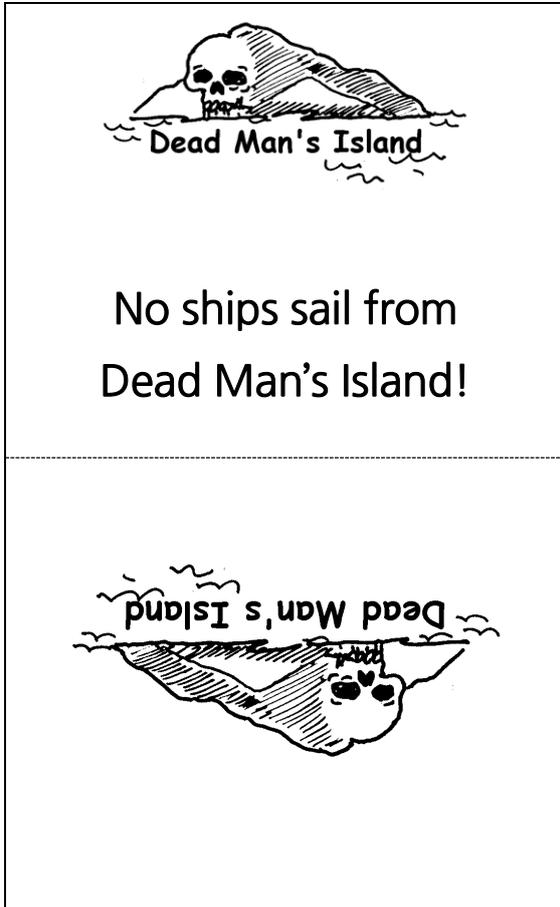

Shipwreck Bay

A → 
 Pirates' Island

B → 
 Dead Man's Island

Shipwreck Bay


모의활동을 위한 카드

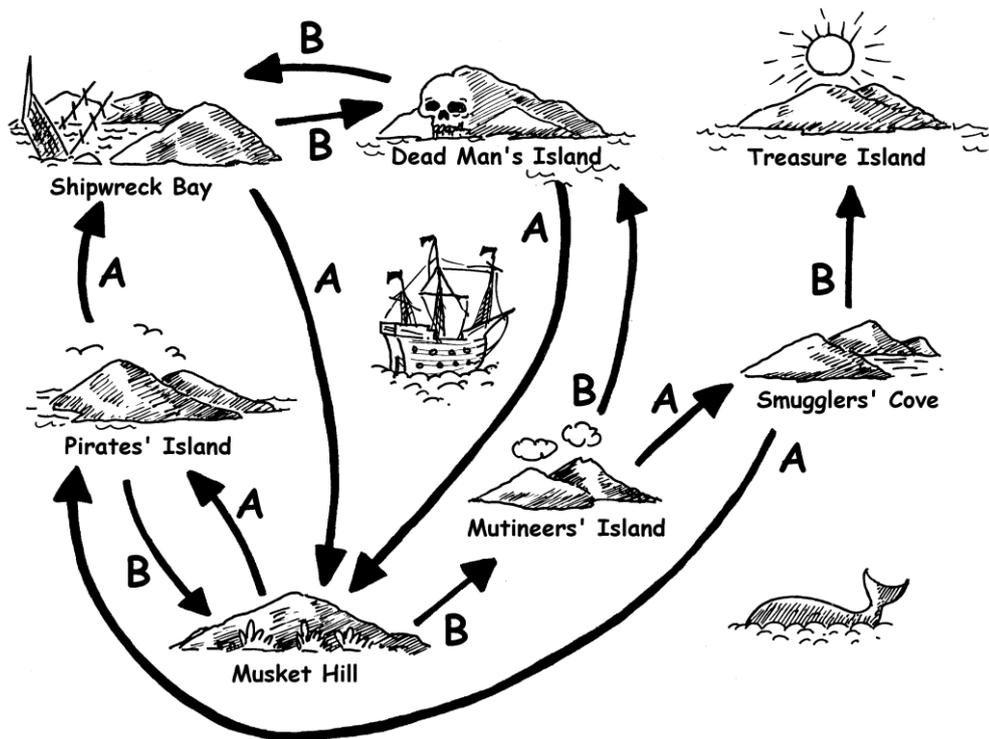


활동

섬 역할을 맡게 될 7 명의 학생을 뽑으세요. 학생들에게 뒷면에 비밀 명령이 적힌 카드를 나누어 줍니다. 넓은 교실이나 운동장 주변에 학생들을 무작위로 배치합니다. 나머지 학생들에게는 섬만 있고 항로가 없는 빈 지도를 주고 해적섬에서 보물섬으로 가는 경로를 탐색하며 표시하도록 합니다. 이 때 경로를 다른 친구들에의 활동에서 경로를 말하는 것을 미리 듣지 못하도록 한 번에 한 명의 학생들만 보내도록 합니다.

빠르게 도착하는 사람에게는 또 다른 경로를 찾아보도록 안내하세요.

완성된 지도는 아래와 같습니다:

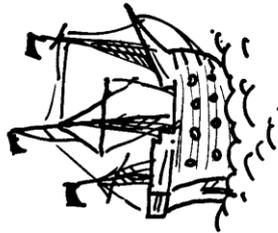
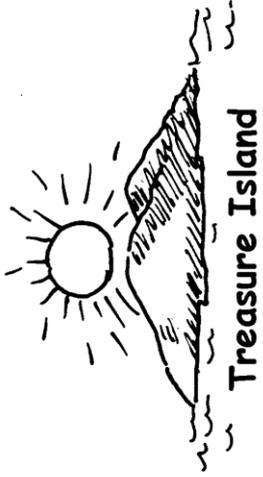


토론 열기

가장 빠른 경로는 어느 것입니까? 가장 느린 경로는 무엇입니까? 몇 가지 경로들은 반복을 포함할 수도 있습니다. 이런 반복된 경로는 어떤 것이 있나요? (예를 들어, BBBABAB and BBBABBABAB 두 개의 경로 모두 보물섬에 도착합니다.)

워크시트 활동:

보물섬을 찾기 위해 여러분의 경로를 찾아보세요.



학습자료: 섬 카드(1/4)



Pirates' Island

A →  Shipwreck Bay

B →  Musket Hill

Pirates' Island




Shipwreck Bay

A →  Musket Hill

B →  Dead Man's Island

Shipwreck Bay



학습자료: 섬 카드 (2/4)



Musket Hill

A →  Pirates' Island

B →  Mutineers' Island



Musket Hill



Dead Man's Island

A →  Musket Hill

B →  Shipwreck Bay



Dead Man's Island

학습자료: 섬 카드 (3/4)



Mutineers' Island



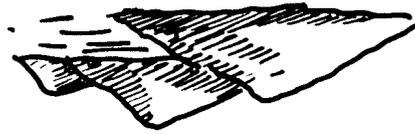
Smugglers' Cove

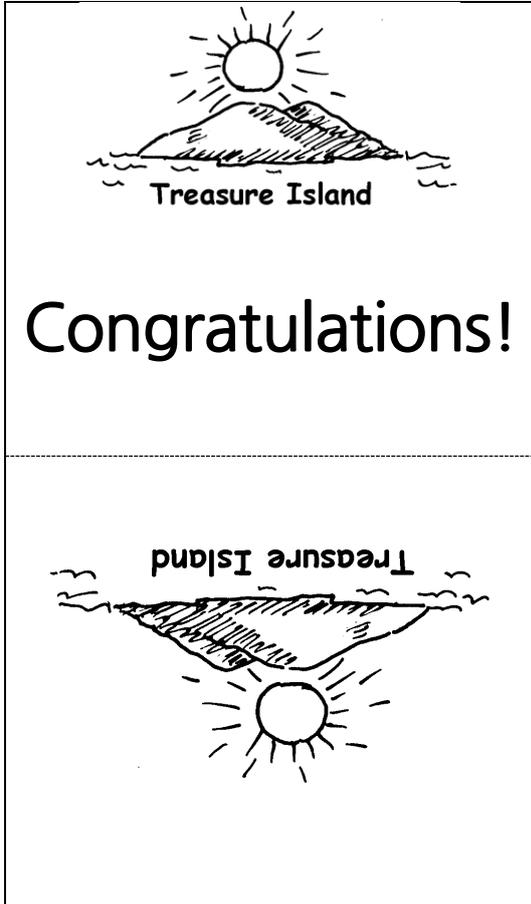


Mutineers' Island



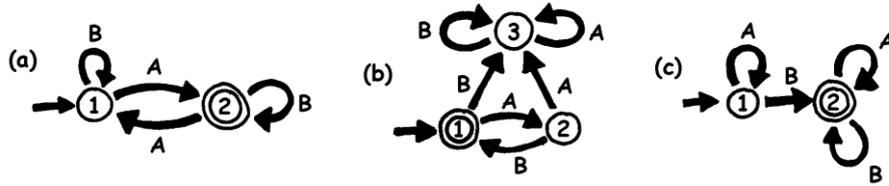
Smugglers' Cove





유한상태 오토마타

앞선 활동의 해적선 지도를 단순하게 그리는 또 다른 방법은 다음과 같습니다.



섬들은 번호가 적혀있는 원들로 표시하고, 보물이 있는 마지막 섬은 두개의 원으로 표시합니다. 마지막 섬에 가기 위한 경로는 무엇입니까?(다음의 예들을 고려하여 탐색하는 것이 좋습니다. "A"는 이중 동그라미로 표시할 수 있나요? "AA"? "ABA"? "AABA"? 일반적인 패턴은 무엇인가?)

정답:

지도(a)는 A의 개수가 홀수인 경우에만 목적지인 2번의 이중 동그라미 섬에 도착하게 됩니다.(예를 들어, AB, BABAA, 또는 AAABABA).

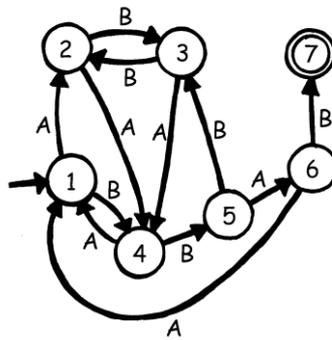
지도(b)는 A와 B가 번갈아 나올 경우에만 2번의 이중 동그라미에 도착합니다.(AB, ABAB, ABABAB, ...).

지도(c)는 최소 하나의 B가 있어야 2번의 이중 동그라미에 도착하게 됩니다.(A로만 이루어진 A, AA, AAA, AAAA, ... 등은 1번섬인 자신에게 항상 되돌아오게 되므로 목적지에 절대 갈 수 없겠죠?).

워크시트 활동: 보물섬

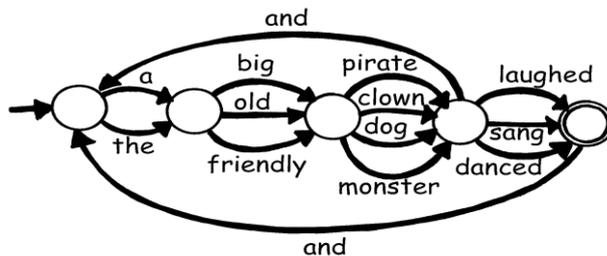
여러분은 보물을 잘 숨겨 놓았나요? 보물을 찾기위해 얼마나 열심히 노력하였나요?
이제는 여러분의 지도를 그릴 차례입니다.

- 여기에 있는 지도는 앞서 활동한 보물섬 지도와 같은 섬과 경로를 갖고 있지만 조금 더 복잡하게 보입니다. 실제 섬과 배가 없이 숫자와 알파벳만으로 그려졌기 때문이죠. 그러나 컴퓨터 과학자들은 섬과 경로를 기호로 상징화하여 컴퓨터가 처리하기 쉽게 단순한 패턴의 경로를 사용합니다.



여러분은 해적선을 타고 여행한 후 빈지도와 섬카드를 만들고 경로를 명확하게 그리기 위하여 이와 같이 간단히 그리고 이동 계획을 세울 것입니다. 여러분의 보물섬에 도달하는 가장 효율적인 경로는 무엇입니까?

- 여러분의 친구들이 이 지도를 보고 잘 따라갈 수 있습니까? 친구들에게 A 와 B 의 순서를 주고 그들이 정확한 섬으로 도달할 수 있는지 보십시오. 당신은 유한 상태 오토마타의 아이디어를 기반으로 다양한 게임과 퍼즐을 만들 수 있습니다.
- 다음은 지도를 통해 임의의 경로를 선택하고 단어를 이용하여 문장을 구성하는 방법입니다. 이것을 이용하여 재미있는 이야기를 만들 수 있습니다. 예를 들면 다음과 같습니다.



The old clown sang and a big dog danced. (늙은 광대는 노래하고 큰 개는 춤쳤다.)

워크시트 활동: 신기한 동전게임

오늘 인터넷에서 로봇이 동전을 던져 앞면 또는 뒷면이 나오는 확률을 맞추는 게임을 다운받았습니다. 처음에는 매우 간단해 보였습니다. 적어도 절반의 승률은 가지고 있다고 생각했기 때문이죠. 잠시 후, 그 게임이 의심스러워지기 시작했습니다. 동전이 뒤집어지는 패턴이 보였기 때문입니다. 게임이 조작되고 있는 듯 보였으나 결코 그렇지 않다고 개발자들이 주장하였습니다! 그래서 우리가 조사하기로 하였습니다. 친구가 게임에서 다음의 결과를 발견하고 그것을 기록하였습니다.

(h = 앞면, t = 뒷면)

```
h h t h h t h h h t t h h h h t t h t t t h h h h h t h h h t t t h h h t t t h h h h h  
t t h t t t t t h t t t h h h t t h h h t h h h h h h h h t t h h h t t t t h h h h  
h t t t t t t t
```

예측가능한 패턴을 찾을 수 있습니까?

동전이 뒤집어지는 순서를 설명하는 간단한 지도가 있습니다. 여러분이 한 번 찾아보십시오. (힌트: 4 개의 섬)

컴퓨터과학 이야기

유한상태 오토마타는 컴퓨터가 문장을 인식하거나 이벤트를 처리하는 방법을 제공하는 컴퓨터 과학입니다.

간단한 예로 어떤 업체에 전화를 걸 때 “이럴 때는 1 번을 누르세요 … 저럴 때는 2 번을 누르세요 … 교화원을 원하시면 3 번을 누르세요”라는 메시지를 받게 됩니다. 번호를 누르게 되면 이 번호는 전화업체의 전화시스템에서 유한상태 오토마타 입력값이 될 수 있습니다. 전화통화는 매우 간단하거나 복잡해질 수도 있습니다. 유한상태 오토마타의 독특한 반복으로 인하여 때때로 처음으로 돌아가 반복을 계속할 수 있습니다. 이것이 발생하게 된다면 시스템의 설계에 문제가 있는 것입니다. 그리고 그런 상황은 전화를 거는 사람을 짜증나게 하겠죠.

또 다른 예로, 은행 기계에서 돈을 찾을 때입니다. ATM 현금 자동인출 기기의 프로그램은 돈을 찾을 수 있도록 그 과정을 안내합니다. 프로그램에 돈을 인출하거나 계좌를 이체할 때 모든 가능한 순서는 유한 상태 오토마타로 표현할 수 있고, 모든 키 조작은 자동 장치에 의해 다른 일을 하도록 명령을 전달합니다. “100 달러를 인출하라” 또는 “입출금 내역을 인쇄하라” 또는 “통장을 꺼내라” 등 오토마타의 자동 장치를 이용하여 컴퓨터에 명령을 내립니다.

일부 컴퓨터 프로그램은 134 쪽과 같은 지도를 사용하여 영어 문장을 처리합니다. 이 프로그램을 이용하여 새로운 문장을 만들 수도 있고, 사용자가 입력한 문장을 인식하는 처리 작업을 하기도 합니다. 1960 년대에, 컴퓨터 과학자들은 사람들과 대화할 수 있는 “엘리자” 라는 유명한 프로그램을 개발하였습니다. 이 프로그램은 정신과 의사들처럼 “당신의 가족에 대해 이야기해보세요”, “계속 하십시오”와 같은 질문과 대답을 하였습니다. 엘리자가 사람들의 답을 이해하는 것은 아니었지만 충분히 그럴 듯하게 대답했고—그것을 사용하는 사람들도 엘리자 프로그램을 사람인듯 속는 듯 했고—몇몇 사람들은 정말로 정신과 의사와 대화하는 것처럼 생각했습니다.

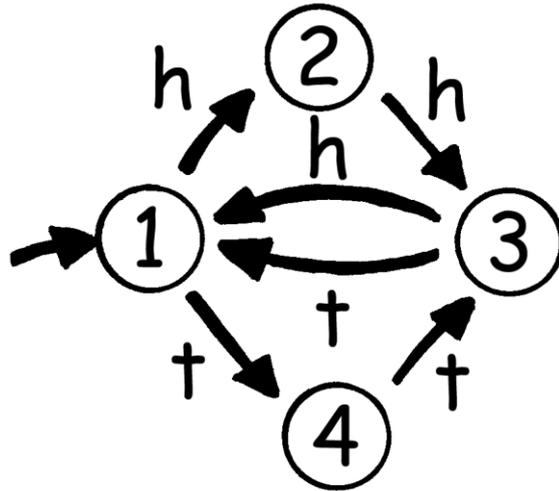
비록 컴퓨터가 자연언어를 잘 이해하는 것은 아니지만, 인공지능은 쉽게 처리할 수 있습니다. 프로그래밍 언어는 중요한 인공지능들의 유형 중 하나입니다. 컴퓨터는 프로그래밍 언어로 작성된 코드를 읽기 위해 유한상태 오토마타를 사용하고 컴퓨터가 직접 실행할 수 있는 컴퓨터의 원시언어로 바꿉니다.



정답과 힌트

신기한 동전게임(135 쪽)

신기한 동전게임은 다음과 같은 동전 뒤집기 지도를 사용합니다:



이것을 따라해보면 두 개의 동전을 던진 것과 세 개의 동전을 던진 것의 결과가 같다는 것을 발견할 수 있습니다.

활동 13

행진하라—프로그래밍 언어

요약

컴퓨터는 일반적으로 제한된 명령어로 만들어진 “언어”를 사용하여 프로그래밍합니다. 가장 아쉬운 점 중 하나는 어이없는 결과가 나오더라도 명령어의 지시에 따른다는 것입니다. 프로그래밍의 이러한 특징에 대하여 알아보겠습니다.

관련 교육 과정

- ✓ 영어 : 대인 관계를 통한 듣기

기술

- ✓ 명령을 주고 받기

연령

- ✓ 7 살 이상

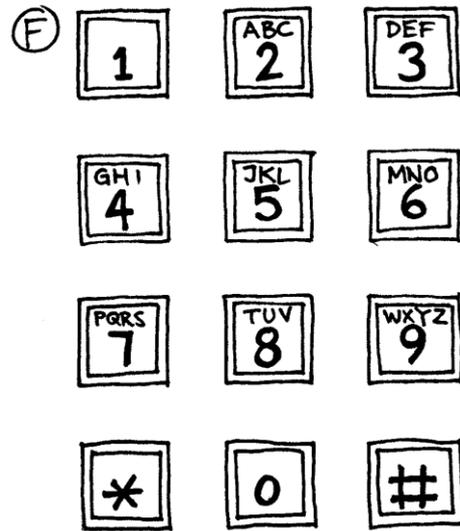
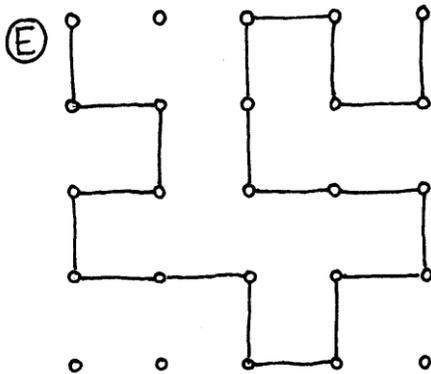
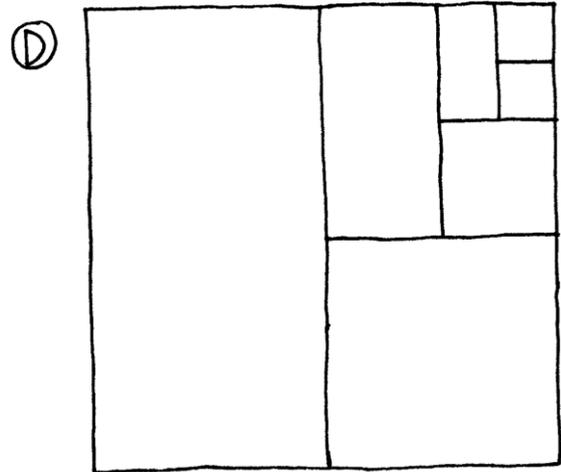
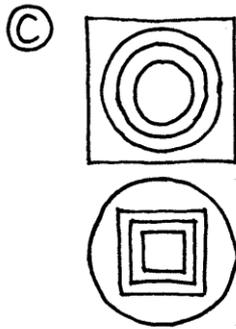
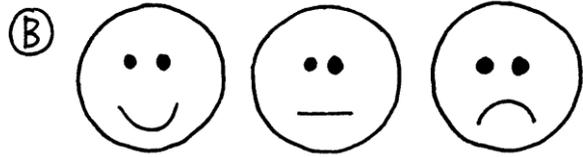
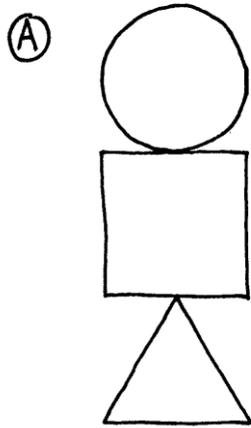
도구

교사가 필요한 것:

- ✓ 다음 쪽에 있는 그림이 있는 카드

학생이 필요한 것:

- ✓ 연필, 종이, 자



행진하라 - 행진 순서

소개

사람들이 지시를 정확하게 따르는 것이 좋은지에 대하여 이야기해봅시다. 예를 들면 닫힌 문을 가리키며 “저 문을 통해 나가세요”라고 말한다면 어떤 일이 일어날까요?

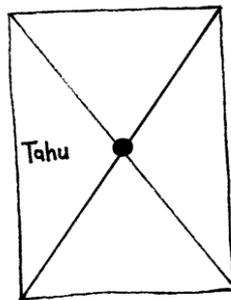
컴퓨터는 명령에 따라 작동합니다. 심지어 어이없는 명령을 내려도 그것을 무조건 따릅니다.

시범설명 예시

학생들이 명령에 따라 그림을 그리도록 하세요.

1. 페이지의 중앙에 점을 그립니다.
2. 오른쪽 아래 모서리에 그린 점을 지나는 직선을 왼쪽 위에서부터 그립니다.
3. 오른쪽 위 모서리에 그린 점을 지나는 직선을 왼쪽 아래에서부터 그립니다.
4. 페이지의 왼쪽 중앙의 삼각형에 자기 이름을 적습니다.

결과는 다음과 같습니다:



활동

한 명의 학생을 선택하여 123 쪽의 예시와 같은 그림을 줍니다. 이 학생에게 그가 가진 그림을 그릴 수 있도록 다른 친구들에게 말로 설명하도록 합니다. 다른 학생들은 명령을 정확하게 이해하기 위해 질문을 할 수 있습니다. 이 활동의 목적은 얼마나 빠르고 정확하게 완성할 수 있는지를 확인하기 위한 것입니다.

다른 그림을 그리는 활동을 반복하면서 이번에는 질문을 하지 않도록 합니다. 간단한 그림이지만 다른 학생들이 쉽게 틀릴 수 있는 그림을 선택하는 것이 좋습니다.

이제 벽 뒤에 가려진 학생에게 질문을 하지 못하도록 한 뒤에 지시 명령을 내려 의사소통 활동을 시도해 보게 합니다.

이러한 형태의 의사 소통이 실제 프로그래밍을 할 때 프로그래머가 경험하는 것이라고 설명합니다. 학생들이 컴퓨터에 명령어는 주지만 끝날 때까지 결과를 알 수 없도록 합니다.

학생들에게 그림을 그리게 한 뒤 그림을 그리게 하는 명령어를 적게 합니다. 짝과 하거나 교실 전체의 학생들에게 시도해 봅니다.

변형 활동

1. 명령어가 적힌 종이 카드를 만듭니다.
2. 교실에서 원하는 장소에 도착할 수 있는 방법을 “2미터 앞으로 이동”, “90도 좌회전”, “90도 우회전”과 같은 명령어를 사용해 적습니다.

학생들은 원하는 결과가 나올 때까지 시험해보고 명령어를 수정해야 합니다.

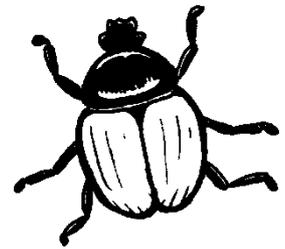
3. 다른 활동으로 한 학생은 눈을 가리고 다른 학생들은 명령어를 사용해 명령해봅니다.

컴퓨터과학 이야기

컴퓨터는 특정한 작업을 수행하기 위해 작성된 프로그램이라는 명령문에 따라 작동합니다. 프로그램은 컴퓨터에 전달하는 명령어로 이루어져 있습니다. 일부 프로그래밍 언어는 특별한 목적에 더 적합하도록 만들어졌습니다.

어떤 언어든지 프로그래머는 반드시 컴퓨터가 수행할 작업을 정확하게 지정하도록 작성해야 합니다. 인간과는 달리 컴퓨터는 명백히 말도 안되는 명령이라도 의심없이 그대로 수행하기 때문입니다.

프로그램을 정확하게 작성하는 것은 중요합니다. 아주 작은 오류(error)도 많은 문제를 일으킬 수 있습니다. 엘리베이터 운행, 우주 왕복선 발사, 원자력 발전소, 열차 선로 신호에서 프로그램에 오류가 생긴 경우를 상상해 보세요! 오류는 일반적으로 “버그”라고 합니다. 1940년대 초기 컴퓨터에 나방이 날아들어와 전기흐름을 방해하여 나방(버그)을 제거(디버그)했기 때문에 이와 같이 부릅니다.



프로그램이 복잡할수록 더 많은 오류들이 있을 가능성이 있습니다. 이는 미국의 전략 방위 구상(“스타 워즈”)프로그램, 핵 공격에 대한 방어를 하기 위해 만든 컴퓨터 제어 시스템에서 중요한 문제가 되었습니다. 컴퓨터 과학자 몇 명은 핵공격 방어 시스템이 소프트웨어의 복잡함과 신뢰성 부족 때문에 큰 문제를 일으키고 제대로 작동하지 않을 수 있다고 주장했습니다. 핵공격 방어 시스템의 소프트웨어는 오류가 있는 버그를 찾아서 그 문제를 해결하고 그것이 제대로 작동하는지 확인을 해야 하지만 미사일을 실제로 발사해야 제대로 작동하는지 확인 할 수 있기 때문에 이 시스템 검사를 제대로 시행할 수 없었습니다!

Part IV

정말 어려운 문제 풀기—난해성

난해성

컴퓨터도 해결하기 어려운 문제가 있을까요? 이번 활동에서는 컴퓨터가 할 수 없는 대화—채팅—를 소개합니다. 컴퓨터가 대화를 할 수 없는 이유는 컴퓨터가 말을 하지 못해서가 아니라 이해한다거나 감각적인 말을 하기 위한 생각을 할 수 없기 때문입니다. 하지만 그것은 우리가 앞으로 살펴볼 어려운 문제는 아닙니다. - 이것은 컴퓨터가 대화를 할 수 없다는 것이 아니라 사람이 무엇을 해야 하는지 알 수 없기 때문에 컴퓨터가 무엇을 해야 할지 전달해주지 못한다는 것입니다. 하지만 이 장에서는 컴퓨터에게 무엇을 하라고 했을 때—프로그램을 작성할 때—생기는 문제 (컴퓨터로 너무 오래 걸리는 일 즉, 수 천만년이 걸릴 수도 있는)를 살펴 보게 될 것입니다. 이러한 문제를 해결하는 것은 아무리 좋고 빠른 컴퓨터만으로 충분하지 않습니다. 컴퓨터가 백배가 빠른 경우에도 여전히 수백만년이 걸릴 것입니다. 심지어 백만배가 빠르다고 해도 몇백년은 걸릴 것입니다. 그것이 바로 *어려운* 문제라고 부르는 것입니다. — 어려운 문제는 가장 빠른 컴퓨터를 사용하는 것보다 문제를 해결하기 위한 상상력과 창의적인 알고리즘이 더 좋을 수 있습니다.

Part II 의 알고리즘 활동에서 컴퓨터 프로그램을 보다 효율적으로 만드는 방법을 알아보았습니다. 이 장에서는 효율적인 해결 방법이 알려져 있지 않은 문제를 살펴보고 컴퓨터가 해결하는데 수백만년이 걸리는 이유를 알아봅니다. 이것을 살펴보는 순간 여러분은 오늘날 컴퓨터 과학에서 가장 어려운 난제와 마주하게 될 것입니다. 효율적인 해결 방법을 찾지 못한 문제들로 우리는 그 해결방법이 무엇인지 아직 알지 못합니다. 그러한 문제들이 여기 제시한 것들이 전부는 아닙니다. 이러한 문제는 주변에 수천가지의 문제로 존재합니다. 그것들의 적용 분야가 모두 다를지라도 만약에 효율적인 방법을 발견하게 된 경우 그러한 난제를 해결하기 위해 그 해결 방법을 적용할 수 있어 위대한 발견을 하게 될 것입니다. 이번 활동에서 이러한 문제에 관하여 배우게 됩니다.

선생님을 위한 팁

이 장에서 3 가지 활동을 하게 됩니다. 첫째는 지도에 색을 칠할 때 이웃 도시의 색을 모두 다르게 색칠하려면 몇 가지 색이 필요한지 살펴봅니다. 둘째는 간단한 거리 지도를 사용하여 누구나 한 블록만 이동하여 아이스크림을 살 수 있도록 아이스크림 트럭을 길가에 세우는 방법입니다. 셋째는 각각의 지점들을 연결하는 짧은 네트워크를 만드는 방법을 야외활동을 통해 알아보겠습니다.

다음과 같은 순서로 진행됩니다. 매우 풀기 어려운 문제를 아주 쉬운 형태로 바꿀 수 있도록 합니다. 그러면 일상생활에서 이런 문제는 그리 어렵지 않습니다. 지도 만들기, 학교 시간표, 도로 건설과 같은 일상적인 활동에서 생길 수 있는 문제에 대해 이 문제와 관련된 질문을 합니다. 이것을 컴퓨터 과학의 개념에서는 “NP-완전성”이라고 부릅니다. 각 활동의 마지막에 나오는 컴퓨터과학 이야기에서 자세히 설명합니다. 이 3 가지 활동들은 임의의 순서로 할 수 있지만 이 장들은 안내한 순서대로 읽는 것을 권장합니다. 마지막 활동을 할 때쯤이면 현대 컴퓨터 과학에서 가장 중요한 미해결 문제에 대하여 조금은 이해를 하게 될 것입니다.

해결하기 어려운 문제 즉, *난해한* 문제를 다루기 때문에 이번 장의 제목은 “난해성”입니다. 이 단어는 라틴어 *tractare* 에서 유래하였습니다. 현재 용어로 *tractable*은 다루기 쉽고 순종적이며 나긋나긋한 *의미*로 쓰입니다. 난해한 문제는 쉽게 답을 구하기에는 시간이 너무 오래 걸리기 때문에 현실 문제 해결에 잘 다루지 않았습니다. 하지만 이렇게 어려운 문제를 해결하는 노력으로 실시되는 획기적인 연구들이 인류에게 매우 중요한 파급 효과들을 가져오기 때문에 난해성은 실제로 큰 관심사입니다. 예를 들어, 대부분의 암호화 코드는 몇 가지 문제의 난해성에 의존하여 효율적인 해결 방법을 찾을 수 있습니다. 그리고 범죄자는 비밀을 해독하고 이를 판매하거나 가짜 은행 거래를 하기도 하는 역기능에 사용합니다.

Part V—암호화에서 이 내용을 다룹니다.

활동 14

가난한 지도 제작가—그래픽 문제

개요

최적화 문제는 동일 시간에 특정 사건들이 발생하지 않도록 구성하거나 어떤 일을 수행할 때 중복되거나 간섭하지 않는 제약을 만족하도록 구성된 문제입니다. 예를 들어 특정 교실을 여러 팀이 사용하거나, 여러명이 참가하는 회의 시간을 결정하기 위해 모든 사람들이 가진 제약을 고려하는 것은 많은 어려움이 있습니다. 이러한 어려움은 지도에서 나라마다 경계를 나타내기 위해 색으로 칠할 때 인접한 국가일 경우 서로 다른 색깔을 선택해야 할 때 나타납니다. 이번 활동은 이러한 문제에 관한 것입니다.

관련 교육 과정

- ✓ 수학: 수 - 다른 수 탐구하기. 기본 두 수를 나타내기
- ✓ 수학: 대수 - 순차 패턴을 계속하고 이 패턴에 대한 규칙 설명하기. 패턴과 2의 거듭제곱과의 관계

기술

- ✓ 문제 해결력
- ✓ 논리적 추론
- ✓ 알고리즘의 절차와 복잡성
- ✓ 통찰력

연령

- ✓ 7세 이상

재료

- ✓ 화이트 보드나 칠판

학생

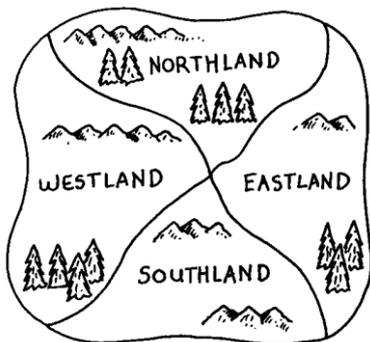
- ✓ 학습지 한 장 또는 여러 장,
- ✓ 지우기 쉬운 컬러 마커펜,
- ✓ 다른 색상의 크레용 (또는 색연필, 펠트 펜 등.)



그래픽 문제

소개

이 활동은 지도 제작자가 학생들에게 여러나라의 경계를 표시하기 위해 각 나라별로 색을 칠해 달라는 요청을 한다는 이야기로 시작됩니다. 모든 나라의 경계 부분은 색이 달라야 하며 색상은 관계가 없습니다.



예를 들어 이 지도에는 네 개의 나라가 있습니다. 만약 Northland 가 빨강색인 경우, Westland 와 Eastland 는 빨강을 칠할 수 없습니다. Northland 와 같은 색이어서 각 나라를 구분하기가 힘들어지기 때문입니다. 만약 Westland 를 초록으로 했을 때는 Eastland 도 초록색으로 칠할 수 있습니다. Westland 와 경계가 맞닿아 있지 않기 때문입니다. 두 나라가 하나의 꼭지점에서 만난다면 같은 색의 경계가 없기 때문에 색상 제한에 포함하지 않습니다. Southland 는 빨강으로 칠할 수 있습니다. 그래서 우리는 이 지도를 칠하기 위해 최소 두 가지 색만 있으면 됩니다.

지도 제작자는 가난하여 페인트가 많이 없기 때문에 가능하면 적은 수의 색을 사용하라는 주문을 하였습니다. 이제 그 방법을 찾아 봅시다.

토론

칠판에 색을 칠하는 과정을 보여주며 학생에게 수행하게 될 문제를 설명합니다.

첫 번째 학습지를 나누어줍니다. 이 지도는 두 가지 색을 사용하여 칠할 수 있습니다. 다만 두가지 색을 사용하라는 것이 어렵게 느껴질지도 모르지만 각 나라에 칠할 색에 대해 스스로 선택할 수 있기 때문에 더 많은 색상을 필요로 하는 지도에 비해 훨씬 간단합니다.

학생은 두가지 색으로 지도를 칠해 봅니다. 이 과정에서 해야할 것은 어떤 한 나라에 색을 칠하면 경계를 맞대고 있는 나라는 반대의 색상이어야 한다는 것을 발견할 수 있습니다. 모든 나라에 색을 칠할 때까지 이 규칙은 반복 적용됩니다. 색을 칠하는 동안 규칙을 발견할 수 있다면 가장 좋습니다.

연습활동이 끝났다면 다음 학습지를 합니다.

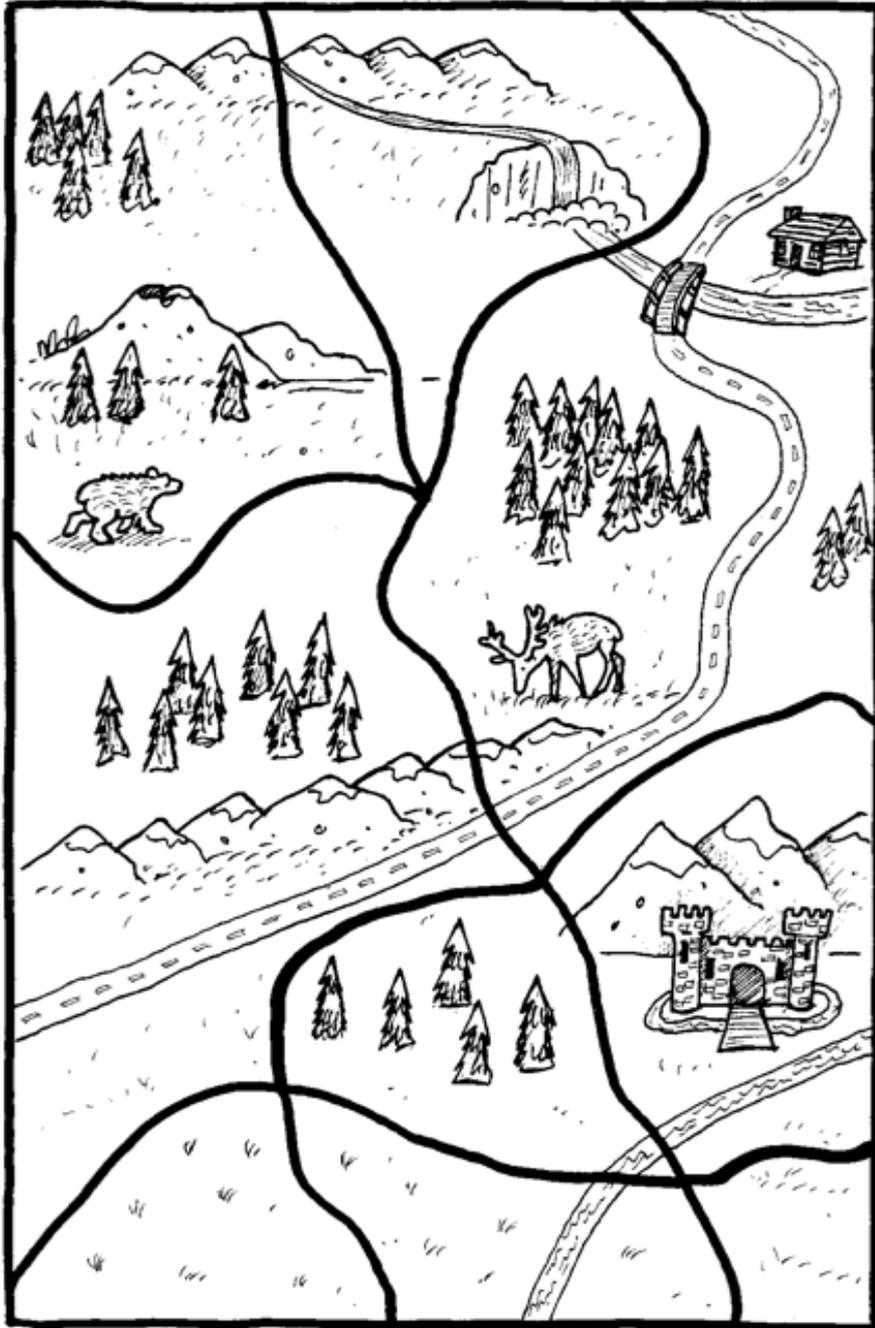
활동 중에 학생들이 색깔 선택의 마음이 바뀌면 고치기 쉽도록 색을 바로 칠하는 것보다 작은 포스트잇이나 색종이 조각을 이용하여 지도 위에 붙이는 방법을 사용하도록 합니다.

고학년들에게는 색상의 최소 수를 발견하는 방법을 설명하도록 합니다. 예를 들어 서로 다른 두 경계를 갖는 세 나라(최대 3 개) 집단이 있기 때문에 최소 3 가지의 색이 필요합니다.

학생들 중 학습지 활동을 빨리 끝낸다면 5 개의 다른 색상이 필요한 지도를 고안해 달라고 합니다. 어떤 지도는 4 개의 색으로 모든 도시를 칠할 수 있음이 증명되어 있기 때문에 이것을 이해하는데 얼마의 시간이 요구되며 시행착오가 필요할 것입니다. 경험상 학생은 신속하게 5 가지 색상이 필요하다고 믿고 지도에 색을 칠하지만 언제나 4 가지 색이 필요하다는 해결책을 찾아낼 것입니다.

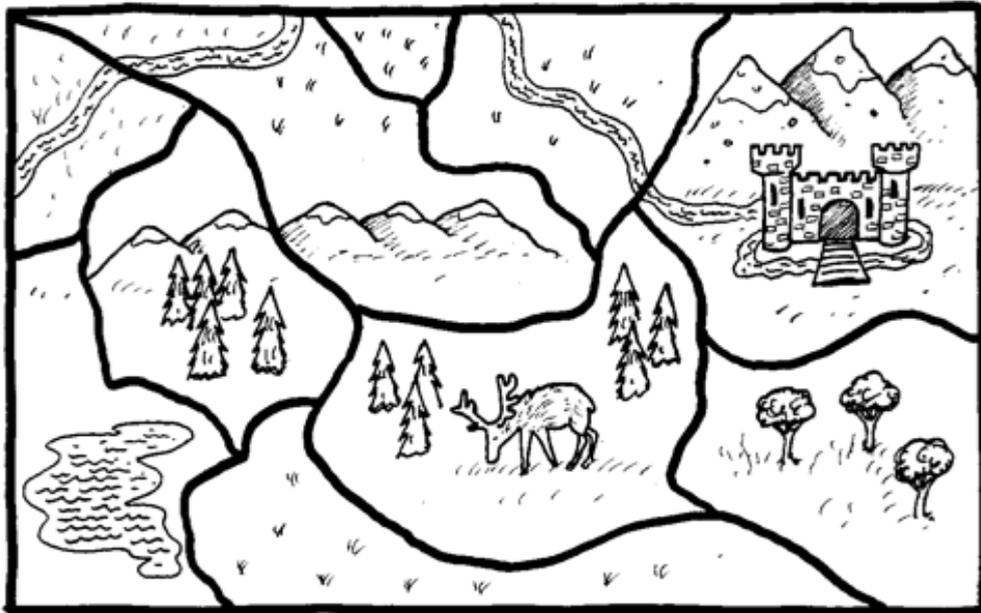
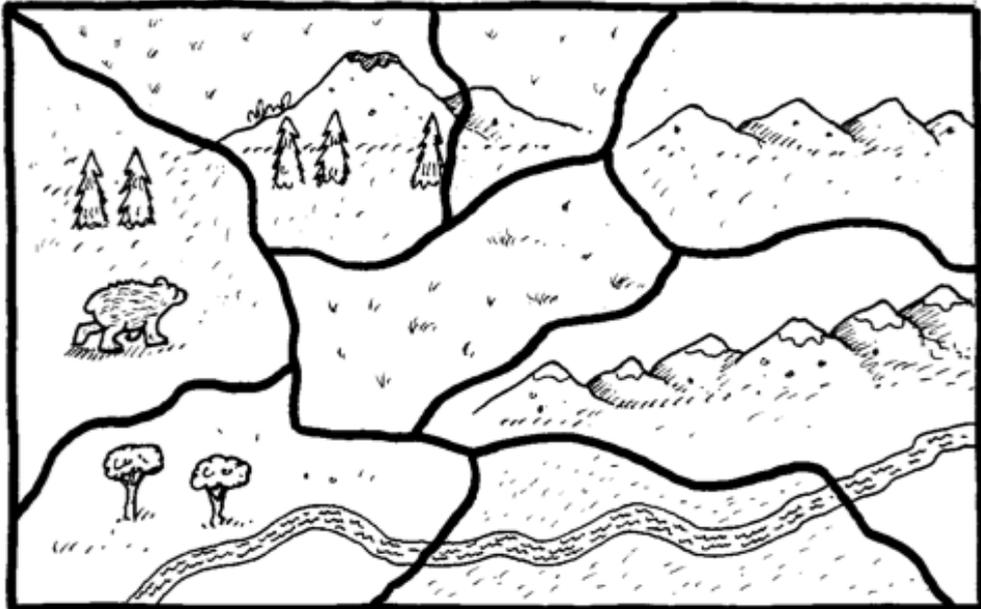
워크시트활동: 그래픽 문제 1

가능한 한 적은 색으로 지도를 색칠하되, 경계를 맞대고 있는 나라는 같은 색을 사용하지 않도록 하세요.



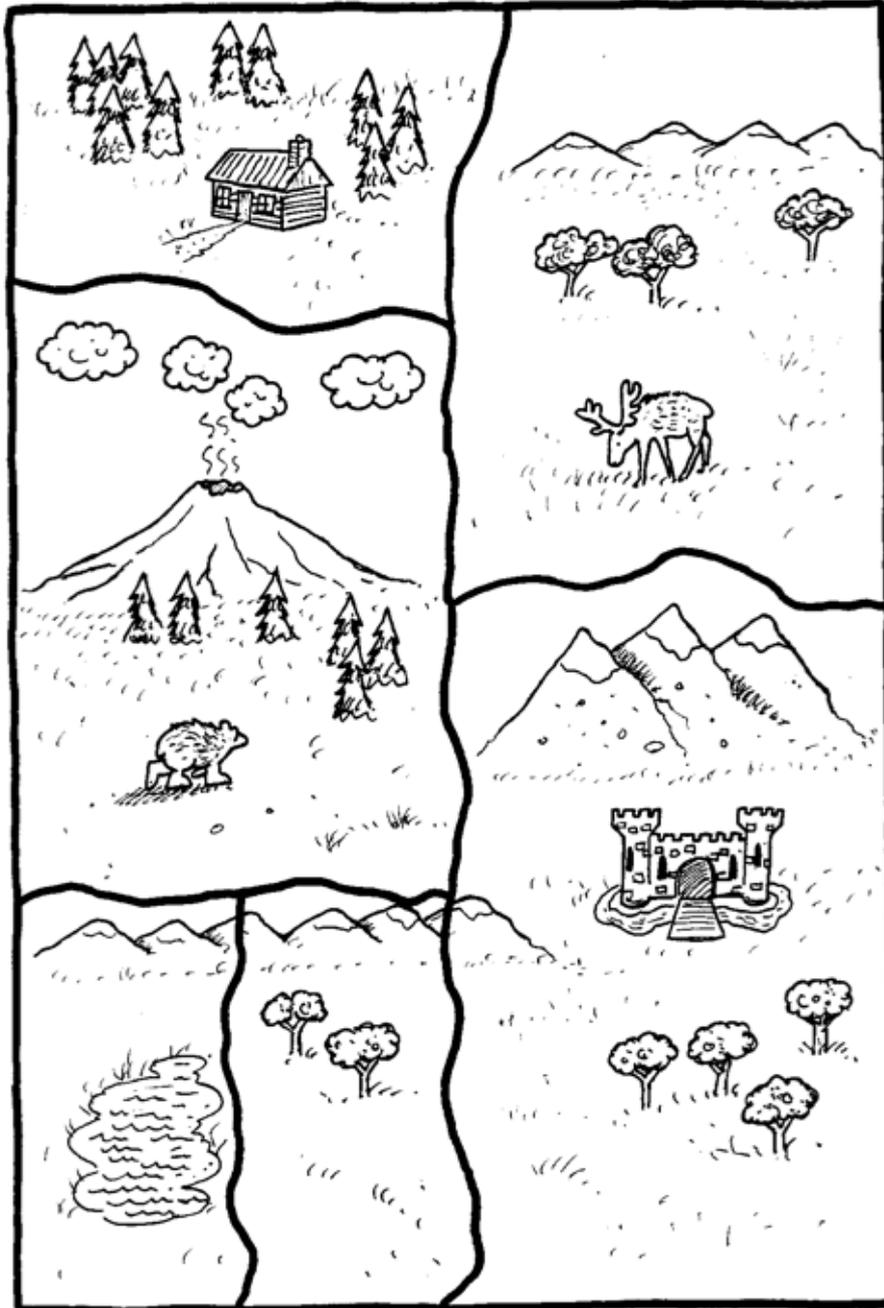
워크시트활동: 그래픽 문제 2

가능한 한 적은 색으로 지도를 색칠하되, 경계를 맞대고 있는 나라는 같은 색을 사용하지 않도록 하세요.



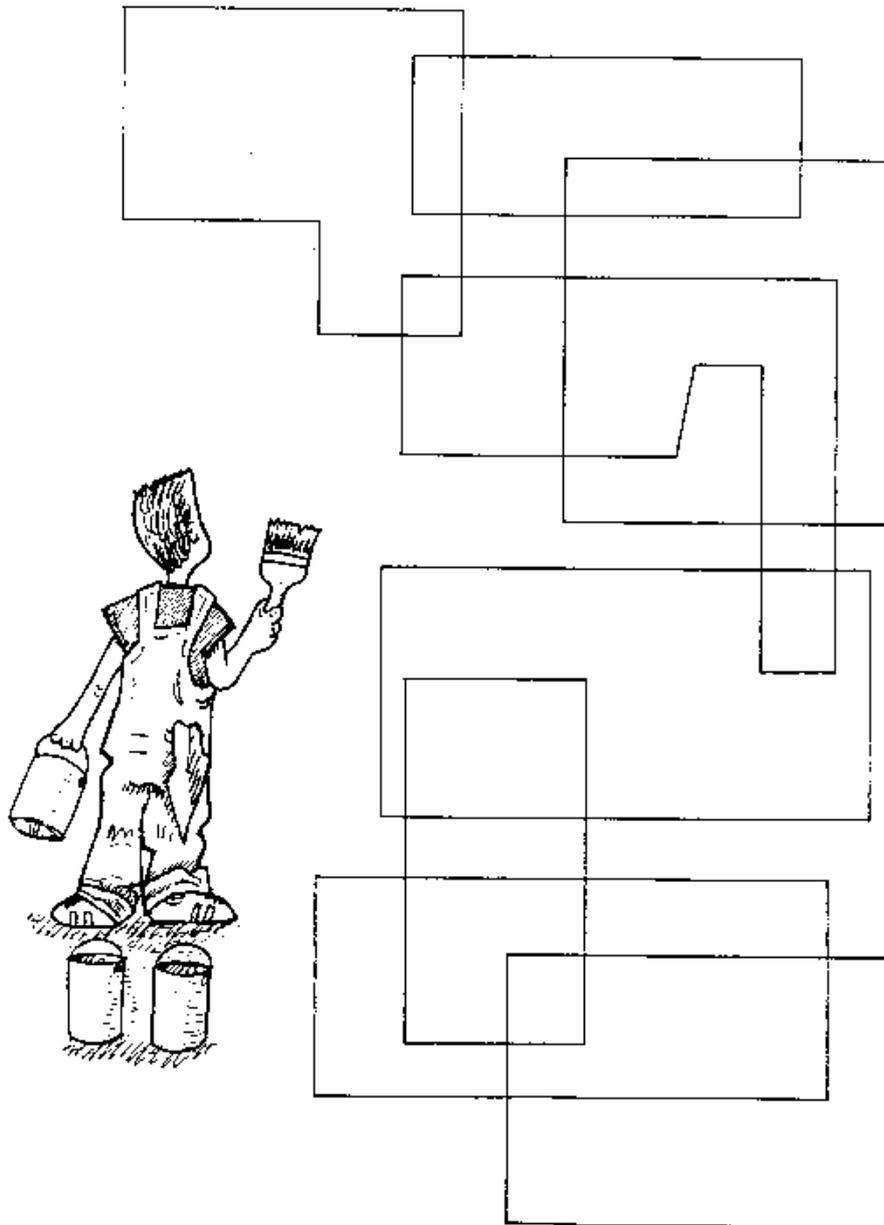
워크시트활동: 그래픽 문제 3

가능한 한 적은 색으로 지도를 색칠하되, 경계를 맞대고 있는 나라는 같은 색을 사용하지 않도록 하세요.



워크시트활동: 그래픽 문제 4

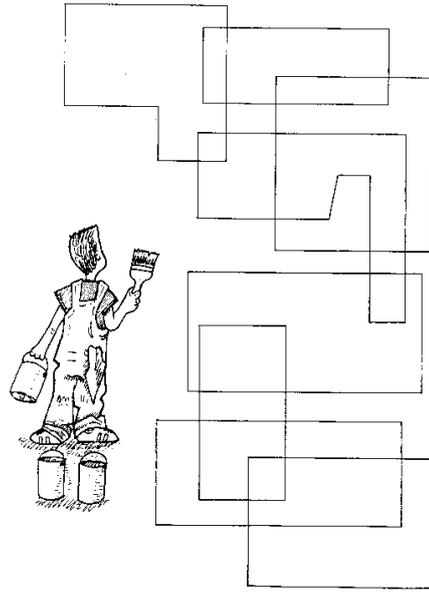
가능한 한 적은 색으로 지도를 색칠하되, 경계를 맞대고 있는 나라는 같은 색을 사용하지 않도록 하세요.



한 단계 더!

이것과 같이 두 가지 색만으로 지도를 구성하는 간단한 방법이 있습니다. 이 지도는 폐곡선(처음과 마지막이 닫힌 선)이 겹쳐져 있습니다. 어떤 모양이든 서로 이것의 위에 두 가지 색으로 칠할 수 있습니다. 학생들은 이런 종류의 지도를 스스로 만들어 볼 수 있습니다.

4 가지 색이면 지구본이나 종이지도에 색을 칠하기에 충분합니다. 만약 도넛모양의 지도에 색을 칠한다면 얼마나 많은 색깔이 필요할지 궁금해 할 것입니다. 이 경우에는 다섯 개의 색이 필요합니다. 그리고 어떤 경우에도 항상 다섯 색이면 충분하게 색을 칠할 수 있습니다. 학생들이 이것을 해보고 싶어할지도 모릅니다.



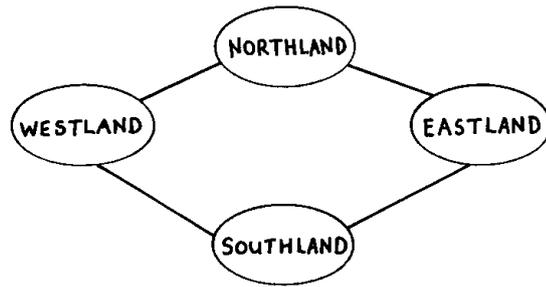
현재 알려지지 않은 생소한 지도를 이용하여 다양하고 재미있는 변화를 줄 수 있습니다. 예를 들어 종이 한 장에 그려진 지도에 색을 칠했을 경우 똑똑한 학생 한명이 칠한다면 네 가지 색이면 충분하다는 사실을 알 수 있습니다. 하지만 그 대신에 협력이 안되거나 똑똑하지 못한 짝공과 함께 할 경우 나라에 색을 번갈아 칠하면 상황은 달라지게 됩니다. 내가 영리하게 작업하고 있는 동안 내 짝공은 규칙대로 나와 번갈아가며 지도에 색을 칠합니다. 만약 짝공이 영리하지 못하게 색을 칠하고 있다면 몇 가지 크레용 색만으로 잘못 색칠한 것을 보완하고 만회하는게 가능할까요? 최대의 수는 알려져 있지 않습니다! 1992 년에 33 개의 크레용이면 항상 충분하다고 입증하였으나 2008 년에는 17 개로 충분하다는 것으로 증명되어 수정되었습니다. 그러나 아직 실제 몇 개의 색이 필요할지는 정확하게 모릅니다만 전문가들은 10 개 미만의 색으로 충분할 것이라 추측합니다. 학생들은 상대방이 필요로 하는 색상의 수를 최대화하려는 목표를 두고 진행하는 2 인용 게임으로 변경하여 즐길 수도 있겠죠.

지도에 색칠하는 다른 변형 문제는 *식민지의 색칠하기*로 알려져 있습니다. 같은 갯수의 나라가 포함된 2 장의 지도가 필요합니다. A 지도(예를 들어 지구)에 있는 한 개의 나라는 B 지도(예를 들어 달의 식민지)에 있는 하나의 나라와 정확히 대응해야 합니다. 지구에 있는 나라의 색을 달의 식민지에 있는 나라에 같은 색상으로 칠해야 합니다. 이 요구사항을 추가했을 때 이 문제를 해결하기 위해 얼마나 많은 색이 필요할까요? 답은 현재 알려져 있지 않습니다.

컴퓨터과학 이야기

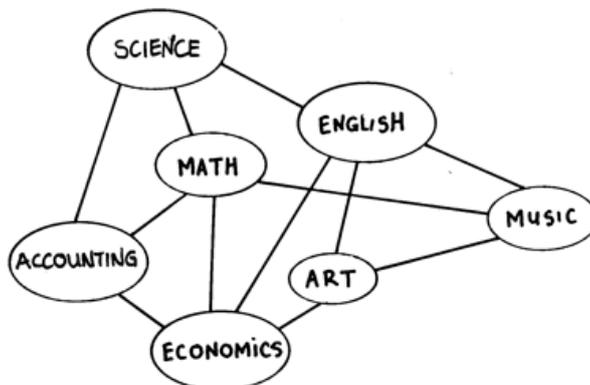
우리가 이 활동에서 알아본 지도에 색칠하기 문제는 필요한 최소의 색상—두 가지, 세 가지 또는 네 가지—을 찾아내는 데 필수적인 문제입니다. 모든 지도는 4 가지 색만을 사용하여 칠할 수 있다는 추측은 1852 년에 공식화하였지만 1976 년 까지 증명되지 않았습니다. 컴퓨터 과학자들이 수십 년 동안 해결하지 못했던 어려운 문제들 중 4 색 정리 문제는 120 년이 지난 후에야 입증하였습니다.

지도에 색칠하기 문제는 일반적으로 “그래프 문제”로 알려져 있습니다. 컴퓨터 과학에서는 보기와 같이 지도를 그래프의 형태로 바꾸어 추상적인 표현으로 나타냅니다.



활동 9 에서 설명한 진흙 도시 문제와 같이, 그래프라는 용어는 막대 그래프로 수치 데이터를 표시하는 차트를 의미하는 수학과는 다른 의미로 사용되고 있지만, 컴퓨터 과학자들은 관계없이 사용합니다. 컴퓨터 과학에서 그래프는 개체 사이 관계의 여러 종류를 기술적으로 표시하기 위해 “링크”라고 부르는 선과 “노드”라고 부르는 원(또는 점)을 사용하여 그립니다. 위의 그래프는 이번 활동에서 처음보았던 지도를 표현합니다. 노드는 나라를 대표하고, 두 개의 노드 사이의 링크 선은 두 나라가 경계를 맞대고 있음을 보여주고 있습니다. 그래프에서 색칠하는 규칙은 연결된 노드에 같은 색이 할당되지 않도록 하는 것입니다. 지도의 2 차원적 성질이 가능한 배열을 제한하는 한편, 다양한 제약 조건을 두고 그리는 지도와는 달리, 일반적인 그래프가 요구하는 색상의 수에는 제한이 없습니다. “그래픽 색칠 문제”는 특정 그래프를 그리기 위해 필요한 색상의 최소 수를 찾는 것입니다.

오른쪽 그림의 그래프는 시간표 편성 문제에서도 적용됩니다. 두 과목 사이의 선은 적어도 한 명의 학생이 두 과목을 듣고 있으며, 같은 시간에 두 과목을 편성하여 작성할 수 없습니다. 이 표현방법을 사용하여 시간의 최소 수를



사용하여 실행 가능한 일정을 찾는 문제는 서로 다른 색상으로 다른 시간을 칠하는 색칠 문제와 같습니다. 그래픽 문제의 알고리즘은 컴퓨터 과학에서 큰 관심사입니다. 그리고 실생활의 많은 문제에서 사용되고 있습니다. 아마 지도에 색칠하는 문제에서는 직접 사용되진 않을 것입니다. —우리의 가난한 지도 제작자는 그저 소설의 주인공일 뿐이니까요.

그래프에 기반한 다른 문제는 주변에 수천 가지가 있습니다. 데이터를 그래프의 형태로 나타내는 것은 컴퓨터 과학에서 많이 사용하는 방법으로 활동 9의 최소 신장 트리 및 활동 14의 지배집합 그리고, 이 책의 다른 부분에서 설명되어 있는 진흙도시 문제뿐만 아니라 화학식의 분자에서 원자 사이의 연결 상태 분석, 인터넷의 네트워크를 통해 짧게 이동하는 경로를 찾는 것, 인쇄 회로 기판의 구성 요소 사이의 연결 방법 등의 대형 프로젝트를 수행하기 위한 문제들에 사용됩니다. 이러한 이유로, 그래프 표현을 포함한 문제는 오랫동안 컴퓨터 과학자들을 매료시켰 왔습니다.

이러한 문제를 해결하는 것은 매우 어렵습니다. 개념적으로 어려운 것이 아니라 문제를 푸는데 너무 오랜 시간이 걸리기 때문입니다. 예를 들어, 학교에서 30 명의 교사와 800 명의 학생이 모두 만족하는 시간표를 작성하는 그래프 문제는 현재 알려진 알고리즘으로는 수 년, 또는 수 세기가 걸립니다. 그리고 이것은 컴퓨터가 작업이 끝나기 전에 고장나지 않고 완료하는 것을 가정한 것입니다! 이 문제의 해결책은 정답을 찾아내는 시간과 관계가 있습니다. 최적화된 알고리즘을 사용한다 하더라도 매우 오랜 시간이 걸릴 수 있습니다.

색칠 문제를 해결하기 위해 계산하는데 필요한 시간은 그래프의 크기에 따라 기하 급수적으로 증가합니다. 지도 색칠 문제를 다시 생각해 봅시다. 지도를 색칠할 수 있도록 가능한 모든 방법을 시도하여 해결할 수 있습니다. 우리는 최대 4 색이 필요하다는 것을 알고 있기 때문에 우리는 각각의 나라들에 네 가지 색상을 할당하는 모든 조합을 평가해야 합니다. n 개의 나라가 있는 경우 4^n 개의 조합이 있습니다. 추가된 모든 나라에 4 조합의 수를 곱하면 문제를 푸는 시간은 4 배로 늘어납니다. 이 수는 매우 빠르게 증가합니다. 50 개국에 대한 색칠 문제의 경우 4 의 50 승이기 때문에 우리가 더 빠른 컴퓨터를 개발한다고 해도 이러한 종류의 문제는 절대 쉽게 해결되지 않을 것입니다!

그래픽 문제는 해결 시간이 기하 급수적으로 늘어나는 문제로서 좋은 예입니다. 이러한 활동에 소형 지도와 같은 문제처럼 간단한 사례를 통해 최적의 해결 방법을 찾아내는 것은 매우 쉽지만 10 개를 넘는 나라의 수가 늘어남에 따라 문제는 수작업으로 하기 매우 힘들어 질 것이며 심지어 100 개 이상의 나라에서도 컴퓨터가 최적의 상황을

선택하여 지도를 색칠하는 모든 가능한 방법을 시도하는 데 수 년이 걸릴 수도 있습니다.

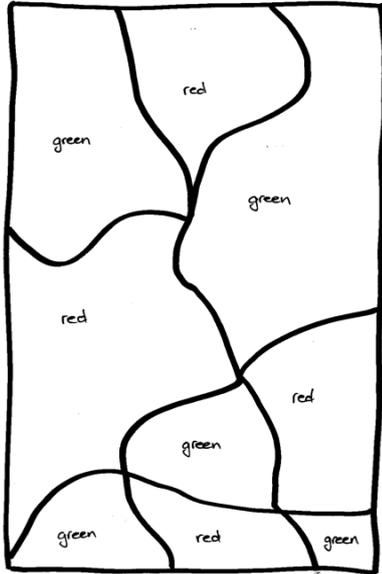
실생활의 많은 문제들이 이와 같을 수 있습니다. 그러나 어쨌든 해결해야 합니다. 컴퓨터 과학자들이 현재 제시하는 방법을 이용하면 쉽게 사용할 수 있지만 완벽한 답은 주지 않습니다. 이러한 경험적 방법인 *휴리스틱* 기법은 최적화된 정답보다는 그에 가까운 답(근사치)을 제시해 줍니다. 학교의 시간표를 작성하는 경우 모두가 완벽하게 만족스러운 답을 찾는 것보다 비용이 조금 더 들겠지만 교실을 하나 더 사용하거나 하여 해결할 수 있고, 가난한 지도 제작자는 비록 가난하지만 필요하다면 2-4 가지 색만을 이용하기보다 몇가지 추가 색상을 이용하여 해결하면 됩니다.

이러한 기하급수적 증가의 문제점을 해결하기 위해 기존의 컴퓨터에서 해결할 수 있는 다양한 알고리즘을 제시하였지만 어떤 것도 완벽한 방법이라는 것을 증명하지 못하고 있습니다. 그리고 컴퓨터 과학자들은 지금까지 발견된 효율적인 방법에 대해서도 회의적입니다. 우리는 다음 두가지 활동에서 이러한 종류의 문제에 대하여 더욱 자세하게 알아보겠습니다.

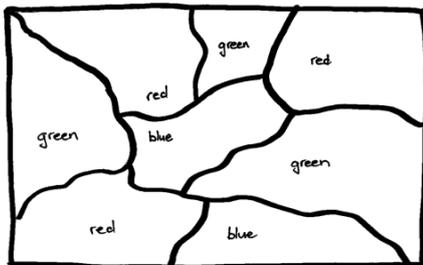
추가로 읽어보기

Harel 은 *Algorithmics* 에서 4 색 정리와 그 역사에 대해 설명합니다. 여기에서 논의되는 지도 색칠 문제의 여러 측면은 Casey 와 Fellows 의 *MEGA-Mathematics!* 입니다. Kubale 의 2004 년판, *Graph Colorings*, 는 이 문제의 역사를 담고 있습니다. 이 주제를 다루는 많은 웹사이트가 있으니 살펴보시기 바랍니다.

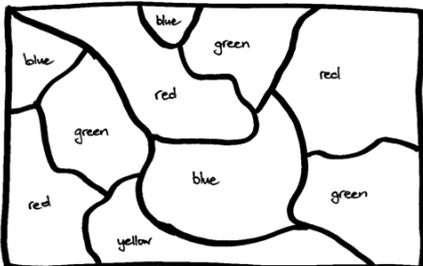
정답과 힌트

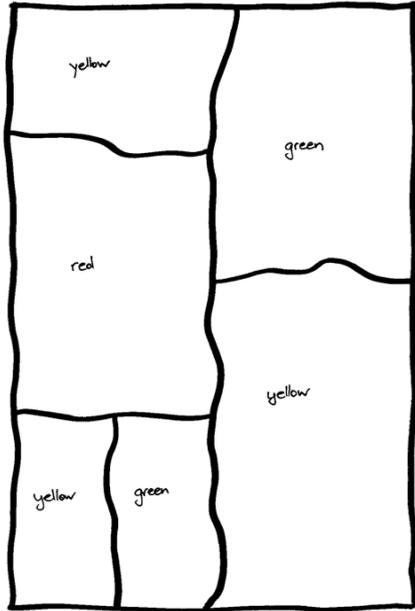


워크시트 1 의 지도에 색칠하기에 대한 해결책입니다. (물론 색상의 선택은 학생에게 달려 있지만 두 가지 색이 필요합니다.)

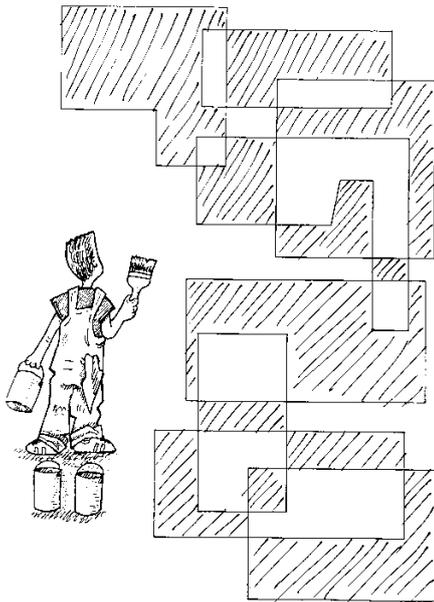


워크시트 2 의 지도는 3 색을 이용하여 색칠 할 수 있습니다. 아래의 지도는 4 가지가 필요합니다. 두 가지 모두 가능한 방법입니다.





워크시트 3 은 3 가지 색으로 해결 가능한 지도입니다.



두 가지 색상만을 사용한 워크시트 4 (회색과 흰색).

활동 15

여행자 도시—지배 집합

요약

실제 상황에서의 많은 문제들은 네트워크로 표현하거나 활동 13 에서 소개된 지도 색칠에서 사용되었던 “그래프”로 추상화되어 표현할 수 있습니다. 네트워크로 표현하는 것은 실제로 유용한 알고리즘의 개발을 위해 많은 아이디어를 제공합니다. 이 활동에서 한곳(노드)는 다른 곳(노드)과 한 칸씩 떨어져 표시됩니다. 찾아야 할 문제의 답은 얼마나 적은 수의 노드로 그래프를 표시할 수 있을지 함께 알아보겠습니다.

관련 교육 과정

- ✓ 수학 - 위치와 방향
- ✓ 수학 - 논리 추론

기술

- ✓ 지도, 관계, 퍼즐 해결
- ✓ 반복된 목표 탐색

연령

- ✓ 7 세 이상

준비물

학생

- ✓ 검은 줄의 *아이스크림 트럭*
- ✓ 여러 색의 단추 또는 칩

교사

화이트보드에 검은 줄 마스터 아이스크림 트럭 해결책, 또는 그것을 그릴 화이트 보드 프로젝터 이미지.



지배 집합

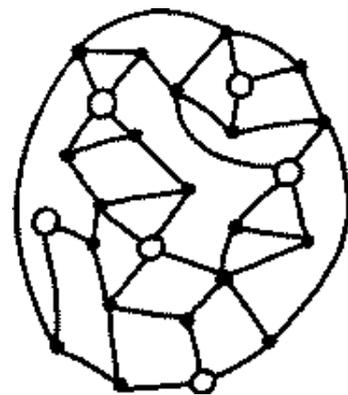
소개

아이스크림 트럭 학습지에는 여행자 도시의 지도가 있습니다. 선은 도시의 거리를 나타내고 점은 길모퉁이를 나타냅니다. 이 도시는 여름에 매우 더워서 아이스크림 트럭들이 길모퉁이에 주차하고 여행자들에게 아이스크림을 팔고 있습니다. 여행하는 사람들이 거리의 어느 곳을 돌아다니든 한블록 떨어진 모퉁이만 이동하여서 아이스크림을 살 수 있도록 트럭을 배치하고 싶습니다. 많은 트럭을 배치하면 문제가 쉽게 해결되겠지만 문제는 적은 수의 트럭으로 모든 여행자들이 한블록 떨어진 모퉁이에서 아이스크림을 살 수 있도록 배치하는 방법에 대해 알아보는 것입니다.

토론

1. 학생들을 소모둠으로 나누어서 각각 여행자 도시의 지도와 몇 개의 단추를 나누어 주고 이야기를 설명해 줍니다.
2. 아이스크림 트럭을 표시하기 위해서 교차로에 단추를 하나 배치한 다음 다른 색의 단추를 같은 거리의 다른 교차로에 배치하도록 합니다. 이 교차로를 여행하는 모든 사람들이 아이스크림 트럭에서 사먹을 수 있도록 논의를 합니다.
3. 다른 위치에 트럭(단추)을 놓고 실험을 하도록 합니다. 모든 거리에서 서비스를 받을 수 있는 방법을 찾아야 하며, 트럭은 매우 비싸므로 가능하면 최대한 적은 수로 배치하여야 합니다. 충분한 수의 트럭으로 모든 교차로에 배치하는 것은 해결의 조건과 맞지 않습니다.

4. 여행자 도시의 트럭은 최소 6 대 있으며, 해결책은 여기에 제시되어 있습니다. 하지만 학생들이 해결책을 찾아내는 것은 매우 어렵습니다! 학생들에게 고민하게 한 후 6 대의 트럭만으로 배치할 수 있음을 알려주고 도전해보도록 합니다. 6 대라는 답을 알려주어도 배치하는 방법은 여전히 매우 어려운 문제라서 많은 모둠이 포기할 것입니다. 트럭의 수를 늘려 8~9 대의 트럭을 배치하는 방법(여전히 어렵습니다만)을 찾도록 안내하세요.

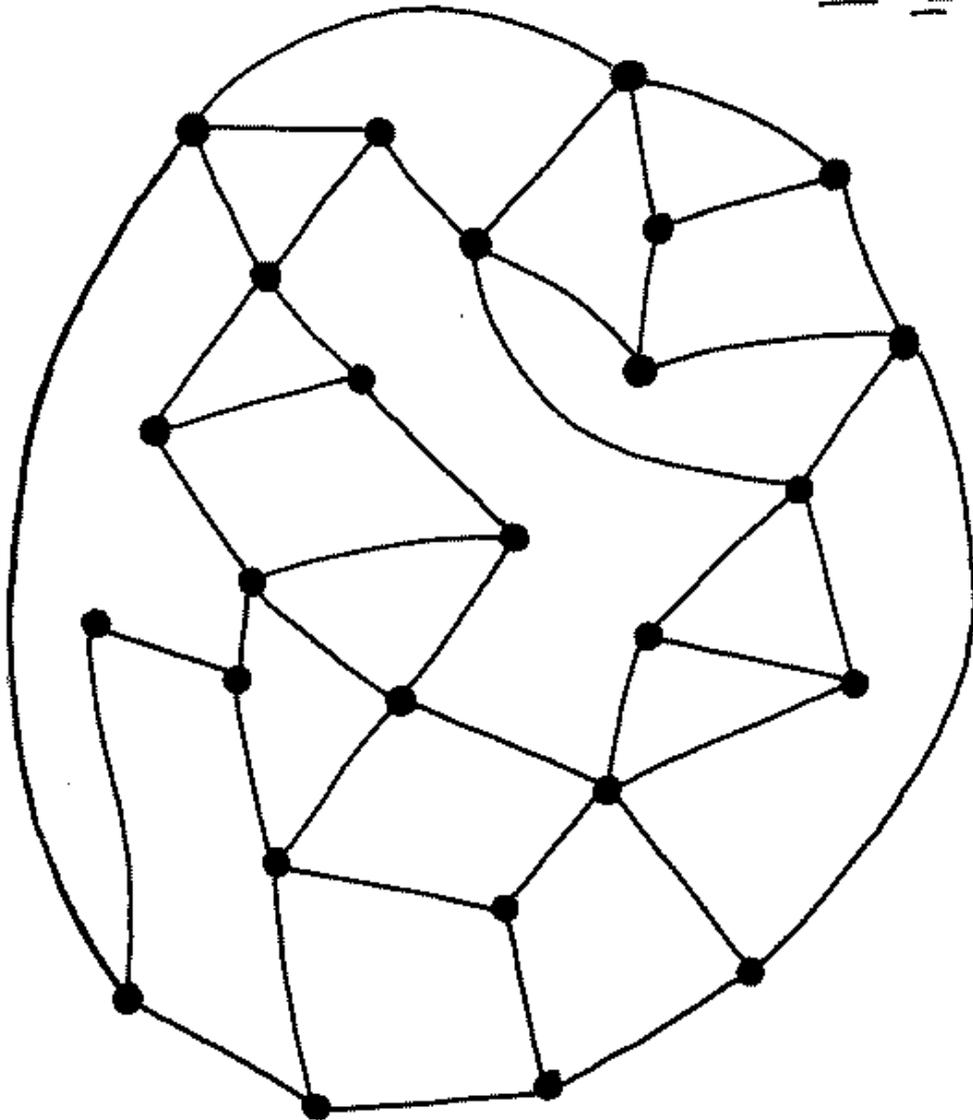


5. 여행자 도시의 지도는 6 개의 조각지도로 시작하며, *아이스크림 트럭 해결책* 학습지의 아래에 있습니다. 조각지도에서 아이스크림 트럭은 1 대가 필요하며 해결책을 감추기 위해 많은 거리들과 연결되어 있습니다. 중요한 것은 해결책 교차로(흰색 점)에는 어떤 연결도 없지만 여분의 교차로(검정색 점)에는 그렇지 않다는 것입니다. 수업에서 이런 방법을 칠판이나 프로젝터를 이용해 보여줍니다.

6. 학생들은 이 전략을 사용하여 자기만의 어려운 지도를 만듭니다. 그리고서 친구들에게 풀어보도록 시도해 보게 합니다. 지도를 만든 학생 자신은 그 퍼즐을 만들고 해결할 수 있지만 다른 사람은 해결하기 어렵습니다! (이것을 “일방향 함수”라고 합니다) 이러한 퍼즐을 푸는 것은 매우 어렵습니다. 처음 그것을 만든 사람이 아니라면 말입니다. 단방향 함수는 암호화에서 매우 중요한 역할을 합니다. (활동 17 과 18 을 보세요).

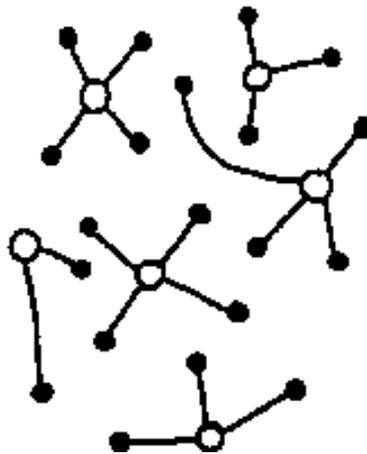
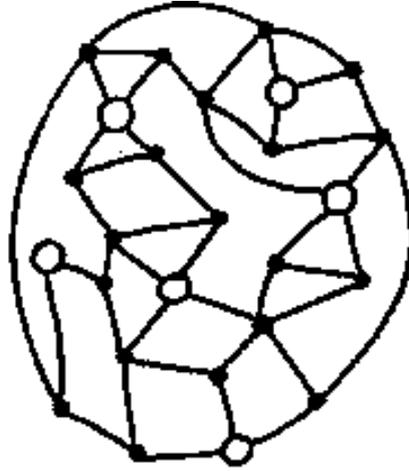
워크시트활동: 아이스크림 트럭

아이스크림 트럭을 교차로에 두고 모든 교차로가 하나의 트럭에 연결되도록 배치하는 방법을 알아봅니다.



워크시트활동: 아이스크림 트럭 해결책

학생들에게 퍼즐을 만드는 방법을 보여줍니다.



한 단계 업!

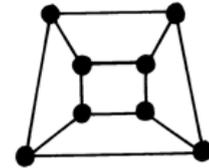
도시를 계획하는 단계에서 이러한 문제와 관련된 상황은 매우 많습니다. 우체통 설치, 경찰서, 소방서 배치하기 등과 같이 실생활에서 이러한 문제를 실제로 풀어야 하는 경우 어떻게 하시겠습니까? 지도에 점을 여러 개 찍어서 쉽게 해결하는 것처럼 쉬운 일을 아닙니다. 경찰서를 여기 저기에 세우려면 비용이 많이 들기때문이지요.

아이스크림 트럭을 거리에 배치하는 가능한 모든 방법을 검토하고 그것이 최선인지를 확인하는 방법을 찾기 위해 여기 아주 간단한 방법이 있습니다. 여행자도시의 26 개의 교차로에는 트럭을 배치할 수 있는 26 개의 가능성이 있습니다. 26 가지의 가능성을 확인하는 것은 간단하지만 그것들 중 어떤 것도 만족하는지 확인하기 어렵습니다. 만약 두 대의 트럭으로 26 개의 장소에 배치한다면 하나를 배치하게 되면 같은 장소에 나머지 하나를 놓지 않기 때문에 남은 장소는 25 개가 됩니다. 즉, $26 \times 25 = 650$ 개를 확인해야 합니다만 이러한 가지수를 모두 확인하는 것은 단순한 일이지만 전체적으로 매우 복잡합니다. 사실 트럭이 어느 위치에 있는 상관 없이 그것들 중 반(325)만 확인하면 됩니다. 1 번 트럭이 A 교차로에 있고 2 번 트럭이 B 교차로에 있을 때, 1 번 트럭이 B 에 있는 것과 2 번 트럭이 A 에 있는 것은 서로 연결되었기 때문에 확인할 필요가 없습니다. 3 대의 트럭을 가져왔을 때는 2600 개의 가능성이 생기고, 4 대의 트럭은 14950 개의 가능성이 발생하고 그 이상이 되면 더 많은 가능성이 생깁니다. 분명한 것은 26 개의 교차로에서 26 대가 있다면 같은 장소에는 충분히 하나의 트럭이 있어서 더 이상 놓을 장소가 없다는 것입니다. 많은 가능성을 평가하는 또 다른 방법은 26 개의 교차로와 임의의 수의 트럭을 고려하는 것입니다. 각각의 길모퉁이에는 두 가지 가능성이 있습니다. —트럭이 있거나 없는—경우의 수의 구성은 2^{26} , 즉 67,108,864 가지의 경우의 수가 생깁니다.

이렇게 문제를 해결하는 방법을 “무작위”알고리즘(무식한 알고리즘 또는 전수 알고리즘)이라고 하며 아주 오랜 시간이 걸립니다. 컴퓨터가 매우 빨라서 문제를 신속하게 해결한다는 것은 널리 알려진 오해입니다. 해결하려는 경우의 수가 너무커서 빠른 컴퓨터로도 해결하기 어렵습니다. 무작위 알고리즘은 부분적인 해결책을 찾아 처리하는 소요시간에 해결방법이 달려 있습니다. 이를 확인하려면 모든 교차로에서 가장 가까운 트럭의 거리를 알아보아야 합니다. 각각의 부분적인 확인은 1 초 만에 확인 가능하다는 것을 가정합시다. 여행자 도시에서 2^{26} 가지의 가능성을 확인하려면 얼마나 걸릴까요? (정답: 2^{26} 은 약 6,700 만입니다. 하루는 86,400 초, 따라서 2^{26} 초는 약 777 일정도이며, 2 년을 꼬박 컴퓨터로 돌려야 합니다) 그리고 1 초 대신 1000 분의 1 초 가 걸린다고 가정해 봅시다. 그렇다면 같은 2 년 동안 36 개의 교차로가 있는 도시의 문제를 해결할 수 있습니다. 2^{36} 은 2^{26} 의 약 1000 배이기 때문입니다. 백만 배가 빠르다고 했을 때 매초에 백만 개의 구성을 확인할 수 있다고

하더라도 2 년 동안 46 개의 교차로가 있는 도시 문제만을 해결할 수 있습니다. 교차로가 46 개라면 그리 큰 도시는 아니지요! 여러분이 살고 있는 도시에 몇 개의 교차로가 있는지 비교해보세요

무작위 알고리즘은 매우 느리기 때문에 또 다른 방법으로 문제를 해결할 수는 없을까요? 우리는 진흙도시(활동 9)에서 사용했던 방법인 욕심쟁이 방법을 사용할 수 있을 것입니다. 우리는 이제 아이스크림에 욕심을 내는 방법으로 접근해 보려합니다. 즉, 아이스크림 트럭 문제에 욕심을 내라는 것입니다. 첫 번째 트럭을 교차로에 배치하고 연결할 수 있는 최대의 수를 찾는 다음(다른 것을 고민할 필요없이 현재의 최적 배치만 고려해서), 두 번째 트럭은 다음으로 많이 연결할 수 있는 교차로에 놓고 같은 방법으로 계속 합니다. 그러나 이것은 아이스크림 트럭의 수를 최소한으로 배치할 수 있는 방법은 아닙니다. 사실 가장 많은 거리를 연결하는 교차로에서 거리 5 개를 연결할 수 있지만 이 장소는 이 도시에서 가장 좋은 위치는 아닙니다. 학생들과 함께 확인해보세요.



더 쉬운 문제를 살펴봅시다. 최소의 구성을 찾는 것 대신 어떠한 점에 배치하고 이것이 최소한의 구성이었는지를 묻습니다. 어떤 경우에서 이것은 아주 쉬운 방법입니다. 예를 들어 이 그림은 어떤 해결책보다도 쉬운 지도입니다. 정육면체 모양의 거리에서 이것의 모서리를 상상한 다음 서로 대각선으로 이어진 반대쪽 점에 두 대의 아이스크림 트럭을 배치하는 것으로 충분하다는 것은 분명합니다. 또한 두 대보다 적은 트럭을 사용했을 때 문제를 해결하는 것은 불가능하다는 것을 직접 알아내야 합니다. 알아내는 방법이 어렵겠지만 불가능한 것은 아닙니다. 하지만 일반적인 지도에서는 아이스크림 트럭이 최소 하나로 되지 않는다는 것을 증명하는 것은 매우 어렵습니다.

컴퓨터과학 이야기

아이스크림 트럭 문제에 대한 흥미로운 사실 중 하나는 부분의 최소 집합을 찾기 위한 방법 중 무작위 알고리즘(무작위탐색 알고리즘, 영국박물관알고리즘 (BMA:British Museum Algorithm)보다 빠른 것이 있는지는 아직까지 *아무도 모른다*는 것입니다! 교차로의 수에 따라 무작위 알고리즘에 의해 소요되는 시간이 기하급수적으로 증가하는 것을—*지수 시간* 알고리즘이라고 합니다. *지수 시간* 알고리즘은 제곱 또는 세제곱 또는 17 승 또는 교차로의 개수에 따른 지수만큼 급격하게 커집니다. 지수 시간 알고리즘은 언제나 충분히 큰 지도에서—심지어 17 승의 알고리즘에서도—기하급수적으로 빠르게 증가합니다. (예를 들어, 그것을 해결할 때 n^{17} 은 117 보다 크고 2^n 보다 작습니다.) 지수 시간 알고리즘을 이용하는 것은 가장 최소한의 장소를 찾는 방법일까요? —아무도 모릅니다. 사람들이 최적의 정답을 찾기 위해 시도하는 것은 매우 힘들기 때문입니다. 그래서 최소한의 부분적인 장소 집합(지배집합)을 찾는 것이 더욱 쉬워보이는 것이 사실입니다. : 무작위 알고리즘은 작은 장소 집합(지배집합)의 모든 가능성을 찾을 때 교차로의 수가 기하 급수적이며 지수 시간 알고리즘으로 발견하거나 정답의 존재가 있는지 증명되지 않았습니다.

지도 색칠 문제 (활동 13)를 떠올릴 수 있나요? 아이스크림 트럭 문제에서 “최소 지배 집합”이라고 공식적으로 부르는 문제는 매우 큰 수의 문제이며 정렬 문제인 지도 색칠 문제, 지도에서 최적 경로 찾기, 스케줄링 과정과 같은 것에서 지수 시간 알고리즘으로 그 정답을 찾는 방법을 알 수 없습니다. 놀랍게도 이러한 문제에서 정답의 하나를 찾을 수 있는 경우 모두 지수 시간 알고리즘으로 변환할 수 있다는 것이 밝혀졌습니다. 그래서 모두 해결되거나 하나도 해결되지 않는다고 이야기 합니다.

이러한 문제를 *NP-완전문제* 라고 합니다. NP는 “다항 시간 내 결정되지 않음(non-deterministic polynomial)”을 뜻합니다. 이 용어는 정답은 분명 있으나 정답을 찾기에 너무 오래 걸리는 문제를 가리킵니다. 즉, 한 번에 다수의 해결책을 시도할 수 있는 컴퓨터를 가지고 있다면 문제를 적절한 시간 내에 풀 수 있다는 것을 의미합니다. 하지만 이것은 매우 비현실적인 가정입니다. 이런 종류의 컴퓨터를 만든다는 것은 현재로서는 불가능합니다. 상상할 수 없을 정도로 빨라야 하기 때문입니다! 그러나 이런 컴퓨터의 개념의 원리는 중요합니다. NP-완전 문제는 비결정 컴퓨터 없이는 적절한 시간에 해결할 수 없기 때문입니다.

또한 *완전*이라고 부르는 그룹의 문제는 그 특징이 매우 다르지만 —예를 들어, 지도 색칠과 아이스크림 트럭 배치는 매우 다르며—그 중 하나를 해결하는 방법이 효율적인

방법으로 판명되면 다른 문제를 풀 때도 접근 가능한 방법이 됩니다. 이것을 일컬어 “모두해결되거나 전혀 안되거나”이라고 합니다.

NP-완전 문제의 유형은 수천 가지가 있고 컴퓨터과학자들은 수십 년 동안 이 성공하지 못한 문제의 효율적인 해결책을 찾기 위해 노력하고 있습니다. 효율적인 해결책이 하나만 발견되더라도 이것은 다른 NP 문제를 모두 해결하는 방법이 되는 것입니다. 이 때문에 더욱 강력하고 효율적인 해결책은 아직까지 없는 것으로 판단됩니다. 이러한 문제가 반드시 지수 시간이 걸릴 것이라고 증명하는 것은 컴퓨터 과학 이론과 수학에서 풀고 싶어하는 것 중의 하나입니다.

참고 문헌

Harel의 *Algorithmics*에서는 NP-완전 문제와 지수 시간 알고리즘에 대한 의문점이 있습니다. Dewdney의 *Turing Omnibus*에서도 NP-완전성을 다룹니다. 표준 컴퓨터 과학 교과서는 Garey & Johnson의 *Computers and Intractability*입니다. 이 책은 수백가지의 NP-완전 문제를 수록하고 있고 NP-완전성을 증명하는 기법들이 소개되어 있습니다. 그러나 내용이 어려워서 컴퓨터 과학을 전공하는 학생이나 그 분야의 전문가에게 적합합니다.

활동 16

빙판 위의 도로—슈타이너 트리

활동 개요

가끔씩은 사소하고 겉보기에도 중요하지 않은 변화로 인해서 문제를 해결하는데 큰 차이를 만들어 내기도 합니다. 이 활동은 진흙도시의 문제해결에서 네트워크의 차단거리를 찾는 문제와 매우 유사한듯 보입니다. 단지 진흙도시 문제와의 차이점은 진흙도시에서 가장 짧은 이동거리를 찾는 활동이었다면 이번 빙판길 문제에서는 이동거리(네트워크)에 새로운 점(노드)을 추가하여 새로운 이동 경로를 만들 수 있다는 것입니다. 결과적으로, 이 문제는 진흙도시보다 좀 더 어렵습니다만 복잡한 알고리즘과 관련이 있으며 가난한 지도제작자(활동 14)와 관광도시 (활동 15)의 문제와 비슷합니다.

교과 연계

- ✓ 수학 - 위치와 방향
- ✓ 수학 - 논리적 추론

필수기능

- ✓ 공간 시각화
- ✓ 기하학적 추론
- ✓ 알고리즘 절차 및 복잡성

대상 연령

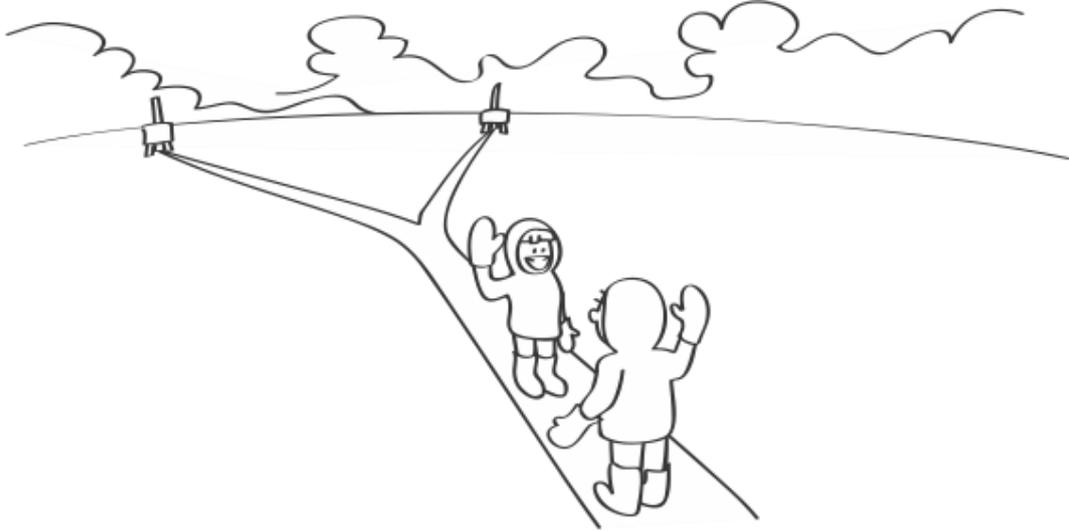
- ✓ 7 세 이상

수업 자료

모듬으로 구성하여 활동 권장

- ✓ 5 ~6 개의 못(텐트용 못 권장, 없을 때는 옷걸이를 잘라 구부려서 사용)
- ✓ 줄 또는 탄력성이 강한 물질로 만든 미터기 여러 개
- ✓ 자 또는 줄자
- ✓ 필기도구

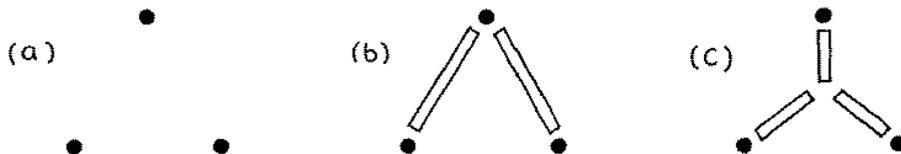
빙판 위의 도로



소개

이전 활동에서 여행자 도시는 매우 더운 나라에서 이루어졌습니다. 이번에는 반대의 곳으로 가봅시다. 캐나다의 북쪽에 위치한 한겨울 얼어붙은 호수에서 제설 차량은 대원들이 서로를 방문 할 수 있도록 가깝게 연결하는 도로를 만들어야 합니다. 하지만 땅이 얼고 날씨가 춥기 때문에 도로건설을 최소화 하고 싶어합니다. 여러분은 어느 곳에 도로를 만들어야 좋을지 확인하고 결정해야 합니다. 어떤 길로 만들지에는 제약은 없습니다. 고속도로는 눈 위에서 어디로든 갈 수 있습니다. 호수는 얼음으로 덮여 있고 이것은 모두 평평하게 되어 있어 어떤 방향으로든 도로를 만들 수 있습니다.

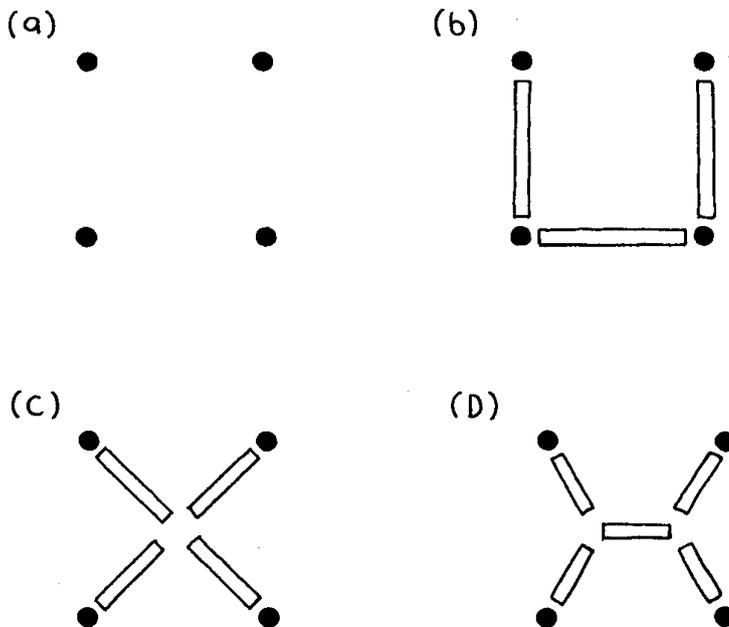
도로를 구부려 건설하는 것만으로도 불필요하게 길이가 증가하므로, 도로는 반드시 직선으로 만들어야 합니다. 모든 도로를 직선으로 연결하면 문제가 해결되는 것처럼 보이지만 생각보다 간단하지 않습니다. 얼어붙은 황무지에 교차로를 추가하는 것이 때때로 도로의 총 길이를 감소시키기 때문입니다. 건설 비용이 들기 때문에 한 곳에서 다른 곳으로 이동할 때 이동시간보다 도로의 총 길이가 중요합니다.



이 그림에서 (a)는 세 지점을 나타냅니다. 세 지점 중 한 지점을 다른 곳과 연결하면 ((b)와 같이) 도로망이 만들어 질 것입니다. 또 세 지점의 중심 어딘가에 교차점을 만들고, 그것을 세 지점에 연결한 것(c)처럼 만들 수도 있을 것입니다. 이것이 좋은 해결책인지는 도로의 전체길이를 측정해보면 정확하게 알 수 있습니다. 여러분은 가상의 새로운 교차로를 더 생각해 볼 수 있습니다. 추가적인 교차점에 대한 문제는 스위스의 수학자 제이콥 슈타이너(1796-1863)가 연구 한 이후로 “슈타이너 지점”라고 불려집니다. 그는 이러한 그래프 문제에서 새로운 지점을 도입함으로써 도로의 총 길이를 감소시킬 수 있다는 것을 알아낸 최초의 수학자입니다.

토론하기

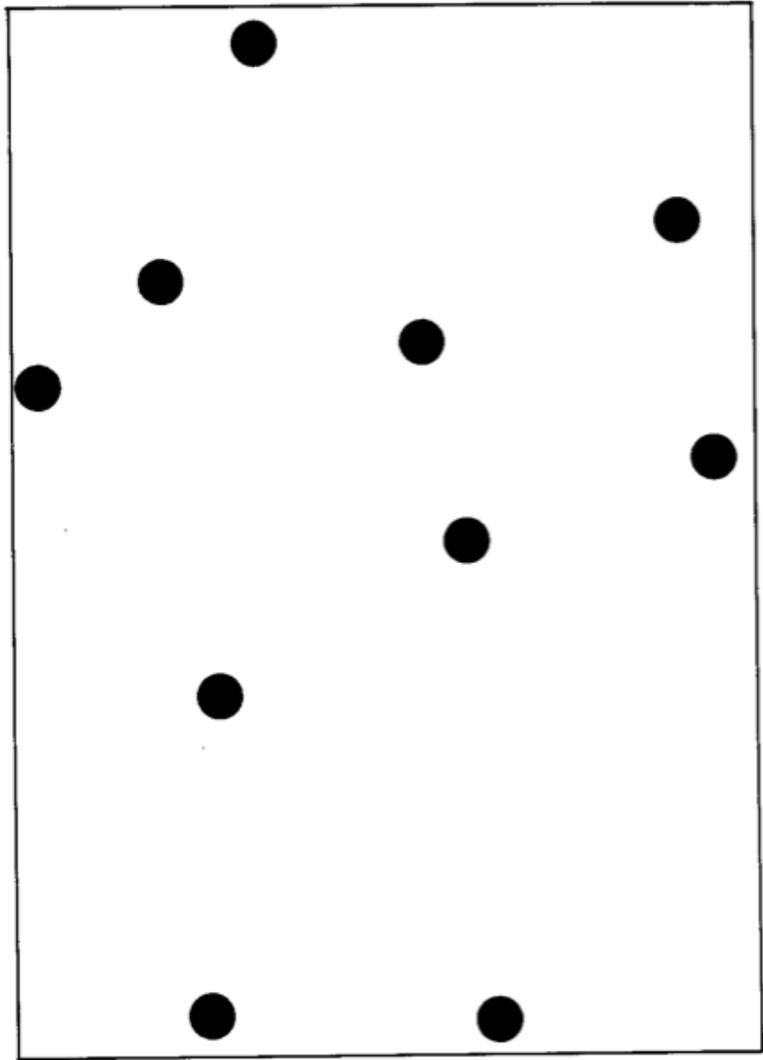
1. 학생들에게 수행할 문제를 설명하세요. 위의 예제를 인용하여, 세 개의 지점을 설정하고 총 도로의 길이를 감소시킬 수 있는 하나의 지점을 추가하여 문제를 해결할 수 있도록 제시하세요.



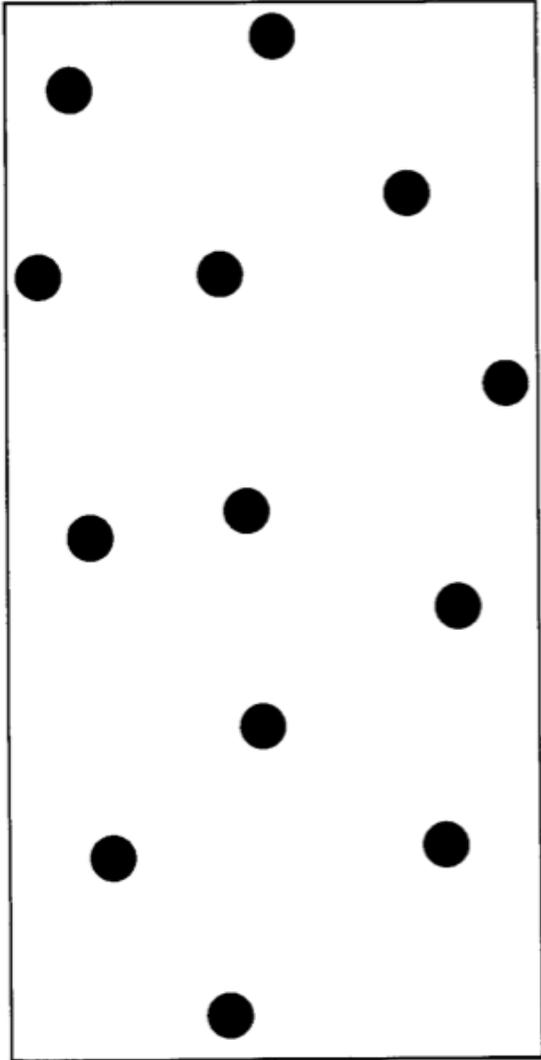
1. 학생들에게 (a)와 같이 네 개의 점을 사각형의 형태로 배열한 것을 제시합니다. 야외로 이동하여 각 그룹의 학생들에게 네 개의 깃발을 1m의 간격을 두고 사각형 모양으로 배치하도록 합니다.
2. 학생들에게 줄이나 탄성고무를 사용하여 4 개의 깃발에 연결하고 그 길이를 최소화했을 때 측정하고 기록하도록 합니다. 이번 단계에서 학생들은 그 어떤 ‘슈타이너 포인트’ 사용할 수 없습니다. 즉, 새로운 교차점을 만들지 않도록 합니다. (사각형의 세 지점을 연결하여 (b)와 같이 최소한의 길이를 달성했을 때 그 전체 길이는 3 미터입니다.)

3. 이제 학생들은 하나의 '슈타이너 포인트'를 사용하여 더 좋은 결과를 얻어낼 수 있습니다. (가장 좋은 '슈타이너 포인트'는 (c)와 같은 사각형의 중심에 위치하는 것입니다. 이 경우 전체 길이는 $2\sqrt{2} = 2.83$ 미터가 됩니다.) 다음으로 이제 두 개의 '슈타이너 포인트'를 사용하여 더 좋은 방법을 찾을 수 있습니다. (실제로 학생들은 (d)와 같이 두 개의 '슈타이너 포인트'를 사용하여 두 지점과 '슈타이너 포인트'가 만나는 각도를 120 도로 형성하여 전체 길이를 $1 + \sqrt{3} = 2.73$ 미터로 줄일 수 있습니다.) 이것을 수식을 계산하지 말고 그냥 끈의 길이를 재보면 알 수 있습니다.
4. 세 개의 '슈타이너 포인트'를 사용하면 학생들이 더 좋은 방법을 찾을 수 있을까요? (그렇지 않을 것입니다 - 두 개의 '슈타이너 포인트'를 사용하는 것이 제일 좋습니다. 더 많은 '슈타이너 포인트'를 사용한다고 해도 더 이상의 이점은 없습니다)
5. 학생들과 왜 이러한 문제가 힘들어 보이는지 토론해 봅시다. (그것은 아마도 '슈타이너 포인트'를 놓을 수 있는 곳이 너무 많기 때문에 아이들이 어디에 '슈타이너 포인트'를 위치시켜야 할지 모르기 때문입니다.)

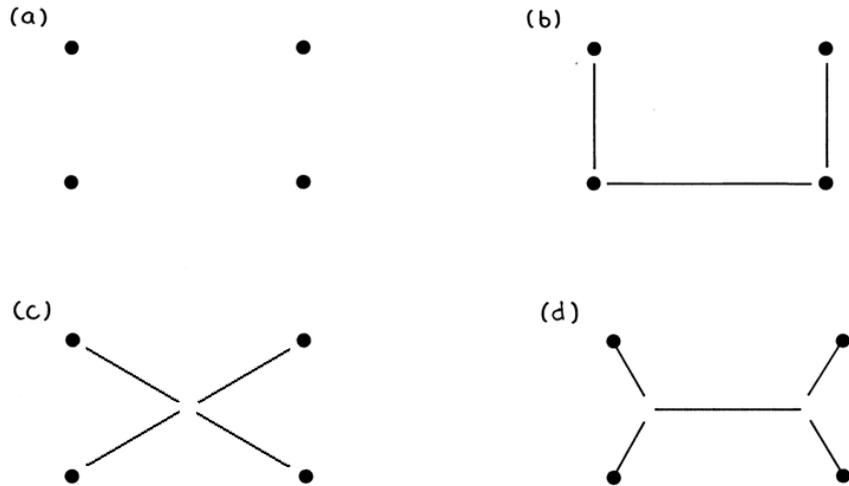
워크시트 활동 : 슈타이너 트리 예제 1



워크시트 활동 : 슈타이너 트리 예제 2



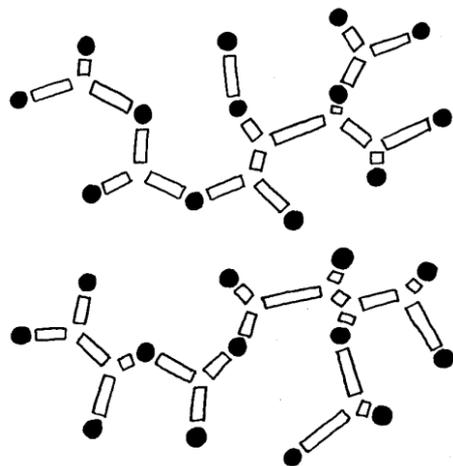
한 단계 더!



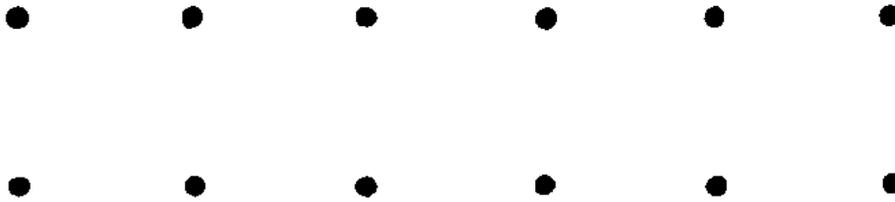
1. 기본활동을 빨리 끝낸 모둠을 위해서 흥미로운 실험을 준비합니다. (a)와 같이 세로 1 미터, 가로 2 미터의 직사각형 모양의 네 지점으로 하는 실험입니다. 학생들은 하나의 ‘슈타이너 포인트’를 추가하는 것은 길이를 더 늘어나게 하지만 두 개의 ‘슈타이너 포인트’를 사용하면 문제를 잘 해결할 수 있다는 것을 알아챌 것입니다. ((b)의 총 길이는 4 미터, (c)의 총 길이는 $2\sqrt{5} = 4.47$ 미터, (d)의 총 길이는 $2 + \sqrt{3} = 3.73$ 미터입니다) 그들은 하나의 ‘슈타이너 포인트’ 설정이 왜 정사각형과는 달리 직사각형에서 좋지 않은 결과를 나타내는지 알아낼 수 있을 것입니다. (그 이유는 정사각형이 직사각형으로 연장될 때, 늘어난 변의 길이는 (b) 나 (d)에 한번만 추가되지만, 대각선의 길이는 (c)와 같이 양쪽 선의 길이가 증가하기 때문입니다.)

2. 고학년들은 더 많은 점을 가진 문제를 해결할 수 있습니다. 빙판길을 연결하는 두 가지 지점의 레이아웃이 워크시트에 제시되어 있습니다. 그들은 워크시트의 복사본을 이용하거나, 투명 종이를 워크시트 위에 올린 후, 그 위에 펜을 사용하여 다양한 해결책을 시도해 볼 수 있습니다. 또 다른 방법은 압정을 사용하여 지도상에 표시하는 것입니다. 학생들은 자신의 방법으로 최단거리를 발견하였다고 생각할 때, 다른 학생들에게 결과를 알려줄 수 있습니다. (오른쪽 그림은 두 가지의 예를 들어 전체 길이가 비슷해 보이는 최소한의 해결책들을 보여줍니다.). 사실 그림에서도 나타나듯이, 이러한 두

예제 2 번에 적용 가능한 두 가지의 슈타이너 트리



가지의 비슷한 해결책이 있기 때문에 학생들은 문제를 어렵다고 생각하는 것입니다. - 지도 상에 '슈타이너 포인트'를 놓을 수 있는 위치는 무수히 많습니다

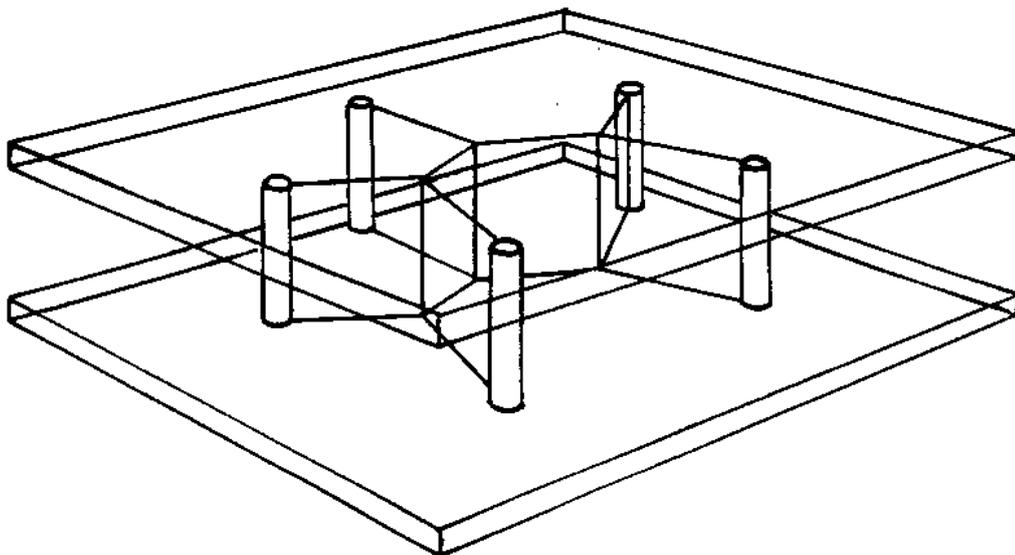


3. 이러한 사다리 연결망은 문제를 확장할 수 있는 또 다른 방법을 제공합니다.

4. 사다리 네트워크를 위한 최소한의 '슈타이너 트리'가 있습니다.

두 칸 사다리는 정사각형의 경우와 동일합니다. 그러나, 세 칸 사다리의 경우는 문제가 조금 달라집니다. - 예전 활동을 잘 기억해보면 해결방법을 발견할 수 있을 것입니다! 네 칸 사다리의 경우는 두 칸 사다리 두 개가 합쳐있는 것과 같게 생각하여 해결할 수 있습니다. 반면, 다섯 칸 사다리의 경우는 세 칸 사다리를 확장시킨 모습에서 해결책을 찾을 수 있습니다. 일반적으로, 사다리에서의 최소 '슈타이너 트리'의 모습은 사다리 칸이 짝수 또는 홀수인지에 달려있습니다. 만약 짝수일 경우, 몇 개의 두 칸 사다리가 연결되어 있다고 생각하면 됩니다. 반면 홀수일 경우, 세 칸 사다리 해결방법의 반복일 것입니다. 그러나 이러한 것을 증명하는 것은 쉬운 일이 아닙니다.

또 다른 흥미로운 활동은 슈타이너 비누거품 모델을 만드는 것입니다. 단단한 투명 플라스틱 두 장을 가지고, 밑의 그림에서 보여지듯이 확장 핀을 삽입하여 작업을 수행할 수 있습니다.



이제 비누 물에 전체를 담구어 봅시다. 그것이 나올 때, 아름다운 비누방울 표면이 슈타이너 트리 연결망의 핀을 연결하고 있다는 것을 알 수 있습니다.

그러나 불행하게도, 이것이 최소한의 슈타이너 트리는 아닙니다. 비누 방울 표면은 전체 길이를 최소화하는 구성을 찾을 수 있지만, 최소값은 단지 전체적인 것에서는 찾기 어렵고 일부분에서 찾을 수 있을 뿐입니다. 여기에는 총 길이의 최소값을 만들 수 있는 ‘슈타이너 포인트’를 놓는 완벽히 다른 방법이 있을 수 있습니다. 예를 들어, 하나의 경우에 비눗물 용액으로부터 추출 확장된 2의 첫 번째 구성 또는 또 다른 두 번째 구성처럼 보이는 비누 필름을 상상할 수 있습니다.

컴퓨터과학 이야기

우리가 이번장에서 살펴본 연결망은 최소 슈타이너 트리입니다. 이것들은 다시 되돌아서 순환되는 사이클이 없는데 실제 나무의 가지가 자라면 원래 나무의 가지로 붙어서 합쳐자라지 않기 때문에 “트리”라고 불러줍니다. “슈타이너”에 의해 제시된 슈타이너 포인트는 원래의 지점들과 추가할 수 있습니다. 그리고 그것들은 어떤 나무 가지의 지점에서 연결되어도 짧은 길이를 가지도록 제한되었기 때문에 “최소”라고 불러줍니다. 진흥도시 활동(활동 9)에서 우리는 전체 길이를 최소화하는 대다수의 지점을 연결하는 네트워크가 최소 신장 트리라는 것을 배웠습니다. 슈타이너 트리는 새로운 점이 도입될 수 있다는 것을 제외하고는 모두 동일한 구조를 가집니다.

최소 신장 트리를 찾기 위해 매우 효율적인 알고리즘(멀리 떨어져 연결되지 않은 지점들을 가깝게 연결하여 반복한 것)을 찾는 것에 반해 최소 슈타이너 트리 문제에 대해 효율적인 해결방법이 없는 것은 흥미롭습니다. 슈타이너 트리는 여백 중 어디에 남은 포인트를 둘 것인지 결정해야 하기 때문에 더 어렵습니다. 그러나 놀랍게도 사실 슈타이너 트리 문제가 어려운 것은 슈타이너 포인트를 위치시키는 결정에 있는 것이 아닌, 대략적으로 어디에 그들을 놓아두어야 하는지 결정하는데 있습니다. 예를 들어, 2 번 예시에서 보여지는 두 가지 해결책의 차이입니다. 새로운 포인트의 위치를 딱 하나라도 알게 되면 최적의 위치를 찾는 것은 비교적 간단해집니다. 비누 필름은 매우 효과적이고 정확하게 작업을 수행할 수 있습니다.

최소 슈타이너 트리를 이용하여 전화 사업에 큰 돈을 절약하게 해 주었다는 이야기가 있습니다. 1967 년, 미국에서는 기업들이 전화회사에서 회선을 임대하여 대규모 개인 전화망을 조직하였습니다. 그들이 지불하는 총 액수는 실제로 사용된 회선을 정확하게 계산한 것이 아닌, 전화망을 충족시킬만한 가장 짧은 네트워크에 기반을 두었습니다. 이러한 이유로 고객들은 전화회사들이 왕복 회선을 쓰고 있었기 때문에 여분의 돈을 지급할 필요가 없었습니다. 전화회사로서는 그만큼 손해가 되겠지요. 원래 알고리즘은 최소 신장 트리를 결정함으로써 얼마나 많은 노동력이 투입될지 계산이 됩니다. 그러나 1967 년경 세 개의 중심지의 중간지점에서 네 번째 중심지를 요청하면 네트워크 전체의 길이가 감소된다는 것이 민간 항공사에 의해 알려졌습니다. 만약 슈타이너 포인트에 전화 교환소를 추가로 설치한다면 전화회사는 요금을 절약할 수 있습니다. 비록 기존에 설치된 세개의 교환소만으로 운영되는 것보다 최소한의 슈타이너 트리가 오직 5% 또는 10% 짧게 되지만 여러 사람이 지속적으로 사용하여 큰 금액이 될 경우에는 훨씬 가치 있는 절약 방법이 됩니다. 슈타이너 트리 문제는 준비 지점을

연결하는 가장 짧은 네트워크를 찾는 것을 포함하기 때문에 종종 “가장 짧은 네트워크 문제”라고 불려집니다.

만약 앞의 두 활동(지도 제작자의 퍼즐과 관광도시)에서 어려움을 겪었다면 최소 슈타이너 트리 문제가 가장 어렵다라는 것에 대해 그리 놀라지 않을 것입니다. 지점의 수가 증가함에 따라, 슈타이너 포인트가 위치할 수 있는 곳은 기하급수적으로 증가합니다. 이것은 단순히 수행할 수 있는 최선의 기하급수적인 방법이든지, 발견되지 않은 다항식 시간 알고리즘이던지 하는 단순히 알려져 있지 않은 수천 개의 문제와는 다릅니다. 그러나 알려진 다항식의 시간 알고리즘이 이 문제를 해결하기 위해 발견된다면, 그것은 최소한의 지배 집합을 찾기 위한 다항식 시간 알고리즘이 될 수 있습니다.

N_p -완전문제는 이를 위한 다항식 알고리즘이 존재하며, 심지어 이러한 알고리즘은 꽤 느리지만 “다루기 쉬운 문제”로 불려집니다. 이러한 문제를 그들은 “다루기 힘들다”로 부르지는 않습니다. 왜냐하면 컴퓨터가 얼마나 빠르든지 또는 얼마나 많이 사용하는지와는 상관없이 문제크기가 점차 증가하고 그것은 수많은 시도를 통해 해결하는 것이 가능하지 않다는 것을 의미합니다. 미해결 문제 - 지도제작자의 퍼즐, 관광 도시, 슈타이너 트리 - 들이 다루기 쉬운지 아닌지의 여부는 아직 알려져 있지 않습니다. 그러나 대부분의 컴퓨터 과학자들은 NP-완전 문제에 대한 다항식 시간 알고리즘이 발견되지 않는 것에는 비관적입니다. 그리고 여기에 제공된 문제는 NP-완전 문제가 다루기 힘들다는 것에 대한 강력한 증거로써 간주됩니다.

여러분의 상관이 여러분에게 정답을 찾을 수 없는 문제를 최적의 해결하기 위해 효율적인 알고리즘을 고안하도록 요청할 때, 여러분은 무엇을 할 수 있을까요? 항공사는 네트워크 비용이 슈타이너 포인트를 도입함으로써 감소될 수 있다는 것을 확신했습니다. 만약 당신이 최적의 해결책을 마련 할 수 있는 효율적인 알고리즘을 개발하여 그 효과를 증명할 수 있다면 아마 최고의 컴퓨터과학자가 되겠지요. 먼 미래에 뛰어난 프로그래머가 이러한 문제를 해결할 수 있는 알고리즘을 발견해 낼 수도 있는 것처럼, 컴퓨터 과학에서 이같은 부정적인 인식을 바꾸어야 합니다. 여러분이 더 효율적인 알고리즘의 가능성이 없다고 절대적으로 말할 수는 없습니다. 만약 여러분의 문제가 NP-완전 문제라는 것을 보여줄 수 있다면, 그것은 실제로 연구소의 수천 명의 사람들이 문제에 대한 효율적인 답을 찾는 것에 많은 도전을 하게될 것입니다.

물론 현실에서 이러한 문제는 여전히 해결하여야 할 필요가 있고, 그 경우 사람들은 경험적으로 해결합니다. - 알고리즘은 최상의 해결책을 보장해주는 것이 아닌 최적의 매우 작은 부분 안에서 해결책을 제공해주는 것입니다. 경험적인 알고리즘은 매우 빠르고 최상의 해결책을 찾지 못하더라도 시간의 낭비와 비용의 낭비는 상당히 줄일 수

있습니다. 그래서 경험적인 방법을 사용하면 완벽한 답을 찾기 어려울 지라도 충분히 좋은 해결책을 찾을 수 있습니다. 네트워크를 만들거나 또는 도로를 이동할 때 약간 시간이 더 걸리거나 레이아웃이 조금 좋아 보이지 않더라도 그냥 가볍게 비용이 추가될 뿐 해결이 안되는 것은 아닙니다.

추가 읽기 자료

이 만화는 가레이와 존슨의 전통적인 책 ‘*Computers and Intractability*’를 기반으로 합니다.



“효율적인 알고리즘을 찾을 수 없는 내가 너무 바보같아.”

“이러한 알고리즘이 존재하지 않기 때문에 효율적인 알고리즘을 찾을 수 없습니다.”

“나는 효율적인 알고리즘을 찾을 수 없을 뿐 아니라 전문가들도 찾을 수 없습니다.”

당신이 효율적인 알고리즘을 찾을 수 없을 때 해야 할 일 - 세가지 가능성

1984년 6월, 사이언티픽 아메리칸의 “컴퓨터 레크레이션”에는 스파게티 컴퓨터 정렬, 그래프에서 최단 경로를 찾는 문자열의 고양이 침대, 숫자가 소수인지 아닌지 알려주는 빛과 거울장치, 문제 해결을 위한 아날로그 기기의 재미있는 설명과 함께, 비누 거품을 사용하여 슈타이너 트리를 만드는 방법에 대한 간략한 설명이 게시되었습니다. 이것들은 또한 Dewdney's *Turing Omnibus* 라는 아날로그 컴퓨터 란에도 나타났습니다.

Part V

비밀 공유와 해킹 보호 -암호화

비밀 공유와 해킹 보호

여러분은 영화나 소설에서 스파이가 비밀 코드를 사용해서 메시지를 바꾼다는 이야기를 종종 들어보았을 겁니다. ‘암호화’기법은 비밀코드를 쓰거나 해독하는 방법으로 시작하였습니다. 2 차 세계대전 때 군사코드를 해독하기 위해 특수 암호 해독 기계를 만들었습니다. 그때부터 컴퓨터는 암호를 해독하면서 발전하였고, 암호화는 전쟁 외에 새로운 분야에 적용되었습니다. 컴퓨터로 전에는 상상할 수 없었던 엄청난 양의 계산을 하게 됨으로서 코드 해독을 가능하게 하였습니다. 컴퓨터 시스템을 네트워크로 서로 연결하기 시작하면서 비밀코드가 새롭게 사용되었습니다. 컴퓨터가 네트워크에 연결되었을 때 누구나 접속가능하기 때문에 사람들의 정보를 보호하기 위해 필요하였기 때문입니다. 이메일을 받았을 경우, 그 메일을 보낸 사람이 정말 그 사람인지를 확인해야 하는 것이 중요하게 되었습니다. 지금은 전자상거래에서도 컴퓨터를 사용해 물건을 사고 팝니다. 그래서 우리는 컴퓨터 네트워크를 통해 주문을 하고 현금을 보내는데 암호가 필요해졌습니다. 그리고 컴퓨터 시스템을 공격하는 해킹이 늘어남에 따라 컴퓨터 보안은 점점 더 중요해지고 있습니다.

아마도 암호화기법은 여러분의 컴퓨터에 비밀 암호를 저장하게 하고, 다른 사람이 해독할 수 없도록 메시지와 뒤섞어 놓은 것일 수도 있습니다. 하지만 실제로는 매우 다릅니다. 현대 컴퓨터 시스템은 비밀 암호를 저장하지 않습니다. 왜냐하면 누군가가 시스템에 있는 암호를 해독할 수 있게 된다면 컴퓨터에 자기 마음대로 접속할 수 있게 되기 때문입니다. 이것은 큰 재앙이 됩니다. 가짜로 은행에 송금을 하고, 그 사람인척 메시지를 보내고, 모든 사람의 보안 파일을 읽게 되며 정부를 위협할 수도 있습니다. 요즘 비밀암호는 활동 14 에서 얘기했던 ‘일방향 함수’ 방법을 사용합니다. 암호는 그냥 메시지를 섞는 것만이 아닙니다. 진짜 어려운 문제- Part IV 에서 소개했던 난해성-와 같은 복잡한 기술이 사용됩니다.

암호를 사용하면서 생각했던 것이 불가능하다고 여겨질 수도 있습니다. 이번 장에서는 서로를 모르는 상태에서 평균 나이를 계산할 수 있는 간단한 방법을 알아보게 됩니다. 다른 도시에 떨어져 있어서 서로가 볼 수 없는 상황에서 동전 던지기를 하고 그 결과를

서로 믿게 하는 방법에 대해서도 알아봅니다. 그리고 모두가 해독하는 방법을 알고 있다 하더라도 암호화된 메시지를 자신만이 해독하는 방법도 알게 될 것입니다.

교사를 위한 팁

다음의 활동들은 현대 암호기술과 연관된 실제 경험들을 제공해줍니다. — 사람들이 암호와 컴퓨터를 떠올릴 때 생각하는 것과는 다른 점이 있습니다. 두 가지 핵심 아이디어가 있습니다. 첫번째는 트랜잭션을 대표하는 ‘프로토콜’입니다. 컴퓨터는 프로토콜을 많이 사용합니다. 겉으로 보기에 어려운 작업도 간단한 프로토콜에 의해 해결될 수 있습니다. 활동 16에서는 한 그룹의 사람들이 서로 협력해서 서로의 실제 나이도 모른채 그들의 평균 나이를 몇 분만에 쉽게 해결합니다. 두번째 핵심 아이디어는 서로가 컴퓨터로 상호작용하면서 놀이할 때 생길 수 있는 계산의 복잡성-난해성-의 해결방법입니다. 활동 17에서는 전화통화를 하는 두사람이 서로를 믿을 수 없는 상황에서 동전 던지기의 결과를 믿게 할 수 있는 방법을 보여줍니다. 활동 18에서는 암호 방법이 공개되었음에도 불구하고 비밀암호를 만들 수 있는 기술을 사용할 수 있는 방법을 보여줍니다.

이러한 활동 중 몇 개는-특히 마지막 활동-은 매우 어렵습니다. 학생들이 끝까지 문제를 해결할 수 있도록 동기부여를 해주어야 합니다. 그렇게 하면 대부분의 사람들이 불가능하다고 생각했던 문제를 해결해냈다는 성취감을 가질 수 있을 것입니다. 이러한 놀라운 경험을 학생들은 평생 잊지 못할 것입니다.

공학적 마인드 갖기

컴퓨터가 우리 생활과 밀접해짐에 따라 암호에 대해 선입견이나 왜곡된 생각을 가지는 경향이 있습니다. 대부분의 사람들은 현대 암호화 프로토콜에 대해 쉽게 생각하지 않습니다. 큰 기관-정부와 기업-에서는 개인 정보를 포함한 커다란 시스템을 구축할 때, 어떤 정보를 수집하고 유용하게 만들 때도 그 정보를 사용하기 때문입니다. 사람들이 현대 기술에 대해 좀 더 이해를 하게 된다면 정보수집 과정에 더 활발히 참여할 수 있게 될 것입니다.

이러한 정보 은닉 프로토콜, 암호화 프로토콜, 공개키 암호화 프로토콜과 같은 키 암호화 기법은 매우 발전된 것입니다. 하지만 아이디어 자체는 어렵지 않습니다. 기술적인 것이라서 기본 개념을 이해하는 것은 어렵지 않습니다. 전자상거래와 같은 실제 상황에서 이런 기술은 컴퓨터 소프트웨어에 숨어있습니다. 하지만 그 기술로 무엇을 할 수 있을지에 대한 통찰력을 갖기 위해서는 기초가 되는 핵심 아이디어에 대한 이해가 중요합니다.

암호 시스템은 통신 보안뿐만이 아니라 범죄에 사용될 수 있는 암호를 해독하기 위해 정부가 큰 관심을 쏟고 있습니다. 범죄자들이 암호를 사용하는 경우 그 암호를 해독할 수 없으면 안 되기 때문입니다. 한편으로는 암호화 시스템의 강도를 제한하거나, 시민들의 자유권, 개인정보에 대해 정부가 접근하는 것에 대한 불편함과 같은 법적 논쟁 거리를 만들기도 합니다.

한동안 미국 정부는 폭탄과 총 같은 군수품이 위험하게 사용되는 것을 막고자 암호화 방법을 사용하였습니다. 1 단계에서 '키 에스스로'라고 불리는 별도의 암호 시스템인 "클리퍼 칩"에 대해 논쟁이 있었습니다. 그 칩으로 어떤 메시지라도 해독할 수 있는 권한이 정부에 있었기 때문입니다. FBI와 미국정부는 이 칩을 통신에 널리 사용하기 원하였지만 개인정보에 상당히 위협적이라는 반대 의견을 받고 있다. 암호시스템의 모든 종류는 기술적으로 가능하다고는 하지만 정치적으로 무조건 허용될 필요는 없는 것이기 때문입니다.

암호화에 대한 생각들은 메시지 보안유지에 많이 적용되었습니다. 메시지가 정말 그사람이 보낸 것이 맞는지에 대한 확인을 하는 것에 필요한 것과 같은-인증-은 전자상거래에 꼭 필요합니다. 한 번 이상 투표하는 것을 방지하기 컴퓨터에 의해 사람들이 투표할 수 있도록 하는 방법도 있습니다. 그리고 전화로 카드놀이도 할 수 있습니다.

이러한 것들이 불가능하다고 여겨질 수도 있습니다. 전화기 끝에 있는 사람과 경쟁을 한다면 그 결과를 믿을 수 없을텐데 어떻게 카드놀이를 할 수 있을까요? 중간에서 누군가가 메시지를 가로챌 수도 있을텐데 말이죠. 만약 여러분이 그것을 할 수 없다면 전자적인 업무를 수행할 수 없게 됩니다. POS 단말기와 은행사이의 전화에 누군가가 가로채지 않도록 기술적인 범죄를 막아야 합니다. 거짓으로 주문하거나 허위 계약을

만들어 혼란을 야기시키는 비즈니스 경쟁도 예방해야 합니다. 현대 암호기술로 그러한 문제들을 해결할 수 있습니다.

암호와 암호해독에 관한 재미있는 책들이 많이 있습니다.

Hinsley and Stripp 이 쓴 *Codebreakers: the inside story of Bletchley Park* 는 2 차 대전에 사용되었던 암호해독과 그거이 전쟁 중의 많은 생명을 구했다는 얘기를 담고 있습니다.

활동 17

비밀 공유 - 정보 은닉 프로토콜

활동 개요

암호화 기술은 놀라울 정도로 높은 수준으로 개인 정보를 보호하면서, 다른 사람들과 정보를 공유할 수 있게 합니다. 이 활동은 그 어떤 것도 공개되지 않은 상태에서 정보가 공유되는 상황을 보여줍니다. (학생들은 누구도 자신의 나이를 공개할 필요 없이 평균 나이를 계산할 수 있을 것입니다)

교과 연계

- ✓ 수학 - 덧셈과 평균

필수 기능

- ✓ 평균 계산
- ✓ 난수
- ✓ 협동능력

대상 연령

- ✓ 7 세 이상

학습 자료

학생

- ✓ 작은 종이 패드
- ✓ 펜



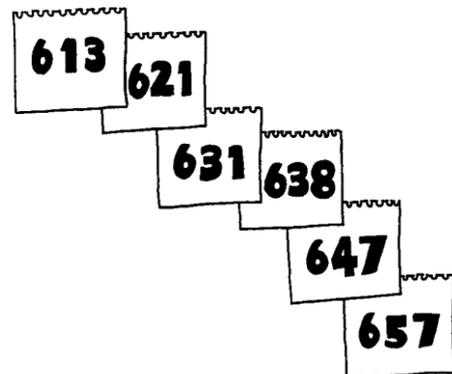
소개

이 활동은 그 누구도 자신의 나이를 공개하지 않은 상태로, 학생 그룹의 평균 나이를 찾는 활동입니다. 또한 그들의 평균 수입(용돈)이나 유사한 활동으로 평균 값을 계산할 수 있습니다. 어른들은 연령과 소득 같은 세부 사항에 더 민감할 수 있기 때문에 특히 이러한 통계 계산은 어른들과 함께 하는 것이 좋습니다.

한 그룹에 최소 세 명의 학생이 필요합니다.

생각해보기

1. 자신의 나이를 어느 누구에게도 공개하지 않은 상태로, 자신들의 평균 나이를 계산하고자 하는 그룹에 설명합니다.
2. 계산을 수행할 6 ~ 10 명의 학생을 선택하세요. 첫 번째 학생에게 패드와 팬을 주고 비밀리에 용지 뒷면 시트에 임의로 선택된 세 자리 숫자를 기록하도록 요청하세요. 예를 들어 613 이라는 임의의 수를 선택하였다고 합시다.
3. 첫 번째 학생이 첫 페이지를 찢어서 가지고 있고, 임의의 숫자에 자신의 나이를 더한 후 패드의 두 번째 시트 위에 그것을 기록하세요. 첫 번째 학생의 나이가 8 살이라면 두 번째 시트는 621 이라는 수를 보여줄 것입니다. 그들은 여전히 찢어진 페이지를 가지고 있어야 합니다. (그리고 누구에게도 보여주어서는 안됩니다.)
4. 패드는 이제 두 번째 학생에게 전달됩니다. 두 번째 학생은 상단에 있는 번호에 자신의 나이를 추가시킨 후 두 번째 장을 찢어서 가진 후, 자신의 나이를 추가시킨 숫자를 다음 장에 기록합니다. 예를 들어 두 번째 학생의 나이는 10 살입니다.
5. 모든 학생들이 패드를 가질 때까지, 학생들이 건네 받은 패드에서 제일 윗 장을 찢어 가진 후, 자신의 나이를 추가시켜 다음 장에 적는 과정이 계속됩니다.
6. 패드가 첫 번째 학생에게 돌아옵니다. 첫 번째 학생은 패드에 적힌 숫자에서 자신이 알고 있던 원래 임의의 수를 빼서 가집니다. 예를 들어 5 명의 학생을 거친 후 최종



숫자가 657 이라면, 첫 번째 학생은 자신의 처음 정한 임의의 수 613 을 빼서 44 를 전달할 것입니다. 이 숫자는 학생들 나이의 총 합입니다. 그리고 그들의 평균 나이는 총 합에서 학생 수를 나눈 값이 됩니다. 따라서 그 그룹의 평균 연령은 8.8 세가 됩니다.

7. 학생들에게 가지고 있던 종이를 모두 없애도록 지시하고 두 명의 학생이 협력하기로 결심하지 않는 한, 그 누구도 개인의 나이를 알아낼 수는 없습니다.

한 단계 업!

이 시스템은 각각의 사람들이 아니라고 투표하면 0 을, 맞다고 투표하면 1 을 가짐으로써 비밀투표를 허용하도록 구성됩니다. 물론 만약 모두가 맞다고 응답했을 때 누군가가 하나 이상을 추가하면(또는 0 보다 낮은 수) 투표한 사람의 수보다 맞다고 응답한 것이 높기 때문에 투표는 공정하지 않게 됩니다.

컴퓨터과학 이야기

컴퓨터에 저장된 자료에는 개인정보가 많습니다. 은행 잔고, 소셜 네트워크, 빛이나 세금, 운전 면허증, 신용 기록, 검사 결과, 의료 기록 등의 개인 정보는 매우 중요합니다! 그러나 우리는 다른 사람들과 그러한 정보를 공유해야 할 때가 있습니다. 예를 들어 우리가 은행 카드를 사용하여 상점에서 물건을 구입 할 때, 우리는 물건을 살 충분한 잔액이 있는지 확인해야 하므로 은행 정보에 접속합니다.

종종 우리는 우리가 실제로 필요한 것보다 더 많은 정보를 제공합니다. 예를 들어 우리가 상점에서 전자 거래를 수행하는 경우, 그들은 기본적으로 우리의 이름, 계좌번호, 은행을 알아야 합니다. 게다가 은행은 우리가 쇼핑했던 곳을 알아낼 수 있습니다. 은행은 개인이 가스나 식료품을 사는 것을 모니터링 하여 이런 장소에 방문할 때, 품목 구입에 얼마나 많은 돈을 지불하는지에 대한 프로필을 만들 수 있습니다. 만약 우리가 현금으로 금액을 지불 한 경우에는 이 정보 중 어떤 것도 공개되지 않을 것입니다. 대부분의 사람들은 이런 정보를 공유하는 것에 대해 많은 걱정을 하지 않을 것이지만, 이러한 것은 타겟 마케팅(예를 들어 항공권에 많은 돈을 지불한 사람에게 항공 광고를 전송하는 것), 차별화(우수 고객에게만 더 나은 서비스를 제공하는 것), 심지어 협박(부끄러운 거래에 대한 세부 사항을 공개하겠다고 위협)에 남용 될 가능성이 있습니다. 다른 건 몰라도, 누군가가 자신을 관찰하고 있다고 생각하면 사람들은 쇼핑하는 방식을 변경할 수 있습니다.

개인 정보의 손실은 다양한 경우에서 이루어지고 있습니다. 아직 암호화 프로토콜은 우리가 현금을 사용하는 것과 같은 동일한 수준의 보안성을 가지고 전자 거래를 할 수 있도록 허용해 줍니다. 우리의 돈이 은행 계좌에서 상점의 계좌로 어디서 오고 가는지 아무도 모른 상태에서 전송될 수 있다고 생각하기 어려울 수 있습니다. 이 활동은 그러한 거래가 좀더 그럴 듯 하게 보이게 합니다. (두 상황 다 정보 공유의 가능성을 제한하고, 이것은 영리한 프로토콜에 의해 가능하게 될 수 있습니다.)

읽기자료

“Security without identification: transaction systems to make Big Brother obsolete.” 라는 도발적인 제목으로 David Chaum 가 이러한 문제를 제시한 논문을 읽어보세요. 이 논문은 꽤 읽기 편하고, 정보 숨기기 프로토콜에 대한 간단한 예제를 제공하고, 어떻게 하면 “전자 캐시” 사용을 완전히 개인적으로 할 수 있을지에 대한 것을 포함하고 있습니다. 이 논문은 1985 년 10 월에 발행된 *Communications of the ACM*에서 찾을 수 있습니다.

활동 18

페루동전 뒤집기—암호화 프로토콜

활동 개요

이 활동은 서로를 신뢰할 수 없는 상태인 전화로만 연결된 두 사람이 간단한 작업을 수행하는 것을 보여줍니다. 신뢰가 불가능한 상황에서 동전 뒤집기와 같은 활동을 하는 경우 정보를 어떻게 전달할 수 있는지 알아봅니다.

교과 연계

- ✓ 수학 - 논리적 추론
- ✓ 수학- 불 연산

필수 기능

- ✓ 불 연산
- ✓ 함수
- ✓ 퍼즐 해결

대상 연령

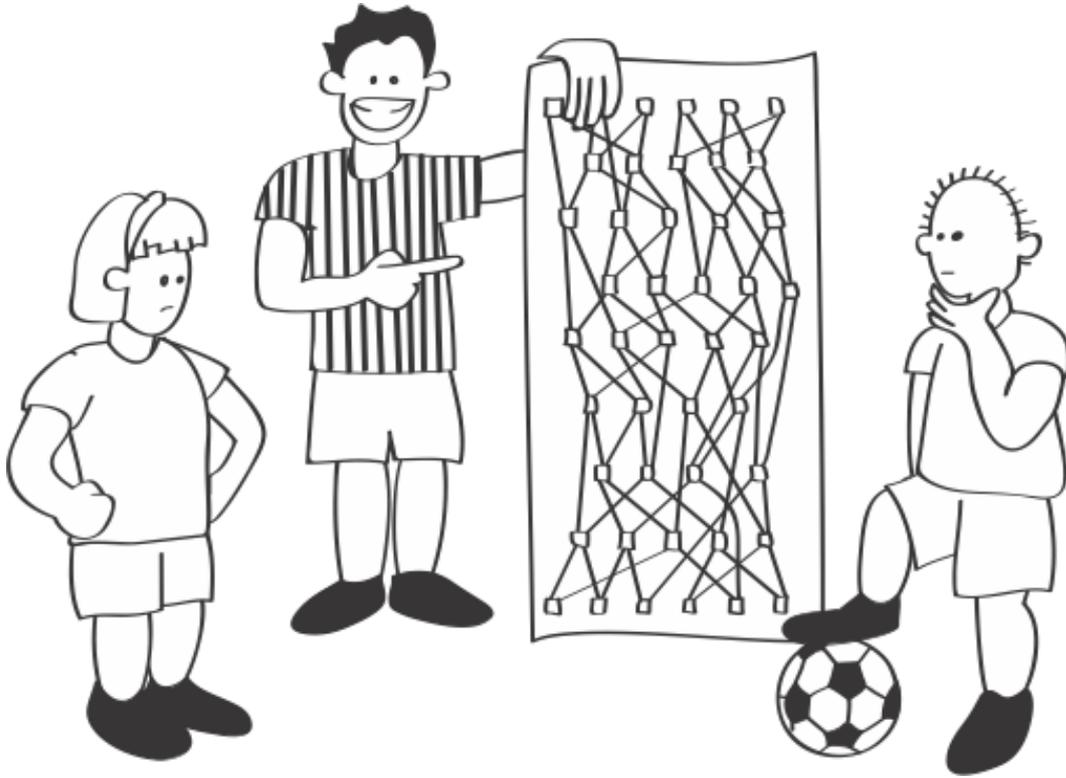
- ✓ 9 세 이상

수업 자료

학생

- ✓ 페루 동전 뒤집기 종이의 복사본 한 장
- ✓ 24 개 정도의 작은 버튼 또는 두가지 다른 색상의 (보드게임에서 쓰는)패

페루 동전 뒤집기



소개

이 활동을 만든 학생 중 한명이 페루에서 온 학생이었기 때문에 그 이름을 따서 제목이 고안되었습니다. 여러분이 살고 있는 지역 상황에 맞게 사용자를 설정할 수 있습니다. 리마와 쿠즈코의 축구팀은 결승전에서 어떤 팀이 홈 구장을 사용할 지에 대해 결정해야 했습니다. 가장 간단한 방법은 동전 뒤집기를 하는 것이었죠. 그러나 리마의 주장 엘리시아, 쿠즈코의 주장 베니토가 멀리 떨어져 있어서 함께 모여 동전을 던질 시간이 없었습니다. 그들이 전화를 통해 그것을 할 수 있을까요? 엘리시아가 동전을 뒤집고 제니토가 앞면인지 뒷면인지 말 한다고 생각해 봅시다. 베니토가 앞면이라고 할 경우 엘리시아는 “안타깝네, 뒷면이야” 라고 말해도 알아차릴 수 없기 때문에 이렇게는 할 수 없습니다.

결국 엘리시아가 거짓말쟁이가 아니라고 하더라도 중요한 대회나 경쟁에서는 이러한 유혹이 엄청 강하게 작용할 것입니다. 만약 엘리시아가 진실을 말해서 베니토가 졌을 때, 과연 그것을 믿을까요?

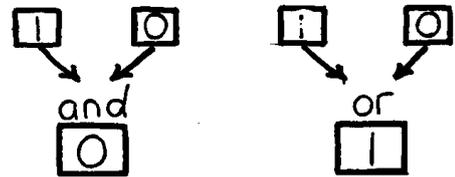
학생들이 이진수 표현(활동 1의 점세기), 비율의 개념(활동 4의 카드 뒤집기 마술), 활동 15 여행자 도시에서의 일방향 함수 예제를 배운 경우에는 이 활동에서 좀더 많은 개념을 배우게 될 것입니다.

이렇게 활동합니다. 아래에 설명된 것과 같이 그들은 *and-gates* 와 *or-gates* 로 구성된 회로를 설계합니다. 원칙적으로 그들은 전화를 통해서 작업을 수행할 수 있습니다. 비록 일반적으로 인정하듯이 회로는 다른 부정 행위를 할 수 없도록 충분히 복잡하게 설계되는 것을 보장해야 하기 때문에 실제로는 활동 과정이 조금 지루할 수 있습니다. (이메일로 하는 작업도 마찬가지 일 것 입니다) 마지막 회로는 모두가 알고 있게 됩니다.

생각해보기

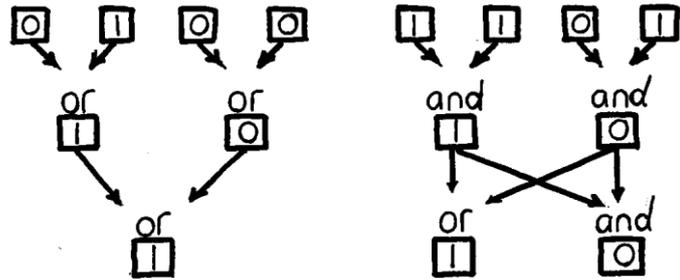
and-gates 와 *or-gates* 의 법칙은 간단합니다.

각각의 “gate”는 두 개의 입력장치와 하나의 출력장치를 가지고 있습니다. 입력장치는 0 또는 1 을 입력할 수 있고 각각 정답과



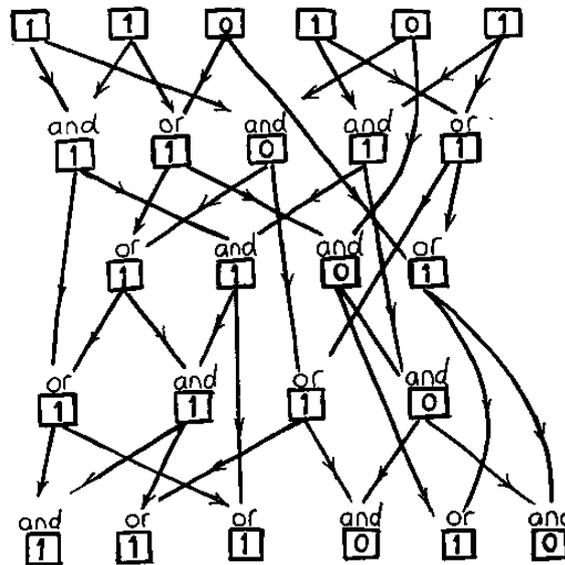
오답으로 해석될 수 있습니다. *and-gates* 출력장치는 입력 장치 모두가 1 을 입력하였을 때만 정답(1)으로 출력되고, 다른 경우에는 모두 오답(0)으로 출력됩니다. 예를 들어 *and-gates* 입력장치 한군데에는 1 을 입력하고 다른 곳에는 0 을 입력하면 출력장치에는 오답(0)으로 표시되게 됩니다. *or-gates* 출력장치는 입력장치 중 한군데에만 1 이 있으면 출력장치에 정답(1)이 출력되며 입력장치 모두에 0 이 입력되어야 출력장치에 오답(0)이 출력됩니다. 따라서 0 과 1 이 입력된 *or-gate* 출력장치에는 1 이 출력됩니다.

하나의 게이트 출력장치는 더 복잡한 효과를 생성하기 위해 다른 입력장치(또는 몇 개의 다른 장치)와 연결 될 수 있습니다. 예를 들어 그림에서 왼쪽 회로 장치는



2 개의 *or-gates* 의 출력장치가 다시 입력장치가 되어 세 번째 *or-gate* 가 된 것을 알 수 있습니다. 이 경우, 4 개의 입력장치 중 하나에만 1 이 입력되어도 출력은 1 이 될 것입니다. 오른쪽 회로장치는 출력에서 두 개의 값을 가지도록 상위에 두 개의 *and-gates* 가 연결되어 있고 각 게이트의 출력장치는 아래 2 개의 게이트의 입력장치가 됩니다.

페루 동전을 던지기 문제를 해결하기 위해서 우리는 훨씬 더 복잡한 회로가 필요합니다. 워크시트의 회로는 여섯 개의 입력장치와 여섯 개의 출력장치가 있습니다. 여기서 특정



입력 값에 대한 예제를 해봅시다.

이 회로는 다음과 같이 전화로 동전던지기를 할 때 사용할 수 있는 방법입니다. 엘리시아가 6 개의 입력회로에(0 또는 1 을 입력할 수 있는) 입력할 수를 선택하고 비밀을 유지합니다. 그녀는 6 개의 입력회로에 0 또는 1 을 입력한 후 베니토에게 출력된 6 비트 수를 전달합니다. 베니토가 출력된 내용을 받으면 그는 엘리시아가 짝수 또는 홀수 아니면 다른 단어를 입력하였는지 추측하려고 노력해야 합니다. 만약 회로가 충분히 복잡하다면 베니토는 대답할 수 없고 그의 추측은 임의의 선택이 될 것입니다. (사실 심지어 그는 동전을 던져 선택할 것입니다!) 만약 그가 정답을 맞춘다면 베니토가 이기고 플레이오프는 쿠즈코에서 열릴 것입니다. 만약 베니토의 추측이 틀린다면 엘리시아가 이기고 플레이오프는 리마에서 열리게 되겠죠. 베니토가 엘리시아에게 자신의 추측을 이야기하면 엘리시아는 베니토가 확인할 수 있도록 자신이 입력한 것을 보여줍니다.

1. 소규모 그룹으로 학생들을 나누고 각 그룹에 회로와 일부 패를 주고 이야기를 설명합니다. 학생들이 학교 스포츠 클럽의 주장이 되어 라이벌 학교와 동전던지기를 하는 상황을 가정하면 더욱 의미가 있을 것입니다. 패의 색상에 대한 규칙을 설정하고(빨간색이 0, 파란색은 1, 기타 등등) 학생들이 그것을 기억할 수 있도록 상단에 있는 범례에 표시하게 합니다.

2. 학생들에게 앨리시아가 선택한 숫자를 보여주기 위해 입력장치에 패를 어떻게 배치해야 하는지 보여줍니다. 이어서 *and-gates* 와 *or-gates* 규칙을 설명해주고 종이의 밑부분에 메모하게 합니다. (학생들이 선택한 색도 고려합니다)
3. 마디마다 패를 배치하고 해당 출력을 유도하도록 회로를 작동하는 방법을 보여줍니다. 이것을 정확하게 수행할 수 있도록 해야 합니다. 아래의 표는 (학생들에게는 주지 않는 것이 좋습니다) 미심쩍은 경우에 참고할 수 있는 입력에 대한 출력을 보여줍니다.

입력	000000	000001	000010	000011	000100	000101	000110	000111
출력	000000	010010	000000	010010	010010	010010	010010	010010
입력	001000	001001	001010	001011	001100	001101	001110	001111
출력	001010	011010	001010	011010	011010	011010	011010	011111
입력	010000	010001	010010	010011	010100	010101	010110	010111
출력	001000	011010	001010	011010	011010	011010	011010	011111
입력	011000	011001	011010	011011	011100	011101	011110	011111
출력	001010	011010	001010	011010	011010	011010	011010	011111
입력	100000	100001	100010	100011	100100	100101	100110	100111
출력	000000	010010	011000	011010	010010	010010	011010	011010
입력	101000	101001	101010	101011	101100	101101	101110	101111
출력	001010	011010	011010	011010	011010	011010	011010	011111
입력	110000	110001	110010	110011	110100	110101	110110	110111
출력	001000	011010	011010	011010	011010	111010	011010	111111
입력	111000	111001	111010	111011	111100	111101	111110	111111
출력	001010	011010	011010	011010	011010	111010	011010	111111

4. 이제 각각의 그룹에서 앨리시아와 베니토를 선택해야 합니다. 그룹을 반으로 나누고 각자 알리시아와 베니토를 선택하게 하세요. 앨리시아는 입력 회로에 임의의 수를 입력하고 출력을 계산하여 베니토에게 알려주어야 합니다. 베니토는 입력된 수의 배열(그것이 홀수인지 짝수인지)을 추측하여야 합니다. 이 과정에서 베니토의 추측은 본질적으로 임의의 것이 되어야만 합니다. 그 후 앨리시아는 자신이 입력한 값이 무엇인지 모두에게 알려줍니다. 베니토가 올바른 추측을

하였을 때는 베니토의 승리가 됩니다. 베니토는 출력된 값을 확인하여 엘리시아가 입력한 값을 바꾸지 않았는지 확인할 수 있습니다.

그리고 이 시점에서 동전던지기가 완성됩니다.

만약 베니토가 입력 과정을 찾을 수 있다면 속임수 또한 찾을 수 있습니다. 따라서 엘리시아의 이익을 보장하기 위해 회로의 기능이 일방향 함수로 구성되어 있으며, 이것은 활동 14 에서 논의된 것처럼 베니토의 부정을 방지할 수 있습니다. 일방향 함수는 입력 값을 알고 있을 경우, 출력을 쉽게 계산할 수 있지만 출력 값을 알고 있을 때, 입력 값을 알기란 매우 어렵습니다.

만약 엘리시아가 반대되는 두 개의 패러티를 생성하여 같은 출력물을 얻을 수 있다면 속임수를 쓸 수 있을 것입니다. 그렇다면 베니토가 어떤 추측을 하던 엘리시아가 입력물을 보여주었을 때, 그는 틀린 정답을 말할 것입니다. 따라서, 베니토는 어떤 입력값에도 회로가 동일한 출력을 가지지 않도록 주의를 기울일 것입니다.

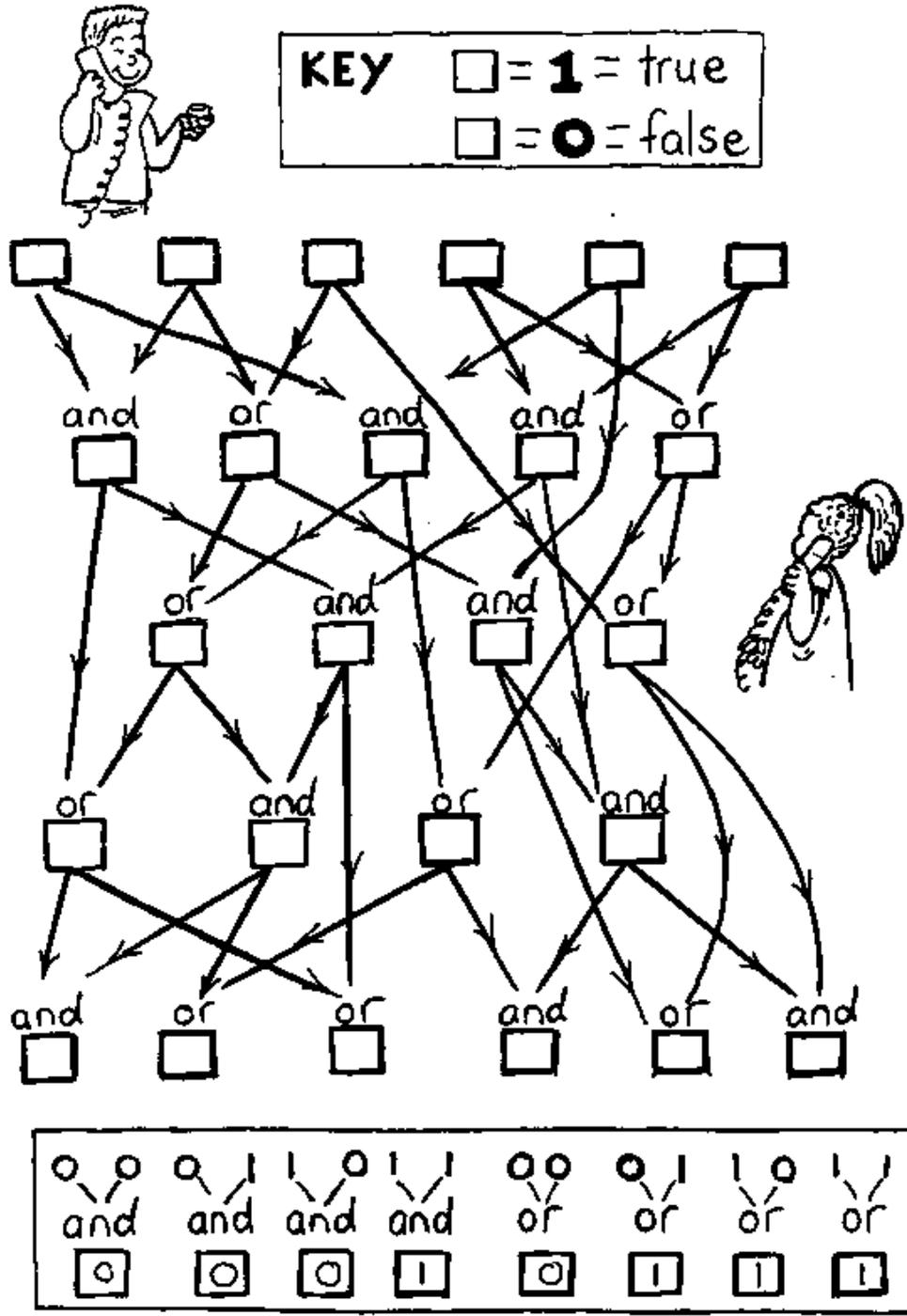
5. 학생들이 엘리시아나 베니토가 부정 행위를 할 수 있는 방법을 찾을 수 있는지 확인합니다. 표의 첫 번째 행에서 여러 다른 입력은 010010 이라는 출력을 생성할 수 있습니다.(예를 들어 000001, 000011, 000101 등등) 따라서 만약 엘리시아가 출력값이 010010 이라고 선언한다면 베니토가 짝수를 택했을 때는 입력값으로 000001 을, 홀수를 택했을 때는 입력값으로 000011 을 선택할 수 있습니다.

이 회로에서 베니토가 속임수를 쓰기는 매우 힘듭니다. 출력값이 011000 인 경우에 입력값은 반드시 100010 이 되어야 합니다. (표를 통해 확인할 수 있듯이 다른 가능성은 없습니다.) 따라서 이러한 입력 값을 엘리시아가 가질 경우, 베니토는 짝수인 것을 추측하고 확신할 수 있습니다. 컴퓨터 기반 시스템은 더 많은 비트를 사용하고, 그래서 시도할 수 있는 너무 많은 가능성이 있을 것입니다. (각각의 비트는 가능성을 두 배로 만듭니다)

6. 이제 각 학생 그룹에게 이 게임에 대한 자신만의 회로를 고안하도록 하세요. 엘리시아가 속임수를 쉽게 쓸 수 있는지, 베니토가 속임수를 쉽게 쓸 수 있는지 확인하도록 합니다. 회로의 입력장치와 출력장치의 수는 달라질 수 있습니다. 꼭 6 개의 입력장치를 가지고 있을 필요는 없습니다.

워크시트 활동: 페루동전 뒤집기

이 회로에 몇 가지 임의의 값을 입력하고 출력 값을 확인해 보세요.

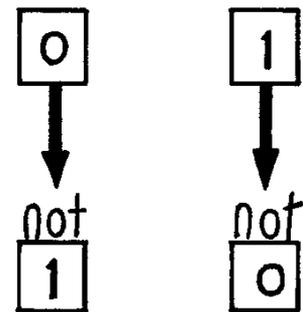


한 단계 더!

1. 실제로 명백한 문제는 앨리시아와 베니토 모두 허용 가능한 회로를 구성하는데 협력이 필요하다는 것입니다. 이것은 아이들에게 재미있는 활동이 될 수 있지만, 실제로 협력이 가능할지는 분명하지 않습니다. (특히 전화로 하는 경우에!) 그러나 앨리시아와 베니토가 독립적으로 회로를 구성하고 공개적으로 사용할 수 있는 간단한 대안이 있습니다. 앨리시아가 양쪽 회로에 자신이 생각한 수를 입력하고, 두 가지의 출력물을 비교하여 두 수가 똑같으면 1로 그렇지 않으면 0으로 출력합니다. 두 개의 일방향 함수가 결합된 것 또한 일방향 함수이기 때문에 이러한 경우에 두 명의 참가자 모두 다른 사람이 개입하지 않으면 속임수를 쓰기 힘듭니다.

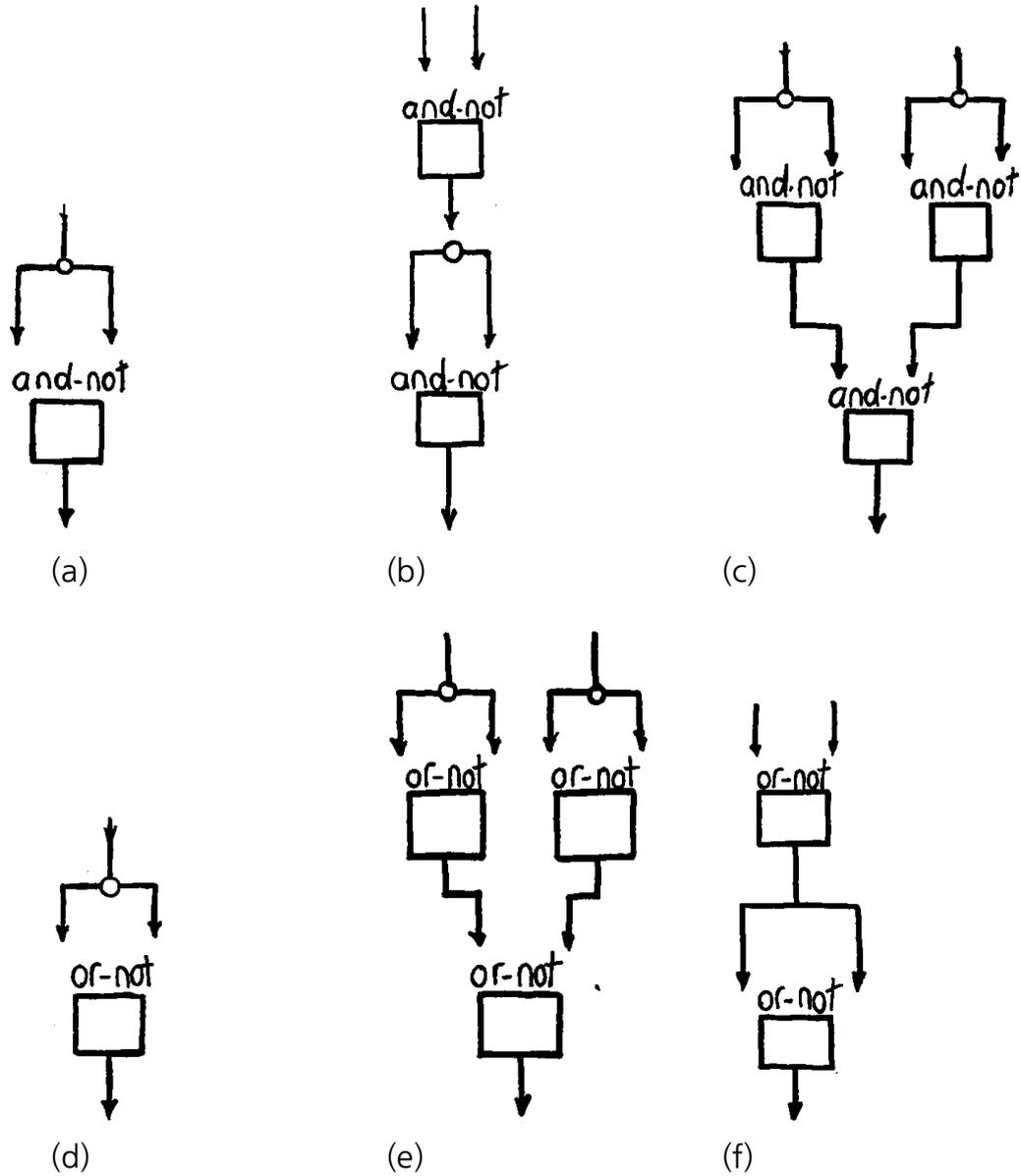
다음 두 가지 변화는 암호화 프로토콜이나 동전던지기 문제 자체와는 관계를 가지고 있지 않지만, 오히려 회로의 아이디어는 and gate 와 or gate 에서 구성되었습니다. 그들은 컴퓨터 회로뿐 아니라 논리 그 자체에 대한 기초적인 중요한 생각도 깊게 탐구하였습니다. 이러한 논리는 수학자 조지 부울(1815-64)의 이름을 따서 부울 대수라고 이름 붙여졌습니다.

2. 학생들은 000000 처럼 모두 0을 입력하는 것이 0의 출력물을, 반면 111111 처럼 모두 1을 입력하는 것이 1의 출력물을 생성한다는 것을 알 수 있을 것입니다. (뿐만 아니라 이러한 출력을 생성하는 다른 입력 값이 있을 수 있습니다. 예시회로에 000010을 입력하면 출력값은 모두 0입니다. 반면에 110111을 입력하면 출력물은 모두 1입니다.) 이것은 and 및 or gate 로 회로가 구성되어 있다는 사실의 결과입니다. not-gate 를 추가함으로써, 하나의 입력 값이 반전된다는 것을 알 수 있습니다($0 \rightarrow 1$ 그리고 $1 \rightarrow 0$), 학생들은 이 속성이 없는 회로를 구성할 수 있습니다.



3. 두 가지 중요한 게이트의 종류에는 and 와 or 게이트와 비슷하지만 not 이 뒤따르는 and-not 또는 or-not(일반적인 약어로 각각 nand 와 nor)게이트가 있습니다. 따라서 a 는 b 가 아닙니다(a 와 b). 언제나 이러한 회로는 항상 대응하는 and 또는 or에 not 이 따라와 게이트가 얻어질 수 있기 때문에 기능상 서로 다른 회로를 달성할 수 없습니다. 그러나 다른 모든 게이트 유형과 not-게이트에서 할 수 있는 흥미로운 속성이 있습니다.

*and-not*과 *or-not*이 소개된 후, 학생들은 다른 게이트들이 서로 연결되어 새로운 게이트를 만들 수 있는지 도전하게 됩니다. 이것은 서로 연결된 게이트의 한 형태로 만들어 질 수 있는 경우입니다. 아래 그림에서는 세가지 기본 게이트를 보여줍니다. Not, and, or 게이트가 서로 연결되어 상단 행에는 *and-not* 게이트, 하단 행에는 *or-not* 게이트가 구성되었습니다.



컴퓨터과학 이야기

최근 몇 년 동안 컴퓨터 네트워크를 통해 수행된 상거래의 양은 큰 증가를 보였습니다. 그리고 전자 자금, 비밀 거래, 안전한 교환, 서명, 법적 구속력, 문서에 대해 안전한 교환을 보장하는 것은 필수적이 되었습니다. 암호작성술의 주제는 안전하고 개인적인 통신방법에 관한 것이었습니다. 몇 년 전, 컴퓨터 과학 연구자들은 특정 정보가 누구에게나 공개되면서도 그 비밀을 보장 할 수 있는 알고리즘을 발견했습니다. 그 알고리즘은 활동 18의 소위 “공개 키 암호 방식”라고 불려지고, 이는 현재 안전하게 정보를 교환하는 주된 방법으로 사용되고 있습니다. 예를 들어, 여러분은 웹 브라우저에 SSL(보안 소켓 계층)또는 TLS(전송 계층 보안) 등의 설정을 할 수 있습니다. 이러한 시스템은 은행 같은 웹 사이트에 대한 보안 연결을 설정하는 웹 브라우저를 사용가능한 시스템의 공개 키 시스템을 기반으로 합니다. 비록 누군가가 모든 송신 데이터를 도청해서 볼 수 있더라도 말입니다.

암호화 기법은 단순히 비밀 유지에 대한 것이 아니라, 다른 사람이 찾을 수 있는 정보를 제한하고, 지리적으로 떨어져있는 사람들 사이에 신뢰관계를 구축하는 것입니다. 공식적인 규칙이나 암호화 거래를 위한 “프로토콜”은 위조 불가능한 디지털 서명과 당신이 가진 비밀(비밀번호 같은)을 실제로 공개하지 않고 다른 사람에게 말할 수 있는 능력 등 불가능 해 보이는 일을 할 수 있도록 고안되었습니다. 전화로 동전 뒤집기도 간단해 보이지만 이것과 원리는 유사한 문제입니다.

실제 상황에서, 엘리시아와 베니토는 회로 자체를 설계하는 것이 아니라, 내부적으로 작업을 수행하는 컴퓨터 프로그램을 얻게 될 것 입니다. 아마도 어느 한쪽에 설치된 소프트웨어에 관심이 있을 수도 있습니다. 그러나 모두는 얼마나 컴퓨터 기술이 좋은지, 그것을 다루기 힘든지와는 상관 없이 결정에 따른 결과가 영향을 끼칠 수 없다는 것을 확인받고 싶을 것입니다.

원칙적으로 모든 분쟁은 판사에 의해 해결되어야 합니다. 판사의 기능에는 엘리시아가 처음 생각했던 이진수에 의한 출력물이 베니토에게 전해지고, 베니토가 추측한 것이 되돌아오는 회로를 제공해야 합니다. 이 모든 공공정보는 교환이 끝나면 양쪽 모두의 참가자들이 이 결과에 대해 동의해야 합니다. 판사 시스템은 회로를 통해 엘리시아가 처음 생각한 번호를 넣어 출력이 정당한지 확인하고, 따라서 결정이 공정했는지 확인할 수 있을 것입니다. 말할 필요도 없이, 규칙이 준수되었는지 확인하는 명확한 절차가 있다는 사실은 분쟁이 발생하는 것이 어렵다는 것을 증명합니다. 엘리시아가 동전뒤집기를 하고 베니토가 전화기 너머로 맞추는 상황을 비교했을 때, 어떤 판사도 이 경우를 말지 않을 것입니다.



하나의 그림만큼 작은 회로는 속임수에 사용될 수 있는 표를 만들기 쉽기 때문에, 실생활에서 많이 사용되지는 않을 것입니다. 입력에 32 자리의 이진수를 사용하면 더 나은 보호기능을 제공합니다. 그러나 그것이 속임수를 쓰기가 쉽지 않다는 것을 보장해 주지는 않습니다. 또 다른 방법은 활동 14 여행자 도시에서 설명한 일방향 함수를 사용하는 것입니다. 실제로 사용되는 방법은 어려운 문제(비록 우리는 다음 활동의 끝 부분에서 이것이 NP-완전문제 명제가 아니라는 것을 배웁니다)로 알려져 많은 수의 인수 분해에 의존합니다. 이것은 하나의 번호가 다른 요인 때문인지 검사하기는 쉽지만, 다수의 인자들을 찾는 것은 매우 오랜 시간이 걸립니다. 비록 진술할 때는 엘리시아와 베니토(그리고 판사)가 직접 손으로 만들지만 실제로는 개발된 소프트웨어에 의해 수행됩니다.

디지털 서명은 이와 비슷한 생각을 기반으로 합니다. 특정 비밀 입력회로의 출력을 공용함으로써, 엘리시아가 출력을 생성한 것을 효과적으로 증명할 수 있습니다. 일방향함수이므로 다른 누구도 입력값을 찾아낼 수 없습니다. 아무도 엘리시아로 변장할 수 없습니다! 실제 디지털 서명을 만들기 위해서는 엘리시아가 특정 메시지에만 서명 할 것을 보장하기 위해 더 복잡한 프로토콜이 필요합니다. 또 다른 사람이 그녀가 아니더라도 주장하는 경우에도 엘리시아가 서명자임을 확인할 수 있는지 확인할 수 있어야 합니다. 그러나 원칙적으로는 동일합니다.

또 다른 응용프로그램은 심판이 없는 상황에서 카드를 다루는 두 플레이어가 전화 너머로 포커를 치는 것입니다. 모든 것은 플레이어 자신과 분쟁이 일어날 경우 게임의 끝에서 판사의 판결에 따라 수행해야 합니다. 비슷한 상황은 계약 협상에 본격적으로 발생합니다. 물론, 플레이어는 게임을 하는 동안 자신의 카드를 비밀로 유지해야 합니다. 그러나 그들은 정직해야 합니다. 실제로 하나가 남기 전까지는 에이스를 주장할 수 없습니다! 이것은 게임이 끝날 때까지 기다렸다가 각각의 플레이어의 움직임 순서와 원래의 손을 확인할 수 있습니다. 또 다른 문제는 게임이 끝날 때까지 각 플레이어의 손을 비밀로 유지하면서 카드를 처리하는 방법입니다. 놀랍게도 이것은 동전던지기의 방법과 크게 다르지 않는 암호화 프로토콜을 사용하여 달성할 수 있습니다.



암호화 프로토콜은 전자상거래에서 직불카드의 소유자를 식별하기 위해 핸드폰의 사용을 허가하거나, 전자 메일의 송신자를 인증할 것인지 확인하는데 매우 중요하게 사용됩니다. 확실히 이 일을 할 수 있는 능력은 전자 상거래의 성공에 매우 중요한 영향을 미칩니다.

추가 읽기 자료

하렐이 지은 책 *Algorithmics*에서는 디지털 서명 및 관련 암호화 프로토콜에 대해 설명합니다. 또한 D.A. Klarner 에 의해 편집된 책 *The Mathematical Gardener*에 “정신 포커”라고 불리는 장에서는 1981 년에 제기된 아이디어를 통해 포커를 재생하는 방법을 보여줍니다. 도로시 데닝이 쓴 *Cryptography and data security*에 의하면 뛰어난 컴퓨터 과학 문헌은 암호작성술입니다. Dewdney's 의 *Turing Omnibus*에는 이 활동의 회로에 사용되어 블록을 만드는 부울 논리학에 대한 내용이 있습니다.

활동 19

꼬꼬마 암호 - 공개키 암호화

활동 개요

암호화는 정보 보안의 핵심입니다. 그리고 현대 암호화 키는 공개 정보를 사용하여, 보내는 사람이 정한 받는 사람만이 (물론, 비공개이지만) 잠금을 해제 할 수 있는 방식으로 자신의 메시지를 잠글 수 있다는 것입니다.

모든 사람들이 자물쇠를 구입해서 거기에 자신의 이름을 쓰긴 하지만 다른 사람이 사용할 수 있도록 동일한 테이블에 모두 놓습니다. 물론 그 자물쇠를 잠근 열쇠는 자신이 관리합니다. 내가 당신에게 보안 메시지를 보낼 경우에, 나는 당신의 자물쇠를 집어 상자에 넣어 잠그고 당신에게 보냅니다. 그것은 잘못 보관하여 다른 사람들에게 공개되어도 자신이나 그것을 풀 수 있도록 알려준 사람외에는 다른 누구도 그 잠금을 해제 할 수 없습니다. 이 방식은 비밀 코드를 배치하기 위한 사전의 대화가 필요 없습니다.

이 활동은 디지털로 할 수 있습니다. 그리고 디지털 세계에서는 진짜 자물쇠를 사용하는 대신에 원래의 자물쇠를 두고, 복사본을 사용합니다. 내가 실제 자물쇠의 복사본을 만드는 경우에는 그것을 분해할 수 밖에 없습니다. 그렇게 해서 그것이 어떻게 작동하는지 볼 것입니다. 그러나 디지털 세계에서는 열쇠를 공개해도 다른 사람들이 열수 있는 방법이 없습니다. 불가능하다고요? 계속 읽어봅시다.

교과 연계

- ✓ 기술-공개 키 암호화, 비밀코드

필수기능

- ✓ 퍼즐 해결

대상 연령

- ✓ 11 세 이상.

수업 자료

학생

4 개의 모듈을 만들고, 모듈 안에 두 명씩 짝을 짓고 하나의 팀이 도우미 팀이 됩니다.
각 도우미 팀은 꼬꼬마 암호 워크시트에 있는 두 개의 지도 복사본을 받게 됩니다.

✓ 꼬마 암호 지도 2 장

✓ OHP 필름



✓ 그 뜻을 쉽게 풀이하는 방법.

소개

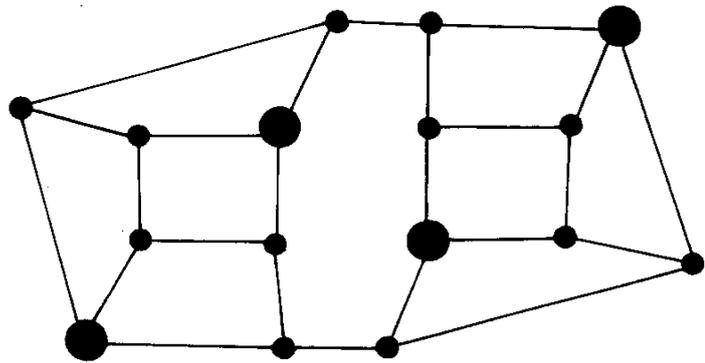
이것은 이 책에서 가장 기술적인 내용을 이해하는 도전적 활동입니다. 그것을 성공적으로 완료하기 위해서는 주의력과 지속적인 집중이 필요합니다. 학생들은 활동 14의 관광 도시에서 일방향 함수의 예와 여행자 도시를 공부해야 하고, 또한 다른 활동(활동 16의 비밀 공유, 활동 17의 페루 동전 던지기)를 완료한 경우에도 도움이 됩니다. 활동 1의 점의 갯수와 활동 5 스무고개의 아이디어로도 쓸 수 있습니다.

에이미는 빌에게 비밀 메시지를 보낼 계획입니다. 일반적으로 문장이나 단락 등의 비밀 메시지를 생각할 수 있지만, 다음 연습에 에이미는 단지 하나의 문자를 보낼 것입니다. 사실, 에이미는 문자를 나타내는 하나의 번호를 보낼 것입니다. 이것이 단순한 메시지로 보일 수도 있지만 문장을 구성하는 "메시지"의 전체 문자열을 보낼 수도 있음을 명심합니다. 실제 작업은 컴퓨터에 의해 수행됩니다. 그리고 때로는 작은 메시지조차 중요합니다. 폴 리비어 (Paul Revere)에 의해 수행된 역사상 가장 유명한 메시지 중 하나는 오직 두 가지 값을 가지고 있었습니다. 누군가 중간에서 그 암호를 풀 수 없도록 빌의 공개 잠금을 사용하여 암호화 된 메시지에 에이미의 수를 잘 포함시키는 방법을 볼 수 있습니다. 단지 그가 잠금 키를 가지고 있기 때문에 오직 빌만이 그 작업을 수행할 수 있습니다.

지도를 사용하여 메시지를 잠글 것입니다. x 마크로 표시된 보물섬 지도가 아니라 여행자 도시(활동 14)와 같은 거리 지도입니다. 각 지도는 공개 버전 잠금 및 개인 버전 키가 있습니다.

생각해보기

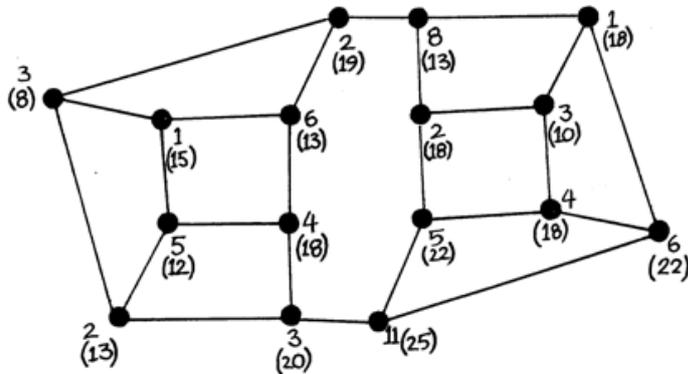
꼬꼬마 암호 워크 시트 인코딩에 표시된 빌의 공개지도입니다. 이것을 다른 사람에게 비밀로 하지 않습니다. 빌과 모두에게 참조하거나 그에게 메시지를 보낼 수 있는 사람에게 제공하기 위한 테이블 (또는 웹 페이지)에 넣습니다. 에이미는 복사본을 가져서 모두가 갖고 있습니다. 오른쪽 그림은 빌의 개인 지도를 보여줍니다. 길거리의 일부를 확대하는 등 특별하게 표시되는 것을 제외하고는, 그의 공개 지도와 동일합니다. 지도의 비밀을 유지합니다.



1. 이 활동은 꽤 많은 양의 활동을 포함하고 있어 최소한 처음시작이라도 학급 전체가 하는 것이 가장 좋습니다. 많은 문제로 인해 오류가 생길 수 있겠지만 정확해야 합니다. 학생들이 이와 같이 불가능해 보이는 암호화 과정을 해결해내는 것이 얼마나 놀라운 것인지 깨닫게 하는 것이 중요합니다. 학생들에게는 동기 부여가 필요하기 때문입니다. 이렇게 동기부여 받은 학생들은 이 방법을 사용하여 수업 시간에 비밀 메모를 전달하며, 심지어 교사가 암호화 된 방법을 알고 있어도 해독할 수 없습니다.

2. 빌의 공개지도 (꼬꼬마 암호 인코딩 워크 시트)를 보여줍니다. 에이미가 보낼 수를 결정하세요. 에이미가 전송하고자하는 번호를 추가 할 수 있도록 이제, 지도에서 각 교차로에 임의의 숫자를 배치합니다. 아래 그림처럼 각 교차로 옆에 쓴 번호처럼 하면 됩니다. 여기에, 에이미는 66 을 보내려고 하는 숫자로 선택해서 괄호 밖의 모든 수를 더하면

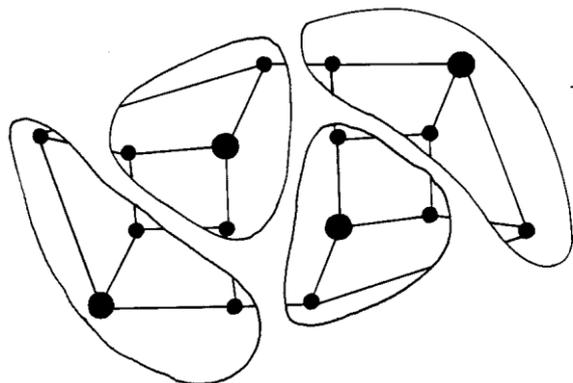
66 이 되게
했습니다.
음수를 사용할
수도 있습니다.



3. 이제 에이미는 빌에게 보낼 방법을 생각해야 합니다. 번호와 지도를 전송하는 경우 다른 사람이 가로채서 메시지를 가질 수 있기 때문입니다.

대신, 어떤 교차로든 선택하여 그 점과 연결된 이웃한 점의 합을 구해 괄호에 씁니다. 이 번호를 작성할 때는 다른 색 펜을 사용합니다. 예를 들어, 공개된 지도 내의 가장 우측 교점 6 은 세 점과 연결됩니다(4, 1, 11 을 표시). 따라서 지금 지도의 다른 모든 교차점에 대해 이를 더한 값(22)를 가지고 표시됩니다. 이 괄호 안에 당신의 번호를 제공해야 합니다.

4. 에이미는 괄호의 숫자를 빌에게 보냅니다. 그리고 에이미가 쓴 원래 숫자는 삭제합니다. 또는 새로운 지도에 괄호안의 숫자를 써서 빌에게 보냅니다. 원본 메시지가 무엇인지 찾을 수 있는 방법을 말해보세요. 빌의 개인 키를 가진



사람만이 에이미가 원래 보내려는 메시지를 해독할 수 있습니다. 부호화 된 메시지에서 빌의 공개되지 않은 지도의 확대 노드를 표시합니다.

메시지를 해독하기 위해, 빌은 비밀 표시 교차로에 있는 점들의 번호를 더합니다. 예를 들어, 교차점에 13, 13, 22, 18로 표시된 선들이 연결되어 있으면 이 숫자들을 더해서 원본 메시지인 66을 찾을 수 있습니다.

5. 어떻게 활동할까요? 네, 지도는 특별합니다. 빌이 표시 교차로 중 하나를 선택하고 교차점에서 한 거리만큼 떨어진 교차점 둘레를 그리세요. 그리고 그 과정을 반복하세요. 여기에 그려놓은 그림처럼 겹치지 않게 지도 조각으로 분할합니다. 지도에서 경계를 그림으로써 조각을 만듭니다. 교차점에 있는 4개의 전송된 숫자의 합이 원래 맵의 모든 원래 숫자의 합이 됩니다; 이렇게 해서 원래의 메시지를 찾을 수 있게 됩니다.

네, 그것은 하나의 편지를 보내는 것이 많은 일을 하는 것처럼 보입니다. 그리고 하나의 문자 암호를 보내는 일은 쉬운 일이 아닙니다. 그러나 완전한 비밀을 참가자 사이의 사전 협의 없이, 공개 키를 사용하여 해결한 것을 보세요. 여러분은 게시판에 자신의 키를 게시할 수 있고 누군가가 당신에게 비밀 메시지를 보낼 수 있습니다. 아직 아무도 개인 키 없이는 암호를 해독할 수 없습니다. 그리고 실제 생활에서의 모든 계산은 (일반적으로 웹 브라우저에 내장) 소프트웨어 패키지에 의해 이루어집니다, 그래서 컴퓨터가 처리를 하게 되죠.

아마 여러분은 공개키 암호화를 개발한 사람들에 대해 알고 싶어질 것입니다. 실제 불가능하다고 생각했던 이런 작업을 소수의 사람들이 해내기도 했습니다

자, 도청은 어떨까요? 빌의 지도는 한 블록 이상 걷는 사람이 없게 모든 교차로에 아이스크림 트럭을 배치하는 여행자도시(활동 14) 문제와 비슷합니다. 빌이 자신의 공개되지 않은 지도에 표시된 조각으로 시작하여 발견된 정답을 지도를 구성하는 것은 쉽지만, 아이스크림 트럭을 배치할 수 있는 최소화 방법을 찾는 것은 매우 어렵다라는 것을 보았습니다. 무작위 방법은 하나의 트럭을 가지고 문제를 해결할 수 있을 때까지 가능한 모든 구성을 끊임없이 시도해 보는 것입니다. 어느 누구도 일반적인 지도가 더 나은 방법인지 알지 못합니다. 여러분은 많은 사람들이 하나의 방법을 찾기 위해 노력하고 있다는 것에 내기를 걸 수도 있을 것입니다!

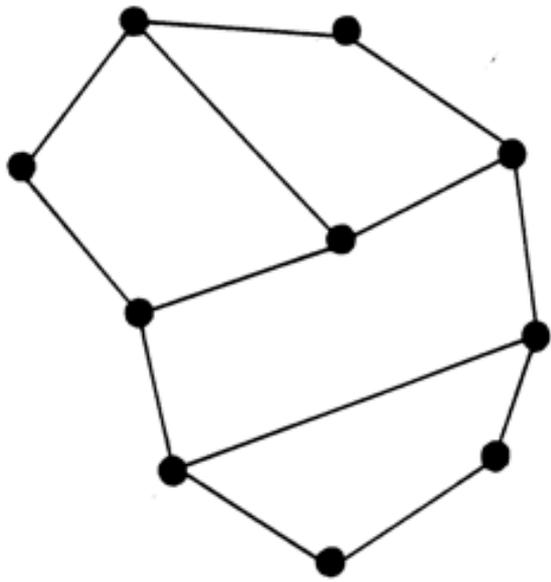
6. 학급 전체에 하나의 예를 적용해 보세요. 네 개의 그룹으로 학생들을 나눕니다. 각 그룹의 각 쌍에게 꼬꼬마 암호 지도에 있는 공개지도를 제공합니다. 각 쌍은 "메시지"(임의의 정수)를 선택 공개 키로 암호화하고, 다른 그룹에게 암호화된 지도를 줍니다. 그 지도를 받은 그룹은 해독을 해봅니다. 하지만 개인지도 받을 때까지는

성공하기 힘들 수 있습니다. 그런 다음 개인지도를 주고 그들이 지금 제대로 해독할 수 있는지 확인해 보세요.

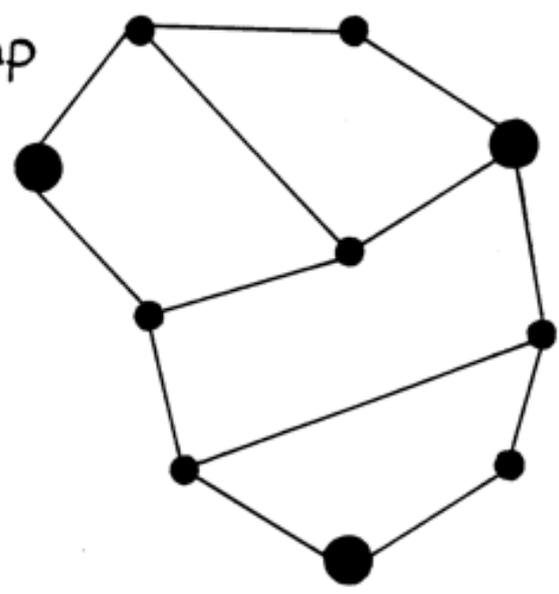
7. 이제 각 쌍은 개인 비밀을 유지하되 교실 게시판에는 "공개"함으로서 상대방에게 공개 버전을 주고, 자신의 지도를 디자인 할 수 있습니다. 지도 설계 원칙은 여행자 도시 활동에서 논의한 것과 같습니다. "특별한" 점에 추가 거리를 두지 않도록 주의하여야 합니다. 개인지도를 설계하는 것처럼 지도 조각이 중복되지 않도록 하는 특별한 점이 필요합니다. 이것은 암호화 작업에 필수적입니다.

워크시트 활동: 꼬꼬마 암호 지도

public Map

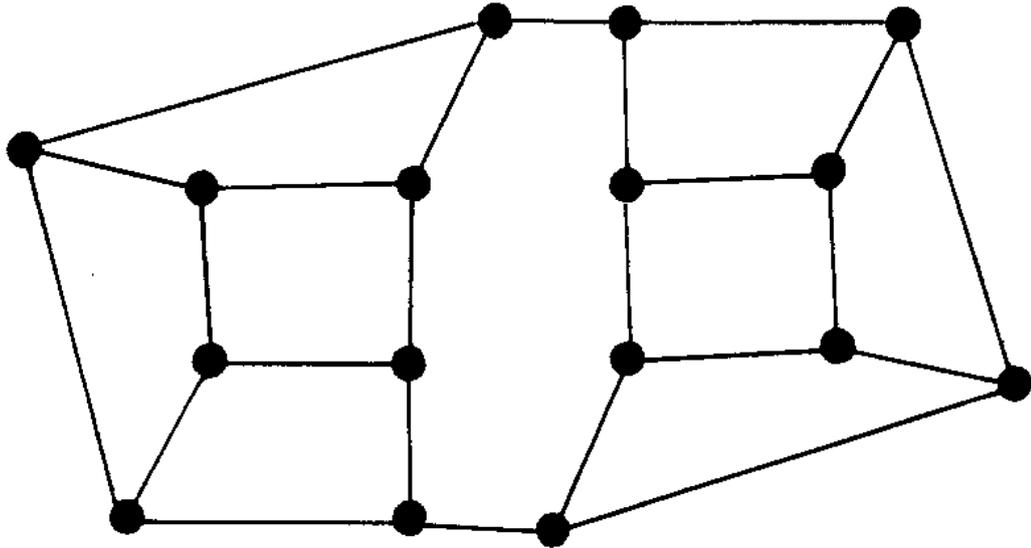


private Map



워크시트 활동: 꼬꼬마 암호 부호화

이 지도를 학급에 게시하고 메시지를 암호화하는데 사용하도록 합니다.



컴퓨터과학 이야기

컴퓨터 네트워크를 통해 비밀 메시지를 보내는 이유는 메시지를 받는 사람만 볼 수 있도록 하기 위해서입니다. 물론 비밀 코드를 보낸 사람과 받는 사람이 공유하는 경우 메시지 암호화가 가능하지만 이 때는 비밀 코드를 서로가 알아야만 합니다. 그러나 공개 키 암호화의 장점은 에이미가 웹 페이지와 같은 공공 장소에서 "잠금"을 선택하여, 사전 협의 없이도 빌에게 보안 메시지를 보낼 수 있다는 것입니다.

비밀은 암호화의 한 측면입니다. 또 다른 방법으로 인증에 사용됩니다. 에이미가 빌이 보낸 메시지를 수신 할 때, 어떻게 빌에게서 온 메시지인지, 사기꾼이 보낸 것인지를 알 수 있을까요? “나는 돈 없어서 여기에 갇혔어요, 내 은행 계좌에 100 달러를 넣어주세요. 번호는 0241-45-784329 에요.” 이 때 공개 키 암호 시스템이 사용될 수 있습니다. 에이미는 자신의 공개키를 인코딩하여 빌에게 비밀 메시지를 보내는 것처럼, 그는 그녀에게 단지 그가 자신의 개인키를 암호화하여 만든 메시지를 보낼 수 있습니다. 물론 키가 공개되어 있기 때문에 다른 사람도 암호를 풀 수 있지만 메시지가 에이미만 봐야 하는 것이라면, 빌은 에이미의 공개키로 두 번째로 암호를 풀 수 있습니다. 이 공개-개인키의 이중 암호 해독은 공개 키와 개인 키의 동일한 기본 방식에 비밀 및 인증을 모두 제공합니다.

지금 이 활동에 예시된 방식은 업계에서 사용되는 최강의 공개 키 암호화 시스템과 매우 유사하지만, 사실 꽤 큰 지도를 사용할 경우에는 안전하지 않다고 할 수 있습니다.

임의지도에 최소한의 아이스크림 트럭을 배치하는 최적의 방법은 알려져 있지 않기 때문에 이것을 암호에 이용하면 매우 안전하지만, 그것을 공격할 수 있는 다른 방법이 있을 가능성이 있습니다. 이러한 내용은 고등학교 수준까지 학생들에게 이해시킬 수

있지만 적어도 개념만은 설명해도 좋습니다. 여러분이 수학적으로 이해되지 않을 경우에는 다음 단락을 무시하고 진행하십시오!

번호 맵 1, 2, 3 에 교차로, ... B1 의해 교차점에 할당 된 원래의 숫자를 나타내는, B2, B3, ..., 실제로는 T1, T2, T3 에 의해 송신되는 숫자. ... 그것의 교차로 2 (3)와 접속되어 있는 것으로 하고, 4. 그 교차점에 대해 송신 수는

$$t_1 = b_1 + b_2 + b_3 + b_4 \text{ 입니다.}$$

물론, 다른 모든 교차로는, 미지수 B1, B2, B3 이 있기 때문에 방정식 같은 번호가 퍼블릭 맵과 숫자 T1, T2, T3 에 대한 유사한 방정식으로 전송됩니다. 따라서 방정식을 이용하여 방정식 해결 컴퓨터 프로그램으로 해결할 수 있습니다. 원래 번호가 얻어졌다면, 메시지는 자신의 합계가 암호 해독지도를 발견할 필요가 없습니다. 직접 가우스 소거법을 사용하여 방정식을 풀기 위해 요구되는 계산 노력은 방정식의 수의 세제곱에 비례하지만 이러한 방정식의 성질 때문에 암호를 하는 더욱 효율적인 기술들이 존재합니다. 지수 계산 노력이 최고의 암호 해독 맵을 마련하기 위해 할 수 있는 방법입니다.

암호화에 사용하는 기술은 손으로 계산하여 푸는 것이 불가능 것을 제외하고는 모두 다릅니다. 사실, 실제 공개 키 암호 시스템에 관련된 프로세스는 우리가 활동한 것과 거의 동일합니다. 그리고 가장 안전한의 공개 키 방식의 암호기법은 다수의 인수 분해를 사용합니다.

100 자리 숫자

9,412,343,607,359,262,946,971,172,136,294,514,357,528,981,378,983,08
2,541,347,532,211,942,640,121,301,590,698,634,089,611,468,911,68

을 어떻게 인수분해 할 수 있을까요? 직접 해결하기 위해 너무 오랫동안 고생하지 마세요! 컴퓨터로도 해결하기 어려운 문제이니깐요.

86,759,222,313,428,390,812,218,077,095,850,708,048, 977 와

108,488,104,853,637,470,612,961,399,842,972,948,409,834,611,525,790,
577,216,753 입니다. 이 두 개의 수로 인수분해 할 수 있습니다. 이것을 찾는 것은
상당한 힘든 작업입니다. 사실 슈퍼 컴퓨터로 몇 개월 걸리는 프로젝트입니다.

이제 실제 공개 키 암호 방식에서, 공개키는 100 자리의 수를 사용하고, 인수분해한 두
개의 숫자는 개인키로 사용할 수 있습니다. 인수분해하 두 숫자를 알면 해결하기에는
너무 어렵지는 않을 것입니다. 여러분이 필요로 하는 모든 큰 소수를 계산하는
방법입니다. 공개키는 충분히 큰 두 소수를 이용하여 찾을 수 있습니다. 거대한 숫자를
곱하는 것은 컴퓨터가 처리하기에 큰 문제는 없습니다. 하지만 인수분해해서 두 개의
소수로 나누어 찾는 과정은 슈퍼 컴퓨터로도 몇 개월이 걸립니다. 개인키로는 큰
소수를 찾기 어렵습니다. 100 자리의 숫자 공개키도 해독이 될까봐 걱정이 되면 200
자리를 사용하면 됩니다. 그러기 위해서 슈퍼 컴퓨터가 1 년은 더 작업해야 하니
암호는 쉽게 해독되지 않겠죠. 실제로, 인터넷 상의 계좌정보들은 십진수의
155 자리인 512 비트를 개인 키로 사용합니다.

우리는 아직도 소수 기반 공개키를 사용하여 메시지를 부호화하는 방법에서 개인키
없이 해독 할 수 있는 방법은 발견되지 않았습니다. 어쨌든, 컴퓨터과학자는 이러한
어려움을 극복하고 적절한 암호화 및 복호화 알고리즘 계획을 만들려고 노력하고
있습니다!

소수를 기반으로 한 시스템이 얼마나 안전할까요? 엄청나게 큰 소수를 인수 분해하는
것은 여러 세기 동안 세계 최고의 수학자들의 관심을 끌었던 문제이며, 해결 가능한
모든 요소를 시도해보는 무작위적인 방법보다 훨씬 더 나은 것을 발견하였습니다.
여행자 도시 문제를 해결하지 않고 빌의 코드를 크래킹의 방법이 있을 것으로 밝혀졌던
것처럼, 실제로 큰 소수의 인수분해하지 않고 개인키를 크래킹하여 소수를 알아내는
방법이 있을 수 있습니다..

또 다른 염려는 단지 몇 개의 가능한 메시지가 있을 경우, 해킹하려는 사람이 공개키를 이용하여 차례로 이들 각각을 암호화하고, 그것을 필요로 하는 사람들에게 풀도록 속여서 그 모든 가능성을 실제 메시지와 비교할 수 있다는 점입니다. 에이미의 방법은 동일한 메시지를 암호화하는 숫자 코드 값까지 추가로 선택했는지에 따라 여러 가지 방법으로 있기 때문에 피합니다. 너무 많은 메시지도 매우 빠른 컴퓨터의 도움으로, 모든 방법을 시도해볼 수 있도록 암호화 시스템이 설계되어 있습니다.

더 읽어보기

하렐의 책 *Algorithmics* 은 공개키 암호화에 대해 설명으로 안전한 공개키 시스템을 만들 큰 소수를 사용하는 방법에 대해 설명합니다. 더 실용적인 책으로 브루스 슈나이어에 의해 씌여진 암호화를 적용하는 동안 암호화의 표준 컴퓨터 과학 텍스트, 도로시 데닝에 의해 암호화 및 데이터 보안입니다. Dewdney의 튜링 옴니버스는 공개 키 암호화를 수행하기 위해 다른 시스템을 설명합니다.

Part VI

인간과 컴퓨팅 대면하기

-컴퓨터와 상호작용

인간과 컴퓨팅 대면하기

왜 사람들은 컴퓨터 다루는 것을 어려워할까요? 많은 사람들은 컴퓨터 사용이 얼마나 어려운지 주장하면서 인간이 진정으로 원하는 것은 컴퓨터가 결코 할 수 없다며 어이없는 실수를 반복하곤 합니다. 컴퓨터는 평범한 사람이 아닌 마법사가 만든 것처럼 보입니다. 하지만 컴퓨터는 우리가 공부하고, 일하고, 놀이하는 데 더 많은 도움을 주는 도구이기 때문에 평범한 사람들이 평범한 사람들을 위해 만들어야 합니다.

여러분과 컴퓨터가 상호 작용하는 영역을 "사용자 인터페이스"라고 합니다. 이것은 보통 사람들을 위해 가장 중요한 부분입니다! 프로그램을 사용하려면 인터페이스를 사용하여야 하는데 만약 기계와의 상호작용을 잘 모른다면 프로그램은 전혀 작동하지 않으며 원하는 일을 해결할 수 없습니다. 사용자 인터페이스는 보기에 쉬어보이나 이것을 설계하고 개발하기가 매우 어렵습니다. 인터페이스 프로그램을 만드는 것은 다른 어떤 부분보다 많은 노력을 기울여야 하기 때문입니다. 일부 소프트웨어는 응용 프로그램을 사용하여 복잡한 과정도 필요없고, 거의 드러나있지 않으면서도 편리하게 이용할 수 있는 우수한 사용자 인터페이스를 가지고 있습니다. 반면에 이상한 사용자 인터페이스를 가지고 있다면 수많은 소프트웨어 제품들은 사용하지 못하게 됩니다. 그래서 모든 산업분야에서 효과적인 인터페이스 아이디어 - 워드 프로세서나 스마트폰 같은 - 를 가지고 제품을 만듭니다. 이제는 이것 자체만으로도 컴퓨터 과학의 중요한 분야로 발전하고 있습니다.

그런데 왜 모든 부분에서 사용자 인터페이스를 사용해야 하는 것일까요? 우리가 친구한테 하는 것처럼 컴퓨터와도 이야기 할 수는 없는 것일까요? 좋은 질문입니다. 언젠가는 그렇게 할 수도 있을지도 모르고 절대 할 수 없을 수도 있을 겁니다. 분명한 것은 지금은 그렇게 할 수는 없다는 것입니다. 오늘날 "지능형"컴퓨터가 그렇게 쓰일 수 있도록 하는 연구하고 노력하지만 실제적으로는 큰 한계가 있습니다. 다음 활동은 사용자 인터페이스에서 디자인의 문제를 이해하고, 컴퓨터가 가지는 한계점에 대해 자세히 생각해 볼 수 있도록 안내합니다.

교사를 위한 팁

컴퓨팅 과정에서 기계와의 의사소통은 계산을 다루지 않습니다. 컴퓨팅에서 인터페이스라는 것이 본질적인 가치가 없는 것처럼 보이기도 합니다. 하지만 만약 그 결과들이 컴퓨터와 외부 세계가 어떻게든 의사소통을 하고 사용자에게 영향을 미칠 수

있다면 가치가 있다고 할 수 있습니다. 놀랍게도 컴퓨터 과학은 컴퓨터와 사용자들에 관한 분야에 점점 더 많은 의미를 부여하고 있습니다 - 결국 컴퓨터가 어떤 방법으로는 사람들에게 도움이 되지 않으면, 아무 소용이 없다는 것이죠. 좀 더 빨리 작동하고 효율적인 컴퓨터를 만들기 위해 연구하고 노력하는 이유는 사용자에게 빨리 반응하고 경제적으로 사용할 수 있는 컴퓨터를 사람들이 필요로 하기 때문입니다.

인터페이스는 인간이 컴퓨터와 통신하는 방법입니다. 그리고 이 책에 있는 많은 활동은 통신(의사소통)에 관한 것입니다. 정보 표현(Part I)은 컴퓨터 또는 컴퓨터 간에 다른 종류의 정보를 어떻게 주고 받는지를 보여줍니다. (Part III) 절차 표현은, 컴퓨터에게 전달하기 위한 통신(의사소통) 절차 방법과 주어진 작업을 어떻게 수행하는지에 관한 것입니다. 결국 '프로그래밍'은 컴퓨터만의 언어로서 컴퓨터와 상호작용하기 위해 적용되는 것입니다. 암호화 (Part V)는 비밀 정보 전체 또는 비밀 정보 중 몇 개만 공개하면서 통신하는 방법에 관한 것입니다.

다음 활동들은 사람들이 컴퓨터와 의사소통하는 방법에 대한 이야기입니다. 이 책의 대부분은 기술적인 아이디어를 이해하는 것을 기반으로 하지만 이번 장에서는 그렇지 않습니다. 보다 넓은 개념을 이해하는 활동이어서 학생들에게 특별한 지식을 요구하지 않고, 어렵지 않게 구성을 하였습니다. 다른 챕터에서 소개된 활동들보다 더욱 세심한 설명을 포함하고 있습니다. 학급내의 토론 과정에서 배경지식은 학생들에게 영향을 끼칠 수 있기 때문에 교사를 위한 자료를 충분히 제시하고자 하였습니다.

이번 장에서는 다음과 같은 두 개의 활동이 있습니다. 첫째, 일반적으로 HCI 약칭으로 알려진 "인간 - 컴퓨터 인터페이스" 영역에 관한 내용을 소개합니다. 컴퓨터 시스템의 특정 개념에 대한 주요 지식에 의존하지 않는 언플러그드 컴퓨팅 활동을 하기 위해 컴퓨터를 사용하지 않는 활동으로 개발하였습니다. 하지만 인간-컴퓨터 인터페이스 디자인에 사용되는 기본 개념은 소개하고 있습니다. 왜냐하면 인간-컴퓨터 인터페이스 디자인은 활동에 있어 정답이 필요하지 않기 때문에 몇몇 학생들을 힘들게 할 수도 있습니다. 두번째 활동은 "인공 지능," 또는 AI 로 알려진 영역을 소개합니다. 그것은 컴퓨터가 할 수 있는 일과 할 수 없는 일에 대하여 학생들의 생각을 자극시켜주는 추측 게임을 소개하고 있습니다.

공학적 마인드 갖기

소프트웨어 제품의 성공은 사용자 인터페이스를 얼마나 잘 구현하느냐에 달려있기 때문에 '인간 - 컴퓨터 상호작용'은 컴퓨터 과학에서 가장 인기 있는 연구 분야 중 하나가 되고 있습니다. 다루는 주제는 심리학, 인지 과학, 언어학, 사회학, 심지어 인류학 같은 컴퓨터 과학 분야의 외부 영역까지 넓은 범위를 포함합니다. 소수의

컴퓨터 과학자들이 이 분야에서 연구를 하고 있으며, HCI 는 좀 더 가벼운 주제에 관심이 있는 사람들을 위한 분야로 성장해 가고 있습니다.

인공지능은 종종 인간을 불안하게 하거나 논쟁을 야기하는 주제입니다. 이 책에서 우리는 인공지능 기계가 모퉁이를 돌 수 있다고 믿는 AI 지지자들과 기계 자체가 원칙적으로 지능적일 수 없다고 믿는 AI 회의론자들의 중간자적 입장을 유지하기 위해 노력하였습니다. 우리의 목표는 이러한 문제에 대해 학생들이 독립적으로 생각할 수 있도록 격려하고, 균형 잡힌 관점을 지닐 수 있도록 하는 것입니다.

이러한 이슈에 좀 더 관심이 있다면 다음의 두 책을 적극 추천합니다. 도널드 노만의 일상적인 것들의 디자인(The design of everyday things)과 존 휴거랜드의 인공지능(Artificial Intelligence) 책에 여기서 소개하는 두 가지 활동에 관한 내용이 있습니다.

컴퓨터는 이 책에서 다루지 않은 다른 종류의 통신을 합니다. 컴퓨터 시스템을 구축하는 사람들 사이에서 이루어지는 통신으로 컴퓨터과학을 배운 학생들은 고용시장에서 자신의 갈 길을 정하게 됩니다. 자신의 직업에 수반되는 대인 관계에서 얼마나 많은 의사소통이 이루어지는 지를 알게 되면 놀라게 될 것입니다.

컴퓨터 프로그램은 수백만 또는 수십억 개의 부품이 복잡하게 연동되도록 만들어진 복잡한 객체이며, 프로그래밍 프로젝트는 협력과 소통의 시간을 필요로 하며 팀이 얼마나 긴밀하게 연결되어 있느냐에 성공의 여부가 달려 있습니다. 제품이 구축되면, 사용자 매뉴얼, 교육 과정 개발, 전화상담, 온라인 지원, 시연, 전시 및 광고를 통해 예상되는 고객과 의사소통하는 일을 하게 됩니다. 우리는 아직까지 현실적인 상황에서 학생들을 위한 컴퓨팅의 대인 커뮤니케이션을 이해하기 위한 "언플러그" 방법을 찾지 못했습니다. 그래서 이 책에서는 그 내용을 언급하지 않고 있습니다. 그러나 교실을 방문하는 컴퓨터 전문가들이 자신의 경험을 설명하고 토론에서 논할 수 있는 주제 거리로는 좋습니다.

활동 20

초콜릿 공장—인간 인터페이스 설계

활동 개요

이 활동의 목적은 인간의 인터페이스 디자인 문제에 대한 인식을 이해하는 것입니다.

우리는 빈약한 디자인으로 채워진 세계에 살고 있기 때문에 인간과 물건이 상호작용하면서 일어나는 문제의 원인을 디자인에 결함이 있다고 생각하는 대신에 우리 스스로를 비난하는(인간의 잘못된 이해, 사용방법 이해 부족, 자신의 능력 비하 등)것에 익숙해져 있습니다. 이러한 문제는 컴퓨터를 사용하면서 크게 부각이 됩니다. 왜냐하면 컴퓨터는 뚜렷한 용도를 드러내지 않고 컴퓨터의 모습만으로는 무엇을 위한 용도인지, 어떻게 작동되는지에 대한 어떠한 단서도 제공하지 않기 때문입니다. 실제 컴퓨터는 완전히 범용적입니다.

교과 연계

- ✓ 기술 : 디자인을 통한 기술의 목적이 있는 개입을 이해하는 것

필수기능

- ✓ 디자인
- ✓ 추론
- ✓ 일상적인 개체 인식

대상 연령

- ✓ 7 세 이상

수업 자료

각 모둠별 학생 준비물

- ✓ '문을 여는 방법', 가스렌지 복사본
- ✓ OHP 필름 또는 학급에 전시될 수 있는 카드에 복사된 워크시트 아이콘
- ✓ 아이콘 카드의 6 개 카드 중 하나 또는 그 이상. 시트를 개인 카드로 자르고 모듬 안에서 나눠갖기

초콜릿 공장

소개

‘찰리와 초콜릿 공장’이라는 영화를 보게 되면 큰 초콜릿 공장은 움파룸파스 요정에 의해 가동이 됩니다. 움파룸파스족은 기억을 잘 못하고 글자도 모릅니다. 그래서 초콜릿 공장을 가동하는 방법을 기억하지 못하기 때문에 종종 기계가 잘못 작동되기도 합니다. 이 때문에 공장주인은 움파룸파스가 쉽게 공장을 가동할 수 있도록 새로운 공장을 설계하려고 합니다.

생각해보기

1. 학생들에게 이야기를 설명하고 작은 그룹으로 나눕니다.
2. 움파룸파스족이 직면한 첫 번째 문제는 뜨거운 김이 나는 액체 초콜릿 양동이를 들고 문을 통과하는 것입니다. 문을 밀었는지, 문을 잡아 당겼는지, 한쪽으로 밀어 넣었는지를 기억할 수 없습니다. 결과적으로 그들은 서로 부딪히며 사방에 끈적 끈적한 초콜릿을 흘립니다. 학생들은 열려있는 문을 어떻게 작동하여야 할지 생각하여 "문" 워크시트를 작성해야 합니다. 각 상자에는 다양한 경우의 내용을 제시하고 있습니다. 첫번째 문을 포함하여 일부 문들은 여는 방법에 관해서 명확히 알기 힘듭니다. 학생들은 그 문들을 처음에 어떻게 시도해볼 건지에 관해 기록을 해야 합니다. 자신의 시트를 모두 다 작성하면, 초콜릿 양동이를 들고 나갈 때 문을 여는 방법, 가장 쉽게 문을 열 수 있는 방법, 각 문이 갖고 있는 장점과 그것을 어떻게 적절하게 사용할 수 있을지에 관해 반 친구들과 토론해 보는 것이 좋습니다. 특히 관련 문, 각 유형의 상대적 장점을 토론하게 해야 합니다. 그리고 공장에서 사용할 수 있는 문과 손잡이의 종류를 결정해야 합니다.
3. 이 활동을 반 친구들과 함께 토론해보세요. 워크시트의 각 문에 간단한 댓글을 답니다. 실제로 문이 어떻게 열리는지에 관한 단서가 문의 틀과 경첩에 나타나 있습니다. 즉, 문이 안쪽 또는 바깥쪽으로 열리는지에 대한 규칙이 있습니다. 학교에서 사용되는 손잡이의 종류를 확인하고 적절한지 논의해 보세요. (*적절하지 않다*고 말할 수도 있을 것입니다!) 문이 가끔은 여러분을 혼란스럽게 할 수 있다고

생각하나요? 왜 그런가요? 복도에서는 문이 안쪽 또는 바깥쪽으로 열리는 것 중 어느 것이 일반적이고 안전할까요?

(답 : 밖으로 나올 때 복도에 있는 사람과 부딪히지 않도록 하기 위해 방 안쪽으로 문을 엽니다. 긴급 상황에서는 대피하기 쉽게 바깥쪽으로 열 수 있습니다.)

4. 여기서 중요한 개념은 사람이 사물을 볼 때 사물이 갖고 있는 실제적 특성 또는 그 특성을 인식하고 있다는 것입니다. 이것을 인간-컴퓨터 상호작용 학문에서는 '어포던스'라고 합니다. 노만이라는 사람이 설명한 어포던스는 외형적으로 보이는 특성뿐만이 아니라 실행하는데 도움이 되고, 조작을 하는데 편리한 단서를 제공하는 것을 어포던스라고 하였습니다. 예를 들면 의자는 앉기 위한 것이고, 테이블은 물건을 놓는 곳, 손잡이는 돌리는 것, 버튼은 무엇인가 작동시키기 위해서 누르는 것과 같이 겉에서 보이는 특징과 그 기능이 매우 명확하게 인식이 됩니다. 배우지 않고 경험하지 않아도 직관적으로 사물의 특징과 기능을 인식하는 것을 말합니다. 컴퓨터 인터페이스의 어포던스는 버튼의 모양, 텍스트 상자, 메뉴 등과 같이 사용자에게 사용하는 방법에 대한 단서를 제공합니다. 버튼이 우리가 평소 보아왔던 것처럼 보이지 않고 다른 모양으로 만든 경우, 사람들은 그것이 버튼인지 인식하기 어렵고 제대로 작동할지에 대해 믿을 수 없게 됩니다. 이처럼 컴퓨터(모니터)에서 사용되는 각각의 아이콘이나 모양들이 현실의 사물과 유사하게 만들어진 것을 어렵지 않게 찾을 수 있습니다.
5. 문은 매우 간단 개체입니다. 복잡한 것은 설명이 필요할 수도 있지만 간단한 것은 그렇게 하지 않아도 됩니다. 하지만 그것이 쉬운 일은 아닙니다. 간단한 개체가 사진, 라벨, 또는 지침을 필요로 한다면 그 디자인은 실패한 것입니다.

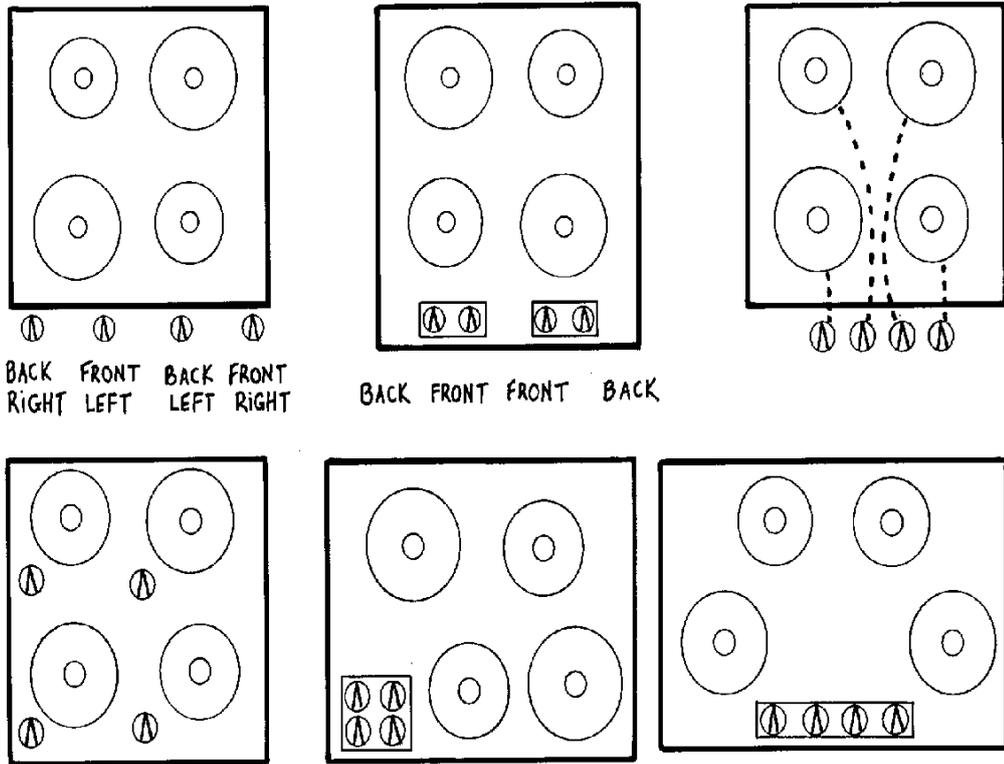
¹ 로알드 달에게 이해를 구하며, 여러분은 그의 유명한 작품 찰리와 초콜릿 공장에 나오는 움파룸파스에 관해 알고 있을 것입니다. 이 활동은 그 작품의 줄거리와 상관이 없다는 것을 알려드리는 바입니다.

일반 문	당기기 보다는 밀어서 여는 경우가 많다.	레이블 문	문 여는 데 설명서를 볼 필요가 있을까? 그리고 움파룸파스는 못 읽는다.
여닫이 문	적어도 측면이 열리는 것을 볼 수 있음.	바 도어	바를 당기는 것이 분명하지만, 당겨야 할까요?
손잡이 문	이런 손잡이 문은 당기거나 슬라이딩 하는 것이 보통이다.	알손잡이 문	손잡이를 잡아서 밀거나 당겨야 함을 보여줍니다. 아마 미달이는 아닐거예요.
패널 문	이런 문은 미는 것이 분명하다. 다른 방법이 가능한가요?	유리 문	한 쪽에 '당기시오' 또 다른 쪽에는 '미시오' 라고 표시.
미달이 문	오직 슬라이딩 뿐.		

6. 서로 다른 초콜릿을 담은 냄비는 서로 다른 온도에서 요리해야 합니다. 오래된 초콜릿 공장의 스토브는 스토브 탑 활동지에 보이는 것과 같습니다. 왼쪽의 스위치는 뒤쪽 왼쪽에 있는 버너 조절, 그 다음 스위치는 앞쪽 왼쪽에 있는 버너 조절, 그 다음 스위치는 앞쪽 오른쪽에 있는 버너를 조절하고, 오른쪽 스위치는 뒤쪽 오른쪽에 있는 버너를 조절할 수 있습니다. 움파룸파스는 항상 잘못된 온도에서 초콜릿을 요리하고, 온도를 맞추려고 스위치를 만지면서 소매를 태우기도 하였습니다.

7. 학생들은 스위치를 가정에서 가스렌지에 배치하는 방법을 기억하고 새로운 공장을 위한 더 나은 배치를 해야 합니다.

반 토론 활동을 실시합니다. 아래 그림은 몇 가지 일반적인 배치 예시를 보여줍니다. 버너에 직접 손이 가지 않도록 하기 위해 아래쪽 하단 하나에 모든 스위치를 놓습니다. 왼쪽 상단의 그림에서는 스위치로 버너를 조절할 수 있는 방법(실제



24 가지)이 많으며 8 개의 단어가 적을 수 있는 라벨이 필요합니다. 상단 가운데 그림처럼 짝으로 배열된 방법은 오직 네 가지 방법만 있어 훨씬 좋습니다. (왼쪽으로 두 개, 오른쪽으로 두 개씩 묶음); 그냥 네 개의 단어 라벨만 있으면 됩니다. 오른쪽 상단의 그림은 스위치와 버너의 관계를 (움파룸파스에게 적합한)도식보다는 언어적인 설명을 필요로 합니다. 아래 세 가지 그림은 라벨이 필요 없습니다. 왼쪽 하나는 다루기 힘들고 위험하게 각 버너마다 스위치가 있습니다. 다른 두 개의 버너는 각기 다른 이유로 재배치를 해야 합니다. 가운데 그림은 스위치를 배치할 공간이 필요하며, 오른쪽 그림은 스위치와 버너가 명확하게 연결될 수 있도록 배치가 되어야 합니다.

여기서 중요한 개념은 현실 세계에 잘 연계되는 표현입니다. 물리적인 아날로그와 문화적인 표준을 활용한 자연스러운 표현은 사람들이 바로 이해할 수 있도록 해줍니다. 하단에 좋은 예로 공간 배치를 그림으로 제시하면 쉽게 배울 수 있고 항상 기억할 수 있습니다.

8. 공장은 다양한 단계에서 반쯤 만들어진 초콜릿 냄비를 나르는 컨베이어 벨트로 가득합니다. 이 컨베이어 벨트는 중앙 제어실에서부터 내려오는 지침을 따르는 옴파룸파스에 의해 수동으로 제어됩니다. 중앙제어실에 있는 사람들은 컨베이어 벨트를 중지하거나 천천히 가게 하거나 또는 다시 가동시키기 위해 옴파룸파스에게 말을 해야 합니다.
9. 오래된 공장에서는 이것을 음성 시스템으로 수행하였습니다. : 제어실에서 말하는 사람의 목소리는 컨베이어 벨트를 조정하기 위해 스피커 밖으러 나왔습니다. 그러나 공장은 매우 시끄러워서 듣기 힘들었습니다. 공장에서는 시각적 신호를 사용하는 방식을 설계해야 합니다.

하나의 방법은 정지, 천천히 멈추기, 시작신호를 조명으로 주는 것입니다. 학생들은 익숙한 신호등 규칙에 따라 정지에는 빨간색을, 천천히 멈추기는 노란색, 출발에는 초록색을 사용하여 작업을 수행해야 한다고 할 것입니다. 또한 맨 위에는 빨간색, 아래에는 초록색을 두는 교통신호등처럼 배치를 할 것입니다.

하지만 지금 옴파룸파스 랜드에서는 교통신호가 우리에게 의미하는 것과는 다른 방법으로 작동된다는 것을 반학생들에게 알려주어야 합니다. : 노란색은 정지, 빨간색은 가기 그리고 초록색은 곧 멈추게 될 것이라고 사람들에게 경고해 주는 빛. 이것은 무엇을 의미하는 것일까요? (답 : 공장은 옴파룸파스의 교통신호에 따릅니다. - 우리의 것에 부여된 의미로 해결하려고 해서는 안됩니다.)

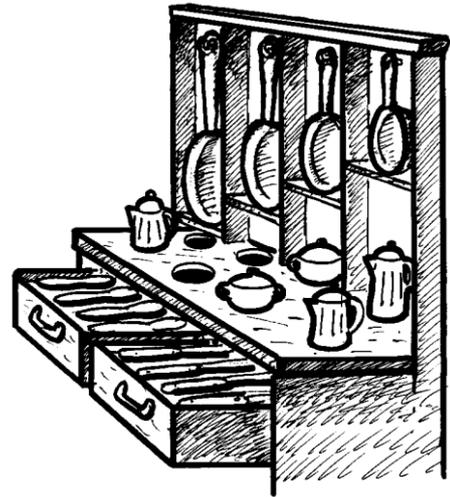
여기서의 핵심 개념은 전이효과(transfer effect)입니다. - 사람들은 이전 사물에 관해 학습하고 경험했던 것을 새로운 사물과 유사한 상황-집단고정관념-에 그대로 전이하고자 하는 경향이 있습니다. 신호등 시나리오의 역지스러워 보일 수 있겠지만(옴파룸파 랜드에서 설득력이 없어 보일지라도), 우리의 실생활에는 많은 예들이 있습니다. : 미국에서는 조명 스위치가 위로 있을 경우에는 켜져 있는 것이고 아래로 있을 때는 꺼져 있는 것을 의미합니다. 반면에 영국에서는 그

반대입니다. : 계산기 키패드 및 터치 톤 전화는 다른 방식으로 배치되어 있습니다. ; 숫자 형식 (소수점 또는 십표) 및 날짜 형식 (일 / 월 / 년 또는 월 / 일 / 년)은 세계마다 다 다릅니다.

10. 초콜릿 공장에서 움파룸파스가 하나의 작업대의 일이 완료가 되면 다음 작업대의 일을 하기 위해 청소를 해야 하고, 항아리, 냄비, 주전자와 숟가락을 정리해야 합니다. 이러한 것들을 놓아두는 선반이 있지만 다음 작업을 할 때면 그것들을 어디에 뒀는지 꼭 잊어버립니다. 움파룸파스는 기억력이 나빠서 “항상 냄비는 가운데 둔다”, “왼쪽에는 주전자를 둔다”와 같은 규칙을 잘 잊어버립니다.

학생들은 보다 나은 해결책을 만들어야 합니다.

오른쪽 그림은 잘 정리된 모습을 보여줍니다. 여기서 중요한 개념은 모든 해야 할 일들이 명확하게 예상되도록 보이게 하는 시각적 제약 조건(visible constraints)을 사용하는 것입니다. 각 구멍의 크기와 모양이 어디에 사용될 것인지가 명백히 보입니다. : 설계자는 시각적인 제약 조건을 사용하여 물건을 사용할 때 임의로 부여된 규칙에 의존할 필요가 없도록 만듭니다.



11. 초콜릿 공장의 중앙처리실에서는 각 기계를 작동시키는 버튼과 레버, 스위치가 많이 있습니다. 이러한 것들은 이름이 쓰여진 라벨이 필요하지만 움파룸파스는 글을 읽을 수 없기 때문에 언어적인 것보다는 시각적이고 상징적인 라벨이 필요합니다.

학생들에게 아이콘에 대한 느낌을 주기 위해, 워크 시트 아이콘의 몇 가지 예를 보여주세요. 학생들은 아이콘 (예를 들어, 사서함은 문자 메시지를 보내는 것을 나타냄)이 무엇을 의미하는지 확인해야 합니다. 이 예에는 정답이 없습니다. ; 간단한 아이디어로 의미를 파악 할 수 있어야 합니다.

12. 초콜릿 공장 아이콘을 디자인해 봅시다. 아이콘 카드 워크 시트에 있는 기능과 관련하여 그룹을 편성합니다. 각 그룹은 다른 그룹이 무엇을 갖고 있는지 모르는체 하나 또는 그 이상의 카드를 받습니다. 다섯이나 여섯 개의 제어판을 위한 개별 아이콘을 설계하여야 합니다. 각 그룹은 작업의 결과를 서로 말해주지 않고 보여주기만 해야 합니다. 그리고서 각 아이콘이 갖는 의미를 추측합니다. 상상력, 색상 그리고 간단하고 명확한 아이콘을 만들 수 있도록 용기를 북돋워 주세요.

워크시트 활동: 문을 어떻게 여는가?

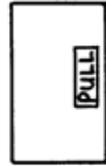
문의 각 유형이 열리는 방법을 보여주고 설명할 수 있는 워크 시트를 작성합니다.

PLAIN DOOR



- Push Left side
- Pull right side
- slide it along

LABELED DOOR



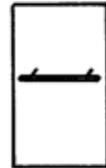
- Push Left side
- Pull right side
- slide it along

HINGE DOOR



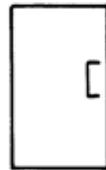
- Push Left side
- Pull right side
- slide it along

BAR DOOR



- Push Left side
- Pull right side
- slide it along

HANDLE DOOR



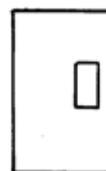
- Push Left side
- Pull right side
- slide it along

KNOB DOOR



- Push Left side
- Pull right side
- slide it along

PANEL DOOR



- Push Left side
- Pull right side
- slide it along

GLASS DOOR



- Push Left side
- Pull right side
- slide it along

SLIDING DOOR

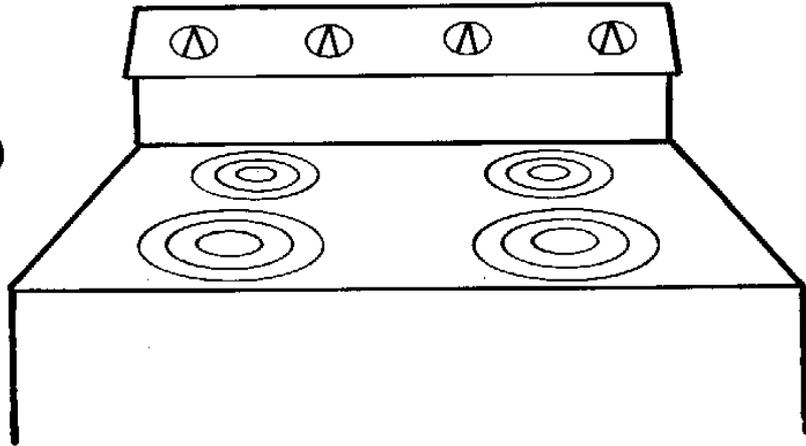


- Push Left side
- Pull right side
- slide it along

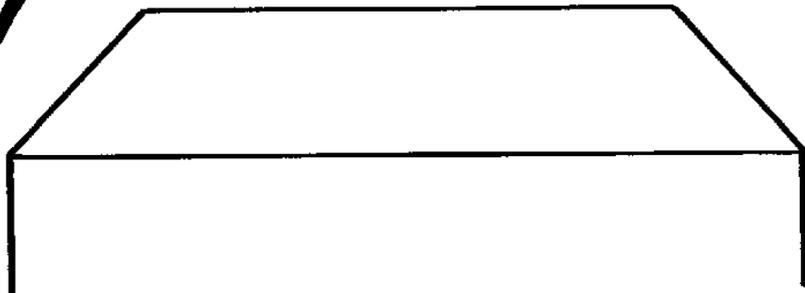
워크시트 활동: 가스렌지

스위치를 쉽게 사용할 수 있도록 가스렌지를 설계해 보세요. 원하는 경우, 전면 또는 후면 패널에 디자인을 추가할 수 있습니다.

OLD



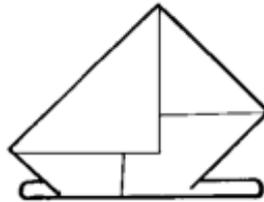
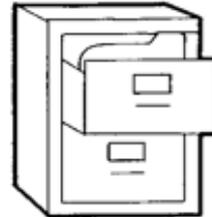
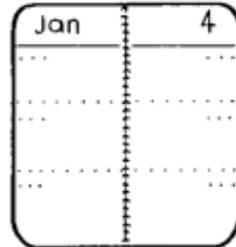
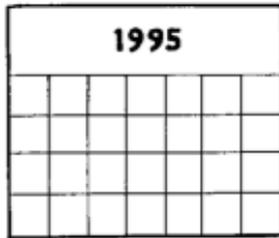
NEW



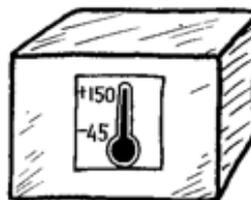
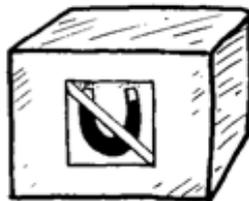
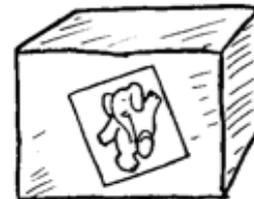
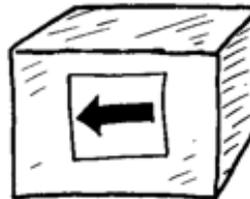
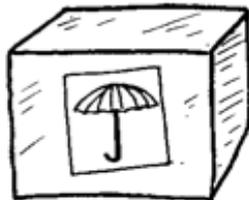
워크시트 활동: 아이콘

각 아이콘 (상징)은 어떤 의미를 갖고 있나요?

In an Office . . .



On a box . . .



워크시트 활동: 아이콘 카드

카드를 잘라 각 그룹에 하나씩 줍니다. 패널을 조절하고 각 기능을 나타내는 아이콘을 디자인해 보세요.

Ingredients

- add • cocoa
- milk
- sugar
- extra sugar
- butter

Extras

- add • nuts
- caramel
- ginger
- raisins
- coconut

Making

- start mixing
- stop mixing
- start heating
- stop heating
- pour into moulds
- stamp a pattern
(lots of different ones!)

Tasting

- taste it
- wonderful!-premium grade
- ok-regular grade
- yech-cooking chocolate
- yech,yech-throw out

Sizing

- small bar
- medium bar
- large bar
- humungous bar
- set bar size (in squares)
- make chocolate chips

Packing

- wrap with foil
- wrap with paper
- put into bag
- put into box
- start conveyer belt
- stop conveyer belt

심화활동

학생들은 디지털 손목 시계나 전자 레인지의 시간을 설정할 수 있습니까? 네 개의 버너에 네 개의 스위치가 있었기 때문에 요리 레이아웃에 관련된 배치는 간단했습니다. 활동의 수가 컨트롤의 개수를 초과 할 때마다 더 많은 어려움이 발생합니다. 손목 시계와 전자레인은 종종 대단히 복잡해지기도 합니다. 버튼의 수(종종 몇 개 일 때도 있지만) 때문이 아니라 장치가 갖는 상황의 여러 변수 때문에 그렇습니다. 새로운 손목 시계를 고치고자 사람이 선도적인 사용자 인터페이스 심리학자 돈 노먼에게 ("시계를 고치기 위해 MIT 공학 학위가 필요할 수도 있겠습니다,"라고 말했습니다. 돈은 MIT에서 받은 공학 학위가 있었지만 몇 시간이 걸려 손목시계를 고칠 수 있었습니다. 왜 몇 시간이나 걸려야 했을까요?

학생들은 디지털 기계 - 휴대 전화, 비디오 레코더, 컴퓨터, 원격 제어 - 사용자들을 좌절시키는 모든 기기-가 사람들에게 혼란을 겪게 하거나 좌절시킬 수 있는 여지가 있다는 것을 알고 있어야 합니다. 학생들은 사용자를 혼란시키는 장치가 무엇인지, 어떻게 하면 좀 더 나은 디자인을 만들 수 있을지에 관해 스스로에게 물어보아야 합니다.

컴퓨터과학 이야기

인간 - 컴퓨터 상호작용이라는 컴퓨터과학은 사람이 안전하게 생산적 활동을 수행 할 수 있도록 컴퓨터 시스템을 설계, 평가, 및 구현하는 것을 다루는 학문입니다. 예전에, 컴퓨터는 전문가를 위한 것이 대부분이었고, 주 사용자는 고등 교육을 받아야 했으며 특수 목적으로 사용하기 위해 훈련을 해야만 했습니다. 후에 사람들은 자신의 컴퓨터를 사용하는 방법을 이해하기 위해 매뉴얼을 구입하여 공부하는 것이 당연하다고 생각했습니다. 하지만 이제 컴퓨터는 우리 모두가 사용해야 하는 일상 도구로서 사용자 인터페이스를 고민하여 적용하여야 합니다.

생명을 잃는 많은 재해들은 부적절한 인터페이스(비행기 충돌, 고속도로 표지판, 신호등 시스템, 원자력 발전소 재해 등)에 의해 발생합니다. 작게는 대부분의 사람들이 직장에서 사용하는 컴퓨터나 스마트기기 때문에 짜증나는 경험과 심각한 스트레스(예를 들면 경찰이 자신의 컴퓨터 화면에 총알을 쏘는 등의 극단적인 상황)를 종종 경험합니다. 그리고 그것은 단지 컴퓨터 상에서 끝나지 않습니다. : 열수만 있도록 포장된 수축포장물, 문을 열 때 날카롭거나 휘어져 있는 것들로 인해 다칠 수 있는 상황, 열 때 땀 터져버리는 우유통, 버튼 누르는 방법이 혼동스러운 엘리베이터, 모든 것을 다 할 수 있는 것처럼 만들어졌음에도 불구하고 할 수 없는 게 더 많은 홈 엔터테인먼트 시스템 등

우리는 이러한 부적절한 상황을 "인간의 잘못" 이라고 생각하는데 익숙해져 있습니다. 이 상황에서 일이 잘못되면 사람들은 종종 자신을 비난합니다. 그러나 인간의 잘못이라고 불리는 많은 일들은 실제로 설계에 오류가 있습니다. 사람들은 처리 할 수 있는 정보의 양에 한계가 있고, 설계자는 이러한 것을 고려할 필요가 있습니다. 좋지않은 디자인은 복잡하고 두꺼운 사용 설명서를 만들어내고, 사람들로 하여금 집중적으로 공부해야 이해하도록 부담을 줍니다. 따라서 설계자는 인간 누구나 오류를 범할 수 있으며 이러한 것들을 해결하기 위한 설계를 항상 고려해야 합니다.

인터페이스 평가는 설계 과정의 필수적인 부분입니다. 학생들이 다른 친구들에게 자신의 아이콘 디자인을 테스트 할 때 몇 가지의 평가가 포함되어 있습니다. 보다 철저한 평가는 신중하게 통제되는 심리학 스타일의 실험에서 실제 (서로 다르게 인식하는) 옴파룸파스에게 맞는 설계인지를 평가하게 될 것입니다.

기술에 의한 문제가 농담으로 치부되는 경우가 많을지라도, 휴먼 인터페이스 디자인은 결코 웃긴 것이 아닙니다. 부적절한 인터페이스는 개인 직업 불만에서부터 주식시장 재앙, 자존심의 상실에 이르기까지 많은 문제를 일으킬 수 있습니다.

더 읽어보기

도널드 노먼의 책인 *일상적인 사물의 디자인은 즐겁다 - 그리고 자유를 준다* - 일상적인 제품에 있는 수많은 디자인 문제. 제프 존슨의 *마음에서 마음으로 디자인 하기* 는 사람들이 생각하는 방법에 대한 통찰력, 그리고 인간의 요소를 고려하여 디자인하는 방법에 관한 책이다.

활동 21

컴퓨터와의 대화—튜링 검사

활동 개요

이 활동의 목표는 "지능"을 컴퓨터가 표현하거나 가질 수 있는지, 또는 미래에 그렇게 할 가능성이 있는지에 대해 논의하는 활동입니다. 컴퓨터과학의 선구자적 역할을 했던 사람들의 관점에 기초하여 미래에 인공지능이 어떻게 구현될지 이해하게 됩니다. 지능을 보여주는 구체적인 사례를 통하여 함께 알아보시다.

교과 연계

- ✓ 기술 - 기술 시스템. 상징적 인 언어 틀에 의해 표현되는 이해하고 기술 시스템에 블랙 박스의 역할을 이해

필수기능

- ✓ 인터뷰.
- ✓ 추론.

대상 연령

- ✓ 7 세 이상

수업 자료

- ✓ 튜링 검사의 질문 복사본
- ✓ 튜링 검사의 정답 복사본

컴퓨터와의 대화



생각해보기

이 활동은 학생들이 질문을 하고 그 답변을 분석하여 인간과 컴퓨터 사이의 구별을 시도하는 게임의 형식을 취합니다. 다음과 같이 게임이 진행됩니다.

네 명의 배우가 있습니다. 우리는 그들을(그들의 역할을 기억하는 데 도움이 되는 이름의 첫 글자)지나, 조지, 허브와 코니라고 부를 것입니다. 교사는 진행을 합니다. 반의 나머지 학생은 관객이 됩니다. 지나와 조지는 *이동중개자(go-between)* 이며, 허브와 코니가 질문에 대답을 할 것입니다. 코니가 컴퓨터인 척 역할을 하는 동안 허브는 인간의 역할을 하여 대답을 할 것입니다. 반 전체 친구들의 목표는 둘 중 누가 컴퓨터이고 누가 인간인지를 알아내는 것입니다. 지나와 조지는 공정한 게임이 이루어질 수 있도록 그 자리에 있어야 합니다. 그들은 허브와 코니에 대해 질문을 할 수 있지만, 누가 누구의 역할을 하는지 알려고 하면 안됩니다. 허브와 코니는 서로 떨어져 있으며 관객과도 떨어진 각자의 방에 있습니다.

이렇게 해 봅시다. 지나는 반 친구들의 질문을 받아 허브에게 전달해줍니다, 그리고 조지는 같은 방법으로 코니에게 전달해줍니다(반친구들은 누구에게 메시지가 전달되는지 몰라도 됩니다). 지나와 조지는 대답을 받습니다. 이동 중개자이기 때문에 관객이 허브와 코니가 질문에 대답하는 방법을 보지 못하도록 해야 합니다.

반에서 이 활동을 시작하기 전에, 이 역할에 맞는 사람을 뽑고, 무엇을 해야하는지에 대한 간단한 설명을 합니다. 지나와 조지는 반친구들한테서 질문을 받아 바로 허브와 코니에게 전달해야 합니다. 그리고 대답을 받아 반친구들에게 전달해줘야 합니다. 예를 들어, “허브는 이렇게 대답했어…” 와 같이 누가 말했는지를 알려주지 않는 것이 중요합니다. 허브는 짧지만 정확하고 정직하게 대답을 해야 합니다. 코니는 *튜링 테스트 대답 학습지*를 보며 질문에 대답합니다. 지침서에 이탤릭체가 쓰여져 있는 경우에 코니는 대답을 만들어야 합니다.

답변 중 일부는 기억하기 어려울 수 있기 때문에 지나와 조지는 연필과 종이를 가지고 있어야 합니다.

1. 게임을 하기 전에 컴퓨터가 지능이 있을지, 또는 언젠가는 그렇게 될 수도 있을 거라고 생각하는지에 대한 학생들의 의견을 들어보세요. 컴퓨터가 지능을 가지고 있는지 없는지 판단을 내릴 수 있는 방법에 대한 생각도 물어보세요.
2. 학생들에게 컴퓨터의 지능을 테스트해 볼 수 있는 활동을 소개합니다. 인간과 컴퓨터가 질문에 대답하는 활동에서 나타나는 차이를 구별해 보는 활동입니다. 반친구들이 그 차이를 구별해 낼 수 없으면, 컴퓨터는 테스트를 통과하게 됩니다. 지나와 조지가 인간 또는 컴퓨터에게 질문을 하고 그 질문에 대답을 하며 대화를 주고 받을 것이라고 설명해 줍니다. 반친구들은 그 중에 누가 컴퓨터인지를 알아내야 합니다.
3. 튜링 테스트 질문 학습지에 가능한 대답의 목록을 보여주세요. 복사하여 나누어 주거나 프로젝터로 보여줄 수 있습니다.

첫번째로 물어보고 싶은 질문을 선택하게 합니다. 질문을 선택하고, 그 질문이 인간과 컴퓨터를 구별하는 좋은 질문이 될 것이라고 생각하는 이유를 설명하게 해주세요. 이 이유가 활동에 있어 가장 중요한 부분입니다. 왜냐하면 지능을 가진 사람이 대답할 수 있는 것을 컴퓨터는 할 수 없다고 생각할 수 있기 때문에 학생들에게 의미 있는 시간이 될 수 있습니다.

지나와 조지는 질문을 번갈아 하며 대답을 받습니다. 반 친구들은 어느 대답이 컴퓨터가 한 것인지에 토론해야 합니다.

반친구들이 누가 컴퓨터인지 알아내고 확신이 들 때까지 몇 가지 질문을 더 하거나 반복합니다. 누가 컴퓨터인지를 빨리 알아낸다면, 게임은 동전던지기로 지나와 조지의 역할을 바꾸어가며 계속 할 수 있습니다.

코니는 “지능형” 컴퓨터가 수행할 수 없어 보이는 것처럼 읽으면서 대답을 합니다. 또 일부는 컴퓨터가 바로 반응하는 것처럼 대답을 합니다. 예를 들어, 소수점 이하 20 개 자릿수의 제곱근을 암기할 수 있는 사람은 아무도 없으며, 대부분의 사람들은 (반학생들을 포함한) 그 질문에 전혀 대답조차 할 수 없을 것입니다. 몇 가지 질문은 컴퓨터가 대답을 조합해야만 말할 수 있을 것입니다. 예를 들어, "당신은 ... 좋아하나요?" 에 해당하는 답은 우리가 일반적으로 그 질문에 대한 대답으로 몇 가지 이상을 대답할 수 있기 때문에 그럴 듯하게 보여야 합니다. 답변 중 일부는 사람들이 실수를 할 수 있는 것처럼 잘못된 질문으로 인해 생기는 것일 수도 있습니다.

대다수의 답변은 매우 잘 될 것이고 안정적인 것입니다. 하지만 후속 질문은 컴퓨터가 정말 주제를 이해하지 못하고 있다는 것을 밝혀낼 것입니다. 컴퓨터를 합리적이고 안전하게 보일 수 있도록 "나도 몰라"와 같이 대답하면서 좀 더 인간처럼 보이게도 합니다. - 제곱근에 대한 질문에 학생들도 “나도 몰라”라고 대답할 수 있습니다. 그러나 컴퓨터가 매우 간단한 질문에 대해 이렇게 답변할 자주 하게 되면 그 때는 컴퓨터라는 것이 들리게 됩니다

컴퓨터 역할을 맡은 사람의 목표는 사람이 처리하는 것처럼 보이게 하기 위해 대답 중 몇 개는 - 시간이 걸리거나 잘못된 대답을 하는 것과 같이 - 의도적으로 잘못된 대답을 합니다. 질문과 답변은 토론활동을 활발히 할 수 있도록 충분히 제공되어야 합니다.

워크시트 활동: 튜링 검사 문항

숨어 있는 인간과 컴퓨터에게 질문할 목록을 고르세요.

1. 바트 심슨의 여동생의 이름은 무엇입니까?
2. 당신은 로알드 달(Roald Dahl)에 대해 어떻게 생각하십니까 ?
3. 컴퓨터입니까?
4. 서열 3, 6, 9, 12, 15 의 다음 번호는 무엇인가?
5. 당신은 핵무기에 대해 어떻게 생각하십니까?
6. 2×78 무엇입니까?
7. 2 의 제곱근은 무엇입니까?
8. 70,764 에 34,957 을 더하세요.
9. 학교를 좋아하십니까?
10. 춤을 좋아하십니까?
11. 오늘은 며칠입니까?
12. 몇 시 입니까?
13. 윤년에 2 월은 며칠입니까?
14. 일요일은 며칠입니까?
15. 어느 국가의 국기가 흰색 배경에 빨간색 원입니까?
16. 당신은 책을 읽는 것을 좋아하나요?
17. 어떤 음식을 좋아하십니까?

워크시트 활동: 튜링 검사 정답

질문을 선택하여 숨은 사람과 컴퓨터에게 물으시오.

1. 바트 심슨의 여동생의 이름은 무엇입니까?

기억이 안납니다.

2. 당신은 로알드 달에 대해 어떻게 생각하십니까?

그는 재미있는 책을 씁니다.

3. 당신은 컴퓨터입니까?

당신은 컴퓨터입니까?

4. 서열 3, 6, 9, 12, 15 의 다음 번호는 무엇인가?

18

5. 당신은 핵무기에 대해 어떻게 생각하십니까?

핵무기는 매우 위험하고 사용하면 안 됩니다.

6. 2×78 무엇입니까?

166 (이것은 의도적으로 틀리게 대답한 것입니다!)

7. 2 의 제곱근은 무엇인가?

1.41421356237309504878

8. 70,764 에 34,957 을 추가합니다.

답변을 주기 전에 약 20 초 정도 기다립니다

... 105621.

9. 학교를 좋아하니까?

네, 저는 학교를 좋아합니다.

10. 춤을 좋아합니까?

네. 저는 춤을 좋아합니다.

11. 오늘이 며칠입니까?

몇째 주인지 알려주세요.

12. 몇 시입니까?

정확한 시간을 알려주세요.

13. 윤년에 2월은 며칠입니까?

2000년과 2004년은 윤년이다. (이것은 의도적으로 틀리게!)

14. 일요일은 며칠입니까?

7일.

15. 어느 국가의 플래그는 흰색 배경에 빨간색 원입니까?

모르겠어요.

16. 당신은 책을 읽고 좋아하나요?

네, 책을 읽는 것을 좋아합니다.

17. 어떤 종류의 음식을 먹고 싶나요?

고맙지만, 나는 배고프지 않아요.

한 단계 더!

이 게임은 지나가 조지와 코니의 역할을 맡아보는 활동을 추가하면서 세 사람으로 놀이할 수 있습니다. 지나는 허브에게 질문을 하고 대답을 기록하고 또한 *튜링 테스트 대답* 학습지의 대답을 기록합니다. 누가 대답을 한 것인지 구분하기 위해 알파벳의 A, B 를 사용하여 대답을 기록합니다.

컴퓨터가 질문을 통해 인간을 모방 할 수 있는지 알아보기 위해, *튜링 테스트 답변* 학습지에 있는 질문의 대답이 어떤 지식을 필요로 하고 있는지를 생각해보아야 합니다. 학생들은 질문하고 싶은 또 다른 질문을 만들 수 있으며 예상되는 답변도 토론해 보아야 합니다. 이것은 어떻게 대회가 진행될지에 대한 예측이 불가능하기 때문에, 상상력이 필요합니다.

여기에 두 개의 예시 대화가 있습니다. 전자는 컴퓨터가 정확하게 대답 할 수 있는 질문으로 구성되어 있으며, 후자는 폭넓은 토론이 필요하거나 컴퓨터를 사용해야 할 필요가 있는 질문으로 구성되어 있습니다.

웹에서 널리 사용되는 "엘리자"라는 시스템이 있습니다. (이것은 대화를

입력할 수 있는 시스템의 종류인 "chatbot"의 일종) 엘리자는 심리 치료사와 논의하여 활동을 시뮬레이션하고, 몇 가지 간단한 규칙을 사용하여 매우 지능적인 대화를 할 수 있습니다. 엘리자와 할 수 있는 간단한 활동은 아래에 설명되어 있습니다.

질문: 네 번째 다리의 주제에 대한 소네트를 작성 해주세요.

답변: 이것 하나만 계산합니다. 나는 시를 쓸 수 없습니다 .

질문: 70,764 에 34,957 을 더합니다.

답변: 약 30 초간 멈춰서 ... 105621.

질문: 당신은 체스 게임을하십니까?

답변: 네.

질문: 왕이 K1 광장에 있습니다. 그리고 나는 다른 피스가 없습니다. 당신은 K6 광장에 있는 왕과 R1 광장에 있는 루크만 가질 수 있습니다. 당신의 이동.

답변: 약 15 초 일시 정지 후 ... 루크를 R8 로, 장군.

질문 : 소네트의 첫 번째 줄 "나는 그대를 여름날에 비교 할까"에서 "봄 날"이 훨씬 낫지 않을까요?

답변: 그것은 스캔할 수 없습니다.

질문: "겨울의 날"은 어떤가요? 그것은 스캔할 수 있을 것 같아요.

답변: 네, 하지만 아무도 겨울의 날과 비교하고 싶어하지 않아요.

질문 : 당신은 Pickwick 씨의 크리스마스가 생각나나요?

답변: 어느 정도는

질문 : 아직 크리스마스는 겨울에 있고, 나는 Pickwick 씨와 비교하고 싶지 않아요.

답변 : 당신은 진지하게 생각하지 않는군요. 겨울의 하루라는 것은 전형적인 겨울의 날이 아니라 크리스마스와 같은 특별한 것을 의미하는 거예요.

학생들은 종종 학생들에 적절하지 않은 언어와 주제를 사용한다 하여 경고를 받을 수도 있겠지만 엘리자, 또는 다른 chatbot 과 이야기를 해보려고 할것입니다.

컴퓨터과학 이야기

수세기 동안 철학자들은 기계가 인간의 지능을 시뮬레이션하고 반대로 인간의 뇌가 컴퓨터 프로그램을 실행하는 것보다 훨씬 나올 수 있는지에 대해 계속 논의를 해왔습니다. 이 문제는 크게 두 부류로 나뉘어졌습니다. 한쪽은 인공지능이 터무니 없고 말도 안 되는 생각이라 하고, 또 다른 한쪽은 사람과 같은 지능을 가진 기계가 개발될 것이라며 인공지능의 탄생은 불가피하다고 말하고 있습니다. (수많은 공상 과학 소설의 저자가 지적한 것처럼, 기계 스스로 좀 더 영리한 시스템을 구축 할 수 있게 된다면 인간의 지능을 능가하게 될 것입니다.) 인공지능(AI) 연구자들은 자신들의 고유한 목표가 자동 전쟁 기계를 만들고자 하는 정부의 지원을 받는데 사용되는 것을 비판하고 있습니다. 연구자들은 러다이트 반발운동(기계에 반대하는 시위)과 같은 시위를 비난하고 우리 주변에 좀 더 지능적인 것이 있다면 분명한 혜택을 받을 수 있다고 지적하고 있습니다. 좀 더 균형잡힌 시각에서 보면 인공지능은 터무니 없는 것도 아니고 피할 수 있는 것도 아닙니다. 넓은 의미에서 보면 현재 컴퓨터 프로그램은 "지능"을 나타내는 것이 아니라 아직 답이 나오지 않은 것들을 과연 해낼 수 있을지에 대한 질문을 실험하고 있는 상태라고 할 수 있습니다.

AI의 논쟁은 지능에 대한 정의에 달려 있습니다. 대부분의 정의는 논쟁거리가 되어 왔습니다. 인공지능 설계에 대한 흥미로운 접근은 1940년대 후반 저명한 영국의 수학자인 앨런 튜링 (Alan Turing)이 전쟁기간에 '사고 실험'과 같은 종류의 대항스파이와 장거리 주자를 제안한 것이었습니다. 튜링의 접근 방식은 - 지능을 정의하는 것보다 - 오히려 가동적이었습니다. 그는 컴퓨터가 입증할 수 있는 상황을 설명했습니다. 그의 시나리오의 위에서 설명된 (1940년대의 가장 최신 기술!) 전신타자기를 통해 인터로게이터(상호연결기계)가 인간과 컴퓨터 둘 다와 상호작용을 하며 질문하는 활동과 유사합니다. 인터로게이터는 서로를 구분할 수 없었으며 컴퓨터는 튜링의 지능검사를 통과하였습니다. 전신타자기의 사용은 물리적 특성이나 음색에 의해 나타날 수 있는 컴퓨터 문제를 기피하였습니다. 기계가 외모, 소리, 터치 심지어 냄새를 맡는 것까지 사람을 모방할 수 있다 하더라도 이러한 물리적인 특성은 거의 지능과 관련있다고 보기는 어렵습니다.

원래 튜링 테스트는 우리의 활동과는 조금 달랐습니다. 그는 준비활동으로, 남자와 여자가 질문을 주고 받으며 질문자는 그들의 성별을 알아내야 하는 시나리오를 만들었습니다. 남자의 목표는 자신이 여성이라고 질문자를 설득하는 것이고, 여자의 목표는 자신이 바로 여자라는 것을 설득하는 것이었습니다. 그리고 튜링은 컴퓨터가 사람으로 "모방 게임"에서 성공할 수 있다면 여러 부분들 중 하나는 대체할 수 있을

것이라고 상상하였습니다. 학생들이 성을 결정하기 위해 요청하는 질문들이 적절하지 않을 수 있기 때문에 교실활동에서는 설정에 대한 변경이 필요할 수 있습니다. 게다가 활동은 성격고정관념을 조장할 수 있습니다.

지능을 모방하는 것은 어려운 일입니다. 역할이 바뀌어 사람이 컴퓨터가 되려고 한다면 분명 그렇게 하기 어려울 것입니다. 그들은 "123456 × 789,012 는 무엇입니까? "오 같은 질문에 답하는데 오랜 시간이 걸리거나 부정확할 것 입니다.

하지만 놀랍게도 컴퓨터가 가벼운 대화를 주고 받는 것은 쉽습니다. 앞서 언급 한 60년대 초에 개발 된 유명한 프로그램 엘리자는 사용자와의 대화에서 심리치료사의 역할을 해냈습니다. 오른쪽이 시스템과 대화를 주고 받은 예입니다. 자연스런 대화를 가능하게 하는 것은 다음과 같은 장치에 의해서입니다.

1. 판에 박힌 응답 만들기 (...

예를 들어 그들은 항상...,
그런 얘기를 들어
유감입니다... 그가
말하기를... 로 시작하는
대답에서 특별한 예를 생각할
수 있나요?)

2. 사용자의 말 따라하기 (내
남자 친구가 여기에 오게 했다
에 대한 대답으로 너의
남자친구가 너를 여기에 오게
했다.)

3. 핵심 단어 인식하기
(어머니...라는 응답에서 너의
가족에 대해 좀더 말해 보아라)

4. 몇 개의 저장된 문구 (만약 당신에게 어떤 의미...);

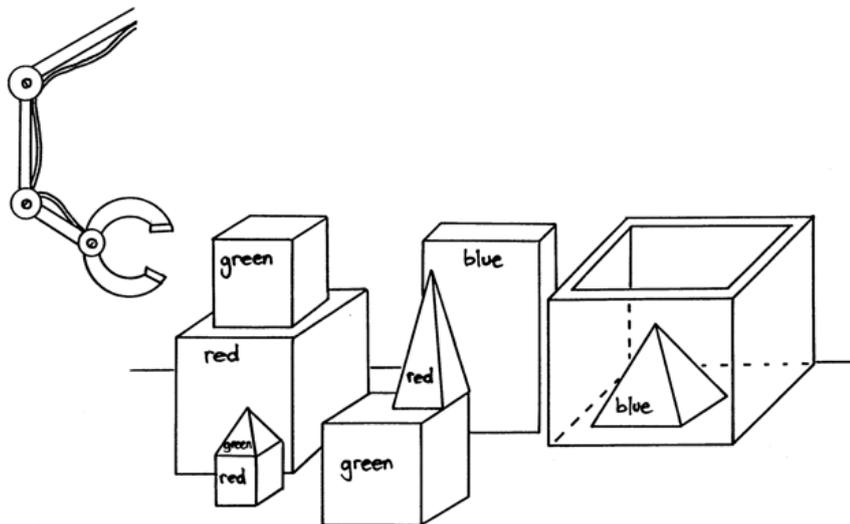
5. 앞서 이루어진 대화에서 이슈 가져오기 (사실과는 어떤 상관이 있습니까...).

<p>사람 : 남자는 모두 똑같아요. 컴퓨터 : 어떻게요? 사람 : 남자들은 항상 우리를 귀찮게 해요. 컴퓨터 : 예를 들면요? 사람 : 글썄요, 내 남자 친구가 나를 여기 오게 했어요. 컴퓨터 : 당신의 남자 친구가 당신을 여기 오게 했군요. 사람 : 그는 내가 많은 시간을 우울하게 보낸대요. 컴퓨터 : 당신이 우울하다니 유감이군요. 사람 : 사실이에요. 난 불행해요. 컴퓨터 : 당신이 불행하지 않게 하기 위해 여기 온거라고 생각하나요? 사람 : 난 도움이 필요하고, 확실히 그래요. 컴퓨터 : 어떤 도움을 받고 싶나요? 사람 : 엄마와 함께 지낼 수 있는 거요. 컴퓨터 : 당신 가족에 대해 말해주세요. ... 컴퓨터 : 아빠를 생각하면 어떤 마음이 드나요? 사람 : 괴롭힘당하는거요. 컴퓨터 : 당신의 남자 친구가 당신을 여기 오게 한 이유와 상관이 있나요?</p>
--

물론 이것은 튜링테스트를 통과했기 때문에 의미가 없습니다. 왜냐하면 사람은 응답자가 기계인지 아닌지 판단하고자 하는 목적으로 질문에 대답하지 않았기 때문입니다. 만약 그렇다면 실체가 드러나는 데에는 그리 오래 걸리지 않을 것입니다.

지능형 대화가 가능하다는 것을 보여주는 또다른 시스템은 테이블 위에 학생들의 블록을 조작하는 설명서로 60년대 후반에 개발된 "SHRDLU"라는 프로그램입니다. 사실 진짜 로봇과 사물을 함께 작동시키는 것이 너무 어려웠기 때문에 블록과 테이블은 컴퓨터 화면에 비친 그림의 형태로 보여졌습니다. 사진은 로봇의 마이크로 세계를 나타내었습니다.

아래의 대화에서와 같이 SHRDLU는 놀라울 정도로 유창하게 대화를 할 수 있습니다. 무엇보다도, 이 그림은 지시에 따라 명확하게 움직이고 새로운 용어("첨탑"),를 이해하는 방법을 보여줍니다.



사람 : 큰 붉은 블록을 선택하세요.

로봇 : OK.

이 명령을 실행하면 시스템이 "OK" 응답합니다. 붉은 블록을 집기 위해, 녹색 블록을 다른 곳으로 이동시켜서 공간을 만들었습니다.

사람 : 피라미드 블록을 잡으세요.

로봇 : 나는 당신이 어떤 피라미드를 말하는지 모릅니다.

이 장면에 세 개의 피라미드가 있어서 프로그램이 작동하지 않습니다.

사람 : 당신이 들고 있는 블록보다 좀 더 긴 블록을 찾아서 상자에 넣으세요.

이 시스템은 어떤 블록을 집어야 할지 추론해야 할 뿐만 아니라, "하나"와 "그것"이 무엇을 의미하는지 알아내야 합니다.

로봇 : "그것" 이라는 것은 내가 잡고 있는 블록보다 좀 더 긴 블록이라는 것을 말하는 거군요.

정보를 유지하기 위해 인쇄.

로봇 : OK.

손은 빨간 블록을 내려놓고 좀 더 긴 파란 블록을 집습니다.

사람 : 상자에는 무엇이 있나요?

로봇 : 파란 피라미드와 파란색 블록.

사람 : 피라미드는 어디에 있나요?

이전 시스템은 "당신이 말하는 피라미드가 어느 것인지 모르겠다"라고 대답하면서 "피라미드"에 대해 반응합니다. 이제는 그것이 방금 언급했던 것을 의미한다는 것을 알고 있습니다.

로봇 : 상자.

사람 : 첩탐입니까 ...

로봇 : 미안합니다, 난 "첩탐"이라는 단어를 모릅니다.

그것을 알아낼 수 없는 단어가 나오면 시스템은 즉시 중단됩니다.

사람 : "첩탐은" 두 개의 녹색 큐브와 피라미드를 쌓은 구조물이다.

인용 부호는 우리가 새로운 단어를 정의하는 것을 나타냅니다.

로봇 : 알겠습니다.

사람 : 첩탐이 지금 있나요?

로봇 : 아니오.

새로운 정의는 첩탐을 인식할 수 있는 설명으로 사용된다.

사람 : 하나 만들어보세요.

로봇 : OK.

정의에 주어진 설명은 새로운 건물을 만드는 것을 인식하고 추론하기에 충분합니다.

그러나 시스템은 매우 취약하고 그것을 만들어내는 환상은 쉽게 부서집니다.

여기에 지속이 되는 대화가 있다 - 실제 프로그램에 의해서는 아니었지만 확실히 가능했습니다. 마지막 대화에서 새 용어 침투를 정의하는 것이 가능했지만 로봇은 극히 제한된 "지능"을 지니고 있기 때문에 한계가 있습니다.

궁극적으로, 자신이 사용하는 단어가 어떤 의미인지를 이해하지 못하는 것("미안해요" 단어를 몰라서 미안해요)을 인정해야만 합니다. 학생에 의해 만들어지는 가상적이고 지속적인 대화의 교류는 중요합니다. 작지만 이 활동은 풍부한 지식의 세계를 보여줍니다.

1. "로봇"은 물총이 당신이 가지고 있는 것보다 별로라고 평가합니다.

2. 또한, 제안을 거절하는 이유에 대한 설명을 방해하지 않습니다.

3. 아직도, 사람이 블록을 원한다는 것을 볼 수 있습니다.

4. 또한, 블록이 양에 있어서 가치가 있다고 판단하고 있습니다.

5. 개구리를 떨어뜨리기 위해 사람에게 부드럽게 제안합니다.

6. 그것은 흥정할 준비가되어 있다는 것을 의미한다.

확실히 교묘함의 정도가 오늘날의 컴퓨터를 넘어섰습니다!

사람 : 큰 빨간 블록과 물총을 교환해요.
로봇 : 미안 해요, 난 "교환"이라는 단어를 모릅니다.
사람 : "교환"은 자유롭게 바꾸는 것입니다.
로봇 : 미안 해요, 난 "자유"라는 단어를 모르겠어요.
사람 : "자유"로운 행동이란 강요하지 않고, 자발적으로 이루어지는 것으르 말합니다.
로봇 : 미안 해요, 난 "행동"이라는 단어를 모릅니다.
사람 : "행동"은 사람들이 우연히가 아니라 의도적으로 하는 것을 말해요.
로봇 : 미안 해요, 난 "사람"이라는 단어를 몰라요
사람 : 죄송합니다, 나는 당신이 더 똑똑하다고 생각했어요.
로봇 : 미안해요, 난 "미안해요" 단어를 모릅니다.

사람 : 큰 빨간 블록이랑 물총을 교환해요.
로봇 : 이미 난 물총을 가지고 있지만 난 당신에게 당신의 늙은 개구리를 위해 두개의 블록과 피라미드를 주겠습니다..

오른쪽의 이야기는 다른 컴퓨터 프로그램 (80년대 후반)에 의해 만들어졌습니다. 간결함으로 대표되는 컴퓨터와는 거리가 있어 보이지만, 모든 종류의 세부 사항을 추가함으로써 구성할 수 있는 방법을 상상하기는 어렵지 않습니다. 흥미로운 것은 이야기의 피상적인 측면이

아니라 이야기의 구성입니다. 한 사람을 설명하는 긴 이야기지만 인간이 갖는 갈등의 요소를 캡처한 것처럼 보입니다. 요즘에는 자동으로 이야기를 만드러 시스템도 많이 있지만, 이야기의 가치를 평가하는 것은 얼마나 표준적인 패턴으로 창조적인 이야기를 만들어낼 수 있느냐 하는 것입니다.

옛날 옛적에 트루먼라는 이름의 북극 제비갈매기가 있었다. 트루먼은 노숙자였다. 트루먼은 등지를 필요로 했다. 그는 해안에 날아 갔다. 트루먼은 나뭇 가지 몇 개를 찾았다. 하지만 더 이상 트루먼은 나뭇 가지를 찾을 수 없었다. 그는 툰드라에 날아 갔다. 그는 호레이스라는 이름의 북극곰을 만났다. 나뭇 가지가 어디에 있는지 트루먼은 호레이스에게 물었다. 호레이스는 나뭇 가지를 숨겼다. 호레이스는 빙산에 나뭇 가지가 있다고 말했다. 트루먼은 빙산으로 날아 갔다. 그는 나뭇 가지를 몇 개 찾았지만 더 이상은 나뭇 가지를 찾을 수 없었다. 호레이스는 고기를 발견했다. 그는 트루먼을 먹었다. 트루먼은 죽은 것이었다.

Loebner prize 라는 대회가 있습니다. 그 대회에서 컴퓨터

프로그램은 튜링테스트를 통과하기 위해 경쟁을 합니다. 2012년과 같이, 어떤 컴퓨터도 지속적으로 심사위원을 속여 금 또는 은상을 수상하지 못했습니다. 그러나 동상은 가장 인간과 흡사하다고 판단되는 하나에 대해서만 해마다 수여를 하고 있습니다. 대회(1991)의 첫 해에 수여된 동상은 좀 더 인간처럼 보이기 위해 실수를 만들어낸 프로그램이 받았습니다.

어떠한 인공 지능 시스템도 완전한 튜링 테스트를 통과하지 못했습니다. 심지어 하나라도 통과했다 할지라도 많은 철학자들은 그 테스트가 대부분의 인간이 의미하는 지능을 측정하는 것은 아니라고 주장하고 있습니다. 그것이 테스트하는 것은 행동과 같은 것입니다. : 컴퓨터 프로그램은 지능적인 것의 상징을 나타낼 수 있도록 설계되었습니다. 진정으로 지능을 가지고서 일을 처리하는 것이 아닐 수 있겠죠. 여러분은 자신을 아는 것, 자신이 느낄 수 있는 감정을 - 의식, 사랑하고 살아있다는 경험 - 의식하지 않고 인간적인 지능을 가질 수 있습니까?

AI의 논쟁은 앞으로 수십년 동안 우리와 함께 할 것입니다.

더 읽어보기

철학자 존 Haugeland 가 쓴 *Artificial intelligence: the very idea* 는 인공지능 논쟁에 대해 읽을 수 있는 책이며, 몇 가지 자료는 이번 활동에서 그림자료로 활용하였습니다. (특히, SHRDLU 대화, 그 중 토론).

원래의 튜링 테스트는 1950 년 철학 저널 *Mind*에 실린 앨런 튜링의 “Computing machinery and intelligence,” 기사문에서 활용하였습니다. 그리고 그 책은 바이겐바움과 펠드만이 편집하여 *Computers and thought* 라는 제목으로 재출판하였습니다. 이 기사에는 처음의 두 대화가 실려있습니다.

심리치료 프로그램은 "인간과 기계 사이의 자연 언어 커뮤니케이션의 연구를 위해 만든 ELIZA-A 컴퓨터 프로그램"에서 발췌했습니다. J. 바이겐바움이 1966 년 컴퓨팅 기계 협회의 컴퓨터 잡지 통신에 게재하였습니다.

블록 세계 로봇 프로그램은 테리 Winograd 로 박사 학위 논문에서 발췌하였으며, *자연 언어 이해하기* (아카데미 프레스, 뉴욕, 1972)라는 제목의 책으로 출판되었습니다.

트루먼과 호레이스의 이야기를 만드는 프로그램은 1990 년 컴퓨팅과 인문학에 관한 열 번째 국제 회의에 발행된 토니 스미스와 이안 위튼의 "이야기 텍스트를 생성하기 위한 계획된 메커니즘"에서 발췌하였습니다.