

고등학교

인공지능 기초

교사용 지도서



(주)삼양미디어

구성과 특징

1. 교육과정

1

2. 내용 체계 및 성취기준

(1) 인공지능의 이해

학년	단원	내용
초등학교	1학년	인공지능이 무엇인지 이해하고 인공지능의 개념과 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
	2학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
중학교	7학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
	8학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
고등학교	1학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
	2학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.

(2) 인공지능의 활용

학년	단원	내용
초등학교	1학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
	2학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
중학교	7학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
	8학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
고등학교	1학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.
	2학년	인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다. 인공지능의 원리를 설명할 수 있다.

2. 연간 지도 계획

2

연간 지도 계획

계획은 64차시

학년	단원	차시	주제
초등학교	1. 인공지능의 이해	1	인공지능의 원리
		2	인공지능의 활용
중학교	1. 인공지능의 이해	1	인공지능의 원리
		2	인공지능의 활용
고등학교	1. 인공지능의 이해	1	인공지능의 원리
		2	인공지능의 활용

교육과정 해설

2022 개정 교육과정의 총론과 성취기준을 제시하여 교육과정 편성에 참고하도록 하였습니다.

연간 지도 계획

64차시(4학점×16주)를 기준으로, 전체적인 구성과 차시를 표로 제시하여 연간 계획이 한눈에 들어오도록 하였습니다.

3

I 인공지능의 이해

단원 개관

인공지능이 발전하여 지능은 스마트폰 속 비사업 분야까지

단원 전개 계획

단원명	차시	주제
1. 인공지능의 이해	1	인공지능의 원리
	2	인공지능의 활용
2. 인공지능의 활용	3	인공지능의 원리
	4	인공지능의 활용

대단원 도입

- 단원 개관** | 대단원에서 배울 내용을 소개하고, 학습 목표와 방향을 명확하게 제시하였습니다.
- 단원 전개 계획** | 소단원명, 학습 주제, 지도 방법 등을 표로 제시하여 수업 계획에 도움이 되도록 하였습니다.

01 인공지능의 원리

4

01 인공지능의 원리

학습 시간: 2시간 (1교시 1시간, 2교시 1시간)

학습 목표

- 인공지능의 원리를 설명할 수 있다.
- 인공지능의 원리를 설명할 수 있다.
- 인공지능의 원리를 설명할 수 있다.

학습 방법

- 교과서 읽기, 관찰, 조사, 실험, 발표, 토론, 발표, 발표, 발표

평가 방법

- 교과서 읽기, 관찰, 조사, 실험, 발표, 토론, 발표, 발표, 발표

차시 시작

교수·학습 계획안 | 소단원마다 교수·학습 계획안을 제시하였습니다.

01 인공지능과 사회 변화

5

01 인공지능과 사회 변화

학습 시간: 2시간 (1교시 1시간, 2교시 1시간)

학습 목표

- 인공지능이 사회에 미치는 영향을 설명할 수 있다.
- 인공지능이 사회에 미치는 영향을 설명할 수 있다.
- 인공지능이 사회에 미치는 영향을 설명할 수 있다.

학습 방법

- 교과서 읽기, 관찰, 조사, 실험, 발표, 토론, 발표, 발표, 발표

평가 방법

- 교과서 읽기, 관찰, 조사, 실험, 발표, 토론, 발표, 발표, 발표

교수 참고 자료와 지도 방법, 문제 해설과 예시 답안

- 교수 참고 자료** | 교사들이 참고할 자료를 교과서 본문 내용과 유기적으로 연계하여 수록하였습니다.
- 지도 방법** | 실제 수업 지도 시 유의할 점이나 도움이 될 내용을 넣었습니다.
- 문제 해설과 예시 답안** | 해 보기, 탐구 활동 등의 문제에 대한 예시 답안과 해설을 수록하였습니다.

지도 방법

인공지능이 현대 사회에 지고 있음을 발견함으로써 가치와 영향력이 중요하게 된다.

예시 답안

사물 인터넷은 사물과 사물이 정보를 주고받는 기술로, 휴대 전화, 자동차, 가전제품 등 다양한 사물이 인터넷을 통해 연결되어 있다. 이는 우리의 생활을 편리하게 만들어 주지만, 개인정보 유출, 사생활 침해 등 다양한 문제를 발생시킬 수 있다.

추가 활동지

6

추가 활동지 2

탐색해결하기

1. 주어진 문제를 해결하는 방법을 찾아라.

2. 주어진 문제를 해결하는 방법을 찾아라.

3. 주어진 문제를 해결하는 방법을 찾아라.

추가 활동지

교과서에 수록하지 못한 활동지를 추가로 제공하여 수업에 활용할 수 있도록 하였습니다.

7

단원 평가 문제

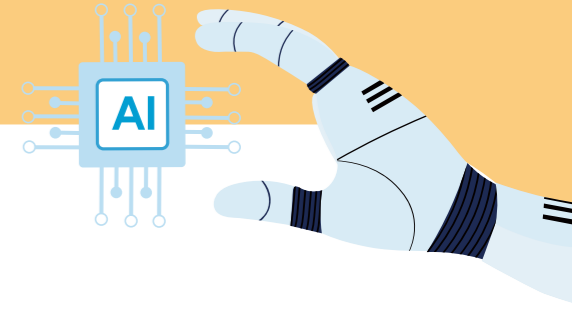
1. 다음 문제를 풀고 답을 설명하라.

2. 다음 문제를 풀고 답을 설명하라.

3. 다음 문제를 풀고 답을 설명하라.

시험 대비용 대단원 평가 문제

각 단원별로 단원을 정리할 수 있는 평가 문제와 상세한 해설을 두어 평가에 활용할 수 있도록 하였습니다.



2022 개정 교육과정 해설 · 성취수준

01	교육과정	08
02	성취수준	15
03	연간 지도 계획	25

I 인공지능의 이해

01	인공지능의 원리	29
02	인공지능과 탐색	41
03	맹목적 탐색과 정보 이용 탐색	53
04	프로젝트 실습 도전! 독도로 가는 길 찾기	69
05	지식의 표현과 추론	85
	· 추가 활동지	99
	· 시험 대비용 대단원 평가 문제	106

II 인공지능과 학습

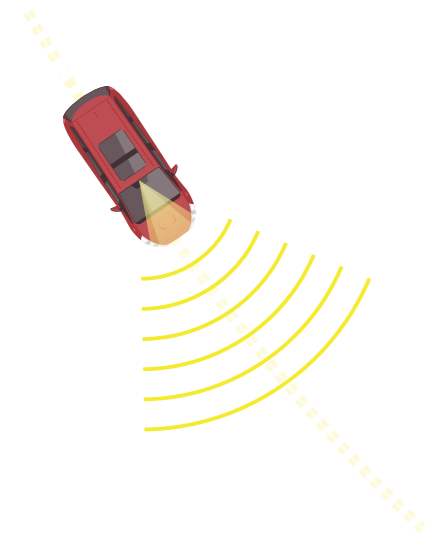
01	기계학습과 데이터	113
02	데이터 전처리와 핵심 속성 추출	127
03	기계학습의 유형과 알고리즘	141
04	기계학습 프로젝트 기계학습을 활용한 문제 해결	155
05	딥러닝의 이해와 활용	177
06	딥러닝 프로젝트 딥러닝으로 손 글씨 분류하기	195
	· 추가 활동지	212
	· 시험 대비용 대단원 평가 문제	220

III 인공지능의 사회적 영향

01	인공지능과 사회 변화	227
02	인공지능과 진로	239
03	인공지능과 윤리	251
	· 추가 활동지	269
	· 시험 대비용 대단원 평가 문제	272

IV 인공지능 프로젝트

01	인공지능과 지속 가능 발전 목표	277
02	인공지능 프로젝트 1 미래의 최저 시급은 얼마일까?	287
03	인공지능 프로젝트 2 이 물은 먹을 수 있는 물일까?	295
	· 시험 대비용 대단원 평가 문제	308





2022 개정

교육과정 해설 · 성취수준

01

교육과정

- ① 성격 및 목표
- ② 내용 체계 및 성취기준
- ③ 교수·학습 및 평가

02

성취수준

- ① 성취기준별 성취수준
- ② 영역별 성취수준

03

연간 지도 계획



01. 교육과정

1 성격 및 목표

가. 성격

‘정보(Informatics)’과는 인공지능으로 정의되는 사회에서 데이터와 정보로 인한 디지털 세상의 변화를 인식하고, 정보의 사회적 가치를 탐구하며, 정보를 처리하는 다양한 원리와 기술에 기반한 컴퓨팅 사고력을 바탕으로 실생활 및 다양한 학문 분야의 문제를 해결하는 능력과 태도를 기르는 교과이다. ‘정보’는 디지털 대전환 시대의 국가·사회적 요구에 부응하여, 컴퓨팅을 활용한 문제 해결을 위해 사회 구성원이 갖추어야 할 필수 역량을 제공한다. ‘정보’의 학문적 기저는 컴퓨터에서 처리되는 데이터와 정보의 원리, 컴퓨팅 시스템을 설계하고 구현하는 기술과 방법, 정보를 다루는 인간 사회에 대한 이해 등을 포괄하고 있다. 즉, ‘정보’는 컴퓨터과학뿐 아니라 데이터 과학, 인공지능, 정보기술, 정보시스템, 소프트웨어 공학 등의 분야를 포괄하는 정보학에 대한 기본 개념과 원리를 기반으로 다양한 학문 분야와 미래 사회의 문제를 해결하는 데 도움이 되는 지식과 기술을 함양한다. 교과의 이러한 특성은 사회 각 분야에서 요구되는 소프트웨어와 인공지능에 대한 기본 소양을 갖추고, 공학뿐만 아니라 자연과학, 인문·사회과학, 예술과 체육 등 다양한 학문 분야에서 문제를 창의적으로 해결하는 인재 양성에 도움을 준다.

‘인공지능 기초’는 학문의 분야 중 컴퓨터과학, 데이터 과학, 정보시스템의 내용을 기반으로, 개인의 삶과 다양한 분야에서 직접적인 영향을 미치고 있는 인공지능에 대한 깊은 이해를 제공하게 된다. 미래 사회의 변화와 불확실성 등으로 인한 진로와 직업의 변화에 능동적으로 대처하며, 인공지능의 주체적 사용자인 학습자가 인공지능을 인간 중심으로 안전하고 책임 있게 사용하는 자기주도성을 갖춘 디지털 민주시민으로 성장하게 한다. 프로젝트 기반의 프로그래밍을 통한 직접 구현, 모델에 대한 평가 등 인공지능에 대한 깊이 있는 학습을 기반으로 대학의 전공과 연계된 기초경험을 제공하도록 한다.

나. 목표

‘인공지능 기초’는 인공지능의 발전에 따른 사회의 변화를 파악하고, 인공지능의 원리에 대한 이해를 바탕으로 인공지능을 다양한 분야의 문제를 창의적으로 해결하기 위한 핵심 도구로서 프로그래밍할 수 있으며, 인공지능의 윤리적 쟁점에 관한 올바른 가치관과 태도를 함양하는 데 중점을 둔다.

- (1) 인공지능에 대한 이해를 바탕으로 실생활의 문제를 인공지능의 관점에서 파악하고, 지능적 판단을 구현하기 위해 탐색과 추론 방식을 적용하는 능력과 태도를 기른다.
- (2) 기계학습을 활용하여 해결할 수 있는 문제를 정의하고, 문제 해결 과정에서 필요한 데이터와 모델을 활용하여 문제를 효과적으로 해결하는 능력과 태도를 기른다.
- (3) 인공지능의 발전에 따른 인간의 삶과 진로의 변화를 탐색하고, 인공지능의 다양한 측면에 대한 비판적인 자세를 바탕으로 인공지능과 관련된 윤리적 문제에 대해 올바른 가치관을 형성할 수 있는 태도를 기른다.
- (4) 인공지능이 다양한 분야와 융합하여 새로운 가치를 창출할 수 있다는 점을 인식하고, 인류가 직면해 있는 문제를 인공지능을 활용하여 해결할 수 있는 능력과 태도를 기른다.

2 내용 체계 및 성취기준

가. 내용 체계

(1) 인공지능의 이해

핵심 아이디어		<ul style="list-style-type: none"> 인공지능은 인간의 지능적인 행동을 모방하는 것으로 실생활에 도움을 준다. 탐색과 추론으로 문제를 해결하는 인공지능을 구현하는 것은 다양한 학문 분야에 활용된다.
범주	구분	내용 요소
범주	지식·이해	<ul style="list-style-type: none"> 인공지능의 원리 인공지능과 탐색 지식의 표현과 추론
	과정·기능	<ul style="list-style-type: none"> 인공지능 기반 문제 해결 사례 탐색하기 탐색 알고리즘을 문제 해결에 적용하기 추론을 통해 새로운 지식을 생성하는 방법 탐색하기
	가치·태도	<ul style="list-style-type: none"> 인공지능의 필요성과 적용 가능성 인식 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세

(2) 인공지능과 학습

핵심 아이디어		<ul style="list-style-type: none"> 기계학습 기반의 인공지능을 구현하기 위해서는 문제 해결에 적합한 데이터와 기계학습 모델을 활용한다. 딥러닝은 다중의 은닉층으로 구성된 인공신경망으로 복잡한 문제를 효과적으로 해결하는 데 활용된다.
범주	구분	내용 요소
범주	지식·이해	<ul style="list-style-type: none"> 기계학습과 데이터 기계학습 알고리즘 인공신경망과 딥러닝
	과정·기능	<ul style="list-style-type: none"> 기계학습을 적용할 문제 정의하기 해결하고자 하는 문제에 적합한 데이터 탐색하기 문제에 적합한 기계학습 알고리즘을 선정하고, 모델 구현하기 딥러닝을 활용한 문제 해결 방법 탐색하고 구현하기
	가치·태도	<ul style="list-style-type: none"> 기계학습에 적용하는 데이터의 중요성 판단 학습을 통한 인공지능의 효과성과 효율성 인식

(3) 인공지능의 사회적 영향

핵심 아이디어	<ul style="list-style-type: none"> 인공지능이 현대 사회에 끼치는 영향력이 커지고 있고, 이에 따라 직업의 변화 속도가 빨라지고 있다. 인공지능을 올바르게 활용하기 위해서는 인공지능에 의해 발생할 수 있는 윤리적 문제에 대한 이해가 필요하다. 	
범주	구분	내용 요소
지식 · 이해	· 인공지능의 발전과 사회 변화	<ul style="list-style-type: none"> 인공지능과 진로 인공지능과 윤리
	· 인공지능으로 해결할 수 있는 사회적 문제 탐색하기	
과정 · 기능	· 인공지능에 의해 변화하는 인간의 삶과 직업의 양상 파악하기	<ul style="list-style-type: none"> 인공지능과 인간의 공존 방안에 대해 탐색하기 인공지능과 관련된 윤리적 딜레마 상황에 대해 논의하기
	· 인공지능과 인간의 공존 방안에 대해 탐색하기	
가치 · 태도	· 진로 및 직업 관점에서 인공지능의 중요성 인식	<ul style="list-style-type: none"> 인공지능의 다양한 측면에 대한 비판적 자세와 윤리적 태도
	· 인공지능의 다양한 측면에 대한 비판적 자세와 윤리적 태도	

(4) 인공지능 프로젝트

핵심 아이디어	<ul style="list-style-type: none"> 인공지능은 다양한 분야와 융합하여 새로운 가치를 창출하는 데 도움을 준다. 인공지능은 지속가능발전목표를 달성하는 데 도움을 준다. 	
범주	구분	내용 요소
지식 · 이해	· 인공지능과 지속가능발전목표	<ul style="list-style-type: none"> 인공지능 문제 해결 절차
	· 인공지능 문제 해결 절차	
과정 · 기능	· 인공지능 프로젝트 주제 탐색하기	<ul style="list-style-type: none"> 인공지능 프로젝트 수행 계획 구안하기 인공지능 소프트웨어 개발 및 평가 방법 설정하기
	· 인공지능 프로젝트 수행 계획 구안하기	
가치 · 태도	· 인류의 지속가능발전에서 인공지능의 중요성 및 가치를 판단하는 태도	<ul style="list-style-type: none"> 인공지능 프로젝트를 수행하는 과정에서 협력적으로 문제를 해결하는 자세 프로젝트를 수행하는 과정에서 윤리 문제 등 사회적 영향 인식
	· 인공지능 프로젝트를 수행하는 과정에서 협력적으로 문제를 해결하는 자세	

나. 성취기준

(1) 인공지능의 이해

- [12인기01-01] 인공지능의 지능적 판단에 대한 이해를 바탕으로 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석한다.
- [12인기01-02] 인공지능에서 탐색의 중요성을 이해하고 문제 해결을 위한 탐색 과정을 설계한다.
- [12인기01-03] 맹목적 탐색과 정보 이용 탐색의 차이를 중심으로 지능적 탐색의 원리를 파악한다.
- [12인기01-04] 지능적 탐색이 필요한 문제를 찾아보고 문제 해결을 위해 정보 이용 탐색 알고리즘을 적용한다.
- [12인기01-05] 규칙과 사실을 활용하여 지식을 표현하고, 새로운 지식을 추론하여 생성한다.

(가) 성취기준 해설

- [12인기01-01] 인공지능의 개념과 특성을 이해하고, 튜링 테스트를 통해 인공지능의 지능적 판단에 대해 고찰하며, 인공지능이 활용된 최신 사례를 분석하여 인공지능의 활용 범위와 중요성을 설명할 수 있어야 한다.
- [12인기01-04] 지능적 탐색을 적용할 수 있는 퍼즐이나 길찾기 문제를 탐색하고, 최상 우선, A* 등의 정보 이용 탐색 방법을 적용한 인공지능 프로그램을 개발할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 인공지능의 다양한 사례를 탐색하여 실제로 많은 분야에 인공지능이 활용되고 있음을 인식하고, 인공지능을 구현하는 방법으로서 탐색과 추론 방식에 대한 이해를 바탕으로 문제를 해결할 수 있는 능력을 함양하도록 교수·학습을 계획한다. 특히 탐색 방식으로 문제를 해결하는 과정에서는 탐색 알고리즘을 적용한 프로그램을 개발하는 방식으로 활동을 구성하도록 한다.
- 실제 사례를 통해 인공지능의 개념을 정확하게 이해하고 있는지, 인공지능의 중요성을 설명할 수 있는지를 평가한다. 또한, 문제 해결에 적합한 탐색 알고리즘을 선택하여 적용하였는지, 개발한 인공지능 프로그램이 문제를 해결하는 데 효과적이었는지에 중점을 두어 평가하도록 한다.

(2) 인공지능과 학습

- [12인기02-01] 기계학습을 적용할 문제를 정의하고, 문제 해결에 필요한 데이터를 선정하여 수집한다.
- [12인기02-02] 수집한 데이터를 가공하여 핵심 속성을 추출한다.
- [12인기02-03] 문제 해결에 적합한 기계학습의 유형과 알고리즘을 선정한다.
- [12인기02-04] 훈련 데이터를 이용하여 학습을 진행하고, 테스트 데이터를 사용하여 성능을 평가한다.
- [12인기02-05] 인공신경망과 딥러닝의 특성에 대한 이해를 바탕으로 활용 분야를 탐색한다.
- [12인기02-06] 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하고, 성능을 평가한다.

(가) 성취기준 해설

- [12인기02-01] 전통적 프로그래밍과 대비되는 기계학습의 특성을 이해하여 기계학습 기반의 인공지능을 적용할 문제를 판단하고, 문제 해결에 적합한 데이터를 수집할 수 있어야 한다. 이 과정에서 나타날 수 있는 데이터의 편향이 인공지능의 학습에 영향을 미칠 수 있음을 고려하여 데이터의 수집과 활용에 공정성을 추구할 수 있어야 한다.
- [12인기02-02] 기계학습에 사용할 데이터를 수집한 후, 결측치와 이상치의 유무를 파악하고 문제 해결에 필요한 속성이 무엇인지 선별한 다음 문제 해결에 적합한 형태로 전처리할 수 있어야 한다.
- [12인기02-03] 기계학습의 유형을 지도학습과 비지도학습, 강화학습 등으로 구분하고, 해결해야 하는 문제의 특성을 고려하여 문제 해결에 적합한 유형을 선정할 수 있어야 한다. 또한, 분류, 예측, 군집 등에 활용하는 기계 학습 알고리즘을 이해하여 문제 해결에 적합한 알고리즘을 선정할 수 있어야 한다.
- [12인기02-06] 딥러닝을 기반으로 하는 컴퓨터 비전, 음성 인식, 자연어처리 기술을 구현하는 방식을 이용하고, 적합한 딥러닝 기술을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결할 수 있어야 한다. 문제 해결에 있어 딥러닝을 활용해 문제를 해결하는 경우와 다른 방법으로 문제를 해결한 경우의 차이점을 설명할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 해결하고자 하는 문제에 따라 기계학습의 사용이 적합한지를 판단할 수 있도록 적절한 문제 상황을 선정하도록

하며, 기계학습에서 데이터의 중요성을 인식하고, 데이터를 수집하는 과정에서 데이터가 편향되지 않게 충분히 안내하도록 한다.

- 기계학습과 딥러닝을 구현하는 과정에서 학습자의 수준에 따라 코드 작성의 방식을 달리 구성할 수 있다. 프로그래밍 역량이 낮은 학생은 이미 작성된 코드를 실행시켜 코드를 이해한 후 차츰 코드 일부를 수정하는 방식으로 학습할 수 있도록 하고, 프로그래밍 역량이 높은 학생은 작성이 완료되지 않은 코드를 스스로 완성하여 실행시킬 수 있도록 하는 등 교수·학습 단계를 세부적으로 구성하여 학습자의 흥미가 유지되도록 한다. 또한, 기계학습과 딥러닝을 코드로 구현하는 학습을 원활하게 진행하기 위해, 다양한 라이브러리 활용 방법을 충분히 안내한다.
- 문제 해결에 사용된 데이터와 기계학습 유형, 기계학습 알고리즘이 해결하고자 하는 문제의 특성에 비추어 적합한지를 평가한다. 또한, 이미 개발되어 있는 딥러닝 프로그램 코드의 활용을 권장하되 실생활 및 다양한 학문 분야의 문제를 스스로 생각하여 해결할 수 있도록 교수·학습을 구성하고, 학생의 학습 과정과 창의적인 아이디어를 중심으로 평가 루브릭을 작성하여 평가하도록 한다.

(3) 인공지능의 사회적 영향

[12인기03-01] 인공지능의 발전으로 인한 사회 변화를 살펴보고, 인공지능으로 해결할 수 있는 사회적 문제를 분석한다.

[12인기03-02] 인공지능에 의해 변화하는 인간의 삶과 직업의 양상에 대해 이해하고 진로를 탐색한다.

[12인기03-03] 인공지능에 대한 비판적 자세를 바탕으로 인공지능과 인간의 공존 방안을 도출한다.

[12인기03-04] 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리 관점에서 분석한다.

(가) 성취기준 해설

- [12인기03-03] 인공지능에 대한 일방적인 수용 또는 거부보다는, 비판적인 자세를 바탕으로 인간과 공존해야 하는 존재로서 인공지능의 역할을 제시할 수 있어야 한다. 인공지능을 대하는 인간의 태도에 대한 윤리적 고찰을 통해 바람직한 공존 방식을 규정할 수 있어야 한다.
- [12인기03-04] 인간의 편향성에 대한 이해를 바탕으로, 알고리즘과 데이터의 편향성으로 인해 인공지능이 사회에 끼치는 영향을 이해하고 인공지능으로 인한 딜레마 상황에서 윤리적인 판단과 선택을 할 수 있어야 한다. 현재까지 정의되어진 인공지능 윤리(지침)와 관련하여 인공지능 개발자, 사용자, 운영·관리자 관점을 살펴보고, 인공지능 기술 활용으로 발생 가능한 윤리적 쟁점에 대하여 사회적 책임과 공정성의 가치를 판단할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 인공지능의 사회적 가치와 영향력이 중요하다는 것과 새롭게 고려해야 할 윤리적 쟁점에 대한 충분한 사회적 논의 과정이 필요하다는 것을 전제로, 인공지능 사회의 구성원으로서 갖추어야 할 인공지능 윤리의식과 가치 판단 능력을 함양하여 인공지능 시대의 사회적 책임과 공정성을 실천할 수 있도록 하는 데 중점을 두고 교수·학습을 구성하도록 한다.
- 인공지능의 발전으로 인한 사회의 변화를 조사하고 인공지능에 의해 변화할 미래 사회의 모습을 예측하는 과정에서 자신의 진로를 구체적으로 설계하기 위해 충분한 자료를 제공하도록 한다.
- 인공지능과 관련한 윤리적 쟁점 사례를 조사하고, 구체적인 사안을 주제로 하는 토론 활동의 과정에서 인공지능 윤리의 중요성과 인공지능과 인간의 관계를 올바르게 설명할 수 있는지를 평가하도록 한다. 이 과정에서 근거를 바탕으로 자신의 주장을 펼치고 타인의 견해를 존중하는 성숙한 토론 문화를 조성하도록 한다.

(4) 인공지능 프로젝트

[12인기04-01] 지속가능발전목표를 해결하기 위해 인공지능을 적용할 수 있는 방안을 탐색하고, 인공지능 프로젝트 활동에 적합한 주제를 도출한다.

[12인기04-02] 인공지능 문제 해결 과정에 기반하여 프로젝트 수행 계획을 구안한다.

[12인기04-03] 인공지능 프로젝트를 수행하는 과정에서 협력적인 문제 해결 자세를 바탕으로 인공지능 소프트웨어를 개발한다.

[12인기04-04] 인공지능의 사회적 영향을 고려하여 인공지능 소프트웨어를 개발하고, 평가 결과를 반영하여 성능을 개선한다.

(가) 성취기준 해설

- [12인기04-01] 지속가능발전목표는 2015년 유엔 총회에서 결의한 것으로 총 17개의 주요 목표와 169개의 세부 목표로 구성되어 있다는 것을 이해하고, 17개의 주요 목표를 통해 현재 인류가 직면하고 있는 위기가 무엇인지 인식할 수 있어야 한다. 인공지능을 활용하여 달성할 수 있는 목표에는 어떤 것이 있는지를 살펴보고 인공지능 프로젝트 활동에 적합한 주제를 도출할 수 있어야 한다.
- [12인기04-02] 문제 정의, 문제 해결에 적합한 데이터 수집 및 전처리, 기계학습 유형과 알고리즘 선정, 기계학습을 통한 모델 생성, 성능 평가 및 수정 등의 인공지능 문제 해결 과정에 따라 프로젝트 수행 계획을 구안할 수 있어야 한다.

(나) 성취기준 적용 시 고려 사항

- 인공지능 프로젝트를 위한 주제 선정 과정에서 인류가 직면해 있는 다양한 위기를 충분히 인식할 수 있도록 하고, 이러한 위기를 극복하는 다양한 방법 중 하나로 인공지능이 중요한 역할을 수행할 수 있다는 점을 강조하도록 한다. 또한, 인공지능을 활용하는 방법적인 측면에서 직접 구현하기가 어려운 부분이 있더라도 학습자의 아이디어를 존중하여 다양한 생각이 발산될 수 있는 활동을 구성하도록 한다.
- 인공지능 프로젝트를 수행하는 과정에서 인공지능 소프트웨어의 완성도뿐만 아니라 주제의 적절성, 문제 해결 방식의 창의성, 다른 사람과의 협업 능력 등 다각적인 측면을 골고루 반영하여 평가할 수 있는 루브릭을 제작하도록 한다.

3 교수·학습 및 평가

가. 교수·학습

(1) 교수·학습의 방향

- (가) 실제적인 삶의 맥락에서 컴퓨팅을 통해 문제를 해결하도록 하는 학습 과제를 제시하여 학습자가 과제를 스스로 해결하는 과정에서 자연스럽게 컴퓨팅 사고력, 디지털 문화 소양, 인공지능 소양을 함양할 수 있도록 지도한다.
- (나) 학습자의 흥미와 다양성을 고려하여 학습 소재, 학습 환경 및 학습 과정에 대한 선택의 기회를 제공하고, 교수·학습의 설계 과정에 학습자 참여 기회를 증진하는 등 학습자 맞춤형 교수·학습을 통해 역량 함양을 위한 깊이 있는 학습 지도 방안을 구성한다.
- (다) 인공지능 기초 과목의 지식·이해, 과정·기능을 활용하여 민주시민교육, 생태전환 교육 등 현 시대가 당면한 여러 사회문제와 더불어 지속가능발전 등의 범교과 주제를 교수·학습 과제로 제시하여 주도성 있는 문제 해결 경험을 제공한다.

02. 성취수준

(라) '인공지능 기초' 과목의 교수·학습은 인공지능의 원리에 대한 이해를 바탕으로 문제를 효율적으로 해결하는 역량을 향상하는 데 중점을 둔다. '인공지능 기초' 과목 내의 내용 영역, 다른 교과 및 비교과 활동과의 융합을 통해 학생의 인공지능 소양이 다양한 분야에 전이될 수 있도록 한다.

(2) 교수·학습 방법

- (가) 교과 역량을 함양하기 위해 문제기반학습, 프로젝트 기반학습, 디자인기반학습, 짝 프로그래밍, 탐구학습 등 각 영역의 핵심 아이디어를 습득하는 데 적절한 교수·학습 방법을 선택하여 활용한다.
- (나) 학습자 개인별로 학습하는 속도가 다양할 수 있음을 고려하고, 최소 성취수준을 보장할 수 있도록 학습관리시스템(LMS)을 활용하여 온라인 학습자료를 제작 및 제공함으로써 학습 격차를 최소화하도록 노력한다.
- (다) 학습자의 디지털·인공지능 소양에 대해 선제적으로 파악하고 학생의 현재 수준에 맞는 방식으로 교수·학습 활동을 구성한다. 프로그래밍을 통한 인공지능 프로그램의 개발 활동을 중심으로 하되, 인공지능 개념 이해에 도움이 되는 소프트웨어나 인공지능 관련 라이브러리를 적극적으로 활용하여 최소 성취수준을 보장하도록 한다.
- (라) 디지털 교육 환경에 적응할 수 있도록 온오프라인 연계 수업, 다양한 디지털 도구의 활용 등을 통해 디지털 도구에 대한 인지적 부담은 최소화하고, 활용에 대한 경험은 높일 수 있도록 수업을 구성한다.
- (마) '인공지능 기초' 과목과 연계된 진로 및 직업을 안내하고 학생이 자신의 진로를 탐색할 수 있도록 교수·학습 과정을 구성한다.

나. 평가

(1) 평가의 방향

- (가) 평가 항목은 컴퓨팅 사고력, 디지털 문화 소양, 인공지능 소양의 하위 요소를 기반으로 구체화한다.
- (나) 평가 내용은 지식·이해뿐 아니라, 과정·기능, 가치·태도의 측면 등을 다면적으로 반영하고 과정을 중시하는 평가를 통해 학생의 성장과 발달을 돕는 평가를 실현한다.
- (다) 구체적인 평가 루브릭을 학생과 함께 구성하는 과정을 통해 학생이 자신의 학습 수준을 파악하고 스스로 학습을 성찰할 수 있는 기회를 제공하여, 적극적이고 능동적인 학습이 이루어지도록 한다.
- (라) 인공지능 프로그램 구현에 프로그래밍 활동이 포함되나 프로그래밍 내용보다는 인공지능의 구현과 활용, 인공지능 활용에 대한 인식에 초점을 맞추어 평가한다.

(2) 평가 방법

- (가) 성취기준을 분석하고 재구성하여 지필평가 및 수행평가, 관찰 평가, 서·논술형 평가 등을 활용하는 종합적인 평가 포트폴리오를 계획하고, 교사 평가, 자기 평가, 동료 평가 등과 같은 다면적 평가를 실행하여 학생 성장에 기여할 수 있는 결과를 제공한다.
- (나) 평가 내용이나 방법에 따라 다양한 디지털 도구(프로그램 자동 평가시스템(online judge 등), 학습관리시스템(LMS) 등)을 활용할 수 있으며, 평가 이전에 학생이 디지털 도구를 다룰 수 있도록 교육하여 평가의 불이익이 없도록 계획한다.
- (다) 인공지능에 관련된 평가는 다양한 방식으로 나타날 수 있으므로 정량적 평가와 정성적 평가 내용을 명확하게 구분한다. 특히 모델 학습과 적용이 반복적으로 이루어지는 프로젝트의 경우 평가 결과가 학생 활동에 즉시 피드백되어 결과물의 개선으로 나타날 수 있도록 평가를 계획한다.

1 성취기준별 성취수준

1 인공지능의 이해

성취기준	성취기준별 성취수준
[12인기01-01] 인공지능의 지능적 판단에 대한 이해를 바탕으로 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석한다.	A 인공지능의 개념과 특성을 올바르게 설명하고, 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 구체적으로 비교·분석하여 인공지능의 필요성과 적용 가능성을 내면화할 수 있다.
	B 인공지능의 개념과 특성을 올바르게 설명하고, 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석하여 인공지능의 필요성과 적용 가능성을 인식할 수 있다.
	C 인공지능의 개념과 특성을 설명하고, 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석하여 인공지능의 필요성과 적용 가능성을 인식할 수 있다.
	D 인공지능의 개념과 특성을 설명하고, 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 부분적으로 비교·분석하여 인공지능의 필요성과 적용 가능성을 수용할 수 있다.
	E 인공지능의 개념과 특성을 인지하고, 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 부분적으로 비교·분석하여 인공지능의 필요성과 적용 가능성을 수용할 수 있다.
[12인기01-02] 인공지능에서 탐색의 중요성을 이해하고 문제 해결을 위한 탐색 과정을 설계한다.	A 인공지능에서 탐색의 중요성을 정확하게 설명하고, 문제 해결을 위한 탐색 과정을 구체적으로 설계할 수 있다.
	B 인공지능에서 탐색의 중요성을 정확하게 설명하고, 문제 해결을 위한 탐색 과정을 설계할 수 있다.
	C 인공지능에서 탐색의 중요성을 설명하고, 문제 해결을 위한 탐색 과정을 설계할 수 있다.
	D 인공지능에서 탐색의 중요성을 설명하고, 문제 해결을 위한 탐색 과정을 부분적으로 설계할 수 있다.
	E 인공지능에서 탐색의 중요성을 인지하고, 문제 해결을 위한 탐색 과정을 부분적으로 설계할 수 있다.
[12인기01-03] 맹목적 탐색과 정보 이용 탐색의 차이를 중심으로 지능적 탐색의 원리를 파악한다.	A 맹목적 탐색과 정보 이용 탐색의 차이를 정확하게 설명하고, 다양한 사례에서 지능적 탐색의 원리를 파악할 수 있다.
	B 맹목적 탐색과 정보 이용 탐색의 차이를 정확하게 설명하고, 지능적 탐색의 원리를 파악할 수 있다.
	C 맹목적 탐색과 정보 이용 탐색의 차이를 설명하고, 지능적 탐색의 원리를 파악할 수 있다.
	D 맹목적 탐색과 정보 이용 탐색의 차이를 설명하고, 지능적 탐색의 원리를 부분적으로 파악할 수 있다.
	E 맹목적 탐색과 정보 이용 탐색의 차이를 인지하고, 지능적 탐색의 원리를 부분적으로 파악할 수 있다.

성취기준	성취기준별 성취수준
[12인기01-04] 지능적 탐색이 필요한 문제를 찾아보고 문제 해결을 위해 정보 이용 탐색 알고리즘을 적용한다.	A 정보 이용 탐색의 적용 방법을 설명하고, 이들을 적용하기에 적합한 문제에 적용하며, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 생활화할 수 있다.
	B 정보 이용 탐색의 적용 방법을 설명하고, 이들을 적용하기에 적합한 문제에 적용하며, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 경험할 수 있다.
	C 정보 이용 탐색의 원리를 설명하고, 이들을 적용하기에 적합한 문제에 적용하며, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 경험할 수 있다.
	D 정보 이용 탐색의 원리를 설명하고, 이들을 문제에 적용하며, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 수용할 수 있다.
	E 정보 이용 탐색의 원리를 인지하고, 이들을 문제에 적용하며, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 수용할 수 있다.
[12인기01-05] 규칙과 사실을 활용하여 지식을 표현하고, 새로운 지식을 추론하여 생성한다.	A 규칙과 사실, 자료와 정보 그리고 지식을 구분하여 정확하게 설명하고, 다양한 방식으로 새로운 지식을 추론하여 생성할 수 있다.
	B 규칙과 사실, 자료와 정보 그리고 지식을 구분하여 정확하게 설명하고, 새로운 지식을 추론하여 생성할 수 있다.
	C 규칙과 사실, 자료와 정보 그리고 지식을 설명하고, 새로운 지식을 추론하여 생성할 수 있다.
	D 규칙과 사실, 자료와 정보 그리고 지식을 설명하고, 지식을 추론하여 생성할 수 있다.
	E 규칙과 사실, 자료와 정보 그리고 지식을 인지하고, 지식을 추론하여 생성할 수 있다.

2 인공지능과 학습

성취기준	성취기준별 성취수준
[12인기02-01] 기계학습을 적용할 문제를 정의하고, 문제 해결에 필요한 데이터를 선정하여 수집한다.	A 기계학습의 개념과 원리를 정확하게 설명하고, 기계학습을 적용할 문제를 정의하며, 문제 해결에 적합한 데이터를 수집하고, 데이터의 수집과 활용에 공정성을 추구하려는 자세를 내면화할 수 있다.
	B 기계학습의 개념과 원리를 정확하게 설명하고, 기계학습을 적용할 문제를 정의하며, 데이터를 수집하고, 데이터의 수집과 활용에 공정성을 추구하려는 자세를 인식할 수 있다.
	C 기계학습의 개념과 원리를 설명하고, 기계학습을 적용할 문제를 정의하며, 데이터를 수집하고, 데이터의 수집과 활용에 공정성을 추구하려는 자세를 인식할 수 있다.
	D 기계학습의 개념과 원리를 설명하고, 기계학습을 적용할 문제를 정의하며, 데이터를 일부 수집하고, 데이터의 수집과 활용에 공정성을 추구하려는 자세를 수용할 수 있다.
	E 기계학습의 개념을 인지하고, 기계학습을 적용할 문제를 정의하며, 데이터를 일부 수집하고, 데이터의 수집과 활용에 공정성을 추구하려는 자세를 수용할 수 있다.

성취기준	성취기준별 성취수준
[12인기02-02] 수집한 데이터를 가공하여 핵심 속성을 추출한다.	A 결측치, 이상치, 독립변수, 종속변수의 개념을 정확하게 설명하고, 데이터를 가공하여 정확하게 핵심 속성을 추출하며, 기계학습에 적용하는 데이터의 중요성을 내면화할 수 있다.
	B 결측치, 이상치, 독립변수, 종속변수의 개념을 정확하게 설명하고, 데이터를 가공하여 핵심 속성을 추출하며, 기계학습에 적용하는 데이터의 중요성을 인식할 수 있다.
	C 결측치, 이상치, 독립변수, 종속변수의 개념을 설명하고, 데이터를 가공하여 핵심 속성을 추출하며, 기계학습에 적용하는 데이터의 중요성을 인식할 수 있다.
	D 결측치, 이상치, 독립변수, 종속변수의 개념을 설명하고, 데이터를 가공하여 핵심 속성을 부분적으로 추출하며, 기계학습에 적용하는 데이터의 중요성을 수용할 수 있다.
	E 결측치, 이상치, 독립변수, 종속변수의 개념을 인지하고, 데이터를 가공하여 핵심 속성을 부분적으로 추출하며, 기계학습에 적용하는 데이터의 중요성을 수용할 수 있다.
[12인기02-03] 문제 해결에 적합한 기계학습의 유형과 알고리즘을 선정한다.	A 기계학습의 유형과 알고리즘의 적용 방법을 설명하고, 문제 해결에 적합한 기계학습 유형과 알고리즘을 선정할 수 있다.
	B 기계학습의 유형과 알고리즘의 적용 방법을 설명하고, 문제 해결을 위한 기계학습 유형과 알고리즘을 선정할 수 있다.
	C 기계학습의 유형과 알고리즘의 개념을 설명하고, 문제 해결을 위한 기계학습 유형과 알고리즘을 선정할 수 있다.
	D 기계학습의 유형과 알고리즘의 개념을 설명하고, 문제 해결을 위한 기계학습 유형을 선정할 수 있다.
	E 기계학습의 유형과 알고리즘의 개념을 인지하고, 문제 해결을 위한 기계학습 유형을 선정할 수 있다.
[12인기02-04] 훈련 데이터를 이용하여 학습을 진행하고, 테스트 데이터를 사용하여 성능을 평가한다.	A 훈련 및 테스트 데이터의 적용 방법을 설명하고, 훈련 데이터를 이용하여 올바르게 학습을 진행한 후 테스트 데이터를 사용하여 적절하게 성능을 평가할 수 있다.
	B 훈련 및 테스트 데이터의 적용 방법을 설명하고, 훈련 데이터를 이용하여 학습을 진행한 후 테스트 데이터를 사용하여 성능을 평가할 수 있다.
	C 훈련 및 테스트 데이터의 개념을 설명하고, 훈련 데이터를 이용하여 학습을 진행한 후 테스트 데이터를 사용하여 성능을 평가할 수 있다.
	D 훈련 및 테스트 데이터의 개념을 설명하고, 훈련 데이터를 이용하여 부분적으로 학습을 진행할 수 있다.
	E 훈련 및 테스트 데이터의 개념을 인지하고, 훈련 데이터를 이용하여 부분적으로 학습을 진행할 수 있다.
[12인기02-05] 인공지능경망과 딥러닝의 특성에 대한 이해를 바탕으로 활용 분야를 탐색한다.	A 인공지능경망과 딥러닝의 원리를 설명하고, 딥러닝의 활용 분야를 다양하게 탐색할 수 있다.
	B 인공지능경망과 딥러닝의 원리를 설명하고, 딥러닝의 활용 분야를 탐색할 수 있다.
	C 인공지능경망과 딥러닝의 활용 방법을 설명하고, 딥러닝의 활용 분야를 탐색할 수 있다.
	D 인공지능경망과 딥러닝의 활용 방법을 설명하고, 딥러닝의 활용 분야를 부분적으로 탐색할 수 있다.
	E 인공지능경망과 딥러닝의 개념을 인지하고, 딥러닝의 활용 분야를 부분적으로 탐색할 수 있다.

성취기준	성취기준별 성취수준
[12인기02-06] 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하고, 성능을 평가한다.	A 컴퓨터 비전, 음성 인식, 자연어처리의 원리를 설명하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 효과적으로 해결한 후 성능을 평가하며, 인공지능의 효과성과 효율성을 내면화할 수 있다.
	B 컴퓨터 비전, 음성 인식, 자연어처리의 원리를 설명하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결한 후 성능을 평가하며, 인공지능의 효과성과 효율성을 인식할 수 있다.
	C 컴퓨터 비전, 음성 인식, 자연어처리의 활용 방법을 설명하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결한 후 성능을 평가하며, 인공지능의 효과성과 효율성을 인식할 수 있다.
	D 컴퓨터 비전, 음성 인식, 자연어처리의 활용 방법을 설명하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 부분적으로 해결하며, 인공지능의 효과성과 효율성을 수용할 수 있다.
	E 컴퓨터 비전, 음성 인식, 자연어처리의 개념을 인지하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 부분적으로 해결하며, 인공지능의 효과성과 효율성을 수용할 수 있다.

3 인공지능의 사회적 영향

성취기준	성취기준별 성취수준
[12인기03-01] 인공지능의 발전으로 인한 사회 변화를 살펴보고, 인공지능으로 해결할 수 있는 사회적 문제를 분석한다.	A 인공지능의 발전으로 인한 사회 변화를 정확하게 설명하고, 인공지능으로 해결할 수 있는 사회적 문제를 다양한 관점에서 분석할 수 있다.
	B 인공지능의 발전으로 인한 사회 변화를 정확하게 설명하고, 인공지능으로 해결할 수 있는 사회적 문제를 분석할 수 있다.
	C 인공지능의 발전으로 인한 사회 변화를 설명하고, 인공지능으로 해결할 수 있는 사회적 문제를 분석할 수 있다.
	D 인공지능의 발전으로 인한 사회 변화를 설명하고, 인공지능으로 해결할 수 있는 사회적 문제를 부분적으로 분석할 수 있다.
	E 인공지능의 발전으로 인한 사회 변화를 인지하고, 인공지능으로 해결할 수 있는 사회적 문제를 부분적으로 분석할 수 있다.
[12인기03-02] 인공지능에 의해 변화하는 인간의 삶과 직업의 양상에 대해 이해하고 진로를 탐색한다.	A 인공지능과 관련 있는 진로 분야를 자세하게 설명하고, 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 다양한 관점에서 구체적으로 탐색하며, 진로 및 직업에 미치는 인공지능의 중요성을 내면화할 수 있다.
	B 인공지능과 관련 있는 진로 분야를 자세하게 설명하고, 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 다양한 관점에서 구체적으로 탐색하며, 진로 및 직업에 미치는 인공지능의 중요성을 인식할 수 있다.
	C 인공지능과 관련 있는 진로 분야를 설명하고, 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 탐색하며, 진로 및 직업에 미치는 인공지능의 중요성을 인식할 수 있다.
	D 인공지능과 관련 있는 진로 분야를 설명하고, 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 탐색하며, 진로 및 직업에 미치는 인공지능의 중요성을 수용할 수 있다.
	E 인공지능과 관련 있는 진로 분야를 인지하고, 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 부분적으로 탐색하며, 진로 및 직업에 미치는 인공지능의 중요성을 수용할 수 있다.

성취기준	성취기준별 성취수준
[12인기03-03] 인공지능에 대한 비판적 자세를 바탕으로 인공지능과 인간의 공존 방안을 도출한다.	A 인공지능과 인간의 공존 방안을 다양한 관점에서 탐색하고, 인공지능에 대한 비판적 자세를 내면화할 수 있다.
	B 인공지능과 인간의 공존 방안을 다양한 관점에서 탐색하고, 인공지능에 대한 비판적 자세를 인식할 수 있다.
	C 인공지능과 인간의 공존 방안을 탐색하고, 인공지능에 대한 비판적 자세를 인식할 수 있다.
	D 인공지능과 인간의 공존 방안을 탐색하고, 인공지능에 대한 비판적 자세를 수용할 수 있다.
	E 인공지능과 인간의 공존 방안을 부분적으로 탐색하고, 인공지능에 대한 비판적 자세를 수용할 수 있다.
[12인기03-04] 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리 관점에서 분석한다.	A 인공지능 윤리적 딜레마의 개념과 사례를 구체적으로 설명하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 깊이 있게 분석하며, 인공지능에 대한 윤리적 태도를 내면화할 수 있다.
	B 인공지능 윤리적 딜레마의 개념과 사례를 구체적으로 설명하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 분석하며, 인공지능에 대한 윤리적 태도를 인식할 수 있다.
	C 인공지능 윤리적 딜레마의 개념과 사례를 설명하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 분석하며, 인공지능에 대한 윤리적 태도를 인식할 수 있다.
	D 인공지능 윤리적 딜레마의 개념과 사례를 설명하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 부분적으로 분석하며, 인공지능에 대한 윤리적 태도를 수용할 수 있다.
	E 인공지능 윤리적 딜레마의 개념을 인지하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 부분적으로 분석하며, 인공지능에 대한 윤리적 태도를 수용할 수 있다.

4 인공지능 프로젝트

성취기준	성취기준별 성취수준
[12인기04-01] 지속가능발전 목표를 해결하기 위해 인공지능을 적용할 수 있는 방안을 탐색하고, 인공지능 프로젝트를 활동에 적합한 주제를 도출한다.	A 지속가능발전목표의 개념과 인공지능을 적용한 사례를 구체적으로 설명하고, 인공지능 프로젝트 활동에 적합한 다양한 주제를 탐색하며, 인류의 지속가능발전을 위한 인공지능의 중요성을 내면화할 수 있다.
	B 지속가능발전목표의 개념과 인공지능을 적용한 사례를 구체적으로 설명하고, 인공지능 프로젝트 활동에 적합한 주제를 탐색하며, 인류의 지속가능발전을 위한 인공지능의 중요성을 인식할 수 있다.
	C 지속가능발전목표의 개념을 설명하고, 인공지능 프로젝트 활동에 적합한 주제를 탐색하며, 인류의 지속가능발전을 위한 인공지능의 중요성을 인식할 수 있다.
	D 지속가능발전목표의 개념을 설명하고, 인공지능 프로젝트 활동에 부분적으로 적합한 주제를 탐색하며, 인류의 지속가능발전을 위한 인공지능의 중요성을 수용할 수 있다.
	E 지속가능발전목표의 개념을 인지하고, 인공지능 프로젝트 활동에 부분적으로 적합한 주제를 탐색하며, 인류의 지속가능발전을 위한 인공지능의 중요성을 수용할 수 있다.

성취기준	성취기준별 성취수준	
[12인기04-02] 인공지능 문제 해결 과정에 기반하여 프로젝트 수행 계획을 구안한다.	A	인공지능 문제 해결 과정을 올바르게 설명하고, 인공지능 프로젝트 수행 계획을 구체적으로 수립할 수 있다.
	B	인공지능 문제 해결 과정을 올바르게 설명하고, 인공지능 프로젝트 수행 계획을 수립할 수 있다.
	C	인공지능 문제 해결 과정을 설명하고, 인공지능 프로젝트 수행 계획을 수립할 수 있다.
	D	인공지능 문제 해결 과정을 설명하고, 인공지능 프로젝트 수행 계획을 부분적으로 수립할 수 있다.
	E	인공지능 문제 해결 과정을 인지하고, 인공지능 프로젝트 수행 계획을 부분적으로 수립할 수 있다.
[12인기04-03] 인공지능 프로젝트를 수행하는 과정에서 협력적인 문제 해결 자세를 바탕으로 인공지능 소프트웨어를 개발한다.	A	프로젝트 수행 계획에 의거하여 절차적으로 인공지능 소프트웨어를 개발하되, 그 과정에서 협력적으로 문제를 해결하는 자세의 중요성을 내면화할 수 있다.
	B	프로젝트 수행 계획에 의거하여 절차적으로 인공지능 소프트웨어를 개발하되, 그 과정에서 협력적으로 문제를 해결하는 자세의 중요성을 인식할 수 있다.
	C	프로젝트 수행 계획 중 일부에 의거하여 인공지능 소프트웨어를 개발하되, 그 과정에서 협력적으로 문제를 해결하는 자세의 중요성을 인식할 수 있다.
	D	프로젝트 수행 계획 중 일부에 의거하여 인공지능 소프트웨어를 개발하되, 그 과정에서 협력적으로 문제를 해결하는 자세의 중요성을 수용할 수 있다.
	E	인공지능 소프트웨어를 개발하되, 그 과정에서 협력적으로 문제를 해결하는 자세의 중요성을 수용할 수 있다.
[12인기04-04] 인공지능의 사회적 영향을 고려하여 인공지능 소프트웨어를 개발하고, 평가 결과를 반영하여 성능을 개선한다.	A	인공지능 소프트웨어를 개발하고 다양한 방식으로 평가하여 성능을 개선하며, 그 과정에서 인공지능의 사회적 영향을 내면화할 수 있다.
	B	인공지능 소프트웨어를 개발하고 다양한 방식으로 평가하여 성능을 개선하며, 그 과정에서 인공지능의 사회적 영향을 인식할 수 있다.
	C	인공지능 소프트웨어를 개발하고 평가하여 성능을 개선하며, 그 과정에서 인공지능의 사회적 영향을 인식할 수 있다.
	D	인공지능 소프트웨어를 개발하고 평가하여 성능을 개선하며, 그 과정에서 인공지능의 사회적 영향을 수용할 수 있다.
	E	인공지능 소프트웨어를 개발하고 평가하며, 그 과정에서 인공지능의 사회적 영향을 수용할 수 있다.

2 영역별 성취수준

1 인공지능의 이해

영역	영역별 성취수준		
I 인공지능의 이해	A	지식·이해	인공지능의 개념과 특성, 인공지능에서 탐색의 중요성과 다양한 탐색 알고리즘의 차이, 탐색 알고리즘의 적용 방법, 규칙과 사실, 자료와 정보 그리고 지식을 구분하여 정확하게 설명할 수 있다.
		과정·기능	인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 구체적으로 비교·분석하고, 문제 해결을 위한 탐색 과정을 구체적으로 설계할 수 있다. 지능적 탐색의 원리를 다양한 사례를 통해 파악하고, 탐색 알고리즘을 적용하기에 적합한 문제에 적용할 수 있으며, 다양한 방식으로 새로운 지식을 추론하여 생성할 수 있다.
		가치·태도	인공지능의 필요성과 적용 가능성을 내면화하고, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 생활화할 수 있다.
	B	지식·이해	인공지능의 개념과 특성, 인공지능에서 탐색의 중요성과 다양한 탐색 알고리즘의 차이, 탐색 알고리즘의 적용 방법, 규칙과 사실, 자료와 정보 그리고 지식을 구분하여 설명할 수 있다.
		과정·기능	인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 구체적으로 비교·분석하고, 문제 해결을 위한 탐색 과정을 설계할 수 있다. 지능적 탐색의 원리를 파악하고, 탐색 알고리즘을 적용하기에 적합한 문제에 적용할 수 있으며, 다양한 방식으로 새로운 지식을 추론하여 생성할 수 있다.
		가치·태도	인공지능의 필요성과 적용 가능성을 내면화하고, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 생활화할 수 있다.
	C	지식·이해	인공지능의 개념과 특성, 인공지능에서 탐색의 중요성과 다양한 탐색 알고리즘의 차이, 탐색 알고리즘의 원리, 규칙과 사실, 자료와 정보 그리고 지식을 설명할 수 있다.
		과정·기능	인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석하고, 문제 해결을 위한 탐색 과정을 설계할 수 있다. 지능적 탐색의 원리를 파악하고, 탐색 알고리즘을 적용하기에 적합한 문제에 적용할 수 있으며, 새로운 지식을 추론하여 생성할 수 있다.
		가치·태도	인공지능의 필요성과 적용 가능성을 인식하고, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 경험할 수 있다.
	D	지식·이해	인공지능의 개념과 특성, 인공지능에서 탐색의 중요성과 다양한 탐색 알고리즘의 차이, 탐색 알고리즘의 원리, 규칙과 사실, 자료와 정보 그리고 지식을 부분적으로 설명할 수 있다.
		과정·기능	인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석하고, 문제 해결을 위한 탐색 과정의 일부를 설계할 수 있다. 지능적 탐색의 원리를 파악하고, 탐색 알고리즘을 문제에 적용할 수 있으며, 새로운 지식을 추론하여 생성할 수 있다.
		가치·태도	인공지능의 필요성과 적용 가능성을 인식하고, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 경험할 수 있다.
E	지식·이해	인공지능의 개념과 특성, 인공지능에서 탐색의 중요성과 다양한 탐색 알고리즘의 차이, 탐색 알고리즘의 원리, 규칙과 사실, 자료와 정보 그리고 지식을 인지할 수 있다.	
	과정·기능	인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 부분적으로 비교·분석하고, 문제 해결을 위한 탐색 과정의 일부를 설계할 수 있다. 지능적 탐색의 원리를 부분적으로 파악하고, 탐색 알고리즘을 문제에 적용할 수 있으며, 지식을 추론하여 생성할 수 있다.	
	가치·태도	인공지능의 필요성과 적용 가능성, 인공지능을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결하는 자세를 수용할 수 있다.	

2 인공지능과 학습

영역		영역별 성취수준	
II 인공지능과 학습	A	지식·이해	기계학습의 개념과 원리, 결측치, 이상치, 독립변수, 종속변수를 정확하게 설명하고, 기계학습의 유형과 알고리즘, 훈련 데이터, 테스트 데이터의 적용 방법 설명하며, 인공신경망, 딥러닝, 컴퓨터 비전, 음성 인식, 자연어처리의 원리를 올바르게 설명할 수 있다.
		과정·기능	기계학습을 적용할 문제를 정의하고 올바르게 학습을 진행한 후 적절하게 성능을 평가할 수 있다. 딥러닝의 활용 분야를 다양하게 탐색하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 효과적으로 해결한 후 성능을 평가할 수 있다.
		가치·태도	기계학습에 적용하는 데이터의 중요성, 인공지능의 효과성과 효율성을 내면화할 수 있다.
	B	지식·이해	기계학습의 개념과 원리, 결측치, 이상치, 독립변수, 종속변수를 설명하고, 기계학습의 유형과 알고리즘, 훈련 데이터, 테스트 데이터의 적용 방법 설명하며, 인공신경망, 딥러닝, 컴퓨터 비전, 음성 인식, 자연어처리의 원리를 설명할 수 있다.
		과정·기능	기계학습을 적용할 문제를 정의하고 올바르게 학습을 진행한 후 성능을 평가할 수 있다. 딥러닝의 활용 분야를 다양하게 탐색하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결한 후 성능을 평가할 수 있다.
		가치·태도	기계학습에 적용하는 데이터의 중요성, 인공지능의 효과성과 효율성을 내면화할 수 있다.
	C	지식·이해	기계학습의 개념과 원리, 결측치, 이상치, 독립변수, 종속변수를 설명하고, 기계학습의 유형과 알고리즘, 훈련 데이터, 테스트 데이터의 개념을 설명하며, 인공신경망, 딥러닝, 컴퓨터 비전, 음성 인식, 자연어처리의 활용 방법을 설명할 수 있다.
		과정·기능	기계학습을 적용할 문제를 정의하고 학습을 진행한 후 성능을 평가할 수 있다. 딥러닝의 활용 분야를 탐색하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결한 후 성능을 평가할 수 있다.
		가치·태도	기계학습에 적용하는 데이터의 중요성, 인공지능의 효과성과 효율성을 인식할 수 있다.
	D	지식·이해	기계학습의 개념과 원리, 결측치, 이상치, 독립변수, 종속변수를 설명하고, 기계학습의 유형과 알고리즘, 훈련 데이터, 테스트 데이터의 개념을 설명하며, 인공신경망, 딥러닝, 컴퓨터 비전, 음성 인식, 자연어처리의 개념을 인지할 수 있다.
		과정·기능	기계학습을 적용할 문제를 정의하고 학습을 진행할 수 있다. 딥러닝의 활용 분야를 탐색하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 해결할 수 있다.
		가치·태도	기계학습에 적용하는 데이터의 중요성, 인공지능의 효과성과 효율성을 인식할 수 있다.
E	지식·이해	기계학습, 결측치, 이상치, 독립변수, 종속변수, 기계학습의 유형과 알고리즘, 훈련 데이터, 테스트 데이터, 인공신경망, 딥러닝, 컴퓨터 비전, 음성 인식, 자연어처리의 개념을 인지할 수 있다.	
	과정·기능	기계학습을 적용할 문제를 정의하고 부분적으로 학습을 진행할 수 있다. 딥러닝의 활용 분야를 부분적으로 탐색하고, 딥러닝을 활용하여 실생활 및 다양한 학문 분야의 문제를 일부 해결할 수 있다.	
	가치·태도	기계학습에 적용하는 데이터의 중요성, 인공지능의 효과성과 효율성을 수용할 수 있다.	

3 인공지능의 사회적 영향

영역		영역별 성취수준	
III 인공지능의 사회적 영향	A	지식·이해	인공지능의 발전으로 인한 사회 변화를 정확하게 설명하고, 인공지능과 관련 있는 진로 분야를 자세하게 설명하며, 인공지능 윤리적 딜레마의 개념과 사례를 구체적으로 설명할 수 있다.
		과정·기능	인공지능으로 해결할 수 있는 사회적 문제와 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 구체적으로 탐색할 수 있다. 인공지능과 인간의 공존 방안을 다양한 관점에서 탐색하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 깊이 있게 분석할 수 있다.
		가치·태도	진로 및 직업에 미치는 인공지능의 중요성과 인공지능에 대한 비판적 자세와 윤리적 태도를 내면화할 수 있다.
	B	지식·이해	인공지능의 발전으로 인한 사회 변화를 정확하게 설명하고, 인공지능 관련 진로 분야를 설명하며, 인공지능 윤리적 딜레마의 개념과 사례를 설명할 수 있다.
		과정·기능	인공지능으로 해결할 수 있는 사회적 문제와 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 구체적으로 탐색할 수 있다. 인공지능과 인간의 공존 방안을 탐색하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 분석할 수 있다.
		가치·태도	진로 및 직업에 미치는 인공지능의 중요성과 인공지능에 대한 비판적 자세와 윤리적 태도를 내면화할 수 있다.
	C	지식·이해	인공지능의 발전으로 인한 사회 변화, 인공지능 관련 진로 분야, 인공지능 윤리적 딜레마의 개념과 사례를 설명할 수 있다.
		과정·기능	인공지능으로 해결할 수 있는 사회적 문제와 인공지능에 의해 변화하는 인간의 삶과 직업의 양상, 인공지능과 인간의 공존 방안을 탐색하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 분석할 수 있다.
		가치·태도	진로 및 직업에 미치는 인공지능의 중요성과 인공지능에 대한 비판적 자세와 윤리적 태도를 인식할 수 있다.
	D	지식·이해	인공지능의 발전으로 인한 사회 변화, 인공지능 관련 진로 분야를 설명하고, 인공지능 윤리적 딜레마의 개념을 인지할 수 있다.
		과정·기능	인공지능으로 해결할 수 있는 사회적 문제와 인공지능에 의해 변화하는 인간의 삶과 직업의 양상을 탐색할 수 있다. 인공지능과 인간의 공존 방안을 일부 탐색하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 부분적으로 분석할 수 있다.
		가치·태도	진로 및 직업에 미치는 인공지능의 중요성과 인공지능에 대한 비판적 자세와 윤리적 태도를 인식할 수 있다.
E	지식·이해	인공지능의 발전으로 인한 사회 변화, 인공지능 관련 진로 분야, 인공지능 윤리적 딜레마의 개념을 인지할 수 있다.	
	과정·기능	인공지능으로 해결할 수 있는 사회적 문제와 인공지능에 의해 변화하는 인간의 삶과 직업의 양상, 인공지능과 인간의 공존 방안을 부분적으로 탐색하고, 인공지능의 활용사례와 윤리적 딜레마 상황을 인공지능 윤리의 관점에서 일부 분석할 수 있다.	
	가치·태도	진로 및 직업에 미치는 인공지능의 중요성과 인공지능에 대한 비판적 자세와 윤리적 태도를 수용할 수 있다.	



4 인공지능 프로젝트

영역	영역별 성취수준	
A	지식·이해	지속가능발전목표의 개념과 인공지능을 적용한 사례, 인공지능 문제 해결 과정을 올바르게 설명할 수 있다.
	과정·기능	인공지능 프로젝트 활동에 적합한 다양한 주제를 탐색하고, 인공지능 프로젝트 수행 계획을 구체적으로 수립할 수 있다. 프로젝트 수행 계획에 의거하여 절차적으로 인공지능 소프트웨어를 개발하고, 다양한 방식으로 평가하여 성능을 개선할 수 있다.
	가치·태도	인류의 지속가능발전을 위한 인공지능의 중요성, 인공지능 프로젝트를 수행하는 과정에서 협력적으로 문제를 해결하는 자세의 중요성, 인공지능의 사회적 영향을 내면화할 수 있다.
B	지식·이해	지속가능발전목표의 개념과 인공지능을 적용한 사례, 인공지능 문제 해결 과정을 설명할 수 있다.
	과정·기능	인공지능 프로젝트 활동에 적합한 다양한 주제를 탐색하고, 인공지능 프로젝트 수행 계획을 수립할 수 있다. 프로젝트 수행 계획에 의거하여 인공지능 소프트웨어를 개발하고, 평가하여 성능을 개선할 수 있다.
	가치·태도	인류의 지속가능발전을 위한 인공지능의 중요성, 인공지능 프로젝트를 수행하는 과정에서 협력적으로 문제를 해결하는 자세의 중요성, 인공지능의 사회적 영향을 내면화할 수 있다.
C	지식·이해	지속가능발전목표의 개념과 인공지능 문제 해결 과정을 설명할 수 있다.
	과정·기능	인공지능 프로젝트 활동에 적합한 주제를 탐색하고, 인공지능 프로젝트 수행 계획을 수립할 수 있다. 프로젝트 수행 계획 중 일부에 의거하여 인공지능 소프트웨어를 개발하고, 평가하여 성능을 개선할 수 있다.
	가치·태도	인류의 지속가능발전을 위한 인공지능의 중요성, 인공지능 프로젝트를 수행하는 과정에서 협력적으로 문제를 해결하는 자세의 중요성, 인공지능의 사회적 영향을 인식할 수 있다.
D	지식·이해	지속가능발전목표의 개념과 인공지능 문제 해결 과정을 부분적으로 설명할 수 있다.
	과정·기능	인공지능 프로젝트 활동에 부분적으로 적합한 주제를 탐색하고, 인공지능 프로젝트 수행 계획을 수립할 수 있다. 프로젝트 수행 계획 중 일부에 의거하여 인공지능 소프트웨어를 개발하고, 평가할 수 있다.
	가치·태도	인류의 지속가능발전을 위한 인공지능의 중요성, 인공지능 프로젝트를 수행하는 과정에서 협력적으로 문제를 해결하는 자세의 중요성, 인공지능의 사회적 영향을 인식할 수 있다.
E	지식·이해	지속가능발전목표의 개념과 인공지능 문제 해결 과정을 인지할 수 있다.
	과정·기능	인공지능 프로젝트 활동에 부분적으로 적합한 주제를 탐색하고, 인공지능 프로젝트 수행 계획을 일부 수립할 수 있다. 인공지능 소프트웨어를 개발하고, 평가할 수 있다.
	가치·태도	인류의 지속가능발전을 위한 인공지능의 중요성, 인공지능 프로젝트를 수행하는 과정에서 협력적으로 문제를 해결하는 자세의 중요성, 인공지능의 사회적 영향을 수용할 수 있다.

IV
인공지능
프로젝트

03. 연간 지도 계획

본 연간 지도 계획은 64차시(4학점×16주)를 기준으로 작성되었으며, 학교 편성 계획에 따라 증감(±1학점×16주)이 가능합니다.

대단원	소단원	차시		교과서 쪽수
		차시	시간	
I 인공지능의 이해	01 인공지능의 원리	2	14	10~19
	02 인공지능과 탐색	2		20~29
	03 맹목적 탐색과 정보 이용 탐색	3		30~43
	04 프로젝트 실습 도전! 독도로 가는 길 찾기	4		44~53
	05 지식의 표현과 추론	2		54~65
	대단원 정리 및 평가 문제	1		66~67
II 인공지능과 학습	01 기계학습과 데이터	2	22	70~81
	02 데이터 전처리와 핵심 속성 추출	3		82~93
	03 기계학습의 유형과 알고리즘	2		94~105
	04 기계학습 프로젝트 기계학습을 활용한 문제 해결	8		106~125
	05 딥러닝의 이해와 활용	3		126~139
	06 딥러닝 프로젝트 딥러닝으로 손 글씨 분류하기	3		140~153
	대단원 정리 및 평가 문제	1		154~155
III 인공지능의 사회적 영향	01 인공지능과 사회 변화	3	10	158~167
	02 인공지능과 진로	3		168~177
	03 인공지능과 윤리	3		178~191
	대단원 정리 및 평가 문제	1		192~193
IV 인공지능 프로젝트	01 인공지능과 지속 가능 발전 목표	2	18	196~203
	02 인공지능 프로젝트 1 미래의 최저 시급은 얼마일까?	6		204~209
	03 인공지능 프로젝트 2 이 물은 먹을 수 있는 물일까?	6		210~219
	대단원 프로젝트	4		220~221
연간 시수		64		

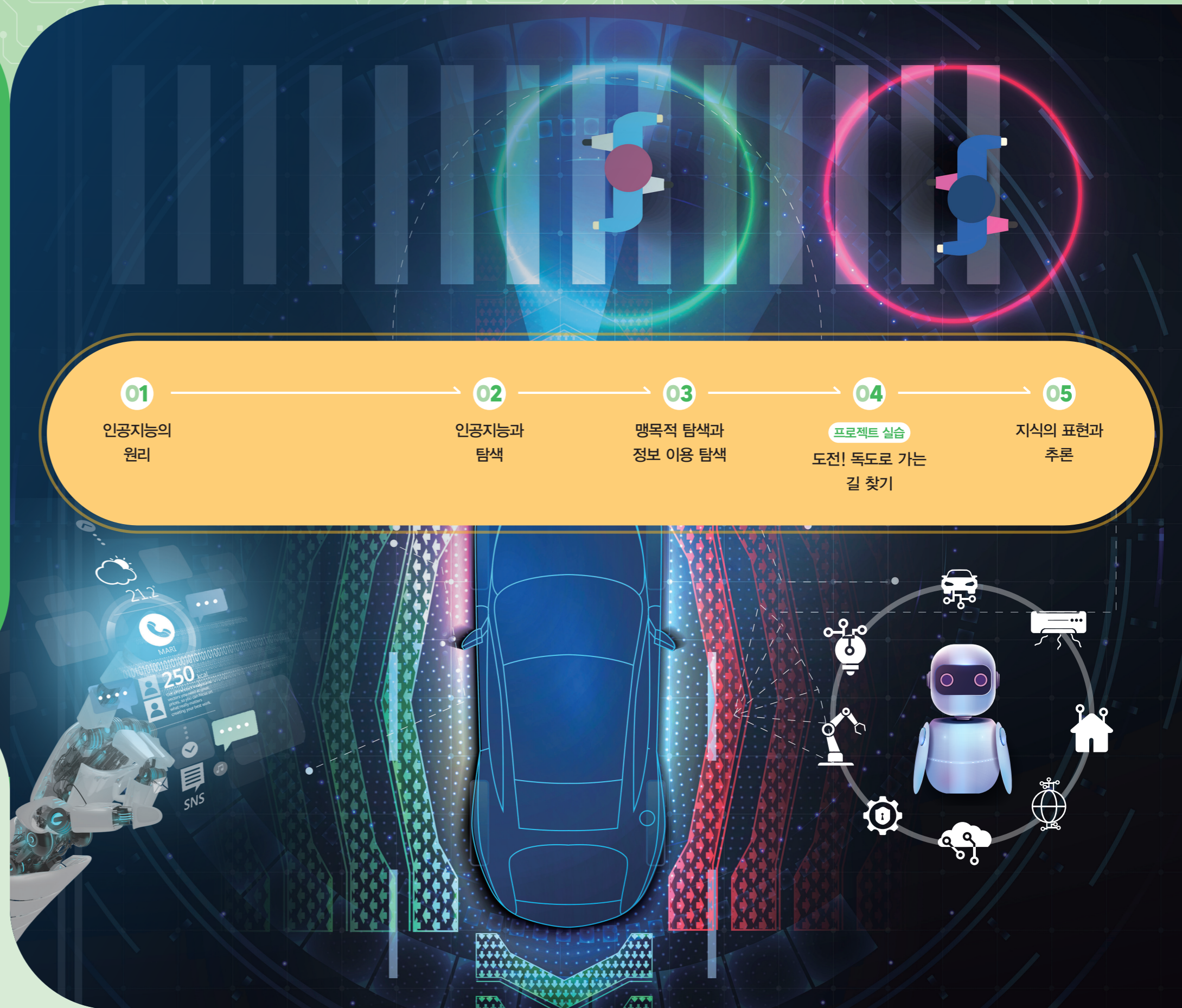
I

인공지능의 이해

단원 개관

인공지능이 발전하여 우리 생활 속에 다양하게 적용되고 있다. 인공지능은 스마트폰 속 비서, 자율주행 자동차, 제조 공정에서 불량품 감지 등 산업 분야까지 다양하게 활용되고 있다.

이 단원에서는 인공지능에서 지능적 탐색의 중요성과 탐색 원리를 이해하여 탐색 프로그램을 구현하고 인공지능이 학습하는 방식인 추론에 대한 이해를 통해 문제를 해결하는 역량을 기른다.



단원 전개 계획

소단원명	차시	학습 주제	지도 방법
01 인공지능의 원리 교과서 쪽수 10~19쪽	2	1. 인공지능의 개념과 특성 •해 보기① 인공지능의 역할 •해 보기② 자율주행 자동차에 적용할 수 있는 인공지능 •해 보기③ 인공지능과 대화해 보기 2. 인공지능을 활용한 문제 해결 •해 보기④ 인공지능을 활용한 문제 해결 사례 찾아보기	• 인공지능의 개념과 특성을 이해하고, 튜링 테스트를 통해 지능적 판단에 대해 생각해 볼 수 있도록 지도한다. • 인공지능이 활용된 사례를 찾아보고 이를 분석하여 인공지능의 활용 범위와 중요성을 이해할 수 있도록 지도한다.
02 인공지능과 탐색 교과서 쪽수 20~29쪽	2	1. 탐색의 이해 2. 문제 해결을 위한 탐색 과정 설계	• 미로 찾기 예시를 통해 탐색의 개념을 이해하고, 다양한 현실적인 사례를 통해 인공지능에서 탐색이 왜 중요한지 이해하도록 지도한다. • 전구 켜기 예제를 통해 문제를 구조화하여 탐색으로 문제를 해결하는 과정을 경험하고, 문제 해결을 위한 탐색 과정을 설계할 수 있도록 지도한다.
03 맹목적 탐색과 정보 이용 탐색 교과서 쪽수 30~43쪽	3	1. 맹목적 탐색 •해 보기① 트리구조 파악하기 2. 정보 이용 탐색	• 8 퍼즐과 길 찾기 예제를 통해 맹목적 탐색과 정보 이용 탐색의 방법을 익히고, 두 방법의 차이점을 통해 지능적 탐색이 무엇인지 이해하여 문제 상황에 적용할 수 있도록 지도한다.
04 [프로젝트 실습] 도전! 독도로 가는 길 찾기 교과서 쪽수 44~53쪽	4	도전! 독도로 가는 길 찾기	• 인공지능 프로그래밍의 프로젝트 절차를 이해하고, 독도로 가는 길 찾기 프로그램을 수행해 봄으로써, 탐색 알고리즘을 적용한 인공지능 프로그램을 개발할 수 있도록 지도한다.
05 지식의 표현과 추론 교과서 쪽수 54~65쪽	2	1. 지식의 표현 •해 보기① 데이터, 정보, 지식 생성하기 •해 보기② 지식을 IF~THEN으로 표현해 보기 •해 보기③ 의미망을 이용하여 지식 표현하기 2. 추론	• 규칙과 사실을 활용하여 지식을 표현하는 다양한 방법을 이해하고, 지식을 다양한 형태로 표현할 수 있도록 지도한다. • 알고 있는 사실을 활용하여 추론을 통해 새로운 지식을 생성할 수 있도록 지도한다.

01

인공지능의 원리

수업 시간: 2시간 교과서 쪽수: 10~19쪽

학습 목표	• 인공지능의 지능적 판단 과정을 설명할 수 있다. • 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석할 수 있다.
수업 방법	강의, 토론, 실습, 발표
준비물	교사 교과서, 단말기, 관련 교수 학습 자료 학생 필기도구, 단말기

단계	교수·학습 방법	지도상의 유의점
도입	생각 열기 〈생각 열기〉와 관련된 동영상을 보고 우리가 사용하고 있는 인공지능이 무엇이 있는지 친구들과 토의한다.	• 인공지능과 인공지능이 적용되지 않은 소프트웨어를 구분하여 이야기하는지 확인한다.
전개	① 인공지능의 개념과 특성 • 인공지능의 원리를 이해하고 인공지능이 활용되는 사례를 통해 우리의 생활이 어떻게 변화될지 이해할 수 있도록 한다. • 인공지능이 지니는 특성을 이해하고 인간과 인공지능, 인공지능이 적용된 소프트웨어와 적용되지 않은 소프트웨어의 특성을 토의한다. • 튜링 테스트의 개념을 이해하고, 이를 바탕으로 인공지능에서 지능적 판단의 의미를 이해할 수 있도록 한다. ② 인공지능을 활용한 문제 해결 • 인공지능을 활용한 문제 해결 사례를 탐색해 보고, 이를 바탕으로 인공지능으로 변화될 사회 모습을 토의한다.	• 인공지능의 특성을 인간의 지능, 인공지능이 적용되지 않은 소프트웨어와의 차이점을 통해 이해할 수 있게 한다.
정리	탐구 활동 • 교실(학교)에서 발생할 수 있는 다양한 문제 중에서 인공지능으로 해결할 수 있는 문제와 그렇지 않은 문제를 구분하고, 해당 문제를 해결하기 위한 인공지능의 역할을 설정할 수 있도록 한다.	• 교실(학교)에서 발생하는 문제 중 인공지능으로 해결할 수 있는 문제를 찾아 해결 방법을 논의할 수 있도록 지도한다. • 본 차시를 정리하고, 다음 차시를 예고한다.
평가	• 인공지능의 개념과 특성을 설명할 수 있는가? • 지능적 판단에 대해 설명할 수 있는가? • 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석할 수 있는가?	• 인공지능의 개념과 특성, 활용 사례에 대한 이해를 바탕으로 인공지능으로 인한 사회 변화를 이해할 수 있도록 지도한다.

성취기준

- 12인기01-01** 인공지능의 지능적 판단에 대한 이해를 바탕으로 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석한다.
- 12인기01-02** 인공지능에서 탐색의 중요성을 이해하고 문제 해결을 위한 탐색 과정을 설계한다.
- 12인기01-03** 맹목적 탐색과 정보 이용 탐색의 차이를 중심으로 지능적 탐색의 원리를 파악한다.
- 12인기01-04** 지능적 탐색이 필요한 문제를 찾아보고 문제 해결을 위해 정보 이용 탐색 알고리즘을 적용한다.
- 12인기01-05** 규칙과 사실을 활용하여 지식을 표현하고, 새로운 지식을 추론하여 생성한다.

01 인공지능의 원리

01

인공지능의 원리

- 학습 목표**
- 인공지능의 지능적 판단 과정을 설명할 수 있다.
 - 인공지능을 활용한 실생활 및 다양한 학문 분야의 문제 해결 사례를 비교·분석할 수 있다.

학습 요소 인공지능의 지능적 판단, 인공지능 문제 해결 사례

📌 생각 열기 생활 속 인공지능

인공지능은 다양한 분야에서 활용되고 있으며, 그 활용 범위는 계속 넓어지고 있다. 매일 사용하는 스마트폰에 내장된 음성 비서도 인공지능의 한 사례다.



❓ 우리의 생활 속에서 찾을 수 있는 인공지능에는 어떤 것이 있을까?

• 제시 의도 •

다양한 분야에 인공지능이 활용되고 있고, 특히 매일 사용하는 스마트폰에 들어 있는 인공지능 비서를 통해 인공지능에 대해 생각해 볼 수 있다.

지도 방법

- 일상생활 속에서 쉽게 접하는 인공지능 활용 사례를 통해 인공지능이 이미 우리 생활에 다양하게 활용되고 있음을 이해할 수 있도록 안내한다.
- 가정이나 회사 등에서 인공지능을 이용한 사례를 직접 탐색해 보고, 인공지능이 무엇인지 생각해 볼 수 있도록 지도한다.
- 인공지능의 개념과 사례를 간단히 소개하는 영상을 동기 유발에 활용한다.

예시 답안

1. 음성 비서(Voice Assistants): 애플의 Siri, 아마존의 Alexa, 구글 어시스턴트 등 음성 인식을 활용한 인공지능 비서는 우리의 음성을 인식하고 응답하여 다양한 일상적인 업무를 도와준다.
2. 이미지 인식: 얼굴 인식을 통한 출입 통제, 자율주행에서의 사물, 신호, 다른 차량, 사람 인식, 의료 분야에서의 질병 진단 등에 활용된다.

10

1 인공지능의 등장

인공지능(AI: Artificial Intelligence)이라는 용어는 1956년 다트머스 회의에서 존 매카시(John McCarthy)가 처음 언급하였고, ‘지성이 있는 기계를 만드는 과학과 공학(the science and engineering of making intelligent machines)’으로 정의하였다. 이후 프랭크 로젠블랫(Frank Rosenblatt)이 퍼셉트론을 제안하였다. 퍼셉트론(perceptron)은 워런 매컬러(Warren McCulloch)와 월트 피츠(Walt Pitts)가 제안한 모델과 ‘헵의 규칙(Hebbian rule)’을 바탕으로 만든 신경망이다. 퍼셉트론은 뇌 구조를 모방한 신경망으로 딥러닝의 원조이다.

2 인공지능의 현재와 두 가지 관점

- 인공지능은 부흥기와 침체기를 거치다가 딥러닝, GPU의

퍼셉트론의 한계가 지적된 이후 허버트 사이먼(Herbert Simon), 파이겐바움(Feigenbaum), 레더버그(Lederberg)는 전문가 시스템을 개발하였다. 전문가 시스템은 논리적 규칙을 바탕으로 특정 분야에 대한 질문에 답하거나 문제를 해결하는 프로그램이다.

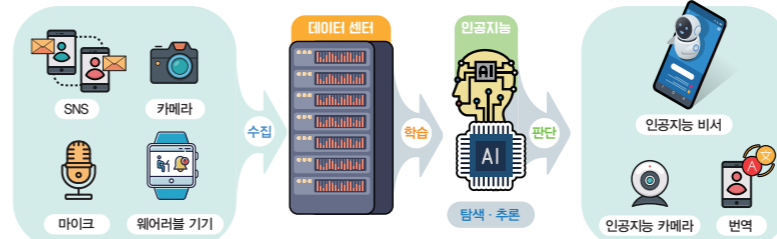
신경망 연구를 주축하게 만들었던 퍼셉트론의 문제점은 인공신경망의 은닉층을 늘린 다층 퍼셉트론으로 해결되었으며, 다층 퍼셉트론의 오류를 줄이기 위한 오류 역전파 방법이 등장하였다.

1 인공지능의 개념과 특성

01 인공지능의 개념

인공지능(AI: Artificial Intelligence)이란 인간의 지능이 갖는 학습, 탐색, 추론 등의 능력을 모방하여 컴퓨터로 구현한 시스템이나 구현하려는 컴퓨터 과학 기술을 말한다. 인공지능은 1950년경 앨런 튜링(Alan Turing)*에 의해 처음 제시되었으며, 현재는 새로운 부흥기를 맞이하여 빠르게 발전하고 있다.

현재의 인공지능은 목적에 맞는 대량의 데이터를 학습하여 주어진 문제에 적절한 판단을 내리는 시스템으로 볼 수 있으며, 이러한 동작을 통해 인간의 판단에 도움을 주고, 단순 작업을 대신하는 등의 편리함을 제공한다. 카메라, 마이크, 각종 감지 센서 등을 통해 입력받은 데이터는 빅데이터로 쌓이게 되고, 이러한 빅데이터를 학습한 인공지능은 탐색과 추론을 통해 판단을 한다.



❶ [그림 1-1] 인공지능의 이해: 인공지능은 SNS, 카메라, 마이크, 웨어러블 기기 등에 의해 생성된 빅데이터로 학습을 하고, 이를 바탕으로 목적에 맞는 판단을 한다.

해 보기 1 인공지능의 역할

- 인공지능이 적용된 제품이나 서비스를 생각해 보고, 인공지능이 어떤 것을 학습했는지 생각해 보자.

인공지능이 적용된 제품이나 서비스	인공지능이 학습한 것
로봇 청소기	실내 공간과 장애물
자율주행 자동차	신호등, 차선, 자동차, 표지판 등
세탁기	세탁물 무게, 습도, 재질, 오염도 등

인공지능의 다양한 정의

- 지성이 있는 기계를 만드는 과학과 공학이다. (존 매카시(John McCarthy))
- 인간의 지적 능력을 기계로 구현하는 과학기술이다. (우리나라 인공지능 국가전략)

*앨런 튜링

영국의 수학자, 암호학자, 논리학자, 컴퓨터 과학자다. 기계가 인간과 같은 지능이 있는지 판단하는 튜링 테스트를 제안하였다.

지도 방법

인공지능은 학자나 단체에 따라 다양하게 정의되고 있음을 안내한다.

* 용어 해설

- 웨어러블 기기: 시계, 옷, 안경 등의 모양으로 몸에 착용할 수 있는 형태의 기기

해 보기 1 지도 방법

- 학교나 집 등 생활하는 공간에서 사용되는 다양한 인공지능 제품을 생각해 볼 수 있도록 지도한다. 또한 해당 제품에서 인공지능이 어떤 역할을 하는지 이야기해 볼 수 있도록 한다.
- 단순한 소프트웨어의 동작을 인공지능으로 착각하지 않도록 주의하여 지도한다.

11

성능 향상, 빅데이터의 증가 등의 영향으로 다시 부흥기를 맞고 있다.

- 기호주의: 인간의 지식을 기호화하고, 그 기호 사이의 관계를 컴퓨터에 입력하여 학습시키면, 컴퓨터도 인간과 비슷한 입력을 받았을 때, 출력 또한 비슷할 것이라는 이론이다.
- 연결주의: 인간의 뇌 구조를 본떠서 인공지능을 구현하려 한 것뿐만 아니라, 인간의 두뇌에서 어떻게 생각하는지도 모방하여 인공지능을 구현하려 하였다.

3 생성형 인공지능(Generative AI)

- 기존 데이터를 소비하고 이를 통해 학습한 다음, 유사한 속성의 다른 데이터를 생성하는 인공지능

• 프롬프트를 작성하면 이에 해당하는 글자, 이미지, 영상 등을 생성하는 딥러닝의 한 형태

- 2014년 이안 굿펠로우(Ian Goodfellow)와 동료들이 생성형 적대적 신경망(GAN: Generative Adversarial Network)을 개발했다. GAN은 두 신경망이 서로 경쟁하는 방식으로 학습을 진행한다. 하나의 신경망은 새로운 데이터를 생성하고, 다른 신경망은 생성된 데이터가 진짜인지, 가짜인지 확인한다. 새롭게 생성된 데이터가 진짜인지, 가짜인지 구분할 수 없을 때까지 학습을 진행한다.

- 활용 사례: 이미지 생성, 이미지 복원, 신약 개발, 동작 흉내, 음성 복원, 애니메이션 캐릭터 생성 등

※ 용어 해설

- **에이전트**: 컴퓨터 과학 분야에서 인간 대신 반복적이거나 특정한 역할을 자동으로 처리해 주는 자동화 시스템
- **지능 에이전트**: 인공지능의 지능적 사고 능력을 결합한 에이전트
- **액추에이터(actuator)**: 입력 신호나 제어 시스템의 출력 신호를 받아, 움직임이나 작동을 일으키는 기기를 의미한다. 액추에이터는 자동화 시스템이나 로봇, 제어 시스템 등 다양한 응용 분야에서 사용된다.

지도 방법

- 로봇 청소기의 동작에서 인공지능이 하는 역할을 이해할 수 있도록 지도한다.
- 로봇 청소기가 모두 같은 동작을 하는 것이 아니라, 적용된 알고리즘에 따라 동작 방식이 서로 다를 수 있음을 이해하도록 지도한다.

이렇게 동작하는 인공지능을 일상에서 이미 지능 에이전트의 형태로 다양하게 경험하고 있다. 스마트폰에 인공지능 대화형 비서가 기본적으로 탑재된 경우가 많으며, 필요에 따라 설치한 다양한 앱 중에 인공지능이 적용된 앱이 많아지고 있다. 번역 플랫폼에도 인공지능이 적용되어 번역의 정확도를 높여 주고, 입력에 오차가 있는 경우에도 정확하게 번역해 준다.



* 액추에이터

지능 에이전트가 판단한 결과를 현실 세계에서 빛, 소리, 움직임 등의 물리적 동작이나 반응으로 만들어 내는 역할을 하며, 구동기라고도 한다.
 ● LED, 버저, 전기 모터, 스피커 등

인공지능이 탑재된 로봇 청소기는 상단에 장착된 카메라를 이용하여 벽면과 천장의 특징을 구별하여 지도를 생성하고, 장애물을 탐지하여 이를 피하며 청소한다.



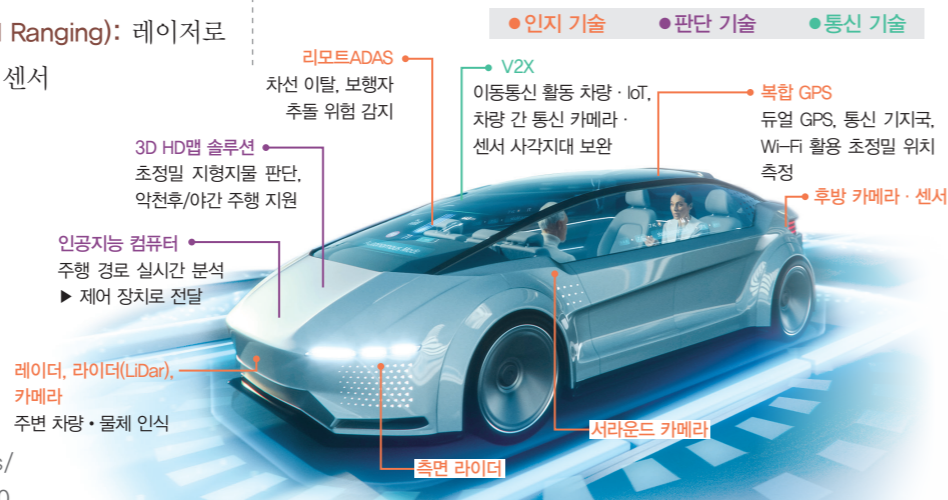
▲ [그림 1-2] 지능 에이전트인 로봇 청소기의 동작 방법 예: 인공지능이 적용된 로봇 청소기는 센서와 카메라를 이용하여 주변 환경을 스캔하고 지도를 생성한 뒤, 지도 정보와 센서의 데이터를 바탕으로 청소를 한다.

4 로봇 청소기에 적용된 센서

- **라이더(LiDAR: Light Detection and Ranging)**: 레이저로 거리를 측정해 지도를 그릴 수 있는 센서
- **3D 센서**: 장애물의 거리와 형상을 파악하는 센서
- **추락 센서**: 바닥과 로봇 청소기의 거리를 측정하여 일정 수준 이상이 되면, 이동을 멈추고 방향을 전환하여 로봇 청소기의 추락을 방지하는 센서

출처 <http://www.bikorea.net/news/quickViewArticleView.html?idxno=17640>

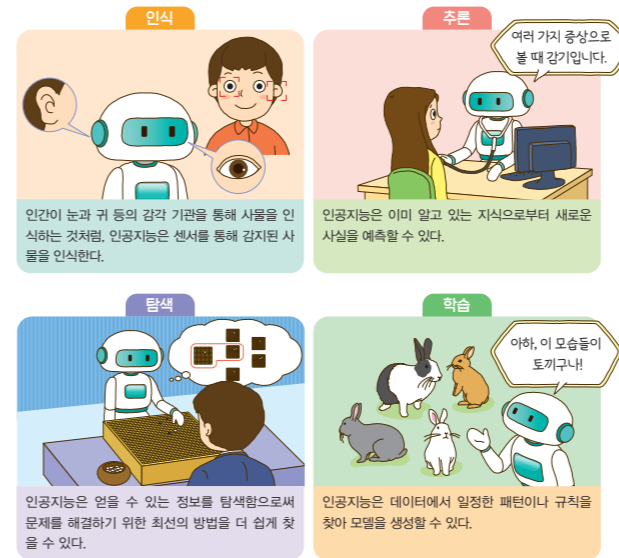
5 자율주행 자동차와 인공지능



▲ SKT 자율주행 자동차의 운영 원리

02 인공지능의 특성

인간의 지능을 모방한 인공지능은 인식, 추론, 탐색, 학습의 특성을 지니며, 데이터를 학습하고, 학습한 데이터를 근거로 추론하고 탐색한다.



▲ 인공지능의 특성

인공지능은 인간의 지능을 모방하려 하지만 인간의 지능과는 다른 특징을 보인다. 인간은 다양하고 복잡한 상황에서도 동시에 여러 가지 일을 처리할 수 있다. 하지만 지금까지의 인공지능은 다양하고 복잡한 상황에서는 한계를 보인다. 또한 특정한 한 가지 일에 특화된 경우가 대부분이다.



6 인간과 인공지능

인공지능은 인간의 지능과 달리 직관을 가지고 판단할 수 없다. 또한 상상하거나 영감을 떠올릴 수 없다. 영상을 만드는 인공지능, 소설을 쓰는 인공지능, 그림을 그리는 인공지능이 등장하고 있지만, 이러한 인공지능은 주어진 데이터와 규칙으로 동작한다. 빅데이터가 없으면 인공지능은 제

지도 방법

인간과 인공지능의 차이점을 지도하는 과정에서, 인공지능에 대해 부정적인 인식이 생기지 않도록 유의한다.

인공지능의 특성별 사례

- 1 **인식**: 얼굴 인식으로 스마트폰의 잠금 화면을 열 수 있다.
- 2 **추론**: 제퍼디 퀴즈쇼에서 우승한 IBM 왓슨은 이미 알고 있는 지식으로부터 문제의 답을 유추한 것이다.
- 3 **탐색**: 바둑을 두는 인공지능 알파고는 탐색을 이용해 최적의 행동을 결정한다.
- 4 **학습**: 쿼드로우즈는 사용자가 그렸던 그림으로 학습해 새로운 그림이 무엇인지 맞춘다.

7 약인공지능 vs 강인공지능 vs 초인공지능

- **약인공지능(Weak AI)**: 특정한 동작이나 문제를 해결하기 위한 인공지능으로 한 가지 또는 특정한 작업에 특화되어 있다.
- **강인공지능(Strong AI)**: 인간 수준의 지능을 가진 인공지능으로 다양한 영역에서 다양한 작업을 수행할 수 있다.
- **초인공지능(Superintelligent AI)**: 인간의 지능을 뛰어넘는 인공지능으로 인간의 모든 지적 능력을 초월한다.

지도 방법

로봇과 인공지능을 혼용하지 않도록 지도한다. 로봇에 인공지능이 탑재되어 동작하는 것을 로봇과 인공지능을 동일시 하여 생각하지 않도록 안내한다.

인공지능이 적용된 소프트웨어

- 인공지능이 적용된 소프트웨어가 무조건 더 좋다고 생각하지 않도록 지도한다. 인공지능의 특성을 이해하고, 이를 바탕으로 인공지능이 적용된 소프트웨어와 적용되지 않은 소프트웨어를 구분할 수 있도록 지도한다.
- 인공지능이 적용되지 않은 소프트웨어는 알고리즘에 따라 항상 같은 동작을 수행한다.
- 인공지능이 적용된 소프트웨어는 데이터 학습을 기반으로 새로운 데이터에 대한 결론을 도출한다. 인공지능의 판단에 따라 동작이 달라질 수 있다.

해 보기 2 지도 방법

자율주행 자동차에 인공지능을 적용할 수 있는 영역을 생각해 보고, 인공지능이 수행할 수 있는 역할을 작성할 수 있도록 지도한다.

로봇 vs 인공지능

로봇은 기계적 장치인 하드웨어로 사전에 정해진 규칙에 따라 스스로 판단하고 행동하는 것을 말하며 기계의 하위 범주에 속한다. 반면, 인공지능은 기계 장치를 동작시키는 소프트웨어로 볼 수 있다.

인공지능이 인간보다 강점을 보이는 부분도 있고, 인간이 인공지능보다 강점을 보이는 부분도 있다. 개와 고양이를 구분하거나 앉았다 일어나는 행동과 같이 인간이 간단히 할 수 있는 일이 인공지능에게는 어려울 수 있다. 하지만 기계 장치에서 동작하는 인공지능은 전기가 공급되는 동안 인간처럼 지치지 않고, 잠도 자지 않으며, 반복적인 작업을 오류 없이 수행할 수 있다. 또한 받아들이는 정보를 빠르게 해석하고 분석할 수 있으며, 특정 영역에 대한 문제를 해결하는 방법이나 해답을 인간보다 빠르게 찾아낼 수도 있다.

인공지능은 일반적인 소프트웨어와도 다른 특징을 지닌다. 일반적인 소프트웨어는 프로그래머가 작성한 알고리즘대로 동작하고 결과를 출력하지만, 인공지능은 데이터를 기반으로 학습한 뒤, 새로운 입력이 있으면 기존 학습을 바탕으로 탐색과 추론을 할 수 있다. 인공지능이 적용되지 않은 소프트웨어와 인공지능이 적용된 소프트웨어의 특징을 비교하면 아래 표와 같다.

인공지능이 적용되지 않은 소프트웨어

- 고정된 단순 작업이나 반복되는 작업을 처리한다.
- 입력 데이터가 같으면 이를 처리하는 시간과 결과가 같다.

인공지능이 적용된 소프트웨어

- 추론이나 예측하는 작업이 가능하다.
- 데이터 기반의 기계학습을 구현한 인공지능의 경우, 학습을 통해 성능이 개선되기도 한다.



알고 기기 악인공지능과 강인공지능

악인공지능은 주어진 역할만 정해진 대로 수행하는 인공지능을 말하며, 악인공지능이 적용된 사례로는 바둑을 두는 알파고, 퀴즈를 맞는 왓슨 등이 있다. 현재까지 우리가 접한 모든 인공지능은 악인공지능이다. 강인공지능은 인간처럼 다양한 일을 할 수 있는 지능을 말하는 것으로, SF 영화나 소설에 나오는 인간과 비슷한 수준의 지능을 가진 인공지능이다.



해 보기 2 자율주행 자동차에 적용할 수 있는 인공지능

자율주행 자동차에 적용될 수 있는 인공지능을 생각해 보고, 인공지능의 역할을 설정해 보자.

인공지능이 적용될 수 있는 영역	인공지능의 역할
전방 사물 인식	전방에 있는 장애물을 인식하여 사물의 종류를 분류한다.
내비게이션	수집된 데이터를 기반으로 안전하고 빠른 길로 주행한다.
교통 신호 인식	신호등을 인식하여 신호에 맞게 주행한다.



해 보기 3 해설 및 예시 답안

교과서 15쪽

1 대화형 인공지능 챗봇(코파일럿(Copilot), 챗GPT(ChatGPT), 뤼튼(wrtn) 등)과 대화해 보자.

8 초인공지능

초인공지능은 모든 부분에서 인간의 능력을 초월하는 인공지능이다. 스스로 인지하고 사고력을 개발하고, 인간의 지능을 능가할 수 있는 인공지능의 한 형태이다. 현재로서 초인공지능은 이론적 가능성에 불과하며, 현실적으로 실현하기 어려울 것이라는 의견이 다수 존재한다.

9 튜링 테스트

• 모방 게임: 남자와 여자, 그리고 질문자가 벌이는 게임이다. 남자와 여자, 질문자는 각각 다른 방에 들어간다. 질문자는 두 명 중 한 명만 남자인 것을 안다. 질문자는 질문을 통해 어느 방에 남자가 있고 여자가 있는지 판단해

야 한다. 질문자가 어떤 질문이라도 할 수 있지만, 질문과 답변은 프린터를 이용한다.

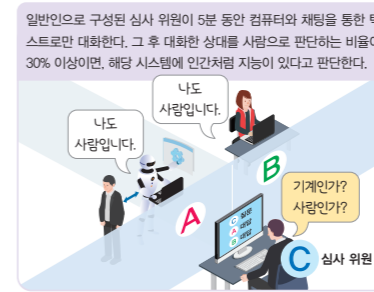
- 튜링 테스트: 이런 게임 진행 중 남자나 여자 중 한 명을 컴퓨터로 바꾼다. 그랬을 때, 컴퓨터로 바뀐 것을 질문자가 알아차릴 수 있는지 검증하는 것이 튜링 테스트이다. 만약, 알아차리지 못한다면 컴퓨터는 인간과 비슷한 수준의 지능이 있다는 것이 앨런 튜링의 주장이다.

03 튜링 테스트

1950년 앨런 튜링이 제안한 튜링 테스트(Turing test)는 기계가 인간과 비슷하게 대화할 수 있는지 판단하는 테스트로, 튜링 테스트를 통과한다면 기계에도 지능이 있다고 봐야 한다는 주장이다. 하지만, 튜링 테스트를 통과했다고 해서 기계에 지능이 있다고 판단하는 것은 한계가 있다는 지적도 있다. 인간이 다양한 감각을 통해 정보를 받아들이는 것처럼 인공지능은 다양한 센서로 주변 상황을 인식하는데, 튜링 테스트는 이런 것들이 빠진 채 문자(text) 대화에만 머물러 있었기 때문이다.

앨런 튜링은 튜링 테스트의 논리적인 방식만 제시했고, 이후 구체적이고 다양한 실험 방법이 추가로 제시되었다. 그러나 2014년 유진 구스트만(Eugene Goostman)*이 테스트를 통과했다고 주장하기 전까지 어떤 인공지능도 튜링 테스트를 통과하지 못했다. 게다가 유진 구스트만도 앞뒤가 맞지 않는 여러 문답을 했으나 13세의 어린 소년이라는 설정으로 기계라는 의심을 비켜 갔다고 볼 수 있다. 이런 텍스트 형태의 대화는 튜링 테스트 통과를 목적으로 일부러 인간처럼 맞춤법을 틀리게 대답하도록 설계된 사례도 있었다.

이처럼 튜링 테스트에 여러 비판 의견이 있지만, 튜링 테스트는 여전히 인공지능의 발전을 측정할 수 있는 하나의 척도이고 인공지능이 인간과 비슷해 보이는지 확인하는 데 충분한 도움을 줄 수 있다.



▲ [그림 1-3] 튜링 테스트

* 유진 구스트만

2014년 영국 레딩 대학(University of Reading)이 개발한 인공지능으로, 13세의 우크라이나 소년으로 설정된 채팅 프로그램이다. 당시 튜링 테스트 통과 기준은 30% 이상이었는데, 심사위원의 33%(1/3)가 유진을 인간으로 착각하여 튜링 테스트를 통과한 최초의 인공지능이라고 주장했다.



해 보기 3 인공지능과 대화해 보기

분석하기

1 대화형 인공지능 챗봇(코파일럿(Copilot), 챗GPT(ChatGPT), 뤼튼(wrtn) 등)과 대화해 보자.

코파일럿 <https://www.microsoft.com/ko-kr/microsoft-copilot> 챗GPT <https://chat.openai.com/auth/login> 뤼튼 <https://wrtn.ai>

질문	답변
프로그래밍과 인공지능의 관계는?	프로그래밍과 인공지능은 깊게 연관된 분야입니다. 인공지능은 프로그래밍과 데이터 분석에 기반을 둔 기술이며, 적절한 프로그래밍이 없는 인공지능 시스템을 구현할 수 없습니다.

2 인공지능의 답변이 지능적인 답변인지 아닌지 생각해 보고, 그렇게 생각한 이유를 적어 보자.

- 지능적인 답변인가? 지능적인 답변이다.
- 그렇게 생각한 이유는? 질문에 적합한 대답을 제시한다. 이것은 질문을 이해했다고 볼 수 있고, 적절한 답을 제시한 것이기 때문에 지능이 있다고 볼 수 있다.

예시 답안

- 질문: 특이점은 언제 올까?
- 답변: 특이점(Singularity)은 인공지능(AI)의 발전이 인간 지능을 능가하고 자기 스스로를 계속 개선해 나가는 기점을 의미합니다. 특이점이 언제 올지에 대한 정확한 예측은 어렵지만, 여러 전문가와 연구자들이 다양한 견해를 제시하고 있습니다(Chat GPT).

예시 답안

- 질문: 특이점은 언제 올까?
- 답변: “특이점(Singularity)”이란 인공지능의 지능이 인간의 지능을 뛰어넘어 급속한 발전이 이루어지는 시점을 의미합니다. 이 시점에서 기존의 예측과 경험에 기반한 사고방식이 더 이상 효과적이지 않게 되며, 인간의 역사와 문명에 큰 변화가 예상됩니다. 특이점의 도래 시기에 대한

지도 방법

인간과 비슷한 대화만으로 지능이 있다고 판단할 수 있는지 생각해 볼 있도록 지도한다. 학자에 따라 지능을 다양하게 정의하고 있음을 안내하고, 인공지능의 어떠한 대답이 지능적 판단으로 볼 수 있는지 생각해 볼 수 있도록 지도한다.



해 보기 3 지도 방법

질문 시 학생의 신분에 맞는 질문이 이루어질 수 있도록 지도한다. 세 명이 한 팀으로, 같은 질문을 각자 다른 인공지능에게 질문하고 대답을 비교할 수도 있다.

15

예측은 다양합니다. 일부 전문가들은 2045년에 특이점이 도래할 것으로 예측하였고, 이로 인해 인간은 영생을 누리게 될 것으로 예측하였습니다. 하지만 이러한 예측은 아직까지 논란의 여지가 있으며, 특이점이 정확히 언제 올지는 아직 불확실합니다(Bing).

2 인공지능의 답변이 지능적인지 생각해 보고, 그렇게 생각한 이유를 적어 보자.

- 지능적인 답변인가? **예시 답안** 지능적인 답변이다.

• 그렇게 생각한 이유는?

예시 답안 온라인에 있는 데이터로 학습한 결과를 잘 조합해서 질문에 적절한 답을 한 것은 지능이 있기 때문에 가능한 것이라 생각한다.

지도 방법

인공지능을 활용한 문제 해결 사례를 다양하게 제시해 준다. 단, 인공지능으로 모든 문제를 해결할 수 있다는 그릇된 인식이 생성되지 않도록 주의한다.

참고

동영상 ▶ 인공지능을 활용한 문제 해결 사례

- 제목: 인공지능 활용
- 영상 내용: 생활 속에서 인공지능 기술을 활용한 사례에 대해 알아본다.
- 주소: <https://youtu.be/4t2QLQ-7-9Q?si=H3mxvWSHbIzEVKq>

* 용어 해설

- **합성곱 신경망(Convolution Neural Network):** 사람의 시각 처리 방식을 모방한 딥러닝 학습 모델로서, 이미지 분류에 적합한 딥러닝 모델이다.

16

2 인공지능을 활용한 문제 해결

01 인공지능을 활용한 문제 해결 사례

*스미싱 문자 메시지(SMS)를 통해 소액 결제를 유도하거나 개인 정보를 빼내 가는 피싱 사기 수법 중 하나다.

인공지능은 생활 속에서 발생하는 다양한 문제를 해결하는 데 활용되고 있다. 은행에서는 인공지능을 이용하여 스미싱(smishing)* 여부를 판단하고, 심리 상담 센터에서는 인공지능이 적용된 앱을 이용하여 어린이의 그림을 분석하고 심리 상태를 파악한다. 또한, 인공지능 돌봄 로봇은 혼자 사는 노인의 대화 상대가 되어 주고, 건강 상태를 파악할 수 있다.

특히 빠른 속도로 고령화에 접어들고 있는 우리나라는 혼자 사는 노인 문제가 사회 문제로 주목받고 있다. 독거노인을 위한 인공지능 스피커는 노인들의 정신 건강 뿐 아니라 신체 건강에도 도움을 준다. 웨어러블 기기와 사물에 부착된 센서가 행동을 감지하여 수집한 데이터를 분석한 뒤, 규칙적인 생활을 할 수 있도록 음성으로 안내하는 인공지능을 활용한 결과, 노인의 평균 걸음 수가 늘어났고 외출 시간도 증가하였다. 또한 우울증 평가 점수도 감소하고 삶의 질에 대한 만족도가 증가하는 등 정신 건강도 향상되는 결과가 나타났다.

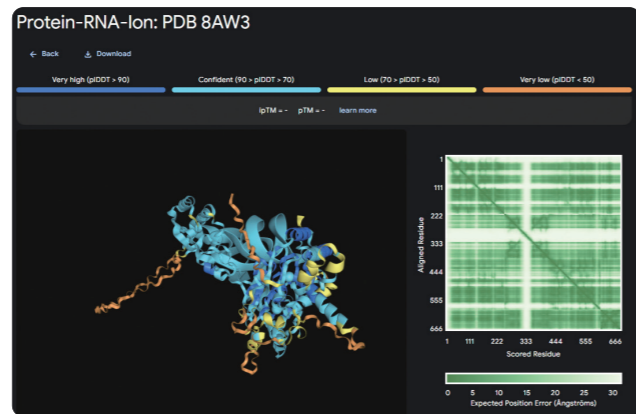
항목	주요 지수 변화량
걸음 수	900~1,773보 증가
외출 시간	6~30분 증가
TV 시청 시간	평균 71% 감소
우울증 평가	4.8 → 2.2 감소
삶의 질 만족도	0.89 → 0.92 증가

▲ [표 1-1] 인공지능 스피커 사용 후, 독거노인의 주요 지수 변화량

출처: 용인텍레스 터치케어 서비스



인공지능은 의학, 외국어 번역 및 문헌학, 예술 등 다양한 학문 분야의 문제를 해결하는 데도 활용되고 있다. 의료 영상을 보고 질병을 진단하고, 오타가 있는 문장도 제대로 번역한다. 변화된 한글을 인공지능에 학습시켜 고서적의 연대를 추정하기도 하고, 작곡도 하고 그림도 그린다. 또한 단백질 구조를 예측할 때도 인공지능이 활



▲ 단백질-RNA-이온: PDB 8AW3 구조

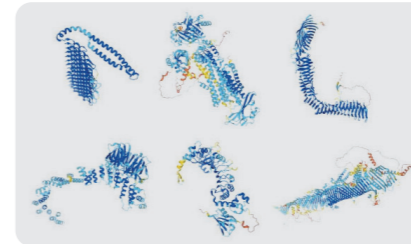
출처: 알파폴드 서버

10 알파폴드3

구글 딥마인드가 공개한 알파폴드3는 구조 예측을 뛰어넘어, 단백질이 우리 몸 안에서 생체 분자와 어떻게 상호작용하는지 예측할 수 있다. 이는 신약 및 치료법 개발 속도를 가속화할 수 있다. 알파폴드3는 알파폴드2에 비해 상호 작용 예측 정확도가 50% 증가하였고, 중요 상호 작용 유형에 대해 정확도가 2배 증가하였다.

알파폴드3를 이용하여 바이오 재료 개발, 내구성 있는 작물 개선, 약물 설계 가속화, 유전체 연구 등 다양한 과학적 발전을 앞당길 수 있다.

용되고 있다. 단백질 구조를 예측하는 학술 대회에서 인공지능 '알파폴드'가 우승하였으며, 코로나 19의 경우 유전 정보가 공개되자 바이러스를 구성하는 단백질 구조 예측에 성공하였다. 알파폴드는 탄생한 지 몇 년 만에 인류가 오랜 시간 밝혀낸 약 2억 개의 단백질 구조를 예측하는 데 성공했다. 이제는 단백질 구조를 예측하는 데 그치지 않고 단백질 구조를 설계하기 시작하였으며, 이는 신약 개발 등의 생명 과학 분야에서 큰 성과를 거둘 수 있다.

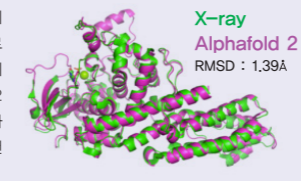


▲ 알파폴드가 예측한 단백질의 3D 구조

알고 가기 알파폴드의 원리

알파폴드(alphafold 2)는 2020년 12월에 열린 단백질 구조 예측 대회에서 월등한 차이로 우승했다. 딥마인드는 2022년 알파폴드를 오픈 소스로 공개했고, 2억 정도의 단백질 구조 예측 결과를 함께 발표했다. 이는 이 시기까지 밝혀진 모든 단백질 구조를 예측한 것으로 볼 수 있다. 알파폴드 2는 합성곱 신경망 구조가 단백질 예측에는 적절하지 않은 것으로 판단하여 떨어져 있는 정보 간의 관계를 파악하는 데 뛰어난 효과를 보여 준 트랜스포머를 적용했다.

• 합성곱 신경망 이미지 처리에 탁월한 성능을 보이는 신경망으로 인접한 정보의 관계를 파악한다.



▲ 알파폴드와 실험으로 얻은 단백질 구조 알파폴드(분홍색)가 예측한 구조와 실험(초록색)으로 얻은 구조가 거의 일치한다.

해 보기 4 인공지능을 활용한 문제 해결 사례 찾아보기

조사·분석하기

1 인공지능을 활용하여 실생활이나 학문 분야의 문제를 해결한 사례를 찾아보자.

지자체에서 무인 회수기를 통해 캔과 페트병을 분리수거하는 장치를 운영하고 있다. 무인 회수기 안에 페트병이나 캔을 넣으면 인공지능이 캔과 페트병을 자동으로 분류하는 장치이다.

2 인공지능을 활용하였을 때와 활용하지 않았을 때의 차이점을 작성해 보고, 친구들과 이야기해 보자.

인공지능을 사용하지 않았을 때는 사람이 직접 분리 수거를 해서 아무렇게나 버리는 사람이 있으면 분류가 잘 되지 않았다. 하지만 무인 회수기는 하나의 입구에 넣으면 자동으로 분류하는 장치로 분리수거가 원활하게 잘 될 수 있을 것이다.

참고 주소

- <https://blog.amariver.com/>
- <https://www.aitimes.com/news/articleView.html?idxno=159481>
- <https://golgi.sandbox.google.com/about>

11 인공지능을 활용한 문제 해결 사례

• 스마트홈 지능형 케어 서비스: 움직임이 없거나 이상함을 감지하면 가족에게 연락이 가도록 설계된 AI 상황 인지 케어 서비스

참고 주소

- <https://www.hellodd.com/news/articleView.html?idxno=97393>

* 용어 해설

- **트랜스포머(transformer):** 구글이 자연어 처리를 위해 만든 모델로, 순차 데이터 내의 관계를 추적해 맥락과 의미를 학습하는 신경망이다. 자연어 처리뿐 아니라 이미지 처리 분야에서도 좋은 결과는 내고 있다.

단백질 구조 예측 대회(CASP: Critical Assessment of techniques for protein Structure Prediction)

1994년 시작하여 2년마다 열리는 국제 행사이다.

해 보기 4 지도 방법

- 인공지능을 활용한 문제 해결 사례를 확인하여 인공지능을 활용할 수 있는 다양한 방안을 생각해 볼 수 있도록 지도한다.
- 인공지능이 적용된 것과 적용되지 않은 소프트웨어를 구분할 수 있도록 지도한다.

17

• 자율주행 자동차: 위험 발생 시 인지와 예측, 대응, 행동이 가능한 인공지능 시스템 내장

• 암 검사: 뉴욕 맨해튼의 한 병원에서 AI를 이용하여 간암, 직장암, 전립선암 등의 발생을 94% 정확도로 예측

참고 주소

- <https://www.aitimes.com/news/articleView.html?idxno=129809>

• 기타 분야: 가짜뉴스 적발, 자연재해 대비, 기후 예측, 기아 구제, 야생 동물 보호 등 다양한 분야에서 인공지능이 활용되고 있다.

※ 용어 해설

• **AI+X**: X의 사전적 의미는 '정해지지 않은'이다. AI+X에서 X는 인공지능과 융합 가능한 다양한 산업을 의미한다고 볼 수 있다. 인공지능 국가 전략에서는 제조 및 산업, 바이오·의료, 스마트 시티·건설, 교통·물류, 에너지, 환경, 네트워크, 농수산, 문화·예술, 법무, 국방 등 전 영역에 걸쳐 AI 융합 프로젝트를 추진한다고 밝혔다.

인공지능의 활용 분야

- **영업**: 판매 여부 또는 판매량 예측, 영업 챗봇, 에이전트 코칭 등
- **서비스**: 음성 인증, 고객 서비스 챗봇, 설문 조사 후 분석 등
- **인사**: 고용, 성과 관리, 인공지능 비서 등
- **헬스케어**: 환자 데이터 분석, 서비스 제공 우선 순위 판단, 조기 진단 및 처방, 의료 시장 분석, 의료 챗봇 등
- **금융**: 사기 탐지, 로보어드바이저, 금융 챗봇, 대출 심사 등
- **제조업**: 생산 공장 최적화, 양품/불양품 판별 등으로 공장의 생산 공정을 개선
- **공공 분야**: 시민 의식주 개선, 정신 건강, 다양한 중독 문제 등의 위험 요소를 예측해 효과적인 정책 지원

*AI+X

인공지능(AI)과 다양한 산업(X)의 융합을 의미한다. AI+교육, AI+의료 등 인공지능과 해당 산업과의 융합을 통칭하는 용어이다.

02 인공지능의 미래 활용 분야(AI+X)*

인공지능 기술이 빠르게 발전하면서 의료, 교육, 운수, 번역, 산업 등 사회 전 분야에서 인공지능이 활용되고 있다. 앞으로 활용 분야는 계속 늘어날 것이다.

AI+의료



질문의 발음을 막는 '예방', 질문을 읽고 있는 사람을 찾아내는 '진단', 증세 악화를 개선하는 '치료'에 모두 적용되어 활용될 수 있다.

AI+비서



인공지능이 개인의 비서 역할을 하는 서비스로 인공지능 스피커가 대표적인 예다. 일정 관리, 외국어 번역, 검색, 음악 관리, 날씨 정보 제공, 사물 인터넷 제어 등을 제공한다.

AI+교육



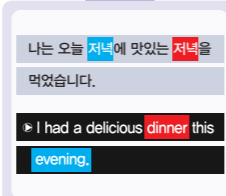
학습자에게 자료 수집, 기초 분석, 시각화 등을 제공하고 교수자에게 수업 현황 보고, 문제에 대한 대안 제시 등 평가 업무와 학급 관리에 도움을 줄 수 있다. 학습자의 개별화 학습, 맞춤형 자기주도 학습이 가능하며, AI 디지털 교과서로 제공될 수 있다.

AI+운수



카메라, 라이다, 레이더 등의 센서를 사물을 인식하고 인공지능으로 최적의 판단을 하여 최종 목적지까지 안전하게 이동한다.

AI+번역



인공신경망을 이용한 기계 번역은 문맥을 파악할 수 있어 보다 자연스럽게 정확한 번역이 가능하다. 딥러닝으로 빅데이터를 학습해 시간이 지날수록 정확도는 높아진다.

AI+산업



공장 내 각종 장비에 센서가 설치되어 데이터를 수집하고, 수집한 데이터를 분석하여 최적의 공정을 설계하고 제어하는 시스템을 통해 생산 기간 단축, 맞춤형 제품 개발, 에너지 효율 증가, 재고 비용 감소 등의 효과가 나타날 수 있다.

소단원 1분 요약

- 1 인공지능은 인간의 지능을 모방한 컴퓨터 시스템으로 탐색, 추론, 학습의 특성을 지닌다.
- 2 튜링 테스트는 인공지능과의 대화를 통해 인공지능에게 지능이 있는지 판단하는 테스트지만, 인공지능의 여러 가지 특징은 배제하고 문자 대화만을 통해 지능을 판단한다는 제약이 있다.
- 3 우리나라는 모든 산업과 인공지능과의 융합을 의미하는 AI+X를 추진하고 있으며, 인공지능은 실생활이나 다른 학문 분야에서의 문제 해결에 다양하게 활용되고 있다.

출처 과학기술정보통신부(2023), 전국민 AI 일상화 실행계획 비전 및 과제

12 전국민 AI 일상화

2023년, 정부는 '전 국민 인공지능(AI) 일상화'를 추진하고, 초거대 AI 경쟁력 강화를 위해 '대한민국 인공지능 도약 방안'을 발표했다. 이것은 AI 기술을 대중화하고 생활과 산업 분야에서 널리 활용하여 국가의 경쟁력을 향상시키는 것을 목표로 한다.

- **AI로 국민 일상을 풍요롭게**: AI 기술을 활용하여 의료, 교육, 교통, 금융 등 다양한 분야에서 혁신적인 서비스와 솔루션을 제공하여 국민의 삶의 질을 향상시킨다.
- **AI 내재화로 산업·일터를 혁신**: 기업이나 산업 분야에서 AI를 적극적으로 도입하여 생산성을 향상시키고 새로운 비즈니스 모델을 개발하여 경쟁력을 강화한다.

- **AI를 가장 잘 사용하는 정부**: 정부는 AI 기술을 적극적으로 활용하여 공공 서비스의 효율성과 투명성을 높이고, 정책 결정에 AI를 활용하여 더 나은 의사 결정을 지원한다.
- **AI 일상화 기반 조성**: AI 인프라 구축, 데이터 개방 및 활용, AI 교육 및 인력 양성, AI 윤리 및 사회적 영향 등 AI 일상화를 위한 기반 환경을 조성한다. 이를 위해 정부는 AI 핵심 인프라 구축, AI 교육 및 인력 양성, AI 윤리 및 사회적 영향 등을 고려한 정책을 수립하고 지원한다.

탐구 활동

교실(학교)에서 발생하는 문제를 해결하기 위한 인공지능 설계하기

1 교실(학교)에서 발생할 수 있는 문제를 찾아보자.

● 겨울철 난방기 가동으로 인해 수업 시간에 조는 학생들이 늘어나는 문제

(1) 점심시간에 많은 잔반이 발생하는 문제

(2)

(3)

2 발견한 문제 중 인공지능이 해결할 수 있는 문제와 해결하기 어려운 문제를 구분해 보고, 그 이유를 써 보자.

해결할 수 있는 문제	이유
● 따뜻한 온도에 조는 학생 많은 잔반 발생	● 온도에 따라 조는 학생이 늘어나는 것을 알 수 있다. 학생 개개인별 식사량을 분석하여 맞춤 배식을 한다.
해결하기 어려운 문제	이유
미인정 결석	학생이 이유 없이 학교를 등교하지 않는 것으로, 해결을 위한 데이터 수집도 어렵다.

3 발견한 문제를 해결하기 위해 인공지능에게 필요한 역할은 무엇인지 논의해 보자.

● 따뜻해서 졸리지 않도록 온도를 조절한다.

학생 개개인의 얼굴과 잔반량을 학습하여 개개인에 맞춤 배식을 할 수 있도록 안내한다.

4 문제 해결을 위해 인공지능이 적용된 지능 에이전트의 형태를 그림으로 그리고, 기능을 설명해 보자.

배식 입구에 카메라, 얼굴 인식, 배식원들에게 식사량 제공, 잔반통 주변 카메라와 잔반량 측정 센서

12쪽의 로봇 청소기 그림을 참고하여 답안을 작성해 보세요.



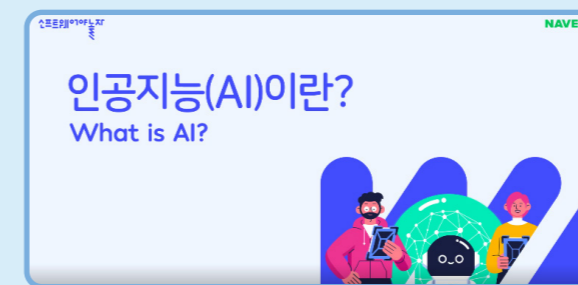
탐구 활동

지도 방법

- 교실이나 학교에서 발생할 수 있는 문제를 찾는 과정에서 친구를 놀리거나 장난으로 문제를 제시하는 학생이 없도록 지도한다. 인공지능을 활용하여 실생활의 문제를 해결하기 위한 경험을 쌓는 과정으로 혼자서 하기 힘들어하는 학생은 팀으로 진행할 수 있도록 안내한다.
- 학교에서 생활하며 불편함을 느꼈거나 개선했으면 하는 부분에 대해 생각해 볼 수 있도록 안내한다.
- 학교에서 발생할 수 있는 문제들: 교권의 추락, 교사와 학생의 관계에서 오는 문제, 학생 사이의 문제, 수업 시간에 자는 학생, 수업 시간에 떠드는 학생, 학교 폭력, 언어 폭력, 체벌, 악성 민원, 따돌림 등

▶ 참고 동영상 인공지능에 대해 알기

- 제목: 인공지능이란 무엇인가?
- 영상 내용: 인공지능에 대한 개념, 활용 분야, 인공지능에 의한 사회 변화에 대해 알아본다.



주소: https://www.playsw.or.kr/artificial/view/playswvtv/680?currentTab=ai_data_video_hello_ai&path=artificial

▶ 참고 동영상 인공지능의 활용

- 제목: 인공지능 활용
- 영상 내용: 인공지능을 활용한 다양한 사례와 활용 방안에 대해 알아본다.



주소: https://www.playsw.or.kr/artificial/view/playswvtv/708?currentTab=ai_data_video_hello_ai&path=artificial

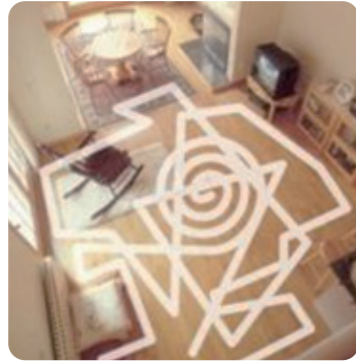
참고 자료

• 교과서 12쪽 관련 내용

• 인공지능이 적용되지 않은 로봇 청소기

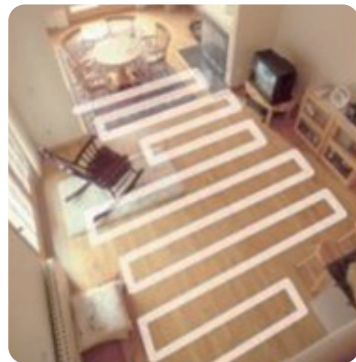
로봇 청소기에 설정된 알고리즘에 따라 움직인다. 랜덤 방식, 매핑 방식, 비전 방식으로 나눌 수 있으며, 장애물을 피해 설정된 동작대로만 움직인다.

1 **랜덤 방식:** 이 방식은 로봇 청소기가 설정된 구역 내에서 무작위로 움직이며 청소를 수행한다. 이 방식은 청소를 효율적으로 수행하지 못할 수 있지만, 간단한 알고리즘으로 구현할 수 있다.



▲ 랜덤 방식

2 **매핑 방식:** 이 방식은 로봇 청소기가 청소할 구역을 지도화하고, 이를 기반으로 효율적인 청소 경로를 생성한다. 이를 위해 로봇 청소기는 센서나 카메라를 사용하여 주변 환경을 감지하고, 이를 기반으로 지도를 작성한다.



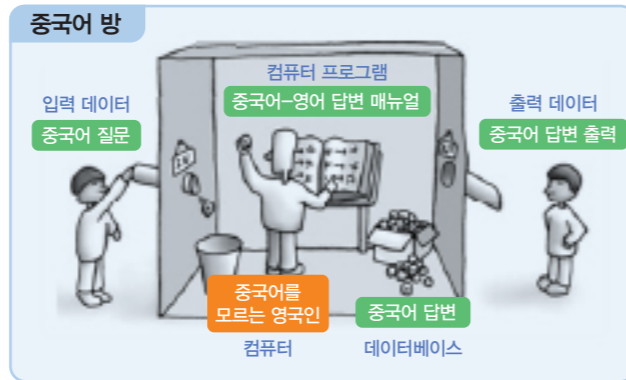
▲ 매핑 방식

3 **비전 방식:** 이 방식은 로봇 청소기가 카메라나 기타 비전 시스템을 사용하여 청소할 구역을 인식하고, 이를 기반으로 청소를 수행한다. 이를 위해 로봇 청소기는 이미지 처리 알고리즘을 사용하여 청소할 구역을 감지하고, 이를 기반으로 청소 경로를 생성한다. 이 방식은 가장 효율적이고 정확한 청소 방식 중 하나이다.

• 교과서 15쪽 관련 내용

• 중국어 방

미국의 철학자 존 설(John Searle, 1932~)은 '중국어 방(Chinese Room)'이라는 사고(思考) 실험을 제안했다. 이 실험은 다음과 같이 설정된다. 영어는 사용할 수 있지만, 중국어는 모르는 사람이 방 안에 있다. 외부에서 중국어로 된 질문지를 전달하면, 이 사람은 영어로 된 지침서에 따라 중국어 답변 중 하나를 선택하여 방 밖으로 내보낸다. 이때 외부에서는 이 사람이 멋진 중국어 시를 짓는다고 감탄한다.



▲ 존 설의 인공지능 비판

이러한 상황에서 “방 안에 있는 사람이 과연 중국어를 이해하였다고 할 수 있는가?”라는 비판이 제기될 수 있다. 이 사람은 중국어를 이해하지 못하고도 중국어 입력값에 해당하는 영어 출력값을 선택했기 때문에, 적절한 중국어 답변이 나왔다고 해서 이것만으로 기계가 생각을 했거나 이해를 했다고 보기는 어렵다는 주장이다.

• 교과서 15쪽 관련 내용

• 튜링 테스트에 대한 이견

현재는 튜링 테스트의 결과를 중요하지 않다고 이야기하는 경우도 많다. 대화형 생성형 인공지능을 조금만 조정하여도 튜링 테스트를 통과할 수 있을 것이라고 말하는 학자도 많이 있다. 2024년 미국의 한 대학에서 'chat GPT'가 튜링 테스트를 통과하지 못하였다고 발표하였다. 통과하지 못한 이유는 수준 낮은 답 때문이 아니라 인간보다 뛰어난 대답을 하였기 때문이다.

02

인공지능과 탐색

수업 시간: 2시간 교과서 쪽수: 20~29쪽

학습 목표	<ul style="list-style-type: none"> 인공지능에서 탐색의 중요성을 설명할 수 있다. 문제 해결을 위한 탐색 과정을 설계할 수 있다.
수업 방법	강의, 토론, 발표
준비물	교사 교과서, 관련 교수 학습 자료 학생 필기도구

단계	교수 · 학습 방법	지도상의 유의점
도입	<p>생각 열기</p> <p>내비게이션을 사용했던 경험, 인공지능이 내비게이션에서 어떻게 쓰이는지에 대해 친구들과 이야기를 나누어 보도록 한다.</p>	<ul style="list-style-type: none"> 실제로 지도 애플리케이션을 실행하여 인공지능이 적용된 부분을 친구와 찾아보게 하며, 흥미를 유발한다.
전개	<p>1 탐색의 이해</p> <ul style="list-style-type: none"> 미로 찾기 예시를 통해 다양한 탐색 방법이 있으며, 탐색의 방법에 따라 효율이 달라질 수 있음을 이해할 수 있도록 한다. 알파고와 내비게이션 등 인공지능에서 탐색이 어떻게 쓰이고 있는지 확인함으로써, 인공지능에서 탐색의 중요성을 깨달을 수 있도록 한다. 	<ul style="list-style-type: none"> 문제 상황에 따라 탐색 알고리즘에 따라 문제 효율이 달라질 수 있으며, 어떤 방법이 가장 지능적인 해결 방법이었는지 이야기를 나누며 직접 깨달을 수 있도록 지도한다. 인공지능에서 탐색이 쓰이는 다른 사례를 찾아볼 수 있도록 지도한다.
	<p>2 문제 해결을 위한 탐색 과정 설계</p> <ul style="list-style-type: none"> 문제 해결을 위한 탐색 과정으로 문제의 상태 정의, 초기 상태와 목표 상태 설정, 수행 작업 파악, 탐색 전략 세우기가 있음을 안내하고, 전구 켜기 문제 활동을 통해 직접 문제를 단계적으로 해결해 보도록 한다. 	<ul style="list-style-type: none"> 전구 켜기 시뮬레이션 프로그램을 통해 직관적으로 파악할 수 있도록 예시를 보여 준다.
정리	<p>탐구 활동</p> <ul style="list-style-type: none"> 전구 켜기 연습 문제를 통해서 문제 해결을 위한 탐색 과정의 모든 단계를 직접 해 보도록 지도한다. 	<ul style="list-style-type: none"> 탐색 결과를 친구들과 나누며, 다양한 탐색 전략에 대해 이야기를 나누도록 한다.
평가	<ul style="list-style-type: none"> 인공지능에서 탐색의 중요성을 설명할 수 있는가? 문제 해결을 위한 탐색 과정을 설계하고 적용할 수 있는가? 	<ul style="list-style-type: none"> 자기 평가 문항을 제시하고, 학습자의 이해도에 따른 피드백을 하여 평가 기준을 달성하도록 돕는다.

인공지능과 탐색

- 학습 목표**
- 인공지능에서 탐색의 중요성을 설명할 수 있다.
 - 문제 해결을 위한 탐색 과정을 설계할 수 있다.

학습 요소 탐색의 중요성, 탐색 과정 설계

• 제시 의도 •

인공지능에서 중요한 개념인 탐색은 다양한 인공지능 시스템에 적용되고, 내비게이션은 그중 가장 친숙한 인공지능 시스템이다. 학생들에게 어렵게 느껴질 수 있는 탐색의 개념을 실생활과 연결하여 더 쉽게 이해할 수 있고, 인공지능에서 탐색이 어떤 식으로 쓰이는지 생각해 볼 수 있다.

지도 방법

- 내비게이션의 사례를 통해 내비게이션에는 인공지능 기술과 탐색이 사용되는 것을 알 수 있도록 지도한다.
- 내비게이션이 없던 시절에 관해 이야기를 나누며, 인공지능과 탐색은 우리 삶을 편리하게 변화시켰음을 알 수 있도록 지도한다.
- 인공지능 탐색에 대한 영상을 활용하거나 지도 프로그램을 직접 실행하여 동기 유발에 사용한다.

예시 답안

교통사고나 도로 공사 등의 교통 상황, 거리와 소요 시간, 도로의 종류 등을 고려한 것이다.

🧠 생각 열기 내비게이션의 길 찾기

내비게이션에 목적지를 입력하면 출발지에서 목적지로 가는 다양한 경로를 탐색하여 안내해 준다.



내비게이션의 AI 추천 경로는 무엇을 고려한 것일까?

20

1 탐색의 개념

국어사전에서 '탐색'이란 단어를 찾아보면, '드러나지 않은 사물이나 현상 따위를 찾아내거나 밝히기 위하여 살피고 찾는 것'이라고 정의되어 있다. 탐색은 문제 해결을 위해 사용하는 가장 기본적인 방법으로, 문제 상황에서 문제를 해결하기 위한 방법을 찾고, 가장 좋은 방법을 선택하여 해결한다. 컴퓨터의 탐색도 문제 상황의 모든 상태 공간에서 문제의 답을 찾아가는 과정이라고 할 수 있으며, 인공지능에서도 탐색이 사용되어 방대한 상태 공간을 줄여 더 빠르게 정답에 가까워진다.

2 탐색의 예

탐색의 예를 보면, 탐색은 어떤 특별한 기술이 아니라 일상생활 곳곳에서 일상적으로 하는 문제 해결 방법임을 알 수 있다. '인터넷에서 검색하기'는 현대인이 문제 해결을 위해 사용하는 가장 흔한 탐색 방법이 되었다. 학생들과 또 다른 예를 찾아보자.

지도 예

선생님: 교과서에 쓰여 있는 것 외에 다른 탐색의 예를 찾아볼까요?
학 생: 쇼핑몰에서 원하는 가게 찾기, 컴퓨터에서 파일 찾기, 퍼즐 맞추기 등이 있어요.

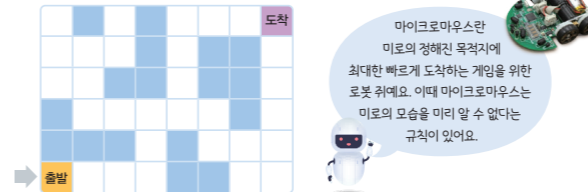
1 탐색의 이해

우리는 해결해야 할 문제가 생기면 여러 상황과 조건을 고려하여 실행 가능한 선택지 중에서 가장 좋은 방법을 선택하여 최적의 해결책을 찾아낸다.

01 탐색의 개념

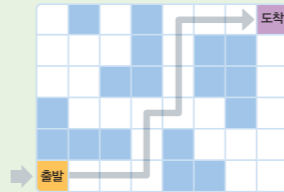
탐색이란 문제 상황에서 답을 찾아가는 과정으로, 컴퓨터과학자들은 인간의 탐색 방법을 모방하여 컴퓨터에 적용하고자 노력해 왔다. '미로 찾기' 문제를 예로 들면, 컴퓨터에 출발지와 목적지를 입력하고, 다양한 탐색 알고리즘을 적용함으로써 목적지까지의 경로를 자동으로 찾아내는 해결 전략을 구현하려고 노력하였다.

마이크로마우스가 아래 그림과 같은 미로를 다양한 알고리즘을 적용하여 탐색하는 과정을 살펴보고 탐색에 대해 알아보자.



1 A 알고리즘

- 1 앞으로 세 칸 이동한다.
- 2 왼쪽으로 90도 회전한다.
- 3 앞으로 두 칸 이동한다.
- 4 오른쪽으로 90도 회전한다.
- 5 앞으로 한 칸 이동한다.
- 6 왼쪽으로 90도 회전한다.
- 7 앞으로 세 칸 이동한다.
- 8 오른쪽으로 90도 회전한다.
- 9 앞으로 세 칸 이동한다.



A 알고리즘은 출발부터 도착까지 모든 과정을 일일이 명령하면 그대로 수행하는 방식으로 그림의 미로를 빠져나갈 수 있다. 가장 간단한 방식이지만 출발지나 도착지가 변경되거나 미로의 모양이 바뀌면 매번 새롭게 명령해야 하는 번거로움이 있고, 미로의 모습을 알아야만 정확한 명령이 가능하기에 미로의 모습을 모르는 상황에서는 수행하기 어려운 알고리즘이다.

21

3 A 알고리즘

A 알고리즘은 전형적으로 순차적으로 문제를 해결하는 방법으로 단순 명령을 그대로 실행하는 기계적인 방법이다. 가장 간단하게 문제를 해결할 수 있지만, 마이크로마우스는 미로의 모양을 모르기 때문에 사용할 수 없는 방법이기도 하다. 또한 문제의 상황이나 조건이 바뀌면 같은 방법으로 해결할 수가 없기에 지능적인 문제 해결 방법이라고 할 수 없다.

지도 방법

- 터틀 프로그래밍(turtle programming) 등 그래픽 프로그램 또는 피지컬 교구인 햄스터 봇(hamster bot) 등을 활용하여 미로를 빠져나가는 프로그래밍을 직접 해 보거나 미로를 빠져나가는 동영상을 보여 주어, 학습의 흥미를 높이도록 한다.
- 알고리즘을 직접 프로그래밍하거나 의사코드(pseudo code)를 작성하여 프로그램을 작성하거나 실행할 때의 효율을 비교해 보는 것도 좋은 방법이다. 이때 미로는 두 개 이상 준비하여 검증의 신뢰도를 높이도록 한다.

탐색의 예

- 도서관에서 책 찾기
- 체스에서 다음 수 찾기
- 인터넷에서 자료 검색하기

▶ 참고 동영상 마이크로마우스 대회 영상

- 제목: 잘못된 데이터 수정하기
- 영상 내용: 2022년에 열렸던 전국 지능형 로봇 대회 마이크로마우스 부분에서 1위를 수상한 마이크로마우스가 미로를 빠져나가는 모습이 담긴 영상이다.



- 주소: <https://www.youtube.com/watch?v=WdYcbnStuos> (2022년 제17회 전국 지능형 로봇 대회 마이크로마우스 부문 1위: 단국대학교 MAZE채널)

지도 방법

A 알고리즘부터 C 알고리즘까지 비교하며, 지능적 탐색의 특징에 대해 이해하도록 지도한다.

*** 용어 해설**

• **우수법과 좌수법:** 우수법(右手法)은 오른손을 벽에 대고 오른쪽으로 회전할 수 있는 곳에서는 모두 회전하며 미로를 탈출하는 방법이고, 좌수법(左手法)은 왼손을 벽에 대고 왼쪽으로 회전하며 미로를 탈출하는 방법이다. 이 방법은 미로에 대한 정보 없이 단순한 규칙에 의해 사용되는 방법으로 무정보 탐색이라고 할 수 있다.

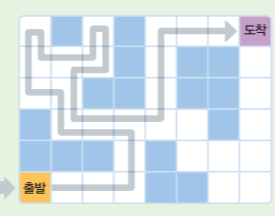
• **휴리스틱:** 문제를 사용하기 위해 직관적이고 경험적인 방법을 사용하는 것을 의미한다. 본문의 C 알고리즘은 휴리스틱(heuristic)을 사용한 알고리즘이다.

좌수법

미로 문제를 해결하기 위해 벽을 따라 이동하는 전략을 사용하는 방법이다. 왼손을 벽에 대고 왼쪽으로 회전할 수 있는 곳에서는 모두 왼쪽으로 회전하며 미로를 탈출한다.

2 B 알고리즘

- ① 왼쪽에 벽이 있다면, 그 벽을 따라 움직인다.
- ② 만약 왼쪽에 빈칸이 있다면, 왼쪽으로 회전 후, 앞으로 직진한다.
- ③ 왼쪽으로 회전할 수 없다면, 앞으로 직진한다.
- ④ 만약 앞이 막혔다면 오른쪽으로 회전한다.
- ⑤ 목적지에 도달할 때까지 ①~④를 반복한다.



B 알고리즘은 왼쪽 벽을 따라가며 왼쪽으로 회전하여 미로 탐색 문제를 해결하는 방법을 사용한다. 이 방식은 미로의 모습을 알지 못해도, 간단한 단일 연결 미로는 반드시 탈출할 수 있는 탐색 방법이다. 하지만 탐색 경로가 항상 최단 경로를 보장하지는 못하며, 출발점이 미로의 가운데 있거나 이중으로 연결된 미로라면 탈출하지 못할 수도 있다는 한계점이 있다.

3 C 알고리즘

- ① 목적지에서 몇 칸이 떨어져 있는지 미로 칸에 숫자를 표시한다.
- ② 길을 따라 이동하고 분기점에서는 작은 숫자의 칸을 선택하여 이동한다.
- ③ 분기점의 숫자가 같다면 미로의 위쪽 칸을 선택한다.
- ④ 진행 방향에 더 이상 길이 없다면 직전의 분기점까지 되돌아오고 ③번과 다른 방향으로 탐색한다.
- ⑤ 목적지에 도착할 때까지 ①~④를 반복한다.



C 알고리즘은 각 미로 칸마다 숫자 정보를 부여하고 탐색하는 알고리즘이다. 사람이 길을 가는 방향을 정한다면 목적지의 반대로 이동하지 않고 목적지가 있는 방향으로 이동할 것이다. 미로 탐색도 목적지에 가까이 가는 방향으로 이동하게 하면, 불필요한 탐색 범위를 줄일 수 있다. 미로의 모습은 알 수 없어도 출발지와 목적지의 위치와 미로의 크기를 알고 있기에 그 정보를 활용하는 것이다. 미로의 칸이 벽인지 길인지 구분할 수 없는 상황에서 단순하게 목적지와 몇 칸 떨어져 있는지 미로를 구조화한 배열에 저장한다. 미로를 탐색하며 벽이 아닌 길을 따라 이동하게 하고, 분기점에서 배열에 저장된 숫자가 작은 칸을 선택하게 하면 목적지와 가까운 방향으로 이동하게 되어 탐색 범위를 좁힌다.

4 B 알고리즘

B 알고리즘은 미로 찾기에서 가장 흔하게 쓰이는 좌수법 알고리즘이다. 왼쪽 손을 벽에 대고 계속 왼쪽으로 회전하는 방식을 사용하는 것으로, 미로의 벽이 하나로 연결되어 있다면 결국에는 벽이 끝나는 지점인 출구에 도착하게 되는 것이다.

5 C 알고리즘

C 알고리즘은 정보를 활용하는 알고리즘이다. 간단하게 정리된 최상우선탐색 알고리즘과 비슷하다고 볼 수 있다. 무정보 알고리즘을 사용했던 A, B 알고리즘과 달리 주어진 정보인 미로의 크기, 출발지와 도착지의 위치를 이용하여

구획을 나누어 가중치를 매기고가중치가 적은 쪽으로 이동하는 방식으로, 인간이 사용하는 방법과 가장 유사하게 지능적으로 이동하였다고 생각할 수 있다.

6 인공지능에서 탐색의 중요성

결국 탐색은 문제의 답을 효율적으로 찾을 수 있게 하는 것이다. 인공지능은 대량의 방대한 데이터를 이용하기 때문에 탐색을 통해 탐색의 규모를 좁혀 속도를 높이는 것이 필수이다. 인공지능에 탐색이 사용되지 않았다면, 느린 계산으로 인해 답을 얻어도 활용하기 어려웠을 것이다.

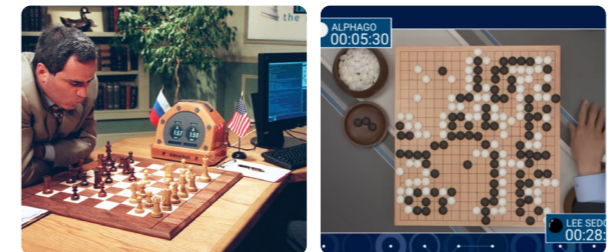
탐색은 최근 의료 분야에서도 중요한 역할을 한다. 인공

이처럼 같은 모습의 미로를 탐색해도 알고리즘에 따라 탐색 범위와 시간이 달라진다. 지능적인 탐색이란 탐색의 범위나 시간을 좁혀 탐색을 최적화한 것을 의미한다. C 알고리즘은 항상 최단 거리를 보장할 수는 없지만 대체로 좋은 경로를 선택하는 효율적인 탐색 방법으로 A, B 알고리즘보다 지능적인 탐색 방법이라 할 수 있다.

02 인공지능에서 탐색의 중요성

복잡한 문제를 탐색으로 해결하려는 노력은 인공지능 연구의 초창기부터 이어져 왔다. 세계 체스 챔피언에게 패배했지만, 계속 도전하여 결국 승리를 거둔 딥 블루(Deep Blue)는 인공지능의 놀라운 발전 가능성을 보여 주었다. 다른 대표적인 예는 알파고(AlphaGo)*다. 바둑은 대략 361!개의 방대한 경우의 수가 존재하기 때문에 성능이 아주 뛰어난 컴퓨터를 이용해도 모든 경우의 수를 계산하기 어렵다. 따라서 바둑은 컴퓨터가 인간을 이길 수 없는 영역으로 여겨졌다. 하지만 알파고는 수많은 경우의 수 중에서 승리에 유리한 수만을 탐색하는 방법으로 탐색 범위를 획기적으로 좁혔다. 또한 딥러닝(deep learning)*으로 방대한 분량의 바둑 전문가 기보* 데이터를 학습하고, 학습의 결과를 다시 탐색에 반영하며, 이길 확률이 높은 수를 선택함으로써 알파고는 인간에게 승리할 수 있었다.

실생활에서도 탐색을 사용하는 다양한 인공지능 사례들이 있는데 그 대표적인 예가 내비게이션이다. 내비게이션은 실시간 교통 정보를 연동하여 목적지까지 가는 최적의 길을 안내한다. 데이터를 기반으로 공사 중인 구간, 사고 현황 등의 데이터를 수집해 막히는 구간을 예측하고, 필요한 경우 우회 도로로 안내한다. 이때 다양한 상황을 반영하여 출발지부터 목적지까지 운행하는 수많은 경로를 모두 탐색하려면 많은 저장 공간과 시간이 필요하기 때문에 내비게이션 시스템을 제대로 제공하기 위해서는 탐색의 범위를 좁혀 가는 지능적 탐색으로 효율적이고 빠른 탐색을 하는 것이 필수다.



▲ 딥 블루와 알파고

*알파고
대표적인 인공지능 바둑 프로그램으로 이세돌 9단과의 경기에서 5전 4승 1패를 기록했다.

*딥러닝
128쪽을 참고한다.

*기보
바둑에서 바둑돌이 놓이는 순서와 위치를 기록한 것을 말한다.

출처 위키백과

지도 방법

인공지능을 생각하면 딥러닝이나 기계학습에 대해서만 생각하기 쉽지만, 탐색 또한 인공지능의 발전에 큰 역할을 했음을 깨달을 수 있도록 지도한다.

*** 용어 해설**

• **딥 블루:** IBM이 만든 체스에 특화된 인공지능 컴퓨터이다. 딥 블루(Deep Blue)는 칩 테스트 프로젝트의 일환으로 당시 세계 체스 챔피언이었던 가리 카스파로프(Garry Kasparov)를 이겼으며, 컴퓨터는 인간을 이길 수 없다는 오랜 패러다임을 깨는 계기가 되었다.

▶ 참고 동영상 인공지능의 시작, 탐색

- **제목:** EBS 이숲[인공지능 첫걸음] 인공지능의 시작, 탐색
- **영상 내용:** 컴퓨터는 어떻게 체스를 둘 수 있을까? 체스 게임에서 컴퓨터는 모든 경우의 수를 계산한 후, 승리를 위한 경우의 수를 탐색한다. 퍼즐 게임에서도 탐색이 사용되는데, 맹목적 탐색과 경험적 탐색을 중심으로 탐색 원리를 이해해 보자.



사람처럼 생각할 수 있는 기계.

• 주소: <https://youtu.be/fG8T07Csz6w?si=X8WXS0Bzj4sH6-AQ>

문제의 상태 정의하기

1 **상태(state)**: 문제를 해결하는 과정에서의 특정 순간을 말한다.

예 미로 내에서의 특정 위치

2 **상태 공간(state space)**: 모든 가능한 상태의 집합이다.

예 미로의 모든 가능한 위치

3 **초기 상태(initial state)**: 문제의 처음 즉, 문제 해결이 시작되는 상태를 의미한다.

예 미로의 입구에서 있기

4 **목표 상태(goal state)**: 문제가 해결된 상태로 답을 구한 상태를 의미한다.

예 미로의 출구로 나옴

5 **상태 변화**: 초기 상태에서 목표 상태로 가는 과정에서의 문제의 상황이 변하는 것을 의미한다.

예 미로에서 있는 위치 변화

지도 방법

• 학생들이 상태, 상태 공간 등의 용어에 대해 이해할 수 있도록 다양한 예시를 들어 설명한다.

• 전구 켜기 문제 외에 다른 문제들을 제시하여 개념을 정확히 이해할 수 있도록 지도한다.

문제의 구조화

복잡한 문제를 쉽게 해결할 수 있도록 관리가 가능한 여러 하위 문제로 문제를 분해하거나 단순화하여 체계적으로 정리하는 작업을 의미한다.

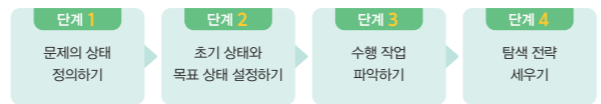
2 문제 해결을 위한 탐색 과정 설계

탐색을 통해 문제를 효과적으로 해결하기 위해서는 문제를 구조화하는 설계 과정이 필요하다. 이후, 문제를 해결하기 위한 탐색 알고리즘을 선택하고 실행하여 해결 가능성을 확인하고 답을 구한다.

인공지능으로 문제를 해결하기 위해 탐색 과정을 설계하려면, 먼저 문제를 명확히 정의해야 한다.

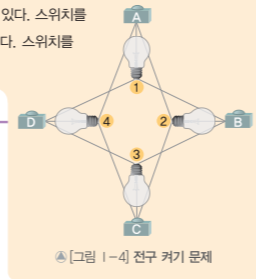
문제 해결을 위한 탐색 과정에서 만나게 되는 상황을 상태라고 하며, 모든 상태의 집합을 상태 공간이라고 한다. 그리고 문제가 주어진 상태, 즉 문제의 처음 상태를 초기 상태라고 하고, 문제가 모두 해결된 상태를 목표 상태라고 한다. 탐색은 '상태 공간 안에서 초기 상태에서부터 목표 상태까지의 경로를 찾는 과정'이라고 정의할 수 있다.

문제 해결을 위한 탐색 과정 설계하기



예제 전구 켜기 문제를 통해 탐색으로 문제를 해결하는 과정을 살펴보자.

전구 4개(①~④)와 스위치 4개(A~D)가 연결되어 있다. 스위치를 누르면, 그 스위치에 연결된 전구 3개의 상태가 바뀐다. 스위치를 눌러서 모든 전구의 불을 켜 보자.



풀이

단계 1 문제의 상태 정의하기

상태 정의는 문제 해결을 위한 첫 단계로서, 문제를 정확히 파악하는 것을 뜻한다. 만약 문제의 핵심을 잘못 파악한다면, 이후의 해결 방법도 신뢰하기 어려울 것이다. 전구 켜기 문제에서는 다음과 같이 상태를 정의할 수 있다.

• 상태: 1번부터 4번까지 전구가 이끄는 위치, 전구와 스위치가 연결된 상태, 전구가 켜진 상태, 전구와 스위치의 개수, 스위치의 눌림 상태

단계 2 초기 상태와 목표 상태 설정하기

그림에서 색이 없는 전구는 전구가 꺼져 있는 상태를, 노란색 전구는 전구가 켜져 있는 상태를 뜻한다. 초기 상태와 목표 상태를 설정해 보자.

7 탐색 과정 설계

탐색을 통해 문제를 효과적으로 해결하기 위해 문제를 구조화하는 설계 과정은 문제를 이해하고 분석하는 데 중요한 역할을 한다. 문제를 구조화하여 명확하게 정의하면, 해결책을 찾기 위한 탐색 알고리즘을 선택하는 데 도움을 준다.

다음 [추가 예제]를 통해 학생들이 탐색으로 문제를 해결하는 과정을 잘 이해할 수 있도록 지도한다.

추가 예제 3리터와 5리터 물통으로 4리터 만들기

➔ 수돗가 옆에 3리터와 5리터의 두 개의 물통이 비어 있다. 두 개의 물통을 이용하여 5리터 물통에 4리터만 담으려고 한다. 어떻게 하면 될까?

[문제 상황] 정확히 4리터의 물을 뿌려야 열리는 비밀의 문이 있다. 물은 한 번만 부을 수 있다. 비밀의 문 옆에는 3리터 물통 1개, 5리터 물통 1개가 있고 옆에는 강이 흐르고 있다. 물통 두 개를 이용하여 5리터 물통에 정확히 4리터의 물을 담아 보자.

풀이 과정

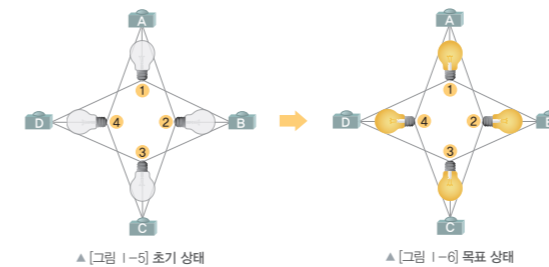
1. 문제의 상태 정의하기

(1) 주어진 문제의 상태를 정의해 보자.

예시 답안 물통의 개수, 각 물통의 크기

해설 문제에서 상태란, 문제가 처한 상황을 의미한다. 이 문제에서는 두 개의 물통이 각 3리터, 5리터의 크기라는 것을 상태로 정의할 수 있다.

• 초기 상태: 1번부터 4번까지 전구가 모두 꺼져 있는 상태
• 목표 상태: 1번부터 4번까지 전구가 모두 켜져 있는 상태



위 그림은 다음과 같이 이진수로 추상화하여 표로 나타낼 수 있다. 0은 전구의 불이 꺼진 상태, 1은 전구의 불이 켜진 상태를 의미한다.

전구	1	2	3	4
상태	0	0	0	0

▶ [표 1-2] 초기 상태

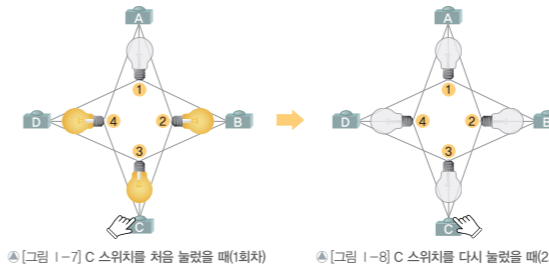
전구	1	2	3	4
상태	1	1	1	1

▶ [표 1-3] 목표 상태

단계 3 수행 작업 파악하기

수행 작업이란 현재 상태에서부터 다음 상태로 진행하기 위한 작업을 뜻한다. 수행 작업을 정의할 때는 문제의 핵심 요소를 고려해야 한다. 전구 켜기 문제에서 고려해야 할 핵심 요소는 무엇일까?

전구를 켜기 위해서는 스위치를 누르는 작업을 해야 한다. 스위치를 누를 때 상태 변화를 파악해 보자. 먼저 스위치를 누르는 횟수를 생각해 보자. 예를 들어, 초기 상태에서 C 스위치를 한 번 누르면 2, 3, 4번 전구의 불이 켜지지만, 두 번 누르면 2, 3, 4번 전구의 불이 꺼져 초기 상태와 같아진다. 즉 스위치를 짝수 번 누르면 처음과 같은 상태가 된다.



*핵심 요소

문제를 해결하기 위해 꼭 필요한 고려 사항으로, 상태 변화에 영향을 주는 데이터나 조건을 의미한다.

수행 작업 파악하기

1 **수행 작업**: 연산자(operator)라고도 하며 문제에서 상태 변화를 일으키는 작업을 말한다.

예 미로에서 한 칸 앞으로 가기, 왼쪽으로 돌기 등

2 **핵심 요소(제약 조건)**: 상태 변화에 영향을 주는 조건이나 데이터를 살펴보면 작업이 어떠한 조건에서 적용될 수 있는지, 적용 결과는 어떠한지 확인한다.

예 앞에 벽이 있다면 오른쪽으로 돌기

2. 초기 상태와 목표 상태를 설정하기

(1) 문제의 초기 상태와 목표 상태를 설정해 보자.

예시 답안

• 초기 상태: 3리터 물통, 5리터 물통이 비어 있는 상태
• 목표 상태: 5리터 물통에 4리터가 담겨 있는 상태

해설 문제 해결의 초기 상태는 문제 해결이 시작되는 상태이므로, 3리터 물통과 5리터 물통이 비어 있는 상태가 초기 상태이다. 목표 상태는 5리터 물통에 정확히 4리터의 물이 담겨 있는 상태이다.

(2) (x, y) 순서쌍을 이용해 간단히 표현해 보자. (앞에서부터 3리터, 5리터 물통 속의 물의 양을 의미한다.)

예시 답안 • 초기 상태: (0, 0) • 목표 상태: (x, 4)

해설 3리터 물통과 5리터 물통 각각의 물의 양을 숫자의 순서쌍으로

표현하면 간단하게 정보를 구조화할 수 있다. 목표 상태에서 3리터 물통의 값은 상관이 없으므로 x로 표현한다.

3. 수행 작업 파악하기

(1) 물통 문제 해결을 위한 첫 번째 수행 작업과 다음 상태를 파악해 보자.

예시 답안

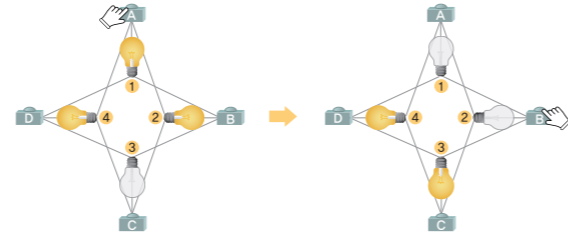
현재 상태	수행 작업	다음 상태
(0, 0)	3리터 물통에 물 담기	(3, 0)
	5리터 물통에 물 담기	(0, 5)

해설 수행 작업을 파악할 때는 문제의 핵심 요소를 고려해야 한다. 이 문제에서는 3리터 물통과 5리터 물통만을 사용하는 것이 핵심 요소이기 때문에, 3리터 또는 5리터 물통에 물을 담는 것이 첫 번째 수행 작업이 될 수 있다.

지도 방법

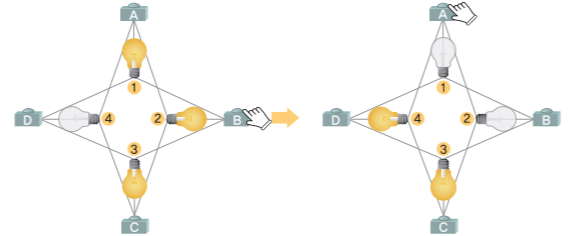
- 수행 작업이 현재 상태에서 다음 상태로 변화를 일으키고, 결국 목표 상태까지 변화하도록 한다는 것을 깨달을 수 있도록 지도한다.
- 첫 번째 수행 작업의 결과를 파악하는 것은 앞으로 문제를 해결하는 방향을 잡는 데 핵심이 되므로, 시간을 들여 꼼꼼히 살펴보도록 지도한다.

이번에는 서로 다른 2개의 스위치를 연달아 누를 경우, 스위치를 누르는 순서가 결과에 영향을 미치지 않아보자. 먼저 A 스위치를 누르면, 1, 2, 4번 전구가 켜질 것이다. 다음으로 B 스위치를 누르면, 1, 2번 전구는 꺼지고, 3번 전구가 켜져 3, 4번 전구가 켜져 있는 상태일 것이다.



▲ [그림 1-9] A 스위치를 먼저 눌렀을 때 ▲ [그림 1-10] 그 이후에 B 스위치를 눌렀을 때

다시 초기 상태로 돌아가서 B 스위치를 먼저 눌러 보자. B 스위치를 누르면, 1, 2, 3번 전구에 불이 켜질 것이다. 다음으로 A 스위치를 누르면, 1, 2번 전구가 꺼지고, 4번 전구가 켜져 3, 4번 전구가 켜져 있는 상태일 것이다. 둘의 결과가 같음을 알 수 있다.



▲ [그림 1-11] B 스위치를 먼저 눌렀을 때 ▲ [그림 1-12] 그 이후에 A 스위치를 눌렀을 때

위의 과정을 통해 스위치를 누르는 순서는 결과에 영향을 미치지 않는다는 것을 알 수 있다. 이러한 조건들을 고려하여 다음과 같이 첫 번째 수행 작업과 결과를 파악할 수 있다.

현재 상태	수행 작업	다음 상태
켜진 전구: 없음 꺼진 전구: 1, 2, 3, 4	A 스위치를 누름	• 켜진 전구: 1, 2, 4 • 꺼진 전구: 3
	B 스위치를 누름	• 켜진 전구: 1, 2, 3 • 꺼진 전구: 4
	C 스위치를 누름	• 켜진 전구: 2, 3, 4 • 꺼진 전구: 1
	D 스위치를 누름	• 켜진 전구: 1, 3, 4 • 꺼진 전구: 2

(2) 가능한 모든 수행 작업을 파악해 보자.

예시 답안

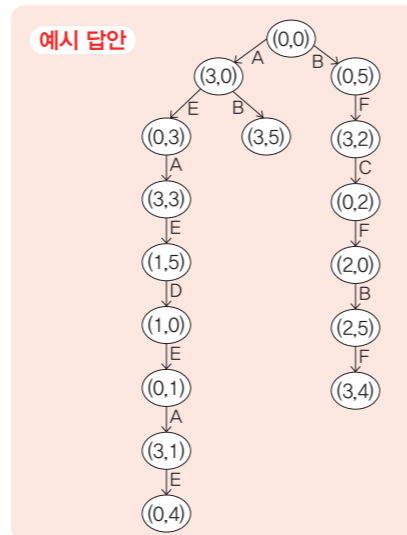
- A: 3리터 물통에 물 담기
- B: 5리터 물통에 물 담기
- C: 3리터 물통 비우기
- D: 5리터 물통 비우기
- E: 3리터 물통의 물 5리터 물통으로 옮기기
- F: 5리터 물통의 물 3리터 물통으로 옮기기

해설 가능한 모든 수행 작업을 나열하면 문제 해결의 여러 경로를 탐색할 수 있다. 각 작업은 물통에 물을 채우거나 비우는 기본 작업과 한 물통에서 다른 물통으로 물을 옮기는 작업으로 구성된다. 이 작업들을 통해 다양한 상태 변화를 만들어 문제를 해결할 수 있다.

4. 탐색 전략 세우기

(1) 물통 문제의 상태 공간 트리를 그려 보자.

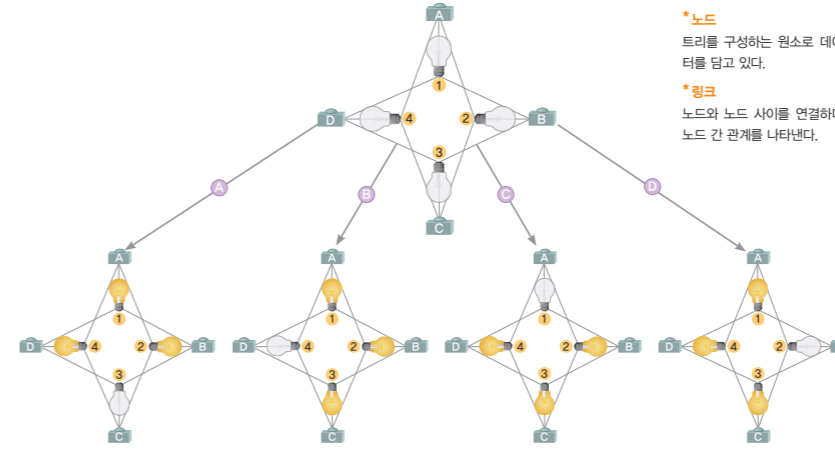
예시 답안



단계 탐색 전략 세우기

탐색 과정 설계의 마지막 단계는 탐색으로 문제를 효과적으로 해결하기 위한 전략을 세우는 것이다. 탐색 전략을 세우지 않는다면, 이미 탐색했던 것을 반복하거나 탐색 과정의 일부를 빠뜨리는 오류를 범할 수 있다. 이를 위해 먼저 문제의 상태 공간을 살펴볼 필요가 있다. 문제의 상태 공간을 구조화하여 표현하는 것은 탐색으로 문제를 해결하기 위해 사용되는 좋은 전략 중 하나이다.

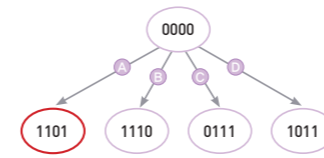
다음 그림은 초기 상태에서 첫 번째 스위치를 눌렀을 때의 상태를 트리구조*로 나타낸 것이다. 이와 같이 탐색 구조를 트리 형태로 나타낸 것을 상태 공간 탐색 트리라고 한다. 이때 노드(node)*는 상태를, 링크(link)*는 작업을 의미한다.



- *트리구조
노드와 링크로 이루어져 있으며 계층적인 정보를 표현할 때 사용한다.
- *노드
트리를 구성하는 원소로 데이터를 담고 있다.
- *링크
노드와 노드 사이를 연결하며 노드 간 관계를 나타낸다.

▲ [그림 1-13] 초기 상태에서 첫 번째 스위치를 눌렀을 때의 상태 변화

위의 트리 그림은 다음과 같이 좀 더 간략하게 표현할 수 있다. 0은 전구의 불이 꺼진 상태를, 1은 전구의 불이 켜진 상태를 의미하며, 4비트 숫자는 순서대로 1, 2, 3, 4번 전구를 의미한다.



▲ [그림 1-14] 더 구조화된 상태 공간 트리

1101은 A 스위치를 눌렀을 때 1, 2, 4번 전구가 켜짐을 뜻해요.

해설 각 상태에서 가능한 상태만 확장하여 상태 공간 트리를 그리고, 이전에 있던 상태와 같은 상태는 추가하지 않는다. 노드에는 상태를 그리고, 링크에는 작업을 추가하여 한 눈에 알아볼 수 있도록 구조화한다.

(2) 상태 공간 트리를 보고 목표 상태에 가장 빨리 도달하는 방법을 적어 보자.

예시 답안

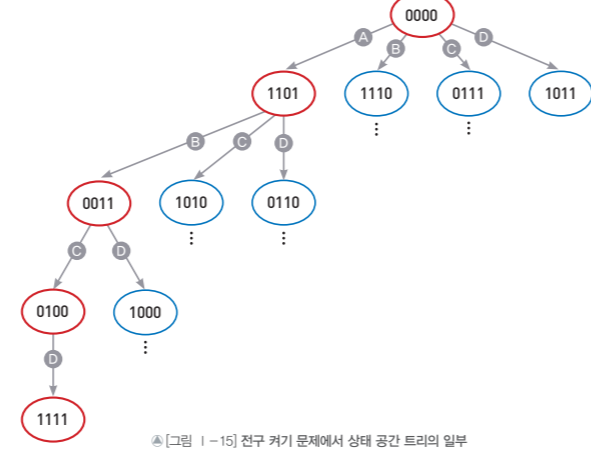
- 1 5리터 물통을 채운다. (0, 0) → (0, 5)
- 2 5리터 물통에서 3리터 물통으로 옮긴다. (0, 5) → (3, 2)
- 3 3리터 물통을 비운다. (3, 2) → (0, 2)
- 4 5리터 물통의 물을 3리터 물통에 옮긴다. (0, 2) → (2, 0)
- 5 5리터 물통을 채운다. (2, 0) → (2, 5)
- 6 5리터 물통에서 3리터 물통으로 옮긴다. (2, 5) → (3, 4) (목표 상태 도달)

해설 상태 공간 트리를 통해 문제를 분석하고, 목표 상태에 도달하기 위한 최단 경로를 찾을 수 있다. 이를 통해 탐색 과정에서 불필요한 상태를 피하고, 최적의 경로를 따라 목표 상태에 도달할 수 있다. 깊이 우선 탐색과 너비 우선 탐색의 결과가 다르므로, 탐색 순서에 따른 효율을 비교할 수 있다.

지도 방법

- 지금은 탐색 트리가 단순하여 답을 찾는 것이 어렵지 않지만, 탐색 트리가 복잡할수록 탐색 전략의 중요성이 커진다. 탐색 순서에 따라 탐색의 효율이 달라질 수 있음을 학생들이 깨달을 수 있도록 지도한다.
- A → B → C → D 순서대로 누르는 사실 깊이 우선 탐색을 적용한 결과이다. 깊이 우선 탐색은 다음 단원의 내용이기여 여기에서 자세히 설명하지는 않고, 이러한 방식을 깊이 우선 탐색이라고 부른다고 알려 주어, 학생들이 다음 단원에 호기심을 가질 수 있도록 지도한다.

다음은 문제 상황을 나타낸 상태 공간 트리의 일부이다. 스위치를 누르는 순서, 켜짐에 대한 조건을 고려하여 상태 공간 트리를 만든다.



☞ [그림 1-15] 친구 켜기 문제에서 상태 공간 트리의 일부

상태 공간 구조화를 통해 단순한 탐색 전략을 세워 보았다.

탐색 전략 A → B → C → D 순서대로 스위치 누르기

이 문제에서는 스위치를 누르는 순서와 상관없이 모두 한 번씩 눌렀을 때, 문제를 해결할 수 있었다.

탐색 전략은 문제 상황에 따라 다르게 정할 수 있다. 탐색할 상태 공간을 구조화하여 탐색의 방향과 범위를 정하고, 어디에서부터 탐색할지 탐색의 순서를 정해야 할 때 있다. 또한 문제에 알맞은 탐색 알고리즘을 선택하는 것도 탐색 전략 중의 하나다. 탐색 전략에 따라 탐색 효율이 달라지기도 하므로 탐색 전략을 잘 세우는 것은 중요하다.

소단원 1분 요약

- 1 탐색은 문제 상황에서 답을 찾는 과정을 의미하고, 인공지능은 탐색을 통해 문제를 해결하며 발전했다.
- 2 탐색을 통해 문제를 효과적으로 해결하기 위해서는 문제를 구조화하는 '탐색 과정 설계'가 필요하다.
- 3 문제를 구조화하여 효율적으로 해결하기 위한 탐색 과정에는 ① 문제의 상태 정의하기, ② 초기 상태와 목표 상태 설정하기, ③ 수행 작업 파악하기, ④ 탐색 전략 세우기가 있다.

8 탐색 과정 설계

1 전방향 탐색(forward search)

시작 지점에서 출발하여 목표 지점으로 나아가는 방식이다. 탐색 공간이 작을 때, 즉 가능한 경로가 많지 않을 때 효율적이다. 시작 지점에서 목표 지점까지의 경로를 직접 찾는다.

예 알파벳 순서대로 단어 찾기: 사전에서 특정 단어를 찾기 위해 A부터 차례대로 단어를 확인하여 찾고자 하는 단어에 도달할 때까지 진행한다.

2 후방향 탐색(backward search)

목표 지점에서 출발하여 시작 지점을 향해 나아가는 방식이다. 목표 지점에서 출발하여 가능한 경로를 거꾸로 추

적하는 것이 더 효율적일 때 사용한다. 특히, 목표 상태가 명확하고 시작 상태가 불명확할 때 유리하다.

예 경찰의 범인 찾기: 경찰이 범죄 현장에서 출발하여 단서를 따라 사건의 시작점(범인의 위치)을 추적하는 것이다.

3 양방향 탐색(bidirectional search)

시작 지점과 목표 지점에서 동시에 출발하여 중간에서 만나는 방식이다. 탐색 공간이 큰 경우, 탐색 범위를 줄이기 위해 양방향에서 동시에 탐색하여 중간에서 만나는 경로를 찾는다.

예 두 친구가 만나는 장소 찾기: 한 친구는 집에서 출발하고, 다른 친구는 학교에서 출발하여 서로를 찾기 위해 중간에서 만난다.

탐구 활동

친구 켜기 연습 문제

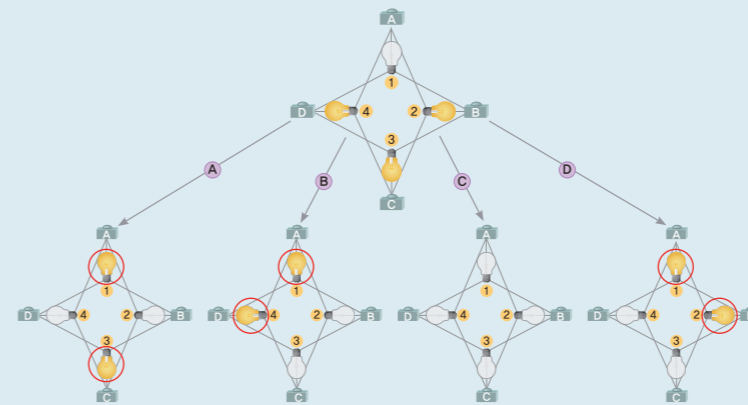
☑ 24쪽에 제시된 친구 켜기 문제에서 1번 친구만 불이 꺼져 있고 나머지 친구들은 불이 켜져 있을 때, 모든 친구의 불을 켜기 위해서는 어떻게 해야 할지 탐색해 보자.

1 초기 상태와 목표 상태를 설정해 보자.

• 초기 상태: • 목표 상태:

2 초기 상태의 다음 상태를 나타낸 표를 채워 보고, 켜진 친구에 색칠해 보자.

현재 상태	수행 작업	다음 상태
• 켜진 친구: 2, 3, 4번 • 꺼진 친구: 1번	A 스위치를 누름	• 켜진 친구: • 꺼진 친구:
	B 스위치를 누름	• 켜진 친구: • 꺼진 친구:
	C 스위치를 누름	• 켜진 친구: • 꺼진 친구:
	D 스위치를 누름	• 켜진 친구: • 꺼진 친구:



3 상태 공간 트리를 완성하고 탐색 경로를 선택해 보자.

4 탐색 결과를 친구들과 비교하며 이야기해 보자.

탐구 활동

지도 방법

- 친구 켜기 연습 문제 탐구 활동은 교과서 24쪽의 친구 켜기 문제를 약간 변형한 것으로 학생들이 스스로 문제를 이해하고, 탐색 과정을 설계해 보도록 구성되었다. 24쪽부터의 본문을 참고하여 학생들이 스스로 문제 해결을 위한 탐색 과정을 설계해 볼 수 있도록 지도한다.
- 다음 상태와 수행 작업의 관계를 파악하도록 안내한다.

- 1 초기 상태 목표 상태를 직접 적어 보며 문제 분석 능력을 기르도록 지도한다.
- 2 스위치를 누르면 친구의 상태가 반전되는 점을 유의하며 답을 채우도록 지도한다.
- 3 문제의 핵심 요소를 고려하여 상태 공간 트리를 완성해 보도록 지도한다.
- 4 상태 공간 트리를 그릴 때 핵심 요소를 잘 파악하고 그렸다면 상태 공간 트리 모양은 같을 것이다. 친구와 비교하며 놓친 부분이 있는지 확인하도록 지도한다. 트리에 수정할 필요가 있다면 고치도록 지도한다.

탐구 활동 해설 및 예시 답안

• 교과서 29쪽

1 초기 상태와 목표 상태를 설정해 보자.

예시 답안

- 초기 상태: 2, 3, 4번이 켜 있는 상태
- 목표 상태: 1, 2, 3, 4번 모두 켜 있는 상태

해설 초기 상태는 현재 문제를 해결하기 시작하는 시점의 상태로, 1번 친구는 꺼져 있고 2, 3, 4번만 불이 켜져 있는 상태를 의미한다. 목표 상태는 문제 해결이 완료된 상태로, 1, 2, 3, 4번 불이 모두 켜져 있는 상태이다. 초기 상태와 목표 상태를 명확히 설정함으로써 문제 해결의 방향성과 목표를 확실히 할 수 있다.

2 초기 상태의 다음 상태를 나타낸 표를 채워 보고, 켜진 친구에 색칠해 보자.

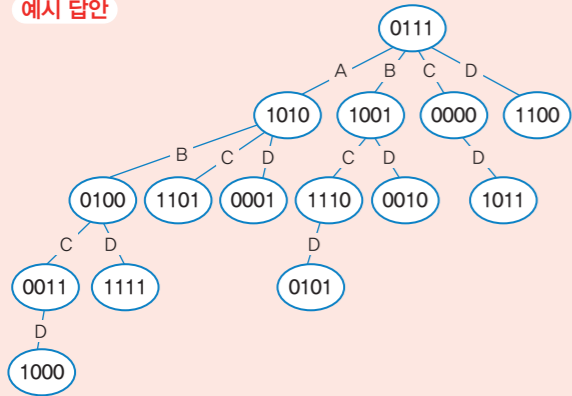
예시 답안 본문에 불이 켜진 부분 색칠 참고

현재 상태	수행 작업	다음 상태
• 켜진 친구: 2, 3, 4번	A 스위치 누름	• 켜진 친구: 1, 3 • 꺼진 친구: 2, 4
	B 스위치 누름	• 켜진 친구: 1, 4 • 꺼진 친구: 2, 3
• 꺼진 친구: 1번	C 스위치 누름	• 켜진 친구: X • 꺼진 친구: 1, 2, 3, 4
	D 스위치 누름	• 켜진 친구: 1, 2 • 꺼진 친구: 3, 4

해설 스위치를 누르면 스위치와 연결된 친구의 상태가 반전되는 것으로 상태 변화가 일어난다. A 스위치로 예를 들면, A 스위치에는 1, 2, 4번 친구가 연결되어 있기에 A 스위치를 눌렀을 때 1, 2, 4번 친구의 상태가 반전되어 꺼져 있던 1번은 켜지고 켜져 있던 2, 4번은 꺼지게 된다. 3번 스위치는 영향을 받지 않으므로 계속 켜져 있는 상태로 남아 있다. 이런 식으로 B, C, D 스위치를 눌렀을 때의 경우도 구하도록 한다.

3 상태 공간 트리를 완성하고 탐색 경로를 선택해 보자.

예시 답안



해설 상태 공간 트리를 그릴 때, 이전에 추가된 노드를 추가하지 않고 그린다. 링크에는 수행 작업을 추가하면 구조화된 상태 공간 트리를 그릴 수 있다.

4 탐색 결과를 친구들과 비교하며 이야기해 보자.

예시 답안

- 나는 중복이 되는 경우는 노드에 추가하지 않았어.
- A, B, D 순서대로 스위치를 누르면 문제가 해결돼. B, D, A나 A, D, B 등 순서를 바꿔도 결과는 같더라.

해설 상태 공간 트리를 그리는 이유는 문제의 해답을 효율적으로 찾기 위함이다. 모든 경우의 수를 트리에 그린다면 스위치를 누르는 순서가 바뀌어도 결과가 같다는 것을 알 수 있지만, 효율성이 떨어질 수도 있기에 중복되지 않도록 그린다.



memo

A large blue memo area with horizontal dotted lines for writing.

03

맹목적 탐색과 정보 이용 탐색

🕒 수업 시간: 3시간 📖 교과서 쪽수: 30~43쪽

학습 목표	<ul style="list-style-type: none"> • 맹목적 탐색과 정보 이용 탐색의 차이를 설명할 수 있다. • 지능적 탐색의 원리를 설명할 수 있다.
수업 방법	강의, 토론, 발표
준비물	교사 교과서, 관련 교수 학습 자료 학생 필기도구

단계	교수 · 학습 방법	지도상의 유의점
도입	<p>생각 열기</p> 보물찾기를 했던 경험, 물건을 찾았던 경험에 대해 친구들과 이야기를 나누어 보도록 한다.	<ul style="list-style-type: none"> • 경험을 나누며 찾는 방법이나 순서에 따라 탐색의 효율이 달라질 수 있음을 느끼도록 지도한다.
전개	<p>1 맹목적 탐색</p> <ul style="list-style-type: none"> • 정보가 없는 상황에서 사용하는 대표적인 알고리즘인 깊이 우선 탐색과 너비 우선 탐색의 공통점과 차이점을 8 퍼즐 예제를 활용하여 비교한다. 	<ul style="list-style-type: none"> • 탐색 트리와 관련한 다양한 새롭고 낯선 용어에 대해 충분히 이해할 수 있도록 지도한다.
	<p>2 정보 이용 탐색</p> <ul style="list-style-type: none"> • 맹목적 탐색과 차이를 통해 인공지능의 지능적 탐색 원리를 깨닫도록 지도한다. 	<ul style="list-style-type: none"> • 항상 정보 이용 탐색이 유리한 것은 아니므로 문제의 형태나 상황에 따라 유리한 탐색을 선택할 수 있도록 지도한다.
정리	<p>탐구 활동</p> <ul style="list-style-type: none"> • 맹목적 탐색의 예제로 사용된 8 퍼즐을 정보 이용 탐색으로 해결한다. 	<ul style="list-style-type: none"> • 탐색 결과를 친구들과 나누며 다양한 탐색 전략에 대해 이야기를 나누도록 한다.
평가	<ul style="list-style-type: none"> • 맹목적 탐색의 원리를 이해하고 설명할 수 있는가? • 정보 이용 탐색의 원리를 이해하고 설명할 수 있는가? 	<ul style="list-style-type: none"> • 자기 평가 문항을 제시하고 학습자의 이해도에 따른 피드백을 하여 평가 기준을 달성하도록 돕는다.

03 맹목적 탐색과 정보 이용 탐색

• 제시 의도 •

'보물찾기'라는 실생활의 경험을 바탕으로 탐색 알고리즘의 개념과 특징을 확인하고 차이점에 대해 예상하여 쉽게 이해할 수 있다.

지도 방법

- [생각 열기]의 삽화를 보며 누가 가장 보물을 빨리 찾았을 것으로 예상되는지 이야기를 나누어 보도록 지도한다.
- 보물찾기에 처음 참여하는 사람과 여러 번 참여했던 사람 중에 누가 보물을 잘 찾을 수 있을지와 그 이유에 관해 이야기하며, 맹목적 탐색과 정보 이용 탐색의 차이점에 대해 예상해 볼 수 있도록 지도한다.

예시 답안

- 보물을 숨길 수 있는 장소와 보물을 숨길 수 없는 장소를 구분하여 보물을 숨길 수 있는 장소부터 찾는다.
- 예전의 경험에서 보물을 찾았던 장소부터 방문하여 찾아본다.

30

03 맹목적 탐색과 정보 이용 탐색

- 학습 목표**
- 맹목적 탐색과 정보 이용 탐색의 차이를 설명할 수 있다.
 - 지능적 탐색의 원리를 설명할 수 있다.

학습 요소 맹목적 탐색, 정보 이용 탐색, 지능적 탐색의 원리

생각 열기 보물찾기 전략

체험학습에서 보물찾기 게임을 했다. 친구들은 각자 전략을 세워 보물을 찾기로 했다.

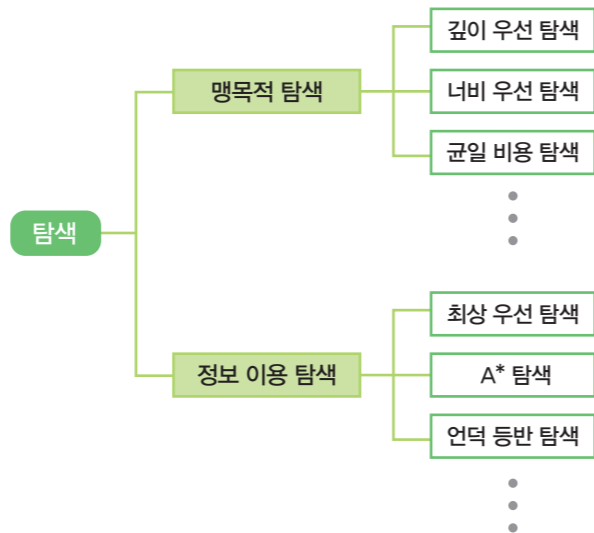


어떻게 하면 보물을 빨리 찾을 수 있을까?

1 탐색의 분류

탐색의 종류를 나눌 때, 정보를 이용하지 않고 단순 로직에 의해 탐색하는가 아니면 정보를 이용하여 탐색하는가에 따라 맹목적 탐색과 정보 이용 탐색으로 나눈다.

맹목적 탐색은 탐색 방향이나 탐색의 방식 등에 따라 분류하고, 정보 이용 탐색은 탐색에 어떤 정보(휴리스틱)를 어떻게 사용할 것인가에 따라 분류한다.



1 맹목적 탐색

맹목적 탐색(blind search)이란 프로그래밍이나 기계학습으로 문제를 해결할 때 목표 상태 이외에 어떤 정보도 주어지지 않은 상태에서 탐색을 하는 것을 말한다. 예를 들어 서랍장에서 물건을 찾을 때 물건의 위치에 대한 사전 정보가 없다면 모든 서랍을 열어 봐야 할 것이다. 운이 좋으면 첫 번째 연 서랍에서 물건을 찾을 수 있겠지만, 이와 반대로 마지막에 열어 본 서랍에서 물건을 찾을 수도 있다. 이처럼 맹목적 탐색은 목표 상태에 도달할 때까지 모든 상태 공간을 탐색하는 알고리즘이다. 현재 상태에서 다음 상태 노드를 방문할 때, 어떤 방향으로 탐색하는가에 따라 깊이 우선 탐색과 너비 우선 탐색 등이 있다.

맹목적 탐색은 아무 사전 정보도 없기 때문에 무정보 탐색(uninformed search)이라고도 한다.

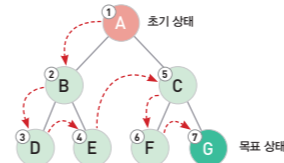
노드의 종류

- 루트 노드: 최상위의 노드이다.
- 부모 노드: 직접 연결된 상위 노드이다.
- 자식 노드: 직접 연결된 하위 노드이다.
- 형제 노드: 나는 왼쪽에서부터 같은 부모 노드를 가진 노드이다.

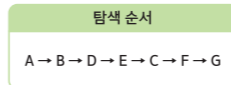
자식 노드에서 답을 발견하지 못했을 때 부모 노드로 되돌아가는 것을 백트래킹(back tracking)이라고 한다.

01 깊이 우선 탐색

깊이 우선 탐색(DFS: Depth First Search)은 상태 공간 트리에서 자식 노드(node)를 먼저 방문하는 수직 방향의 탐색 알고리즘이다. 자식 노드가 두 개 이상일 경우 주로 왼쪽 노드를 먼저 방문한다. 탐색 중에 단말 노드에 도착했는데도 목표 노드를 찾지 못했다면 다시 분기점으로 되돌아가 방문하지 않은 노드부터 다시 수직 방향으로 탐색한다. 만약 목표 노드가 수직 방향으로 깊이 있다면 문제를 빨리 해결할 수 있고, 필요한 저장 공간도 비교적 적게 드는 장점이 있다. 하지만 목표 노드가 없는 경로에 깊이 빠져 버리거나 목표 노드에 이르는 경로가 여러 개일 경우, 지금의 방문 경로가 최단 경로임을 보장할 수 없다는 단점이 있다.



※ [그림 1-16] 깊이 우선 탐색



해 보기 1 트리구조 파악하기

적용하기

• 본문에 제시된 '깊이 우선 탐색' 트리 그림을 보고 다음 물음에 답해 보자.

- 1 B 노드의 자식 노드를 찾아서 적어 보자. D, E
- 2 C 노드의 부모 노드를 찾아서 적어 보자. A
- 3 D 노드의 형제 노드를 찾아서 적어 보자. E

31

지도 방법

맹목적 탐색을 사용하는 이유에 대해 학생과 이야기하며, 맹목적 탐색의 특징에 대해 이해할 수 있도록 지도한다.

지도 예

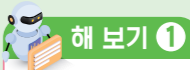
선생님: 맹목적 탐색을 왜 사용할까요? 자유롭게 이야기해 봅시다.

학생: 정보가 전혀 없으면 어쩔 수 없이 맹목적 탐색을 해야 할 것 같아요.

학생: 몇 개 없는 서랍에서 물건을 찾는 것처럼, 문제가 아주 간단하다면 맹목적 탐색도 괜찮을 것 같아요.

• 깊이 우선 탐색의 장점 및 특징을 알 수 있도록 [그림 1-16]에서 목표 상태를 C, D로 바꿔서 탐색 순서를 확인하도록 지도한다.

- 목표 상태가 C일 때 탐색 순서
A → B → D → E → C
- 목표 상태가 D일 때 탐색 순서
A → B → D



해 보기 1 지도 방법

교과서 31쪽 날개에 있는 노드의 종류를 설명한다. 이해가 쉽도록 가계도 등의 트리 구조를 추가 자료로 제시하는 등 개념을 명확히 하도록 지도한다.

1 균일 비용 탐색

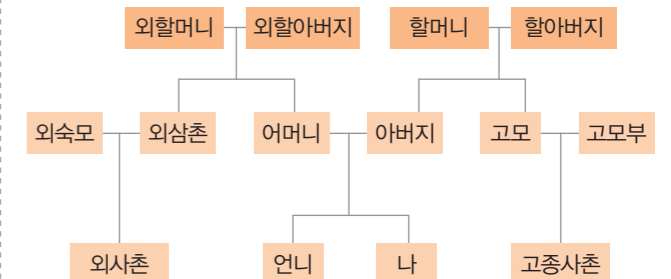
균일 비용 탐색은 너비 우선 탐색(BFS)의 확장으로, 그래프에서 누적 비용이 가장 낮은 경로를 우선 탐색하여 최단 경로를 찾는 알고리즘이다. BFS는 모든 간선의 비용이 같을 때 최단 경로를 찾지만, 균일 비용 탐색은 가중치가 다른 경우에도 정확한 최단 경로를 보장한다. 우선순위 큐를 사용해 비용이 낮은 경로부터 탐색하므로, BFS의 가중치 버전이라고 볼 수 있다.

2 언덕 등반 탐색

언덕 등반 탐색은 현 상태에서 가장 좋은 선택을 반복적으로 선택하며 목표 상태로 나아가는 탐색 알고리즘이다. 현재 상태에서 가능한 다음 상태를 평가하여 값이 더 나은

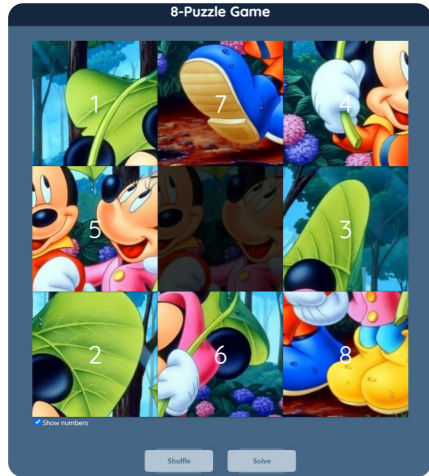
상태를 선택한다. 그러나 지역 최적해에 갇힐 위험이 있으며, 전체 최적해를 반드시 보장하지는 않는다.

2 트리 구조의 예



지도 방법

8 퍼즐을 직접 해 보며 흥미를 유발하고, 배움을 적용해 볼 수 있다. 다음 사이트는 온라인에서 8 퍼즐을 체험할 수 있는 사이트이다.



주소

• <https://murhatsousli.github.io/8puzzle/#/>

예제 8 퍼즐을 깊이 우선 탐색으로 해결해 보자.

8 퍼즐은 1~8까지 쓰여 있는 8개의 숫자판과 한 개의 빈칸으로 구성된 슬라이딩 퍼즐이다. 움직이는 숫자판을 퍼즐의 빈칸으로 이동시키며 숫자를 정렬하여 목표 상태에 도달하는 게임이다. 8 퍼즐 문제 해결에 깊이 우선 탐색이 어떻게 적용되는지 살펴보자.

풀이

8 퍼즐 탐색 과정 설계

단계 1 문제의 상태 정의하기
상태는 퍼즐의 숫자판 위치다.

단계 2 초기 상태와 목표 상태 설정하기
초기 상태 숫자판과 목표 상태 숫자판을 설정한다.

2		3
1	8	4
7	6	5

1	2	3
8		4
7	6	5

초기 상태 목표 상태

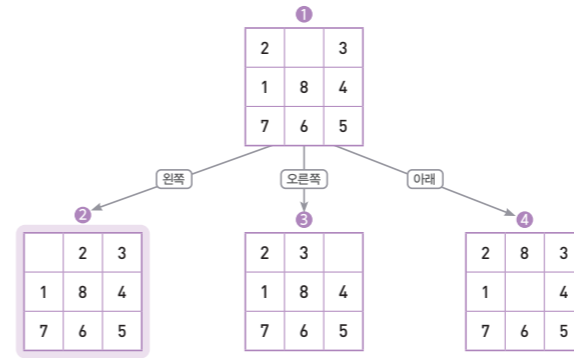
단계 3 수행 작업 파악하기
빈칸 주변 타일의 움직임을 파악하고 정의한다.

- 위: 빈칸의 위에 있는 숫자판을 빈칸으로 옮긴다.
- 왼쪽: 빈칸의 왼쪽에 있는 숫자판을 빈칸으로 옮긴다.
- 오른쪽: 빈칸의 오른쪽에 있는 숫자판을 빈칸으로 옮긴다.
- 아래: 빈칸의 아래에 있는 숫자판을 빈칸으로 옮긴다.

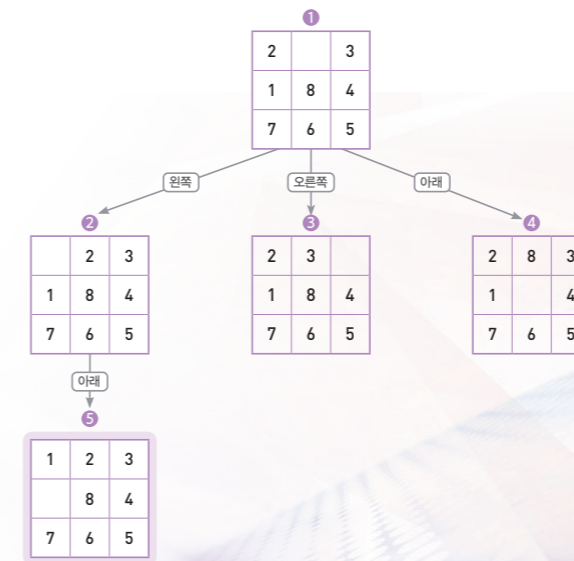
단계 4 탐색 전략 세우기
상태 트리를 다양한 방법으로 탐색한다.

8 퍼즐 깊이 우선 탐색은 현재 상태에서 가능한 작업을 수행하여 상태 공간에 자식 노드를 생성한다. 알고리즘에 따라 첫 번째 자식 노드, 즉 가장 왼쪽의 자식 노드를 방문하여 현재 상태를 자식 노드로 설정하고, 현재 상태와 목표 상태가 같은지 판단한다. 목표 상태가 아니라면 바뀐 현재 상태에서 가능한 작업을 수행하여 자식 노드를 생성하고, 첫 번째 자식 노드를 방문하여 목표 노드인지 판단하는 과정을 반복한다.

그림의 초기 상태를 살펴보면 빈칸의 위에는 숫자판이 없으므로 초기 상태에서 다음 상태를 만들기에는 왼쪽, 오른쪽, 아래의 작업만이 가능하다. 깊이 우선 탐색의 알고리즘에 따라 가장 왼쪽의 자식인 2번 노드를 선택하여 방문한다.



이제 현재 상태는 2번 노드이다. 목표 상태와 다르므로 2번 노드에서 가능한 작업을 수행하여 자식 노드를 생성한다. 빈칸의 위와 왼쪽에는 숫자판이 없고, 오른쪽 작업을 수행하면 결과값이 초기 상태와 같아지므로 수행하지 않는다. 가능한 작업인 아래를 수행하여 자식 노드를 생성하고 방문한다.



* 용어 해설

• **깊이 우선 탐색의 확장:** 탐색 트리에서 자식 노드를 확장(생성)하는 방법은 일반적으로 open 리스트와 close 리스트를 사용한다. open 리스트는 앞으로 방문할 노드의 목록을 의미한다. 깊이 우선 탐색에서는 open 리스트를 스택으로 사용한다. 가장 늦게 입력받은 것부터 출력하여 자식 노드부터 방문하게 한다. close 리스트는 이미 방문한 노드를 의미한다. 같은 노드를 방문하는 일을 막기 위해 사용하고, 파이썬에서는 set이라는 집합 구조를 사용하여 중복이 없도록 만든다.

3 탐색 알고리즘 더 알아보기

교과서에 제시된 8 퍼즐 문제를 다양한 알고리즘으로 탐색할 때, 각 탐색 알고리즘별로 어떤 자료 구조가 어떻게 사용되는지, 알고리즘 설계를 어떻게 하는지 알아보고, 탐색 알고리즘에 대한 이해를 높이도록 한다.

• 탐색 순서

교과서에 있는 대로 위, 왼쪽, 오른쪽, 아래의 순서대로 방문한다.

1-1. 깊이 우선 탐색하기

깊이 우선 탐색은 한 방향으로 깊이 파고 들어가는 탐색이다.

• 사용되는 자료 구조: 스택

스택(stack)은 출입구가 하나인 자료 구조로, 가장 나중에 입력받은 값부터 출력하는 특징이 있다. 트리의 방문으로 설명하자면, 가장 나중에 추가된 노드부터 방문한다. 그러므로 정해진 순서의 반대로 노드를 추가한다.

• 스택 저장 순서: 아래 → 오른쪽 → 왼쪽 → 위

open 리스트에 추가된 역순으로 방문한다. 방문한 노드는 close 리스트에 추가한다.

1 첫 번째 확장

초기 상태의 자식 노드를 확장한다. 초기 상태인 1번 노드를 close에 추가한다. 자식 노드를 open에 추가한다.

- open: 4, 3, 2
- close: 1



2 두 번째 확장

open의 top 위치에 있는 2번 노드를 꺼내 방문한다. 2번의 자식 노드를 open에 추가한다. 이때, 자식 노드가 close 노드와 같다면 추가하지 않는다. 2번 노드를 close 리스트에 삽입한다.

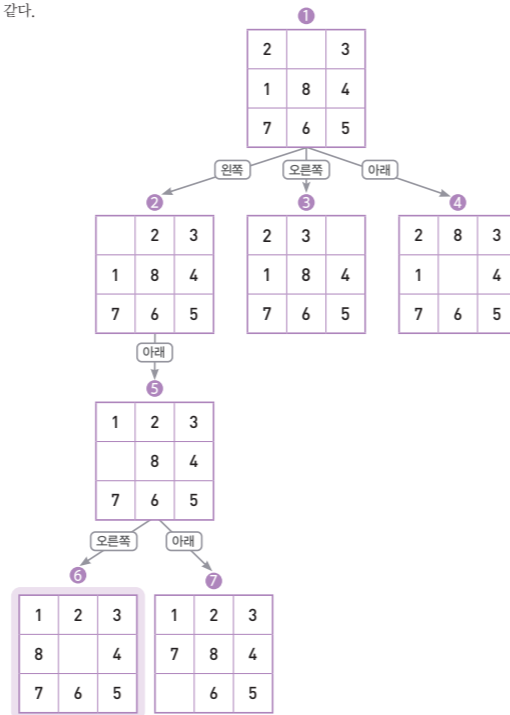
- open: 4, 3, 5
- close: 1, 2



*** 용어 해설**

• **너비 우선 탐색의 확장:** 너비 우선 탐색에서는 자식 노드보다 형제 노드를 먼저 방문해야 하므로, open 리스트를 큐로 사용한다. 가장 먼저 입력받은 것부터 출력하여 형제 노드부터 방문하게 한다. 너비 우선 탐색에서 close 리스트는 깊이 우선 탐색과 마찬가지로 이미 방문한 노드를 의미한다. 확장할 자식 노드를 close 리스트와 비교하여 close 리스트와 같은 노드는 open에 추가하지 않는다.

현재 상태 노드인 5번 노드가 목표 상태가 아니므로 가능한 작업을 수행하여 자식 노드를 생성한다. 이동이 가능한 위, 오른쪽, 아래 중에서 위 작업을 수행하면 결과값이 부모 노드와 같아지므로 수행하지 않고 오른쪽, 아래 작업만 수행하여 자식 노드를 생성한 결과는 다음 그림과 같다.

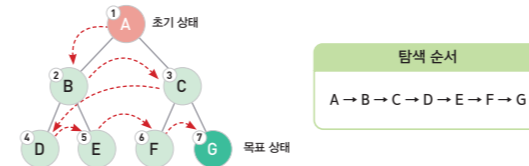


현재 상태인 6번 노드와 목표 상태가 같으므로 탐색을 종료한다.

02 너비 우선 탐색

너비 우선 탐색(BFS: Breadth First Search)은 상태 공간 트리에서 형제 노드를 먼저 방문하는 수평 방향의 탐색 알고리즘이다. 상태 공간 트리의 깊이가 얕고 넓은 경우, 목표 노드를 빨리 찾을 수 있다. 또한 목표 노드를 찾을 때까지 모든 상태 공간을 탐색하므로 목표 노드가 존재한다면 반드시 찾을 수 있고, 최단 경로를 알 수 있다는

장점이 있다. 하지만 노드 수가 많아지거나 목표 노드가 수직 방향으로 깊은 곳에 있다면 탐색 시간이 오래 걸리고, 탐색을 위한 공간과 탐색 범위도 지나치게 넓어지는 단점이 있다.



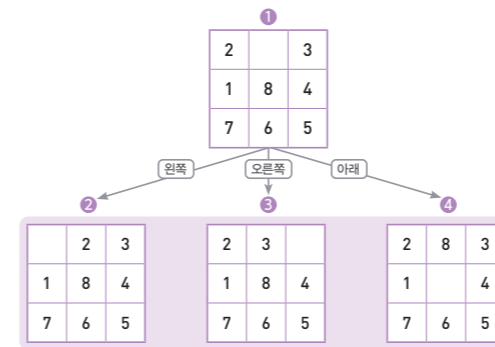
▲ [그림 1-17] 너비 우선 탐색

예제 8 퍼즐을 너비 우선 탐색으로 해결해 보자.

깊이 우선으로 탐색했던 8 퍼즐을 너비 우선으로 탐색하며 비교해 보자.

풀이

첫 번째 수행 결과는 깊이 우선 탐색의 결과와 같다. 초기 상태에서 수행 가능한 왼쪽, 오른쪽, 아래 작업을 수행하면 다음 그림과 같은 상태 공간 트리가 만들어진다.



깊이 우선 탐색이 수직 방향으로 진행되었던 것과 달리 너비 우선 탐색은 수평 방향으로 진행된다. 깊이 우선 탐색은 2번 노드를 방문하고 바로 5번 노드를 방문했지만, 너비 우선 탐색은 2번, 3번, 4번 노드를 모두 방문하며 목표 노드를 찾는다. 목표 노드가 없으면 2~4번 노드의 자식 노드를 생성하고 목표 노드를 찾는다.

지도 방법

너비 우선 탐색의 강점 및 특징을 알 수 있도록 [그림 1-16]에서 목표 상태를 C, D로 바꿔서 탐색 순서를 확인하고, 깊이 우선 탐색의 순서와 비교해 보도록 지도한다.

- 목표 상태가 C일 때 탐색 순서
너비 우선: A → B → C
깊이 우선: A → B → D → E → C
- 목표 상태가 D일 때 탐색 순서
너비 우선: A → B → C → D
깊이 우선: A → B → D

3 세 번째 확장

open의 top 위치에 있는 5번 노드를 꺼내 방문한다. 5번의 자식 노드를 open에 추가한다. 이때, 자식 노드가 close 노드와 같다면 추가하지 않는다.

5번 노드를 close 리스트에 삽입한다.

- open: 4, 3, 7, 6
- close: 1, 2, 5



1-2. 깊이 우선 탐색 결과

6번 노드를 open에서 꺼내 방문한 결과, 목표 상태와 같으므로 탐색을 종료한다.

- open: 4, 3, 7
- close: 1, 2, 5

1 → 2 → 5 → 6으로 탐색되었음을 알 수 있다. 목표하던 노드가 수직 방향으로 존재하여 빨리 찾을 수 있음을 알 수 있다.

2-1. 너비 우선 탐색하기

너비 우선 탐색은 목표 상태를 방문하기 전까지, 같은 레벨에 있는 상태를 모두 확장하여 탐색한다.

• 사용되는 자료 구조: 큐

큐(queue)는 출입구가 두 개인 자료 구조로, 가장 먼저 입력받은 값부터 출력하는 특징이 있다. 트리의 방문으로 설명하자면, 가장 먼저 추가된 노드부터 방문한다.

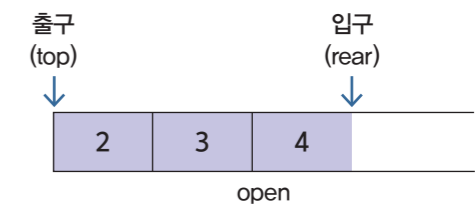
• 큐 저장 순서: 위 → 왼쪽 → 오른쪽 → 아래

open 리스트에 추가된 순으로 방문한다. 방문한 노드는 close 리스트에 추가한다.

1 첫 번째 확장

초기 상태의 자식 노드를 확장한다. 초기 상태 1번을 close에 추가하고, 자식 노드를 open에 추가한다.

- open: 2, 3, 4
- close: 1



*** 용어 해설**

• **평가 함수:** 평가 함수(evaluation function)는 탐색 알고리즘이 상태 공간 트리에서 최적의 경로를 찾기 위해 사용하는 기준이다. 다음 상태가 목표 상태에 얼마나 가까운지를 평가하여, 탐색의 효율성을 높이는 데 도움을 준다. 평가 함수에 따라 탐색의 범위와 속도가 달라지므로 평가 함수 설계는 중요하다.

일반적으로 평가 함수는 $f(n)$ 으로 표기한다. 휴리스틱 함수 $h(n)$ 은 현재 상태에서 목표 상태까지의 예상 비용의 추정치이다. 최상우선탐색에서의 평가 함수는 휴리스틱 함수 그 자체이므로 수식으로 표현하면 $f(n) = h(n)$ 이다.

A* 탐색은 평가 함수에 다른 평가값도 포함시키는데, 바로 비용 함수라 할 수 있는 $g(n)$ 이다. 초기 상태에서 현재 상태까지 소모된 총 비용이다. 즉 시간, 거리, 에너지 등을 의미한다. A* 탐색의 수식은 $f(n) = g(n) + h(n)$ 이다.

예제 '독도로 가는 길 찾기' 문제를 최상우선탐색으로 해결해 보자.

문제 상황 철수는 환경 프로젝트를 위해 독도를 직접 방문하려고 한다. 목적지까지 가장 빠르게 도착하기 위한 경로를 찾아보자.

조건

- 출발지: 서울
- 목적지: 독도
- 교통편: 육로는 차편, 바다는 배편을 이용한다.

풀이

철수는 내비게이션 탐색에 사용되는 휴리스틱 값을 한 지역에서 목적지까지의 직선거리로 정한다는 것을 알고, 지도의 각 지역에서 독도까지의 거리를 직접 재어 측정하고, 그 결과를 표로 정리했다.

$h(n)$ 은 각 노드의 휴리스틱 값을 의미한다.

지역명	$h(n)$	지역명	$h(n)$
광주	16	양양	5
인천	15	제천	5
서울	13	포항	4
춘천	11	강릉항	3
양평	10	후포항	3
대전	9	목호항	2
대구	8	울릉군	1
원주	7		

최상우선탐색은 평가 함수로 휴리스틱 값만을 사용하기 때문에 다음 그림과 같이 추상화 된 지도에 평가 함수값을 표시하여 다음 상태를 선택하기 쉽도록 구조화했다.

1 첫 번째 확장

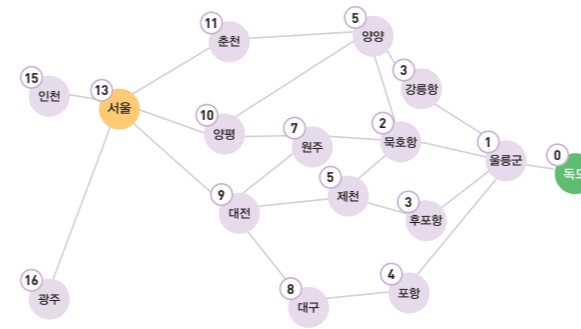
초기 상태의 확장 결과 2번, 3번, 4번 노드가 생성된다. 자식 노드를 큐에 저장하지만, 우선순위가 높은 자식 노드 즉, 목표 상태와 다른 칸이 적은 노드부터 저장한다. 2번 노드는 1, 8칸이 다르므로 2, 3번 노드는 1, 2, 3, 8칸이 다르므로 4, 4번 노드는 1, 2, 8칸이 다르므로 3의 우선순위가 생긴다.

- open: 2, 4, 3
- close: 1

2 두 번째 확장

우선순위가 가장 높은 2번 노드를 선택하여 방문한다. 자식 노드를 확장하여 우선순위대로 open에 추가한다. 2번의 상태에서는 아래로만 수행 가능하여 자식 노드는 5번만 생성된다. 5번은 8칸만 다른 자리에 있어 1의 우선순위를 가진다. 우선순위가 가장 높기에 우선순위 큐의 가장 앞쪽에 저장된다.

- open: 5, 4, 3
- close: 1, 2



이와 같이 정점과 정점을 연결하는 간선으로 구조화한 자료 구조를 그래프라고 한다. 위의 그래프는 지역을 정점으로, 지역 간의 연결을 간선으로 표현하여 복잡한 지도를 단순하게 추상화했다.

'독도로 가는 길 찾기' 탐색 과정 설계

단계 1 문제의 상태 정의하기
상태는 철수가 있는 위치, 지역 간 위치로 정의할 수 있다.

단계 2 초기 상태와 목표 상태 설정하기
초기 상태는 철수가 서울에 있는 상태이고, 목표 상태는 철수가 독도에 도착한 상태.

단계 3 수행 작업 파악하기
• 현재 상태 정점과 연결된 정점 중에서 현재 상태보다 평가 함수값이 작은 정점은 모두 방문 예정 정점으로 정한다.
• 방문 예정 정점 중에서 평가 함수값이 가장 작은 정점을 다음 상태로 정해 방문한다.

단계 4 탐색 전략 세우기
최상우선탐색 또는 A*로 탐색한다.

최상우선탐색에서 다음 상태를 선택할 때 현재 상태보다 평가 함수값이 크다는 것은 목표 상태에서 멀어졌다는 뜻이다. 최상우선탐색은 목표와 가까운 상태를 선택하는 알고리즘이기 때문에 방문이 가능한 다음 상태 중에서 평가 함수값이 가장 작은 노드를 다음 상태로 선택하여 방문한다.

출발지인 서울과 연결된 지역 중 서울보다 평가값이 큰 인천(15), 광주(16)는 탐색 범위에서 제외한다. 서울보다 평가값이 작은 춘천(11), 양평(10), 대전(9) 중에서 평가값이 가장 작은 대전을 선택하여 방문한다.

- open: 5, 4, 3
- close: 1, 2, 5

3 세 번째 확장

우선순위가 가장 높은 5번 노드를 선택하여 방문한다. 5번 노드를 오른쪽, 아래로 이동하여 6번, 7번 노드를 각각 생성한다. 6번 노드의 휴리스틱이 0으로 목표 노드와 일치하므로 탐색을 중단한다.

3-2. 최상우선탐색 결과

결과적으로는 1 → 2 → 5 → 6으로 DFS와 같은 순서로 트리 탐색이 이뤄졌다. 하지만 더 깊이 살펴보면, DFS는 스택을 사용하여 마지막 자식 노드까지 스택에 저장하여 방문하고, 최상우선탐색은 마지막 목표 노드인 자식 노드를 굳이 우선순위 큐에 저장하지 않아도 탐색을 종료할 수 있다는 차이점이 있다.

지도 방법

지도를 그래프로 구조화하고, 문제를 분해하고 탐색 과정을 설계하는 과정에서 1-2단원에서 배운 '탐색 문제 설계 과정'을 떠올릴 수 있도록 안내한다.

*** 그래프**
연결된 객체 즉 정점 사이의 관계를 보여주는 자료 구조다.
• 정점: 트리의 노드와 같은 객체를 의미한다.
• 간선: 트리의 링크와 같이 객체 간의 연결을 의미한다.

문제 상황에 따라 평가 함수값이 큰 정점을 선택하기도 해요. 여기서는 독도까지의 거리가 짧은 것이 더 좋은 선택이므로 평가 함수값이 작은 정점을 선택했어요.

지도 방법

최상우선탐색과 A* 탐색의 탐색 경로와 탐색 비용을 계산해 보고, 평가 함수 차이에 따라 효율이 달라질 수 있다는 것을 이해할 수 있도록 지도한다.

*** 용어 해설**

• A* 탐색 알고리즘의 최적 경로 보장: A* 알고리즘이 최적 경로를 보장하려면,

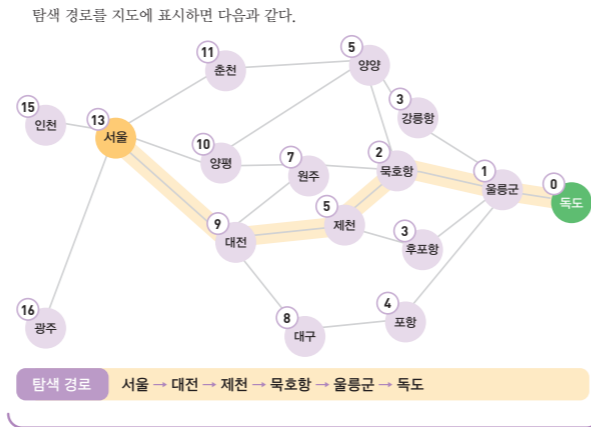
① 인정할 수 있는(admissible) 휴리스틱이어야 한다.

'admissible하다'는 말은 휴리스틱은 항상 실제 비용보다 적거나 같아야 한다는 뜻이다.

$$h(n) \leq h^*(n)$$

$h(n)$ 은 휴리스틱 함수가 추정한 비용을 의미하고, $h^*(n)$ 은 실제 최단 경로 비용을 의미한다. 즉, 휴리스틱 함수가 목표 상태까지 실제 비용을 넘어서지 않아야 한다.

이제 현재 상태인 대전과 연결된 원주(7), 제천(5), 대구(8)를 평가한다. 셋 중 평가 함수 값이 가장 작은 제천을 방문한다. 제천에서 연결된 목호항(2), 후포항(3) 중 평가값이 작은 목호항을 선택한다. 목호항에서 울릉군까지의 연결은 한 가지이고, 울릉군의 평가값이 목호항의 평가값보다 작으므로 울릉군을 방문한다. 울릉군에서 독도를 방문하면 목적지에 도착하였기 때문에 탐색을 마친다.



A* 탐색 알고리즘은 많은 내비게이션이 채택한 효율적인 알고리즘이다.

02 A* 탐색 알고리즘

A* 탐색 알고리즘은 최상우선탐색을 개선한 알고리즘이다. 최상우선탐색은 현재 상태에서 유리한 선택을 하여 탐색 범위를 좁혔지만, 다른 길로 가는 것이 탐색 비용을 줄일 수 있는 경우에도 평가 함수값이 더 크면 선택할 수 없었다. A* 탐색은 앞으로 남은 탐색 거리에, 초기 상태에서 현재 상태에 이르기까지의 비용도 고려하므로 보다 탐색 비용을 줄일 수 있는 알고리즘이다.

예제 '독도로 가는 길 찾기' 문제를 A* 탐색으로 해결해 보자.

A* 탐색은 최상우선탐색과 같이, 현재 상태와 연결된 정점 중 평가 함수값이 가장 작은 정점을 방문한다. 다만 최상우선탐색이 휴리스틱 함수값만을 사용했던 것과 달리, 초기 상태에서 현재 상태까지의 탐색 비용을 반영하여 갱신된 평가 함수값을 사용한다. 칠수는 A* 알고리즘으로 독도에 가는 길을 탐색하기 위해 지역 간 이동 시간을 검색해서 얻은 정보를 반영시킨 새로운 평가 함수값을 얻고 표로 정리했다.

1 첫 번째 확장

초기 상태의 확장 결과 2번, 3번, 4번 노드가 생성된다. 이동 횟수는 1회이므로 $g(n)$ 은 1이다.

자식 노드를 큐에 저장하지만, 우선순위가 높은 자식 노드 즉, 목표 상태와 다른 칸이 적은 노드부터 저장한다.

- 2번 노드는 1, 8칸이 다르므로 2, $f(n) = 2 + 1 = 3$
 - 3번 노드는 1, 2, 3, 8칸이 다르므로 4, $f(n) = 4 + 1 = 5$
 - 4번 노드는 1, 2, 8칸이 다르므로 3, $f(n) = 3 + 1 = 4$
- 의 우선순위가 생긴다.

- open: 2, 4, 3
- close: 1

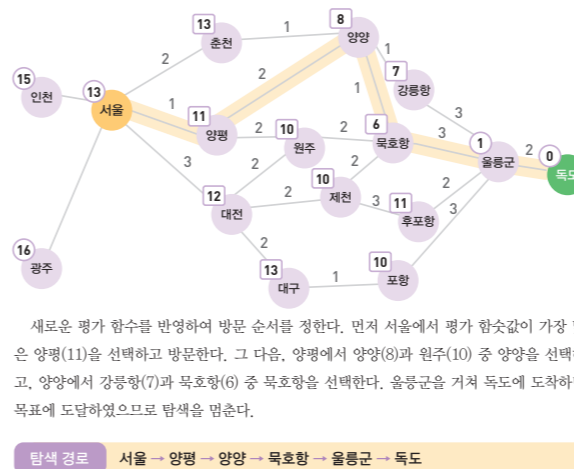
4-1. A* 탐색 알고리즘

8 퍼즐을 A* 탐색하기 위해서 목표 상태와 다른 칸의 수를 휴리스틱 $h(n)$ 으로 정한다. 또 초기 상태에서 현재 상태까지 사용되는 비용인 $g(n)$ 은 이동 횟수로 정한다. 평가 함수는 휴리스틱과 비용을 합하여 $f(n) = h(n) + g(n)$ 으로 정하고, $f(n)$ 이 낮은 노드를 우선순위로 정하고 탐색한다.

• $g(n)$ 은 초기 상태에서 현재 상태까지의 탐색 비용을 의미한다.

현재 상태 경로	다음 상태	$g(n)$	$h(n)$	$f(n) = g(n) + h(n)$
서울	춘천	2	11	2+11=13
	양평	1	10	1+10=11
	대전	3	9	3+9=12
서울-양평	양양	1+2	5	1+2+5=8
	원주	1+2	7	1+2+7=10
서울-양평-양양	강릉항	1+2+1	3	1+2+1+3=7
	목호항	1+2+1	2	1+2+1+2=6

다음 그래프의 간선 위 숫자는 각 정점 간의 이동 거리, 즉 탐색 비용을 의미한다. 또 네모 안의 숫자는 평가 함수값이다. 38쪽 표에 제시된 휴리스틱 값에, 지금까지의 이동 시간을 더하여 얻은 새로운 평가 함수값을 사용한다.



새로운 평가 함수를 반영하여 방문 순서를 정한다. 먼저 서울에서 평가 함수값이 가장 낮은 양평(11)을 선택하고 방문한다. 그 다음, 양평에서 양양(8)과 원주(10) 중 양양을 선택하고, 양양에서 강릉항(7)과 목호항(6) 중 목호항을 선택한다. 울릉군을 거쳐 독도에 도착하면 목표에 도달하였으므로 탐색을 멈춘다.

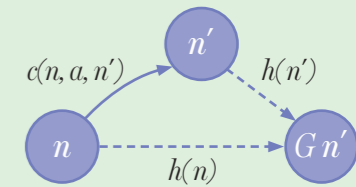
소단원 1분 요약

- 1. 맹목적 탐색이란 목표 상태 외의 다른 정보 없이 일정한 순서대로 목표 상태에 이르기까지 모든 상태 공간을 탐색하는 것으로, 깊이 우선 탐색과 너비 우선 탐색 등이 있다.
- 2. 정보 이용 탐색이란 휴리스틱 정보를 이용해 탐색 범위를 줄이고, 목표 상태까지 가기에 가장 좋은 경로라고 판단되는 노드를 먼저 방문하는 탐색으로 최상우선탐색, A* 탐색 등이 있다.

- open: 5, 4, 3
- close: 1, 2

*** 용어 해설**

• A* 탐색 알고리즘의 최적 경로 보장: ② 일관적인(consistent) 휴리스틱이어야 한다.



$$h(n) \leq c(n, a, n') + h(n')$$

$c(n, a, n')$ 는 노드 n에서 다음 상태 n' 로 가는 비용을 의미하고, $h(n')$ 은 다음 노드에서 목표 상태까지의 예측 비용을 의미한다. 임의의 연결된 두 노드와 목표 노드 사이의 삼각 부등식이 성립하면, 한 노드에서 다른 노드로 가는 비용은 중간 노드를 거쳐 가는 비용보다 크지 않는 뜻으로, 경로 확장 시 불필요한 노드를 다시 확장하지 않게 한다.

출처 Artificial Intelligence: A Modern Approach, 4th edition, Stuart Russell 외를 편집함.

3 세 번째 확장

결과적으로는 1 → 2 → 5 → 6으로 최상우선탐색과 같은 순서로 탐색이 이뤄졌다. 보통의 경우 A* 탐색은 비용을 고려하기 때문에 최상우선탐색보다 A* 탐색이 최단 경로를 찾는 경우가 더 많지만, 이번 예제에서는 같은 결과를 보여 주었다.

교과서 본문에서 8 퍼즐을 깊이 우선 탐색과 너비 우선 탐색으로 비교했다. 이 활동을 통해 맹목적 탐색과 정보 이용 탐색을 비교할 수 있도록 지도한다.

- 1 답을 제시해 주었기 때문에, 학생들과 같이 읽어 보며 개념을 명확히 한다.
- 2 이동 횟수를 앞칸에, 목표 상태와 다른 칸의 수를 세어 두 번째 칸에 쓰고 더하여 f(n)을 완성한다.
- 3 학생들에게 인터넷 검색을 할 시간을 주고, 지능적 탐색이 어디에 쓰이는지 직접 찾아 보고 작성하도록 지도한다.
- 4 휴리스틱에 대해서도 인터넷 검색을 통해 휴리스틱에 관한 다양한 지식을 쌓을 수 있는 기회를 제공한다.

A* 탐색으로 8 퍼즐 해결하기

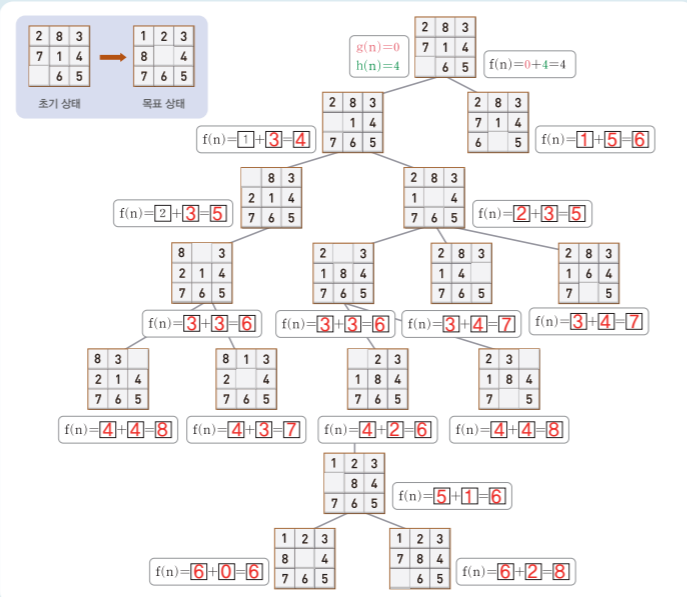
맹목적 탐색으로 해결하였던 8 퍼즐에 정보 이용 탐색을 적용하여 문제를 해결해 보자.

8 퍼즐의 평가 함수는 다음과 같이 정해 보자.

- 휴리스틱 h(n): 8 퍼즐의 각 타일이 목표 상태와 다른 위치에 있는 개수로 정한다.
- 탐색 비용 g(n): 현재까지의 이동 횟수로 정한다.
- 평가 함수 f(n): 현재까지의 이동 횟수와 휴리스틱 값을 더한 값으로 정한다. - 함수식: $f(n) = g(n) + h(n)$

다음은 8 퍼즐에 A* 탐색을 적용하여 문제를 해결하는 과정이다. 알맞은 g(n)과 h(n), f(n)을 적어 보자.

(단, 평가값이 같은 경우에는 모두 방문하여 지식 노드를 생성한다.)



지능적 탐색이 필요한 다른 문제를 인터넷 검색 등을 통해 찾아보자.

평가 함수로 고려할 수 있는 또 다른 휴리스틱에는 무엇이 있을지 이야기해 보자.



ALPHAGO

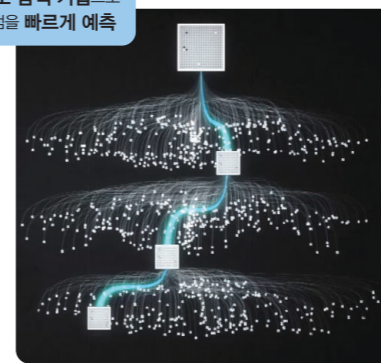
알파고와 몬테카를로 트리 탐색

몬테카를로 방법은 무작위로 선택한 표본을 이용한 확률적 결과를 사용한 값을 계산하는 알고리즘이다. 선거의 출구조사 결과가 실제 선거 결과와 크게 다르지 않은 것처럼, 만약 현재 상태에서 어느 한쪽이 유리한 국면이었다면 유리한 국면에 있는 사람이 이길 확률이 높다는 것이다. 이러한 확률을 게임 트리에 접목한 것이 몬테카를로 트리 탐색이다. 몬테카를로 트리 탐색은 알파고의 결정 과정을 최적화하는 알고리즘으로 선택, 확장, 시뮬레이션, 역전파의 네 가지 단계로 동작한다. 알파고에서 이 네 가지 단계가 어떻게 동작하는지 살펴보자.

첫 번째 단계는 선택이다. 수많은 경우의 수를 계산하여 가장 승리 확률이 높은 수를 선택한다. 두 번째 단계는 확장이자 선택의 결과로 이어지는 다음 수로 가능한 목록이 생겨난 것이다. 세 번째는 시뮬레이션이다. 다음 수에서 이어지는 게임의 결과를 예측해 보는 것이다. 모든 다음 수를 시뮬레이션하기에는 저장 장치의 한계와 시간적인 문제가 있으므로 다음 수 중에서 무작위로 임의의 수를 선택한다. 네 번째 단계는 역전파다. 역전파 단계에서는 시뮬레이션으로 얻은 정보를 현재 노드에서부터 시작 노드, 즉 역방향으로 되돌아가며 그 경로상에 있는 모든 노드의 정보를 갱신한다. 시뮬레이션의 결과, 승리했다면 승리한 상태부터 거슬러 올라가며 지금까지 둔 모든 수에 승점을 부여하는 것이다.

첫 번째부터 네 번째까지 과정을 반복해서 수행한 후 모든 단계가 끝나면 다음 수를 결정하는데 이때 가장 승점이 높은 수가 아닌, 가장 많이 진행된 수를 다음 수로 선택한다. 한두 번은 우연한 승리라고 생각할 수 있지만 승리가 쌓이면 더 이상 우

알파고는 몬테카를로 탐색 기법으로 유리한 지점을 빠르게 예측



연이라고 하기 어렵기에 더 높은 신뢰도를 부여하는 것이다. 몬테카를로 트리 탐색을 통해 알파고는 탐색의 범위를 획기적으로 좁히며 알파고의 성능 발전에 크게 이바지했다.

참고 영상
EBS 이윤, 인공지능 첫 걸음, 알파고의 필수 전략. <https://www.youtube.com/watch?v=QD-pNr-VLbD0>(원출처: EBS)

참고 도서
박상길(2022), 『비전공자도 이해할 수 있는 AI 지식』, 빈다.

3 지능적 탐색이 필요한 다른 문제를 인터넷 검색 등을 통해 찾아보자.

예시 답안
미로 찾기, 강 건너기 문제, 틱택토, 게임 등

출처 <https://www.playsw.or.kr/artificial/textbook/detail/6>

해설 지능적 탐색은 다양한 분야에서 사용되고 있다. 우리 삶의 다양한 분야에서 사용되는 탐색으로 다양한 문제를 해결할 수 있다.

4 평가 함수로 고려할 수 있는 또 다른 휴리스틱에는 무엇이 있을지 이야기해 보자.

예시 답안
맨해튼 거리, 유클리드 거리 등

해설 본문에서는 학생들의 이해와 쉬운 계산을 위해 다른 타일 수로 휴리스틱을 정했지만, 실제로 맨해튼 거리나 유클리드 거리가 조금 더 정교한 휴리스틱을 제공한다.

4 맨해튼 거리 알고리즘

뉴욕 맨해튼의 도로망은 직선의 격자 구조로 되어 있어서 대각선으로는 이동할 수 없고, 수평 및 수직 방향으로만 이동할 수 있다. 맨해튼 거리(Manhattan Distance) 알고리즘은 맨해튼 거리와 같이 수직, 수평 방향으로만 이동이 가

능한 문제에서 거리를 측정하기 위해 사용하는 알고리즘이다. 맨해튼 거리는 수평 이동 거리와 수직 이동 거리의 합으로 구한다. 첫 번째 지점의 좌표가 (x1, y1)이고, 두 번째 지점의 좌표가 (x2, y2)일 때, 식은 다음과 같다.

$$\bullet \text{ 맨해튼 거리} = |x1 - x2| + |y1 - y2|$$

5 8 퍼즐에서 맨해튼 거리

8 퍼즐에서 맨해튼 거리는 각 타일이 현재 상태에서 목표 상태로 이동하는 데 필요한 수평 및 수직 이동의 총합으로 구한다. 각 타일의 현재 상태와 목표 상태 사이의 맨해튼 거리를 계산하고, 모든 타일의 맨해튼 거릿값을 더한다. 초기 상태와 목표 상태가 42쪽 [탐구 활동]과 같다고 했

을 때, 각 타일에 대한 맨해튼 거리는 다음과 같다.

타일 1의 목표 상태는 (0, 0)이므로 (1, 1) → (0, 0), 맨해튼 거리 = |1 - 0| + |1 - 0| = 2이다. 이처럼 맨해튼 거리를 모두 구해 보면,

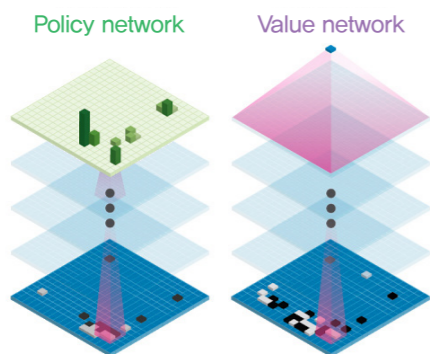
- 타일 2: (0, 0) → (0, 1), 맨해튼 거리 = |0 - 0| + |0 - 1| = 1
- 타일 3: (0, 2) → (0, 2), 맨해튼 거리 = |0 - 0| + |2 - 2| = 0
- 타일 4: (1, 2) → (1, 2), 맨해튼 거리 = |1 - 1| + |2 - 2| = 0
- 타일 5: (2, 2) → (2, 2), 맨해튼 거리 = |2 - 2| + |2 - 2| = 0
- 타일 6: (2, 1) → (2, 1), 맨해튼 거리 = |2 - 2| + |1 - 1| = 0
- 타일 7: (1, 0) → (2, 0), 맨해튼 거리 = |1 - 2| + |0 - 0| = 1
- 타일 8: (0, 1) → (1, 0), 맨해튼 거리 = |0 - 1| + |1 - 0| = 2 이고, 총합은 60이다.



1년 전 이세돌 9단과의 대결에서 승리를 거둔 '알파고'가 더 강해져서 돌아왔다. 이번에는 중국 우전에서 현재 세계 랭킹 1위인 커제 9단과 대국을 펼치고 있다. '알파고'와 관련된 사실 9가지를 정리했다.

1. 왜 하필이면 바둑이었을까

딥마인드가 연구 분야로 바둑을 선택한 이유는 바둑이 인류 역사상 가장 오래되고 깊게 연구된 게임이며, 인공지능 알고리즘을 구축하고 이해하기에 적합한 실험 환경을 만들 수 있어서다. 체스와 달리 바둑은 오랫동안 인공지능의 도전 과제로 남아 있었다. 바둑 특유의 복잡성 때문이다. 체스는 말이 움직이는 방식이 정해져 있고 가치도 각각 다르지만, 바둑은 자유롭게 돌을 놓는 방식이다. 바둑 경기의 경우의 수는 10의 170제곱에 이른다.



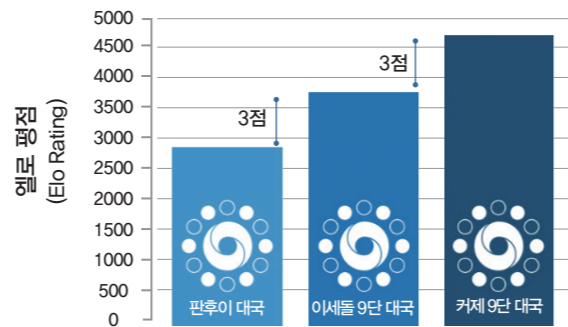
2. 알파고의 학습 방법

알파고는 '지도학습'과 '강화학습'의 강점을 결합한 새로운 기계학습 기법을 사용한다. 지도학습 과정에서는 바둑판에 돌이 놓인 각 위치를 '질문'으로 만들고 특정 수를 가지고 '해답'을 제시한다. 이어 강화학습 과정에서는 수를 선택하기 위한 정책을 개선한다. 즉 '셀프 대국'을 통해 예측의 정확성을 향상하는 것이다. 이 훈련 과정의 결과는 '정책망'으로서, 바둑판의 상태를 추출한 데이터를 활용해 가능한 여러 가지 수를 대상으로 개연성을 분석한다. 다음으로 '가치망'은 셀프 대국에서 특정 위치에 놓인 돌을 보고 승리 확률을 예측하여 -1(상대편의 승리 확실)부터 1(알파고의 승리 확실)까지의 점수를 매긴다.

3. 알파고가 돌을 놓는 방법

상대방이 바둑판에 돌을 놓으면 알파고는 몬테카를로 트리 탐색을 수행해 최적의 수를 결정한다. 각 시뮬레이션은 게임 트리에서 하나의 경로를 지나게 되는데, 여기서 정책망에 따

라 Q값(해당 수의 가치를 예측한 값)이 높으면서 개연성도 높은 수들을 선택한다. 시뮬레이션은 가치망 예측치와 롤아웃으로 종료된다. 롤아웃은 정책에 기반해 매우 빠른 속도로 계산을 수행한다. 알파고는 시뮬레이션에서 각 수의 Q값을 가치망 예측치와 롤아웃 결과의 평균값으로 업데이트한다. 할당된 탐색 시간이 지나면 알파고는 탐색 횟수가 가장 많은 수를 최적의 수로 판단하고, (이 수의 가치망 예측치가 가장 높지 않더라도) 이에 맞추어 플레이한다. 정리하자면, 알파고는 먼저 광범위한 탐색 과정을 거친 뒤 시간이 지남에 따라 최적의 수를 다듬어 결정한다는 의미다. 구글은 "인간의 수준에 가까운 알파고 신경망의 직관과 컴퓨터 시스템의 계산 능력을 통합해 강력한 판단력, 선견지명, 정확성을 가진 플레이어가 탄생하게 된다"라고 설명했다.



4. 지금의 알파고는 예전의 알파고를 석 점 깔고 이긴다

이세돌 9단과 대국한 알파고는 구글 클라우드 상 50개의 TPU(Tensor Processing Unit)를 사용했다. TPU는 구글이 머신러닝을 위해 특별히 제작한 처리 장치다. 1초에 50개의 수와 10만 개의 형태를 탐색할 수 있었다. 현재 커제 9단과 대국 중인 알파고는 '알파고 마스터'라고 불리는 버전이다. 이번 구글 I/O에서 공개된 단일 TPU 머신을 사용하며 2016년 버전 대비 10분의 1의 컴퓨팅 파워를 사용하면서도 더 빨리 계산한다. 이번 버전의 가장 큰 특징은 스스로 학습했다는 점이다. 알파고는 스스로와의 대국을 통해 강화학습을 거쳤다. 알파고 마스터는 2016년에 이세돌 9단과 대국했을 때의 알파고보다 훨씬 강하다. 예전의 알파고가 석 점을 깔아도 이기지 못한다. 물론 이는 현재의 알파고가 이전 버전 알파고의 약점을 잘 알고 있는 이유도 있다. (후략)

출처 돌아온 '알파고'에 관한 9가지 사실, 블로터(2017. 5. 25.), https://www.bloter.net/news/articleView.html?idxno=24921

04

도전! 독도로 가는 길 찾기

수업 시간: 4시간 교과서 쪽수: 44~53쪽

Table with 2 columns: 학습 목표, 수업 방법, 준비물. Content includes learning objectives, teaching methods (lectures, practice), and materials (textbook, computer).

Table with 3 columns: 단계, 교수·학습 방법, 지도상의 유의점. Content includes stages like 도입 (Introduction), 전개 (Development), 정리 (Conclusion), and 평가 (Evaluation) with detailed learning activities and points.

04 도전! 독도로 가는 길 찾기

프로젝트 실습

학습 목표 정보 이용 탐색 알고리즘을 적용한 인공지능 프로그램을 개발할 수 있다.

학습 요소 탐색 방법을 적용한 인공지능 프로그램 개발

• 제시 의도 •

익숙한 예제를 활용하여 학생들의 이해를 돕고, 직접 프로그램을 개발하여 인공지능에서 지능적 탐색이 어떻게 적용되는지 배울 수 있다.

지도 방법

정보 이용 탐색으로 길을 찾았던 것처럼 프로그래밍을 통해 실생활의 문제를 해결할 수 있음을 안내한다.

예시 답안

A* 알고리즘 등을 적용하여 지도, 도로 교통 정보, 최단 경로를 찾기 위한 프로그램을 만들 수 있다.

🧠 생각 알기 도전! 인공지능 프로그래밍

두 친구는 독도로 가는 길에 대해 이야기를 나누다가 독도로 가는 길을 찾는 인공지능 프로그램 개발에 도전하기로 했다.

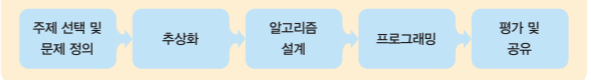


? 내비게이션 프로그램은 어떻게 만들 수 있을까?

도전! 독도로 가는 길 찾기

길 찾기는 지능적 탐색 활동의 대표적인 예다. 최상위탐색을 적용한 '독도로 가는 길 찾기 프로그램'을 개발하고, 인공지능 프로그램에 정보 이용 탐색이 어떻게 적용되는지 살펴보자.

프로그래밍 프로젝트의 절차



1 단계 주제 선택 및 문제 정의

- 주제 선택: 독도로 가는 길 찾기
- 문제 정의: 독도로 가는 길 편리하게 찾기

2 단계 추상화

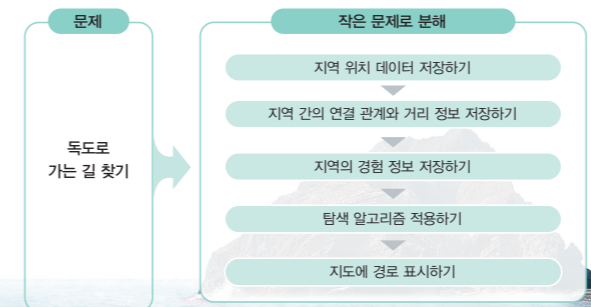
문제의 초기 상태, 목표 상태를 정의하고 더 작은 문제로 분해하여 해결하기 쉬운 문제로 추상화해 보자.

1 문제의 상태 정의



2 문제 분해 및 모델링

복잡한 문제를 작은 문제로 분해하여 모델링하면 다음과 같이 표현할 수 있다.



프로그래밍 프로젝트의 절차

1 주제 선택 및 문제 정의

실생활이나 다양한 학문 분야의 주제를 탐색하여 선정된 후, 해결하고자 하는 문제를 구체적으로 정의하는 단계

2 추상화

문제의 복잡성을 제거하고 해결하기 쉬운 형태로 표현하는 단계

3 문제의 초기 상태 정의

문제의 초기 상태 정의: 문제가 있는 초기 상태와 문제가 해결된 목표 상태를 정의하는 단계
 ② 문제 분해 및 모델링: 크고 복잡한 문제를 해결 가능한 작은 문제로 나누고 글, 그림, 그래프, 표 등을 사용하여 문제를 쉽게 이해할 수 있는 형태로 다시 표현하는 단계

4 알고리즘 설계

문제 분해 결과를 바탕으로 작은 문제별로 문제를 해결을 위한 알고리즘을 설계하는 단계

5 프로그래밍

설계한 알고리즘에 따라 프로그램 언어로 프로그램을 작성하는 단계

6 평가 및 공유

개발 의도에 맞게 프로그램이 동작하는지, 분해한 작은 문제가 각각 해결되어 처음의 큰 문제를 모두 해결했는지 정확성과 효율성을 확인하고, 피드백을 반영하여 완성된 프로그램을 공유하는 단계

지도 방법

프로그래밍 프로젝트의 절차가 낱개에 자세히 나와 있으므로, 본문보다 이 내용을 먼저 읽어 전반적으로 절차를 이해할 수 있도록 지도한다.

추상화의 장점

1 복잡성 감소

문제를 더 작은 단위로 나누거나, 중요하지 않은 요소를 제거하여 처리하기 쉽게 만든다.

2 재사용성 증가

일반화된 문제를 해결하면, 유사한 다른 문제에도 동일한 접근법을 적용할 수 있다.

3 효율적 문제 해결

본질적인 요소에 집중하므로 불필요한 연산과 자원 낭비를 줄일 수 있다.

4 이해도 향상

문제를 단순화하면 더 명확하게 이해하고, 다른 사람과 공유하기 쉽다.

1 내비게이션은 어떻게 발전했을까?

내비게이션은 미국이 보다 정확한 GPS 신호를 민간에 개방함으로써 발전하기 시작했다. 미국은 1970년대에 군사적 목적으로 24개의 GPS 위성을 쏘아 올렸다. 그러나 적대국이 사용하지 않을까 하는 우려에 일부러 최대 100m 이상 오차가 나게 했던 것을 2000년 클린턴 정부 때 오차를 없애고 이를 개방했다. 이때부터 민간에서도 정확도 높은 GPS 신호를 마음껏 사용할 수 있게 되며 적극적으로 활용할 수 있는 기반이 열렸다.

이후 내비게이션 시스템에 교통 정보와 머신러닝까지 연동되면서, 내비게이션 시스템은 더욱 스마트해졌다. 단순히 지도 정보를 보여 주고 현재 위치를 표시해 주었던 초창기 내비게이션이 실시간 교통 정보와 연동하면서 최적의 경로를 안내할 수 있게 되었다. 여기에 머신러닝을 탑재해 도착 시점에 즈음해 해당 구간이 막힐지, 언제 출발하면 막히지 않을지까지 예측해 내면서 편의성을 대폭 키웠다.

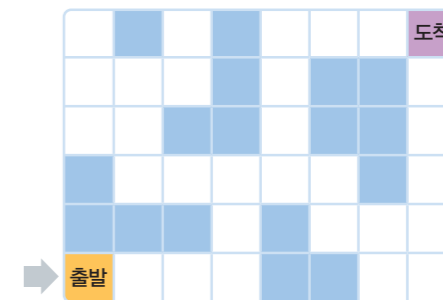
출처 박상길, 『비전공자도 이해할 수 있는 AI 지식』, 반니(2022)

추가 예제 도전! 미로 찾기 프로그래밍

VPython을 이용하여 교과서 21쪽에 제시된 미로를 탐색하는 프로그래밍에 도전해 보자.

참고 미로 찾기 예제 코드

<https://glowsript.org/#/user/jykim/folder/%EC%8B%9C%EB%B2%94/program/maze>
Web VPython 3.2



1단계 주제 선택 및 문제 정의

- 주제 선택: 미로 탐색 알고리즘 비교 프로그램
- 문제 정의: 깊이 우선, 너비 우선, A*, 세 가지 알고리즘을 사용하여 경로를 탐색하고, 시각적으로 나타내는 프로그램 만들기

*** 용어 해설**

- * pip:** pip는 'pip installs packages'의 약자로, 파이썬 패키지를 쉽게 설치, 관리, 업데이트, 삭제할 수 있도록 도와주는 패키지 관리자이다. pip는 패키지의 의존성을 자동으로 처리하며, 다양한 프로젝트에서 일관된 환경을 유지하는 데 유용하다. 가상 환경과 함께 사용하면 프로젝트 별 패키지 관리를 효율적으로 할 수 있다. 주요 명령어로는 install, uninstall, list, upgrade 등이 있으며, 이를 통해 패키지 설치와 업데이트를 간편하게 할 수 있다.
- * import:** import는 파이썬에서 다른 모듈이나 라이브러리의 기능을 현재 코드로 불러올 때 사용하는 키워드이다. 외부 파일이나 패키지의 함수, 클래스, 변수 등을 재사용할 수 있다. 예를 들어, 표준 라이브러리 또는 설치된 패키지에서 특정 기능을 가져와 활용할 수 있다. import는 코드의 재사용성을 높이고, 프로젝트를 모듈화하는 데 중요한 역할을 한다.

3 단계 알고리즘 설계

문제 분해를 바탕으로 알고리즘을 설계해 보자.

문제	알고리즘 설계
지역 위치 데이터 저장하기	지역의 x, y의 위치를 딕셔너리*에 저장하기
지역 간의 연결 관계와 이동 시간 저장하기	2차원 배열에 지역 간 이동 시간 정보 저장하기
지역의 휴리스틱 정보 저장하기	리스트에 지역의 휴리스틱 정보 저장하기
최상우선탐색 적용하기	① 현재 노드와 연결된 노드 중 경로에 없는 노드를 우선순위 큐*에 추가하기 ② 휴리스틱 정보가 가장 작은 노드에 가장 높은 우선순위 부여하기 ③ 우선순위가 가장 높은 노드를 큐*에서 꺼내 방문하기 ④ 방문한 노드를 방문집합과 경로 리스트 변수에 추가하고, 현재 노드까지의 이동 시간을 총 이동 시간 변수에 저장하여 갱신하기 ⑤ 현재 노드가 목적지가 아니고 큐에 남은 노드가 있다면 반복하기 ⑥ 실행이 끝나면 탐색 경로, 총 이동 시간 반환하기
지도에 경로 표시하기	지도에 탐색 경로와 총 이동 시간 표시하기

***딕셔너리**
키와 값을 한 쌍으로 저장하기 위한 자료형을 말한다.
 ① 지역명을 키로 정하고, 평가 함수값을 한 쌍으로 저장하기 '인천': 15

***큐와 우선순위 큐**
큐는 먼저 입력받은 것이 먼저 나오는 선입 선출(First In First Out) 방식을 사용하는 자료 구조다. 입장할 때 줄을 선 순서대로 들어가는 것과 같은 원리다. 우선순위 큐는 큐 안의 데이터에 우선순위를 부여해 우선순위가 높은 것부터 먼저 나오도록 설계된 큐다.

코랩 사용법은 부록 222~223쪽을 참고한다.

***맷플롯립**
데이터를 시각화하기 위해 사용하는 파이썬 라이브러리다.

4 단계 프로그래밍

파이썬(python)과 코랩(colab)을 이용하여 설계한 알고리즘을 바탕으로 독도로 가는 길 찾기 프로그램을 작성한다.

1 지역 위치 데이터 저장하기

길을 찾는 경로를 시각적으로 표현하기 위해 맷플롯립(matplotlib)*을 사용한다. 맷플롯립에서 한글을 출력하기 위해서는 별도의 절차가 필요하다. 탐색을 위해 필요한 큐, 그래프를 그리기 위한 맷플롯립, 한글 사용을 위한 모듈을 불러온다.

```

코드
1 #라이브러리 불러오기
2 import koreanize_matplotlib # 한글을 사용하기 위해 설치
    
```

```

코드
1 import queue # 정보를 저장하기 위해 큐를 사용
2 import matplotlib.pyplot as plt # 시각적인 표현을 위해 사용
3 import koreanize_matplotlib # 설치한 한글 모듈을 불러옴
    
```

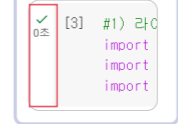
지도에서 노드로 표현된 지역들의 위치 데이터를 사용자 함수 Pos()로 정의한다. Pos() 함수는 딕셔너리 형태로 지역의 이름과 위치의 좌표값이 한 쌍으로 이루어져 있다.

코드

```

1 # 지역의 위치 저장
2 def Pos():
3     city = {
4         '인천': [11, 292], '서울': [100, 257], '광주': [11, 90],
5         '춘천': [205, 300], '양평': [195, 232], '대전': [209, 179], '양양': [295, 315],
6         '원주': [255, 230], '제천': [265, 185], '대구': [250, 80],
7         '강릉항': [332, 252], '목호항': [319, 215], '후포항': [310, 165], '포항': [290, 100],
8         '울릉군': [400, 223], '독도': [450, 200]
9     }
10    return city
    
```

Tip 결과 값이 나타나지 않더라도, 왼쪽에 초록색으로 체크 표시가 나타나면 실행 성공이다.



2 지역 간의 연결 관계와 이동 시간 정보 저장하기

각 지역 간 연결 관계와 지역 간의 이동 시간을 사용자 함수 mTime()으로 정의한다. 한 지역에서 다른 지역으로 연결된 길이 있다면 연결된 지역과 지역까지의 거리를 값으로 딕셔너리를 만든다. 예를 들어 '춘천': [['서울', 2], ['양양', 1]]의 의미는 춘천에서 서울과 양양으로 연결된 길이 있으며, 서울과의 이동 시간은 2, 양양과의 이동 시간은 1이라는 뜻이다. 이때 춘천-서울이 양방향으로 서로 연결되어 있으므로 키가 춘천인 곳에 서울의 값을 추가하고, 키가 서울인 곳에도 춘천의 값을 추가해야 한다.

코드

```

1 # 지역 간의 연결 관계와 이동 시간을 저장
2 def mTime():
3     graph = {
4         '인천': [['서울', 1]],
5         '서울': [['인천', 1], ['춘천', 2], ['양평', 1], ['대전', 3], ['광주', 5]],
6         '춘천': [['서울', 2], ['양양', 1]],
7         '양평': [['서울', 1], ['양양', 2], ['원주', 2]],
8         '대전': [['서울', 3], ['원주', 2], ['제천', 2], ['대구', 2]],
9         '양양': [['춘천', 1], ['양평', 2], ['강릉항', 1], ['목호항', 1]],
10        '원주': [['양평', 2], ['대전', 2], ['목호항', 2]],
11        '제천': [['대전', 2], ['목호항', 2], ['후포항', 3]],
12        '대구': [['대전', 2], ['포항', 1]],
13        '광주': [['서울', 5]],
14        '강릉항': [['양양', 1], ['울릉군', 3]],
15        '목호항': [['양양', 1], ['원주', 2], ['제천', 2], ['울릉군', 3]],
16        '후포항': [['제천', 3], ['울릉군', 2]],
17        '포항': [['대구', 1], ['울릉군', 3]],
18        '울릉군': [['강릉항', 3], ['목호항', 3], ['후포항', 2], ['포항', 3], ['독도', 2]],
19        '독도': [['울릉군', 2]]
20    }
21    return graph
    
```

① 각 지역의 위치는 지도 상의 x, y 좌표값으로 실습에 적절하도록 임의로 정하였다.

② 각 지역 간 이동 시간은 실습에 알맞도록 가능한 값이다.

Tip 2차원 리스트는 복잡한 데이터를 구조화하고 효율적으로 저장하기 위해 사용한다. 한 지역에서 여러 지역으로 연결되어 있고, 연결된 지역의 이름과 이동 시간 등의 정보가 필요하므로 2차원 리스트를 사용하였다. [[]]로 대괄호가 2쌍인 것을 주의하자.

2단계 추상화

1) 문제의 상태 정의

- 초기 상태: 미로를 탐색하지 않은 상태
- 목표 상태: 세 가지 알고리즘을 적용하여 미로를 탐색한 상태

2) 문제 분해 및 모델링

- 문제: 미로 탐색 알고리즘 비교 프로그램 만들기
- 작은 문제로 분해
 - 미로 데이터 준비하기
 - 알고리즘(DFS, BFS, A*) 구현하기
 - 탐색 결과 시각화하기
 - 알고리즘 선택 인터페이스 구축하기

3단계 알고리즘 설계

문제	알고리즘 설계
미로 데이터 설정하기	• 미로의 모양을 2차원 배열로 저장하기 • 미로의 크기를 정의하고, 초기 상태와 목표 상태 설정하기
DFS 알고리즘 선언하기	① 현재 노드의 자식 노드를 스택에 저장하고, 탐색하기 ② 갈 수 없을 때는 스택을 통해 이전 노드로 돌아가기 ③ 목표 상태에 도착하면 경로 반환하기
BFS 알고리즘 선언하기	① 현재 노드의 자식 노드를 큐에 추가하기 ② 큐에서 노드를 꺼내며 순서대로 탐색하기 ③ 목표 상태에 도착하면 경로 반환하기

문제	알고리즘 설계
A* 알고리즘 선언하기	① 현재 노드의 자식 노드를 우선순위 큐에 추가하기 ② 각 노드의 휴리스틱 값을 계산하여 우선순위를 부여하기 ③ 우선순위가 가장 높은 노드를 큐에서 꺼내 탐색하기 ④ 목표 상태에 도착하면 경로 반환하기
경로 시각화하기	탐색이 끝난 후 최종 경로를 화면에 시각적으로 표시하기

4단계 프로그래밍

VPython은 3D 그래픽을 쉽게 생성하고 시각화할 수 있도록 돕는 파이썬 라이브러리이다. 미로 찾기 프로그램을 VPython으로 구현하여 미로의 구조와 탐색 과정을 입체적으로 시뮬레이션하도록 한다.

• 환경 설명: <https://glowsript.org/>

웹에서 쉽게 VPython을 사용할 수 있는 glowsript 환경에서 프로그래밍한다.

코드

Web VPython 3.2

*** 용어 해설**

• **우선순위 큐**: 파이썬 표준 라이브러리에서 PriorityQueue 클래스는 queue 모듈 내에 정의되어 있다. queue 모듈은 파이썬 내장 패키지로, 파이썬 설치 시 함께 제공되기 때문에 별도의 추가 설치 없이 import만으로 사용할 수 있다. PriorityQueue는 우선순위를 설정하기 위해 최소 힙(min heap)을 사용한다. 최소 힙은 부모 노드가 자식 노드보다 항상 작거나 같은 값을 가지는 이진 트리 자료 구조로, 가장 작은 값을 빠르게 꺼낼 수 있는 특징이 있다. 이 구조를 통해 우선순위가 가장 높은 노드를 빠르게 찾고 처리할 수 있다. 연산 후 항상 정렬 과정이 필요하기에, 삽입과 삭제 연산의 시간 복잡도는 $O(\log n)$ 이 된다.

경험 정보는 각 지역에서 목적지인 독도와 직선 거리로 실습을 위해 가능한 값이다.

3 지역의 경험 정보 저장하기

이 프로그램에서 정보 이용 탐색을 적용하기 위해 지역의 휴리스틱 값을 담은 사용자 함수 H()를 정의한다.

```
코드
1 def H():
2     hn = {
3         '인천': 15, '서울': 13, '광주': 16,
4         '춘천': 11, '양평': 10, '대전': 9, '양양': 5,
5         '원주': 7, '재천': 5, '대구': 8,
6         '강릉함': 3, '목호함': 2, '후포함': 3, '포항': 4,
7         '울릉군': 1, '독도': 0
8     }
9     return hn
```

4 최상우선탐색 적용하기

bestFS() 함수는 최상우선탐색을 위해 구현된 사용자 함수다. start, hn, graph, goal 총 4개의 변수를 입력받는다.

- start: 출발지(서울)
- hn: 각 노드의 휴리스틱 값. H() 함수의 반환값을 사용
- graph: 지역 간의 연결 관계와 이동 시간. mTime() 함수의 반환값을 사용
- goal: 목적지(독도)

1 시작 노드를 큐에 저장하고 탐색을 위한 변수 초기화하기

bestFS() 함수는 최상우선탐색을 위해 구현된 함수다. 그래프의 시작 노드를 큐(queue)에 추가하고, 탐색에 필요한 변수를 선언하여 초기화한다.

```
코드
1 def bestFS(start, hn, graph, goal = '독도'):
2     pQueue = queue.PriorityQueue() # 우선순위 큐 생성
3     pQueue.put((hn[start], start, 0)) # 출발지의 이름, 출발지의 휴리스틱 값과 이동 시간을 큐에 추가
4     path = [] # 경로를 저장하기 위한 리스트 초기화
5     best_time = 0 # 총 이동 시간을 저장하기 위한 변수 초기화
6     visited = set() # 이미 방문한 노드를 저장하기 위한 집합
```

코드 Tip
queue를 사용하기 위해서는 모듈을 import해야 한다.
set(): 중복을 허용하지 않고 순서가 없는 자료형. 본문에서는 이미 방문한 도시를 기록해서 중복 방문을 방지한다.

2 탐색하기

현재 노드와 연결된 노드 중 아직 방문하지 않은 노드를 우선순위 큐에 추가하고, 그중에서 우선순위가 가장 높은 노드를 큐에서 꺼내 다음으로 방문한다. 방문한 노드를 visited 집합과 경로 리스트 path에 추가하고, 현재 노드까지의 이동 시간 c_

큐
자료의 삽입과 삭제가 선입 선출(First In First Out)로 이루어지는 자료 구조를 말한다.

이 코드는 교육용으로 이해를 돕기 위해 단순화된 경로 탐색 예제이다.

우선순위 큐
우선순위가 높은 데이터가 먼저 나오는 자료 구조를 말한다.

1) 미로 데이터 설정하기

미로의 모양을 2차원 배열로 저장한다.

```
코드
# 미로 정의(1은 벽, 0은 길, 'S'는 시작점, 'D'는 도착점)
maze = [
    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
    [1, 0, 1, 0, 1, 0, 0, 0, 'D', 1],
    [1, 0, 0, 0, 1, 0, 1, 1, 0, 1],
    [1, 0, 0, 1, 1, 0, 1, 1, 0, 1],
    [1, 1, 0, 0, 0, 0, 0, 1, 0, 1],
    [1, 1, 1, 1, 0, 1, 0, 0, 0, 1],
    [1, 'S', 0, 0, 0, 1, 1, 0, 0, 1],
    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
]
```

```
코드
# 미로의 크기(행과 열 계산)
rows = len(maze)
cols = len(maze[0])

# 이동 방향 순서: 오른쪽, 위, 아래, 왼쪽
direct = [(1, 0), (0, 1), (0, -1), (-1, 0)]

# 시작점과 도착점의 초깃값
start_pos = ''
goal_pos = ''

# 시작점 'S'와 도착점 'D'의 좌표를 찾아 저장
for i in range(rows):
```

time을 총 이동 시간 변수인 best_time에 저장하여 갱신한다. 방문한 노드가 목적지와 같다면 탐색을 종료하고, 탐색 경로 path와 총 이동 시간 best_time을 반환한다.

```
코드
1 while not pQueue.empty(): # 큐에 데이터가 있는 동안 반복
2     current, c_time = pQueue.get()[1:] # 우선순위가 가장 높은 노드를 꺼냄
3     if current in visited: # 현재 노드 visited에 있으면 건너뛰기
4         continue
5     visited.add(current) # 방문한 현재 노드를 visited 집합에 추가
6     path.append(current) # 방문한 현재 노드를 경로 리스트에 추가
7     best_time = c_time # 해당 노드까지의 이동 시간을 총 이동 시간에 더함
8     if current == goal: # 현재 노드가 목적지일 경우
9         break # 반복문 종료
10    for next, cost in graph[current]: # 현재 노드와 연결되어 있는 노드 방문
11        if next not in visited: # 방문하지 않은 노드만 탐색
12            new_cost = c_time + cost # 누적 이동 시간 계산
13            pQueue.put((hn[next], next, new_cost)) # 연결된 노드의 정보를 큐에 추가
14    return path, best_time # 최종 경로와 총 이동 시간을 반환
```

코드 Tip
함수의 리턴값은 함수를 실행한 결과를 반환하는 것을 말한다. 함수가 리턴값을 반환하면 함수의 실행이 종료되고 함수를 호출한 곳에 리턴값을 돌려준다. 리턴값은 변수나 다른 함수의 인자로 쓰인다. 이 코드에서는 Pos() 함수의 city, mTime() 함수의 graph, H() 함수의 hn, bestFS() 함수의 path, best_time이 리턴값이다.

5 지도에 경로 표시하기

탐색 후의 경로를 시각화하기 위해 사용자 함수 drawMap()을 정의한다. 각 지역의 위치는 빨간색 점으로 표시하고 지역의 연결 관계를 회색 선으로 보여 준다. 지역 간의 거리는 보라색 글자로 출력하고 최상우선탐색으로 찾은 경로를 노란색 선으로 보여 주도록 프로그램을 작성한다.

```
코드
1 def drawMap(city, bestfs, graph): # 최상우선탐색이 적용되는 그래프를 시각화
2     for i, j in city.items(): # 모든 지역에 대해 반복
3         plt.plot(j[0], j[1], 'ro') # 지역의 위치에 빨간 점으로 표시
4         plt.annotate(i, (j[0] + 5, j[1]), fontsize = 13) # 지역의 이름을 표시
5         for k in graph[i]:
6             n = city[k[0]] # 이웃 지역의 위치를 저장
7             plt.plot([j[0], n[0]], [j[1], n[1]], 'gray') # 지역 간의 연결선
8             # 지역 간의 거리를 표시
9             plt.annotate(str(k[1]), ((j[0] + n[0])/2, (j[1] + n[1])/2), color = 'purple', fontsize = 13)
10        for i in range(len(bestfs)-1):
11            first = city[bestfs[i]] # 경로의 현재 지역의 위치를 가져옴
12            second = city[bestfs[i + 1]] # 경로의 다음 지역의 위치를 가져옴
13            plt.plot([first[0], second[0]], [first[1], second[1]], 'yellow') # 최상우선탐색 경로 표시
```

*** 함수 해설**

① **plt.plot()**: 선 그래프 생성에 사용하는 라이브러리 함수이다.

- 기본 함수: plt.plot(x, y, **kwargs)
- **kwargs는 플롯의 스타일을 지정하는 다양한 매개 변수를 의미한다.
- 옵션: color(색상), linestyle(선스타일), linewidth(선 두께), marker(데이터 점의 모양) 등

② **plt.annotate()**: 그래프에 주석을 추가하는데 사용하는 라이브러리 함수이다.

- 기본 함수: plt.annotate(text, xy, xytext=None, arrowprops=None, **kwargs)
- 옵션: text(주석 텍스트), xy(주석을 추가할 좌표), xytext(텍스트의 위치), arrowprops(화살표 스타일), fontsize(텍스트 크기), color(텍스트 색상) 등

```
# 시작점과 도착점을 각각 노란색과 자주색 상자로 표시
box(pos=vector(start_pos[0], -start_pos[1], 0), size=vector(1, 1, 1), color=color.yellow)
# 시작점

box(pos=vector(goal_pos[0], -goal_pos[1], 0), size=vector(1, 1, 1), color=color.magenta)
# 도착점
```

```
for j in range(cols):
    if maze[i][j] == 'S':
        start_pos = (j, i)
        maze[i][j] = 0 # 시작점을 경로에 포함
    elif maze[i][j] == 'D':
        goal_pos = (j, i)
        maze[i][j] = 0 # 도착점을 경로에 포함

# 미로에서 벽을 파란 상자로 그리기
for i in range(rows):
    for j in range(cols):
        if maze[i][j] == 1:
            box(pos=vector(j, -i, 0), size=vector(1, 1, 1), color=color.blue)
```

2) DFS 알고리즘 선언하기

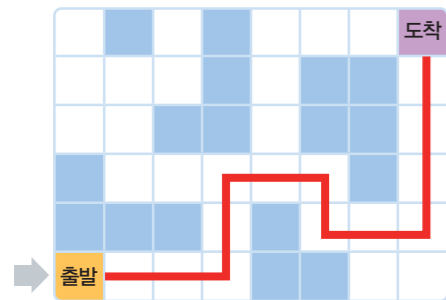
미로를 깊이 우선으로 탐색하는 프로그램을 구현한다.

코드

```
# 깊이 우선 탐색(DFS) 알고리즘 선언하기
def dfs(maze, start, goal, path=[]):
    x, y = start # 현재 위치의 좌표
    if start == goal:
        return path + [start] # 목적지 도달 시 경로 반환
    if maze[y][x] == 1 or (x, y) in path:
        return None # 벽이거나 이미 방문한 경로일 때
    path = path + [start] # 현재 위치를 경로에 추가
    for dx, dy in direct: # 이동 가능한 방향들
        next = (x + dx, y + dy) # 다음 상태 계산
        if 0 <= next[0] < cols and 0 <= next[1] <
            rows: # 미로 범위 안에 있는지 확인
            new_path = dfs(maze, next, goal,
                path) # 재귀 호출(스택처럼 사용)
            if new_path:
                return new_path # 탐색 경로 반환
    return None # 경로를 찾지 못한 경우
```

2-1. DFS 탐색 결과

정해진 순서대로 깊이 우선을 선택하여 탐색한 결과는 다음과 같다. 속도는 빠르지만 최단 경로를 보장하지 않는다.



3) BFS 알고리즘 선언하기

미로를 너비 우선으로 탐색하는 프로그램을 구현한다.

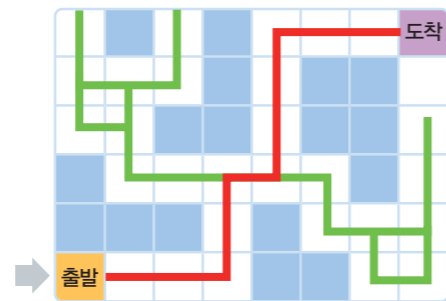
코드

```
# 너비 우선 탐색(BFS) 알고리즘 선언하기
def bfs(maze, start, goal):
    queue = [(start, [start])]
    # BFS 큐 초기화(현재 위치와 경로 포함)
    visited = set([start])
    # 방문한 좌표 집합에 저장 (중복 방지)

    while queue:
        c_state, path = queue.pop(0)
        # 큐에서 현재 상태를 꺼냄
        if c_state == goal:
            return path # 목적지 도달 시 경로 반환
        x, y = c_state
        for dx, dy in direct: # 이동 가능한 방향들
            next = (x + dx, y + dy) # 다음 상태 계산
            if 0 <= next[0] < cols and 0 <= next[1] <
                rows and maze[next[1]][next[0]] == 0 and next
                not in visited:
                visited.add(next) # 방문 기록 추가
                queue.append((next, path +
                    [next])) # 큐에 다음 상태 추가
                # 현재 위치와 이전 위치를 연결하는 경로를 표시
                if len(path) > 1:
                    prev = path[-2]
                    curve(pos=[vector(prev[0],
                        -prev[1], 0.5), vector(c_state[0],
                            -c_state[1],
                                0.5)], color=color.cyan)
                    rate(120) # 시뮬레이션 속도 조절
    return None # 경로를 찾지 못한 경우
```

3-1. BFS 탐색 결과

정해진 순서대로 너비 우선을 선택하여 탐색한 결과는 다음과 같다. 모든 상태 공간을 탐색하기 때문에, 속도는 느리지만 최단 경로를 보장한다.



- 초록색 선: 실제로 탐색한 과정을 나타냄.
- 빨간색 선: 탐색 후에 최종으로 결정한 경로를 나타냄.

4) A* 알고리즘 선언하기

미로를 깊이 우선으로 탐색하는 프로그램을 구현한다.

코드

```
# A* 알고리즘 정의
def astar(maze, start, goal):
    def heuristic(a, b):
        return abs(a[0] - b[0]) + abs(a[1] - b[1])
    # 맨해튼 거리를 휴리스틱으로 사용

    p_queue = [(0, start, [start])]
    # A* 탐색을 위한 우선순위 큐
    visited = set([start]) # 방문한 좌표 기록

    while p_queue:
        p_queue.sort() # 우선순위 큐 정렬 (fn 기준)
        _, c_state, path = p_queue.pop(0)
        # 최소 fn를 가진 상태 꺼냄
        if c_state == goal:
            return path # 목적지 도달 시 경로 반환
        x, y = c_state
        for dx, dy in direct: # 이동 가능한 방향들
            next = (x + dx, y + dy) # 다음 상태 계산
            next = (x + dx, y + dy) # 다음 상태 계산
            if 0 <= next[0] < cols and 0 <= next[1] <
                rows and maze[next[1]][next[0]] == 0 and next
                not in visited:
                visited.add(next) # 방문 기록
                gn = len(path) + 1 # gn는 이동한 거리
                hn = heuristic(next, goal)
                # hn는 휴리스틱 값
                fn = gn + hn # fn = gn + hn
                p_queue.append((fn, next, path
                    + [next])) # 우선순위 큐에 추가
                # 현재 위치와 이전 위치를 연결하는 경로를 표시
                if len(path) > 1:
                    prev = path[-2]
                    curve(pos=[vector(prev[0],
                        -prev[1], 0.5), vector(c_state[0],
                            -c_state[1],
                                0.5)], color=color.cyan)
                    rate(40) # 시뮬레이션 속도 조절
    return None # 경로를 찾지 못한 경우
```

지도 방법

- 평가 및 공유 단계에서는 실제로 프로그램을 실행시키고, 친구들과 이 탐색의 효율에 대해 이야기를 나누며, 사고력을 확장할 수 있도록 지도한다.
- [탐구 활동]에서 프로그램 성능 개선을 위해 개선된 알고리즘을 사용할 예정이므로, 자연스럽게 성능 개선과 연결 지어 생각할 수 있도록 한다.

코드 Tip

- items() 함수는 딕셔너리의 모든 '키-값' 쌍을 반환하는 함수다.
- plt.plot() 함수는 맷플롯립 라이브러리의 파이플롯 모듈에 속한 함수로 데이터를 선 그래프 형태로 시각화한다.
- plt.annotate() 함수는 맷플롯립 라이브러리의 파이플롯 모듈에 속한 함수로 그래프에 주석을 추가하는 데 사용된다.

5 단계 평가 및 공유

정해진 사용자 함수를 호출하면 프로그램이 실행된다. 프로그램을 실행하여 성능에 대하여 친구들과 의견을 공유하고 프로그램을 평가해 보자.

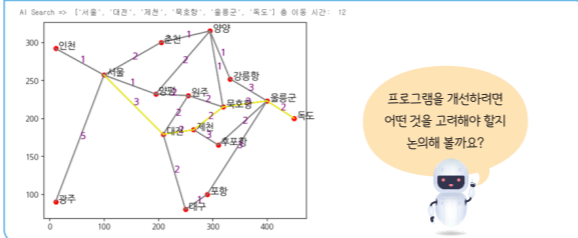
코드

```
1 heuristic = H() # 경험정보 함수 실행
2 graph = mTime() # 지역의 연결 관계 및 이동 시간 정보 함수 실행
3 city = Pos() # 도시의 좌표 설정
4 bestfs, best_time = bestFS('서울', heuristic, graph) # 최상위탐색 실행
5 drawMap(city, bestfs, graph) # 지도에 시각화
6 print('AI Search =>', bestfs, '총 이동 시간:', best_time) # 탐색 결과 출력
```

코드 Tip

H(), mTime() 등의 함수를 변수에 저장하면 함수의 실행 결과값이 반환되어 heuristic, graph 등의 함수에 저장된다. heuristic, graph 변수는 bestFS()의 인자로 쓰인다.

실행 결과



프로그램 성능 평가 결과와 공유한 의견은 프로그램 개선에 사용된다.

성능 평가하기

평가 항목	결과	
문제 해결에 적합한 탐색 알고리즘인가?	○	X
프로그램이 문제를 효과적으로 해결하였는가?	○	X

소단원 1분 요약

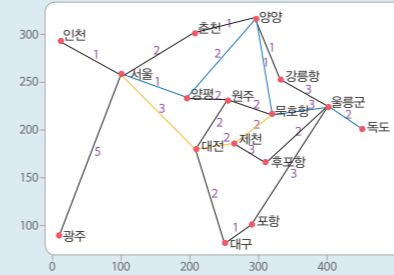
- 1 정보 이용 탐색 프로그램을 개발하려면 문제 정의, 추상화, 알고리즘 설계 등의 구조화 과정이 필요하다.
- 2 설계를 바탕으로 정보 이용 탐색 알고리즘을 적용한 프로그램을 구현하고, 성능을 평가한다.



A* 탐색 알고리즘을 적용한 길 찾기 프로그램 구현

- A* 탐색 알고리즘을 적용한 길 찾기 프로그램을 구현하여 프로그램을 개선하려고 한다. 빈칸(㉠~㉣)을 알맞은 내용으로 채워 프로그램을 완성해 보자.

최상위선 Search => [서울, '대전', '대전', '목포항', '울릉군', '도쿄'] 총 이동 시간: 12
 A* Search => [서울, '양평', '양평', '목포항', '울릉군', '도쿄'] 총 이동 시간: 9



단계 1 A* 탐색과 최상위탐색의 차이점을 파악하고, A* 탐색의 평가 함수 정의하기

- 차이점:
- 평가 함수 f(n) 정의하기
 - 최상위탐색: f(n)=h(n)
 - A* 탐색: f(n)=

단계 2 알고리즘 설계하기

문제	알고리즘 설계
A* 탐색 적용하기	<ol style="list-style-type: none"> 1 현재 노드와 연결된 노드 중 경로에 없는 노드를 우선순위 큐에 추가하기 2 경험 정보와 현재까지의 이동 시간의 합이 가장 작은 노드에 가장 <input type="text" value="높은"/> 우선순위 부여하기 3 우선순위가 가장 높은 노드를 큐에서 꺼내 방문하기 4 방문한 노드를 방문집합과 경로 리스트 변수에 추가하고, 현재 노드까지의 이동 시간을 총 이동 시간 변수에 저장하여 갱신하기 5 현재 노드가 목적지가 아니고 큐에 남은 노드가 있다면 반복하기 6 실행이 끝나면 탐색 경로, 총 이동 시간 반환하기

탐구 활동 지도 방법

최상위선탐색으로 구현했던 프로그램을 A* 탐색으로 해결하는 예제이다. 단순히 코드만 작성하기보다는 두 알고리즘의 차이를 잘 이해할 수 있도록 지도한다.

단계 1

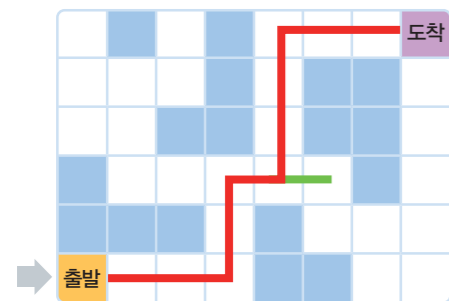
해설 최상위선탐색은 평갯값으로 h(n)만을 사용하지만, A*에서는 초기 상태에서 현재 상태까지 소비된 비용 g(n)을 더하여 평갯값으로 사용한다.

단계 2

해설 경험 정보와 현재까지의 거리의 합은 평갯값을 뜻한다. 경험 정보와 현재까지의 거리의 합이 작다는 말은 목표 상태와 가깝다는 뜻이므로, 평갯값이 가장 작은 노드에 가장 높은 우선순위를 부여한다.

4-1. A* 탐색 결과

A*를 선택하여 탐색한 결과는 다음과 같다. 평갯값이 같을 때는 모두 방문하는 것을 알 수 있다. 빠른 탐색 속도와 좋은 경로를 보장한다.



5) 시각화하기

탐색한 경로 중에서 최적의 경로를 선으로 보여 주어 시각화하는 코드를 작성한다. 또 알고리즘을 선택하고 실행하기 위한 코드를 작성한다.

코드

```
# 최종 경로를 시각화하는 함수
def final_path(path):
    for i in range(len(path) - 1):
        start = path[i] # 경로의 현재 위치
        goal = path[i + 1] # 경로의 다음 위치
        curve(pos=[vector(start[0], -start[1], 0.5), vector(goal[0], -goal[1], 0.5)], color=
            color.red) # 두 점을 연결
        rate(40) # 애니메이션 속도 조절
```

사용자 입력에 따라 탐색 알고리즘 선택

```
algorithm = input("탐색 알고리즘을 선택하세요 (1: DFS, 2: BFS, 3: A*): ")
```

사용자가 선택한 알고리즘에 따라 탐색 수행

```
if algorithm == "1":
    path = dfs(maze, start_pos, goal_pos)
# DFS 실행
elif algorithm == "2":
    path = bfs(maze, start_pos, goal_pos)
# BFS 실행
elif algorithm == "3":
    path = astar(maze, start_pos, goal_pos)
```

A* 실행

```
else:
    print("잘못된 입력입니다.")
# 잘못된 입력에 대한 메시지
path = None
# 최종 경로 시각화
if path:
    final_path(path) # 경로가 있을 때만 시각화
else:
    print("경로를 찾지 못했습니다.")
# 경로가 없을 경우 메시지 출력
```

단계 ③ 2

- 시작 노드를 큐에 저장하고 탐색을 위한 변수 초기화하기 **해설** 평가값 계산을 위한 gn 변수를 선언하고 0으로 초기화한다.
- 탐색하기 **해설** gn을 더해 평가값을 재계산하고, 우선순위 큐에 저장한다.

예제 코드는 학생들의 개념 이해를 돕기 위해 단순화된 형태로 구성되었다. 하지만 실제 환경에서 더 복잡하거나 가변적인 그래프를 처리할 때는 확장성이 좋은 알고리즘 구조가 필요하다. 부모 노드 기록(parent 개념)을 추가하면 탐색 과정에서 각 노드에 도달하게 한 최적의 부모 노드를 설정할 수 있고, 탐색 완료 시 이 정보를 따라 역추적하기 때문에 A* 같은 알고리즘에서 최단 경로를 보장할 수 있다.

역추적이 가능한 수정된 Astar 알고리즘 예시

```
def Astar(start, hn, graph, goal='목도'):
    pQueue = queue.PriorityQueue()
    pQueue.put((hn[start], start, 0))
    parent = {}
    g_cost = {start: 0} # start까지 누적 비용
    astar_time = 0

    while not pQueue.empty():
        f_val, current, c_time = pQueue.get()

        # 현재 노드까지 비용이 이미 기록된 최단 비용보다 크면 무시
        if c_time > g_cost.get(current, float('inf')):
            continue

        if current == goal:
            astar_time = c_time
            break

        for next, cost in graph[current]:
            gn = c_time + cost
            # 더 짧은 경로가 발견될 때 업데이트
            if gn < g_cost.get(next, float('inf')):
                g_cost[next] = gn
                f = hn[next] + gn
                pQueue.put((f, next, gn))
                parent[next] = current

        # 경로 복원
        path = []
        node = goal
        while node != start:
            path.append(node)
            node = parent.get(node, start)
        path.append(start)
        path.reverse()

    return path, astar_time
```

단계 ③ 프로그래밍

실제한 알고리즘을 바탕으로 A* 알고리즘을 적용한 독도로 가는 길 찾기 프로그램을 작성해 보자.

- 준비하기
 - 라이브러리 불러오기
 - 지역별 위치 데이터 설정하기
 - 지역 간의 연결과 이동 시간 정보 설정하기
 - 각 지역의 경험 정보 설정하기
 - ※ 최상위탐색 알고리즘으로 길 찾기 프로젝트(46~49쪽)에서 작성했던 소스 코드를 그대로 사용하도록 한다.
- A* 탐색 알고리즘 구현하기

- 시작 노드를 큐에 저장하고 탐색을 위한 변수 초기화하기

Astar() 함수는 bestFS()를 활용한 사용자 함수다. 기본 구조를 그대로 사용하되 시작 노드에서 현재 노드까지 이동한 시간을 저장하는 변수 gn을 새롭게 선언하고 0으로 초기화한다.

```
코드
1 def Astar(start, hn, graph, goal='목도'):
2     pQueue = queue.PriorityQueue() # 우선순위 큐를 생성
3     pQueue.put((hn[start], start, 0)) # 시작 노드를 큐에 삽입
4     gn = 0 # 시작 노드에서 현재 노드까지 이동한 시간을 저장하는 변수 초기화
5     path = [] # 경로를 저장할 리스트 생성
6     astar_time = 0 # 총 이동 시간을 저장할 변수 초기화
7     visited = set() # 이미 방문한 노드를 저장하기 위한 집합
```

- 탐색하기

이번 단계도 bestFS()의 일부만 수정하여 사용한다. 수정할 부분은 다음과 같다. 현재 노드까지의 이동 시간 c_time을 총 이동 시간 astar_time에 저장한다. 현재까지의 총 이동 시간에 연결된 노드까지의 이동 시간을 더하여 gn 변수에 저장한다. 인접 노드의 휴리스틱 값에 gn을 더한 값을 우선순위로 설정하여 큐에 추가한다.

```
코드
1 <생략>
2 astar_time = c_time # 현재 노드까지의 이동 시간을 총 이동 시간에 저장하기
3 if current == goal: # 현재 노드가 목적지라면
4     break # 반복문 종료
5 # 현재 노드와 연결되어 있는 노드 방문
6 for next, cost in graph[current]:
7     if next not in visited: # 방문하지 않은 노드만 탐색
8         gn = astar_time + cost # 현재까지의 총 이동 시간에 연결된 노드까지의 이동 시간을 더함
9         pQueue.put((hn[next] + gn, next, gn)) # f = hn + gn로 우선순위 큐에 추가
10    return path, astar_time # 최종 경로와 총 이동 시간을 반환
```

2 게임 트리

탐색에서 국면의 전개를 표시하기 위해 게임 트리를 사용해 모델링하기도 한다.

게임 트리(game tree)는 게임의 모든 가능한 상태와 움직임을 트리 구조로 표현한 것이다. 게임 트리는 탐색 과정에서 완전 게임 트리(complete game tree)와 부분 게임 트리(partial game tree)로 나뉜다.

참고 도서 후루카와 히데카즈. 『알파제로를 분석하며 배우는 인공지능』. 제이펍(2019)

3 완전 게임 트리와 부분 게임 트리

완전 게임 트리는 게임이 시작된 순간부터 끝날 때까지 모든 경우의 수를 전부 나열한 트리이다. 많은 경우 이 트리는 너무 크고 복잡하기 때문에 실제로는 계산이 불가능할

단계 ③

- 지도에 경로 표시하기

해설 A*를 적용한 탐색이 지도에 표시되도록 drawSearchMap 함수를 수정한다.

- 프로그램 실행하기

해설 메인 함수에서 astar 함수가 호출되도록 drawMap 함수의 호출 매개 변수에 astar를 추가한다.

- 단계 ⑤

해설 실제로 예제 프로그램으로 지도서에 제공되었으므로 해설은 생략한다.

- 지도에 경로 표시하기

이번 단계도 기존의 drawMap() 함수를 수정하여 사용한다.

 - astar 변수를 추가한다.
 - bestfs 변수를 astar 변수로 바꿔 Astar()로 탐색한 결과가 지도에 출력되도록 한다.

```
코드
1 def drawSearchMap(city, bestfs, @astar, graph):
2     <생략>
3     for i in range(len(astar)-1):
4         first = city[astar[i]]
5         second = city[astar[i+1]]
6         plt.plot([first[0], second[0]], [first[1], second[1]], 'blue')
```

- 프로그램 실행하기
 - bestFS() 함수 호출을 참고하여 Astar() 함수를 호출한다.
 - drawMap() 함수에 astar 변수를 추가한다.
 - 탐색 경로와 총 이동 시간을 화면에 출력하는 부분을 추가한다.

```
코드
1 <생략>
2 astar, astar_time = Astar('서울', heuristic, graph)
3 drawMap(city, bestfs, @astar, graph)
4 print('A* Search => ', astar, '총 이동 시간: ', astar_time)
```

단계 ④ 평가 및 공유

프로젝트 과정을 평가해 보자.

평가 항목	결과	
문제 해결에 적합한 탐색 알고리즘인가?	<input type="radio"/>	<input checked="" type="radio"/>
프로그램이 문제를 효과적으로 해결하였는가?	<input type="radio"/>	<input checked="" type="radio"/>
bestFS에 비해 A*가 효율적인 탐색을 하였는가?	<input type="radio"/>	<input checked="" type="radio"/>

자기평가, 동료 평가

평가 요소	자기 평가			동료 A			동료 B			동료 C		
	상	중	하	상	중	하	상	중	하	상	중	하
알고리즘 이해도												
문제 해결 방식의 창의성												
참여도 및 협업 능력												
프로그램의 완성도												

단계 ⑤ 확장하기

지능적 탐색을 적용하여 프로그래밍할 수 있는 또 다른 예를 찾아보자.

미로 찾기 프로그램 등을 개발할 수 있다.

수 있다. 체스나 바둑과 같은 복잡한 게임에서 완전 게임 트리를 사용하는 것이 비현실적이다.

부분 게임 트리는 탐색의 효율성을 높이기 위해 중요한 경로만 살펴보는 방식이다. 실제로 대부분의 알고리즘은 부분 게임 트리를 활용하여 최적의 수를 찾는다. 품질이 뛰어난 부분 게임 트리를 얼마나 효율적으로 만들어 내는가가 게임 AI의 성능을 결정한다.

4 미니맥스법

미니맥스(minimax)는 두 명의 플레이어가 번갈아 가며 최선의 선택을 하도록 설계된 알고리즘이다. 자신에게 있어서 최선의 수를 선택하고, 상대는 최악의 수를 선택한다는 가정에서 모든 경우의 수를 고려하여 자신에게 가장 유리한

선택을 찾아낸다. 미니맥스는 완전 탐색을 기반으로 최적의 해를 찾지만, 완전 게임 트리에서는 계산 비용이 너무 크다는 단점이 있다.

5 알파베타법

알파베타 가지치기(alpha-beta pruning)라고도 하는 알파베타법은 미니맥스 알고리즘을 개선한 방법으로, 더 이상 탐색할 필요가 없는 가지들을 잘라내는(pruning) 방식이다. 미니맥스는 모든 경우를 탐색하지만, 알파베타는 미니맥스에서 계산하지 않더라도 같은 계산 결과가 나오는 부분은 읽지 않도록 처리하여 꼭 필요한 부분만 탐색하므로, 탐색의 효율성이 향상된다. 이를 통해 부분 게임 트리를 사용하여 더 빠르게 최적의 수를 찾을 수 있다.

코드

```
# [미로 찾기 예제 코드] https://glowsript.org/#/user/jykim/folder/%EC%8B%9C%EB%B2%94/program/maze
Web VPython 3.2
# 미로 정의 (1은 벽, 0은 길, 'S'는 시작점, 'D'는 도착점)
maze = [
    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
    [1, 0, 1, 0, 1, 0, 0, 0, 'D', 1],
    [1, 0, 0, 0, 1, 0, 1, 1, 0, 1],
    [1, 0, 0, 1, 1, 0, 1, 1, 0, 1],
    [1, 1, 0, 0, 0, 0, 0, 1, 0, 1],
    [1, 1, 1, 1, 0, 1, 0, 0, 0, 1],
    [1, 'S', 0, 0, 0, 1, 1, 0, 0, 1],
    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
]

# 미로의 크기(행과 열 계산)
rows = len(maze)
cols = len(maze[0])

# 이동 방향 순서: 오른쪽, 위, 아래, 왼쪽
direct = [(1, 0), (0, 1), (0, -1), (-1, 0)]

# 시작점과 도착점의 초깃값
start_pos = ""
goal_pos = ""

# 시작점 'S'와 도착점 'D'의 좌표를 찾아 저장
for i in range(rows):
    for j in range(cols):
        if maze[i][j] == 'S':
            start_pos = (j, i)
            maze[i][j] = 0 # 시작점을 경로에 포함
        elif maze[i][j] == 'D':
            goal_pos = (j, i)
            maze[i][j] = 0 # 도착점을 경로에 포함

# 미로에서 벽을 파란 상자로 그리기
for i in range(rows):
    for j in range(cols):
        if maze[i][j] == 1:
            box(pos=vector(j, -i, 0), size=vector(1, 1, 1), color=color.blue)

# 시작점과 도착점을 각각 노란색과 자주색 상자로 표시
box(pos=vector(start_pos[0], -start_pos[1], 0), size=vector(1, 1, 1), color=color.yellow) # 시작점
box(pos=vector(goal_pos[0], -goal_pos[1], 0), size=vector(1, 1, 1), color=color.magenta) # 도착점
```

```
# 깊이 우선 탐색(DFS) 알고리즘 선언하기
def dfs(maze, start, goal, path=[]):
    x, y = start # 현재 위치의 좌표
    if start == goal:
        return path + [start] # 목적지 도달 시 경로 반환
    if maze[y][x] == 1 or (x, y) in path:
        return None # 벽이거나 이미 방문한 경로일 때
    path = path + [start] # 현재 위치를 경로에 추가
    for dx, dy in direct: # 이동 가능한 방향들
        next = (x + dx, y + dy) # 다음 상태 계산
        if 0 <= next[0] < cols and 0 <= next[1] < rows: # 미로 범위 안에 있는지 확인
            new_path = dfs(maze, next, goal, path) # 재귀 호출(스택처럼 사용)
            if new_path:
                return new_path # 탐색 경로 반환
    return None # 경로를 찾지 못한 경우

# 너비 우선 탐색(BFS) 알고리즘 선언하기
def bfs(maze, start, goal):
    queue = [(start, [start])] # BFS 큐 초기화(현재 위치와 경로 포함)
    visited = set([start]) # 방문한 좌표 집합에 저장(중복 방지)

    while queue:
        c_state, path = queue.pop(0) # 큐에서 현재 상태를 꺼냄
        if c_state == goal:
            return path # 목적지 도달 시 경로 반환
        x, y = c_state
        for dx, dy in direct: # 이동 가능한 방향들
            next = (x + dx, y + dy) # 다음 상태 계산
            if 0 <= next[0] < cols and 0 <= next[1] < rows and maze[next[1]][next[0]] == 0 and next not in visited:
                visited.add(next) # 방문 기록 추가
                queue.append((next, path + [next])) # 큐에 다음 상태 추가
    # 현재 위치와 이전 위치를 연결하는 경로를 표시
    if len(path) > 1:
        prev = path[-2]
        curve(pos=[vector(prev[0], -prev[1], 0.5), vector(c_state[0], -c_state[1], 0.5)], color=color.cyan)
    rate(120) # 시뮬레이션 속도 조절
    return None # 경로를 찾지 못한 경우

# A* 알고리즘 정의
def astar(maze, start, goal):
    def heuristic(a, b):
        return abs(a[0] - b[0]) + abs(a[1] - b[1]) # 맨해튼 거리를 휴리스틱으로 사용

    p_queue = [(0, start, [start])] # A* 탐색을 위한 우선순위 큐
    visited = set([start]) # 방문한 좌표 기록

    while p_queue:
```

05

지식의 표현과 추론

🕒 수업 시간: 2시간 📖 교과서 쪽수: 54~65쪽

```

p_queue.sort() # 우선순위 큐 정렬 (fn 기준)
_, c_state, path = p_queue.pop(0) # 최소 fn을 가진 상태 꺼냄
if c_state == goal:
    return path # 목적지 도달 시 경로 반환
x, y = c_state
for dx, dy in direct: # 이동 가능한 방향들
    next = (x + dx, y + dy) # 다음 상태 계산
    if 0 <= next[0] < cols and 0 <= next[1] < rows and maze[next[1]][next[0]] == 0 and next not in visited:
        visited.add(next) # 방문 기록
        gn = len(path) + 1 # gn은 이동한 거리
        hn = heuristic(next, goal) # hn은 휴리스틱 값
        fn = gn + hn # fn = gn + hn
        p_queue.append((fn, next, path + [next])) # 우선순위 큐에 추가
# 현재 위치와 이전 위치를 연결하는 경로를 표시
if len(path) > 1:
    prev = path[-2]
    curve(pos=[vector(prev[0], -prev[1], 0.5), vector(c_state[0], -c_state[1], 0.5)], color=color.cyan)
rate(40) # 시뮬레이션 속도 조절
return None # 경로를 찾지 못한 경우

# 최종 경로를 시각화하는 함수
def final_path(path):
    for i in range(len(path) - 1):
        start = path[i] # 경로의 현재 위치
        goal = path[i + 1] # 경로의 다음 위치
        curve(pos=[vector(start[0], -start[1], 0.5), vector(goal[0], -goal[1], 0.5)], color=color.red) # 두 점을 연결
        rate(40) # 애니메이션 속도 조절

# 사용자 입력에 따라 탐색 알고리즘 선택
algorithm = input("탐색 알고리즘을 선택하세요(1: DFS, 2: BFS, 3: A*): ")

# 사용자가 선택한 알고리즘에 따라 탐색 수행
if algorithm == "1":
    path = dfs(maze, start_pos, goal_pos) # DFS 실행
elif algorithm == "2":
    path = bfs(maze, start_pos, goal_pos) # BFS 실행
elif algorithm == "3":
    path = astar(maze, start_pos, goal_pos) # A* 실행
else:
    print("잘못된 입력입니다.") # 잘못된 입력에 대한 메시지
    path = None

# 최종 경로 시각화
if path:
    final_path(path) # 경로가 있을 때만 시각화
else:
    print("경로를 찾지 못했습니다.") # 경로가 없을 경우 메시지 출력
    
```

학습 목표	<ul style="list-style-type: none"> • 규칙과 사실을 활용하여 지식을 표현할 수 있다. • 새로운 지식을 추론하여 생성할 수 있다.
수업 방법	강의, 토론, 실습, 발표
준비물	교사 교과서, 단말기, 관련 교수 학습 자료 학생 필기도구, 단말기

단계	교수 · 학습 방법	지도상의 유의점
도입	<p>생각 열기</p> <p>〈생각 열기〉와 관련된 동영상을 보고, 인공지능이 어떻게 추론하는지 생각해 볼 수 있도록 한다.</p>	<ul style="list-style-type: none"> • 추론과 추론 방법에 대해 이해할 수 있도록 지도한다. 관련 앱이나 프로그램을 활용할 경우, 추론에 대한 이해보다 단순 재미에 치우치지 않도록 유의한다.
전개	<p>1 지식의 표현</p> <ul style="list-style-type: none"> • 인공지능에서 추론의 역할을 이해함으로써 추론을 위한 데이터, 정보, 지식의 차이를 설명할 수 있도록 한다. • 지식의 다양한 표현 방법을 이해하고, 자신이 알고 있는 지식을 다양한 표현 방법으로 표현해 볼 수 있도록 한다. <p>2 추론</p> <ul style="list-style-type: none"> • 연역 추론과 귀납 추론의 차이점을 알고, 추론을 통한 지식 생성 방법을 이해한다. • 지식 추론 방법의 이해를 바탕으로, 주어진 사실로 새로운 사실을 도출하는 추론 과정을 진행하여 제시된 문제를 해결한다. 	<ul style="list-style-type: none"> • 인공지능에서 추론이 어떻게 활용되는지 이해할 수 있도록 지도한다. • 데이터, 정보, 지식의 차이를 알고, 인공지능에서 어떻게 활용이 되는지 이해할 수 있도록 지도한다. • 지식을 다양한 지식 표현 방식으로 표현할 수 있도록 지도한다. • 추론을 통해 새로운 사실을 생성해 볼 수 있도록 지도한다.
정리	<p>탐구 활동</p> <ul style="list-style-type: none"> • 주어진 사실을 통해 새로운 사실을 생성하여, 제시된 문제를 해결할 수 있도록 지도한다. 	<ul style="list-style-type: none"> • 추론의 과정을 통해 문제를 해결하고, 그 과정을 설명할 수 있도록 지도한다. • 본 차시를 정리하고, 다음 차시를 예고한다.
평가	<ul style="list-style-type: none"> • 지식을 규칙과 사실을 활용하여 표현할 수 있는가? • 주어진 사실로 새로운 사실을 도출하는 추론을 할 수 있는가? 	<ul style="list-style-type: none"> • 지식을 표현하고, 제시된 사실로 추론을 통해 새로운 사실을 도출할 수 있는지 확인한다.

지식의 표현과 추론

학습 목표 • 규칙과 사실을 활용하여 지식을 표현할 수 있다.
• 새로운 지식을 추론하여 생성할 수 있다.

학습 요소 지식, 지식 표현, 추론

• 제시 의도 •

인공지능도 추론 능력이 있다. 제시된 사실이나 지식을 통해 새로운 사실이나 지식을 도출할 수 있다. 인공지능이 학습하는 사실이나 지식을 통해 새로운 사실이나 지식을 추론할 수 있는 것을 이해하고, 어떠한 추론 방법이 있는지 생각해 보자.

지도 방법

- 스무고개와 같은 형식으로 교사가 하나의 답을 생각한 뒤 학생들이 질문을 하고 이에 대한 답을 교사가 하는 형식으로 진행할 수 있다.
- 데이터 기반의 추론 프로그램을 활용하여 학생 개개인이 직접 체험해 볼 수 있다.
- 이러한 활동에서 추론하는 방법을 학생이 생각해 볼 수 있도록 지도한다.

예시 답안

학습하는 지식을 바탕으로 문제 해결에 알맞은 결론을 도출한다.

🗨️ 생각 열기 추론을 통한 지식 생성

인간은 기존에 알고 있는 사실이나 지식을 통해 새로운 사실이나 지식을 이끌어 낼 수 있다. 이러한 추론 능력은 인간의 지능을 모방하려는 인공지능도 갖고 있다.



추론이란 이미 알고 있는 사실을 이용하여 새로운 사실을 이끌어 내는 것을 뜻해요.

? 인공지능은 어떻게 추론을 할까?

1 학습과 추론

학습은 인공지능이 데이터를 통해 지식을 얻고, 이를 기반으로 패턴을 인식하거나 예측 모델을 구축하는 과정으로 볼 수 있다. 학습을 위해서는 많은 데이터가 필요하다. 추론은 학습의 결과를 실제 활용하는 것으로 학습한 지식을 사용하여 새로운 상황에 적용하거나 결론을 도출하는 것이다.

2 인공지능과 추론

인공지능에서 추론이란 훈련된 모델이 새로운 데이터에 대해 결론을 도출하는 것을 말한다. 다음은 추론이 사용된 사례이다.

- 대규모 언어 모델(LLM: Large Language Models):** 텍스트 데이터로 훈련된 모델은 새로운 텍스트 구문을 분석하고 해석한다.

- 스팸 분류:** 스팸 메일을 모델이 인식하도록 훈련시킨 뒤 이메일 메시지를 추론하여 스팸 메일을 차단한다.
- 무인 자동차:** 정지 표지판을 학습한 무인 자동차가 처음 주행하는 도로에 있는 정지 표지판을 인식한다.

3 지식의 종류

방법적 지식은 절차적 지식, 과정적 지식이라고도 하며, 해야 할 과제의 절차나 방법을 아는 것을 뜻하는 지식이다. 그렇기 때문에 ‘~을 할 줄 안다’고 말하기 위해서는 그 과정을 수행하기 위해 알아야 할 원리나 규칙을 익히는 것이 필요하다. 명제적 지식은 어떤 사실이나 이론, 원리에 대해 아는 것으로 참·거짓을 구별할 수 있는 문장으로 작성된다. 종류로는 ‘지구는 둥글다’와 같이 사실 혹은 현상을 기술하는 사실적 지식, ‘총각은 결혼하지 않은 성인 남자이다’

1 지식의 표현

01 인공지능과 추론

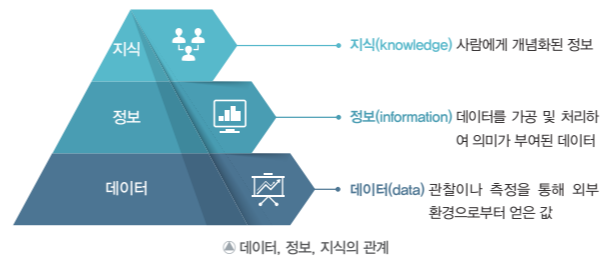
추론은 이미 알고 있는 지식을 토대로 새로운 결론을 이끌어 내는 사고 과정을 말한다. 즉, 알고 있는 사실로부터 모르고 있던 사실을 알게 되는 과정이라 할 수 있다. 예를 들어 A라는 사람의 자녀가 아들과 딸이 한 명씩 있고, 대학생과 고등학생이라는 것을 알고 있다. 그런데 아들이 고등학생이라는 사실을 알게 되었다. 그러면 A의 딸은 대학생이란 것을 추론할 수 있다.

추론은 인공지능에서도 중요한 영역이다. 인공지능은 데이터를 이용한 학습을 바탕으로 새로운 데이터로 추론을 하여 결과를 도출한다. 우리가 사용하는 스마트폰의 음성 인식, 이미지 검색, 이메일의 스팸 필터링 등에서 추론 기능이 사용되고 있다. 다양한 플랫폼의 추천 엔진 역시 추론이 이용된다.

지식 표현과 추론
추론을 하는 데 지식 표현 방식이 중요한 이유는 지식 표현 방식이 추론의 방식을 결정하기 때문이다.

02 데이터, 정보, 지식

데이터는 관찰이나 실험, 조사로 얻은 사실이나 자료를 의미하며, 가공하기 전의 상태이다. 반면, 정보는 다양한 데이터 중에서 사용자의 필요에 따라 처리한 데이터를 말한다. 지식은 개념화된 정보로 정보를 일반화하고 체계화한 것이라 할 수 있다. 예를 들어, 풍속, 풍향, 습도, 온도 등을 기록해 둔 것은 데이터이고, 이러한 데이터를 근거로 “내일 비가 올 확률은 80%”라고 예보하는 것은 정보다. 그리고 그동안의 정보에 대한 경험을 근거로 “비 올 확률이 80% 이상이면 우산을 소지하고 다니는 것이 좋다.”라고 하는 것은 지식이라 할 수 있다.



인간은 지식을 활용하여 다른 사람의 질문을 이해하고 질문에 답할 수 있다. 또한 다양한 상황과 목적에 맞게 지식을 적용할 수 있다. 인공지능도 문제를 해결하기 위해 지식이 필요하며, 인공지능에게 필요한 지식은 기계가 이해할 수 있는 형태여야 한다.

▶ 참고 동영상 인공지능과 추론

- 제목:** 데이터를 이용해 추론하는 인공지능 기계 학습 ‘추론을 통해 똑똑해지는 인공지능’
- 영상 내용:** 탐색, 추론, 학습, 연역적 추론과 귀납적 추론, 지도학습과 비지도학습에 대해 알아 본다.
- 주소:** <https://youtu.be/PTg-lqwZlh0?feature=shared>

지도 방법

- 인공지능에서 추론의 중요성을 이해할 수 있도록 설명한다. 또한 지식 표현과 추론의 관계를 설명한다.
- 데이터, 정보, 지식의 차이를 구분하고 관계를 이해할 수 있도록 지도한다.
- 데이터, 정보, 지식의 다양한 예를 들어 설명한다.

지도 예

- 데이터:** 홈페이지 이용자 수
- 정보:** 지난달, 어제, 오늘의 이용자 수, 월평균 이용자 수
- 지식:** 3월 신입생 영향으로 홈페이지 이용자 수의 급증하여 이때 학교에서 필요한 자료를 홈페이지에 등록하면 효과적이다.

와 같이 용어의 의미를 설명해 주거나, 문장을 구성하는 요소들의 의미상 관계를 나타내는 논리적 지식, ‘민주주의는 바람직한 사회 제도이다’와 같은 도덕적 판단, 가치 판단에 관한 규범적 지식이 있다.

4 지식 피라미드

지식 피라미드(DIKW)는 데이터(data), 정보(information), 지식(knowledge), 지혜(wisdom)의 4개 계층으로 이루어진 피라미드로, 영 단어의 첫 글자를 따서 ‘DIKW 피라미드’라고도 부른다.

- 데이터:** 사실, 값, 문자, 수치 등을 의미하며, 지식 피라미드에서 데이터가 가지는 중요한 특징은 ‘가공되지 않았다’는 사실이다.

예 수도권 날씨

날짜	최저 기온	최고 기온	습도	미세 먼지
8월 5일	28	34	60	보통
8월 6일	27	33	80	좋음
8월 7일	26	33	90	나쁨

- 정보:** 데이터 중에 사용자의 필요에 따라 정제되거나 가공된 데이터를 ‘정보’라고 부른다.
예 8월 8일 수도권의 강수확률은 80%이다.
- 지식:** 서로 관련이 있는 정보들의 패턴을 바탕으로 예측한 결과물을 ‘지식’이라고 볼 수 있다.
예 강수 확률이 80% 이상이면 우산을 가지고 다니는 것이 좋다.
- 지혜:** 패턴화된 지식을 바탕으로 올바른 의사 결정을 할 수 있는 능력을 뜻한다.
예 여름에는 비 올 확률이 80%가 넘는 날이 많으니, 작은 우산 하나는 가지고 다니는 것이 좋겠어.

지도 방법

- 지식 표현에 다양한 방법이 있음을 안내한다.
- 데이터, 정보, 지식의 차이를 이해하고 **해 보기 1** 을 풀어 볼 수 있도록 지도한다.

해 보기 1 지도 방법

- 예시로 주어진 데이터를 보고 정보와 지식을 구분하여 작성할 수 있도록 지도한다. 예시 이외의 다양한 정보와 지식을 작성할 수 있도록 지도한다.
- 정보: 겨울이 강수량이 적은 해가 많다.
 - 지식: 겨울에 강수량이 적어 산불의 위험이 크다.

03 지식 표현

인공지능을 활용하기 위해서는 사람과 컴퓨터가 동시에 이해할 수 있는 형태로 지식을 나타내는 것이 필요하고, 지식을 이러한 형태로 나타낸 것을 지식 표현이라고 한다. 문제 해결을 위해 적절한 지식 표현 방법을 선택하는 것은 매우 중요하며, 표현된 지식은 추론 과정을 통해 새로운 사실을 생성할 수 있다. 지식을 표현하는 방법에는 규칙, 논리, 의미망, 프레임 등이 있다.

해 보기 1 데이터, 정보, 지식 생성하기

• 다음에 제시된 데이터로부터 정보와 지식을 생성해 보자.

데이터	정보	지식															
● 컴퓨터 율품과 고기를 구매하기 위해 가까운 두 마트의 가격을 조사했다. <table border="1"> <thead> <tr> <th>물품</th> <th>A마트</th> <th>B마트</th> </tr> </thead> <tbody> <tr> <td>마우스</td> <td>15,000원</td> <td>18,000원</td> </tr> <tr> <td>키보드</td> <td>18,000원</td> <td>22,000원</td> </tr> <tr> <td>소고기</td> <td>40,000원</td> <td>36,000원</td> </tr> <tr> <td>돼지고기</td> <td>12,000원</td> <td>10,000원</td> </tr> </tbody> </table>	물품	A마트	B마트	마우스	15,000원	18,000원	키보드	18,000원	22,000원	소고기	40,000원	36,000원	돼지고기	12,000원	10,000원	● 마트마다 물건값이 서로 다르게 형성되어 있다. 전자 제품은 A마트가 저렴하고 고기류는 B마트가 저렴하다.	● 전자 제품을 사려면 A마트에 가고, 고기류를 사려면 B마트에 가면 더 저렴하게 물건을 구매할 수 있는 것을 알았다.
물품	A마트	B마트															
마우스	15,000원	18,000원															
키보드	18,000원	22,000원															
소고기	40,000원	36,000원															
돼지고기	12,000원	10,000원															

● 지난 몇 년간의 계절별 평균 강수량을 조사했다.

연도	봄	여름	가을	겨울
2016	277	479	124	87
2017	85	984	102	72
2018	401	559	268	40
2019	112	459	274	136
2020	146	1086	302	30
2021	418	484	250	18
2022	131	1211	410	63

(단위: mm)

● 추가 심화활동
계절별 기온, 겨울철 적설량, 폭염일 등을 조사하여 표로 작성해 보자.

연도	봄	여름	가을	겨울
2018	3	0	2	0
2019	0	0	3	1
2020	2	0	3	2
2021	12	0	0	0
2022	4	0	0	5
2023	15	0	0	0
2024	9	0	0	0

- ① 봄, 가을, 겨울에 비해 여름 강수량이 많고 겨울에 강수량이 적다.
- ② 봄과 가을 강수량은 비슷하다.
- ③ 여름에 비가 많이 와 비 피해가 발생하지 않도록 사전에 대비하여야 한다.
- ④ 여름에 비가 많이 와 비와 관련된 상품은 여름철에 맞춰 준비하는 것이 좋다.
- 봄에 황사가 많이 발생하고 여름에는 발생하지 않는다.
- 봄에는 황사에 대비하여 마스크를 착용하여야 한다.

5 지식 표현

지식 표현은 지식을 컴퓨터가 이해할 수 있는 형태로 변환하는 것이다. 지식 표현의 방법에는 규칙, 논리, 의미망, 프레임, 그래프, 객체 지향 표현 기법 등이 있다.

■ 지식 표현이 어려운 이유

- ① 실세계 대부분의 문제들은 지식으로 표현하기에는 너무 많은 방대한 양의 정보를 가진 경우가 많다.
- ② 미묘하고 애매모호한 지식을 정확하게 표현하기 어렵다.
- ③ 대부분의 지식은 정적인 경우보다 동적인 경우가 많아 상황에 따라 변한다.

■ 지식 표현 시 고려 사항

- ① **지식 표현의 정확성:** 객체* 간의 상관관계를 정확하게 표현
* 객체: 현실에 존재하거나 생각할 수 있는 것
- ② **추론의 정확성:** 추론이 정확하게 될 수 있도록 지식을 표현
- ③ **지식 획득 관리의 효율성:** 새로운 사실을 쉽게 삽입, 수정, 삭제가 가능하도록 지식을 표현
- ④ **추론의 효율성:** 추론 기법을 최적의 방향으로 이끌기 위해 사용되는 특정한 정보와 지식 구조를 결합시키는 능력

지도 방법

- 간단한 지식을 IF~THEN을 이용하여 표현해 볼 수 있도록 지도한다.
- 규칙 기반 표현 방법에 따라 지식을 표현하게 하고, 추론을 통해 새로운 지식을 생성해 보도록 지도한다. 이러한 과정을 통해 컴퓨터를 통한 지식의 표현과 처리가 어떤 절차와 원리로 이루어지는지 경험할 수 있도록 지도한다.

규칙을 이용한 지식 표현

규칙을 이용한 지식 표현은 가장 널리 알려진 지식 표현 방식의 하나로 생성 규칙이라고도 한다. IF~THEN 문장으로 연결된 형태를 취하며 IF 구문의 조건이 만족되거나 발생한 상태이면, THEN 구문이 수행되거나 논리적으로 참이 되는 형태이다. 규칙에서 나타나는 문장을 정교히 하면 논리에서의 술어 계산이 된다.

1 규칙

다양한 지식 표현 방법 중 대표적인 한 가지는 규칙을 이용한 방법이다. 우리가 흔히 사용하는 '만약 <A>라면, 이다.' 또는 '<A>하면, 하다.'와 같은 형태로, IF <A> THEN 로 나타낼 수 있다. <A>는 주어진 정보나 사실에 대응될 조건이고, 는 조건을 만족할 때 하는 판단이나 행동인 결론이다. A가 참이면, B를 수행하는 형태이고, 이를 생성 규칙이라고도 하며, 인과 관계, 추천, 전략, 경험적 지식 등을 표현할 수 있다.

지식 표현이 어려운 이유

- 지식으로 표현하기에 방대한 양의 정보를 가진 경우가 많다.
- 애매모호한 지식을 정확히 표현하기 어렵다.
- 지식은 상황에 따라 변하는 경우가 많다.

구분	지식	지식 표현(IF~THEN)
인과 관계	방새고 공부하면, 다음날 피곤하다.	IF 원인 THEN 결과 IF 방새고 공부했다. (원인) THEN 다음날 피곤하다. (결과)
추천	기온이 영하로 내려가면, 방한이 잘 되는 옷을 입어라.	IF 상황 THEN 추천 내용 IF 기온이 영하로 내려갔다. (상황) THEN 방한이 잘 되는 옷을 입어라. (추천 내용)
전략	차가 시동이 걸리지 않으면, 배터리를 확인한다. 배터리에 이상이 없으면, 연료를 확인한다.	일련의 규칙들로 표현 IF 차가 시동이 걸리지 않는다. THEN 배터리를 확인한다. AND 배터리를 확인을 종료한다. IF 배터리 확인을 종료했다. AND 배터리는 충분하다. THEN 연료를 확인한다. AND 연료 확인을 종료한다.
경험적 지식	액체가 투명하고 100도에서 끓으며, 냄새가 없으면, 액체는 물이다.	경험적 지식을 표현 IF 액체는 투명하다. AND 액체는 100도에서 끓는다. AND 액체는 냄새가 없다. THEN 액체는 물이다.

▲ [표 1-4] 규칙을 이용한 지식 표현

1.1 조건이나 결론이 여러 개인 경우의 지식 표현

조건이 여러 개인 경우		결론이 여러 개인 경우	
형식	예	형식	예
IF (A1) AND (A2) OR (A2) THEN (B)	IF 송곳니가 있다. AND 포유류이다. THEN 육식동물이다.	IF (A) THEN (B1) AND (B2)	IF 물건이 부족하다. THEN 물가가 오른다. AND 경제가 어려워진다.

AND와 OR는 섞어서 사용할 수 있다.

해 보기 2 지식을 IF~THEN으로 표현해 보기

• 내가 알고 있는 지식을 IF~THEN의 형태로 표현해 보자.

내가 알고 있는 지식	지식 표현(IF~THEN)
● 봄이 되면 제비가 돌아온다.	IF 봄이 된다. THEN 제비가 돌아온다.

6 규칙을 이용한 지식 표현의 또 다른 예

상황/행동, 전제/결론, 증거/가설, 원인/결과 등의 경우도 IF~THEN으로 표현할 수 있다.

구분	지식	지식 표현(IF~THEN)
상황/행동	불이 나면 소방서에 신고해라.	IF 불이 났다. THEN 소방서에 신고해라.
전제/결론	송곳니가 있는 포유류이면 육식 동물이다.	IF 송곳니가 있는 포유류이다. THEN 육식 동물이다.
증거/가설	지금까지 아침 해가 떴으면, 내일도 아침 해가 뜬다.	IF 지금까지 아침 해가 떴다. THEN 내일도 아침 해가 뜬다.
원인/결과	밤을 새워 가면서 공부하면, 다음 날 피곤하다.	IF 밤을 새워 가면서 공부했다. THEN 다음 날 피곤하다.

해 보기 2 해설 및 예시 답안

• 내가 알고 있는 지식을 IF~THEN의 형태로 표현해 보자.

내가 알고 있는 지식	지식 표현(IF~THEN)
태풍이 오면 비가 많이 온다.	IF 태풍이 온다. THEN 비가 많이 온다.
동물에 깃털이 있으면 새이다.	IF 동물에 깃털이 있다. THEN 새이다.
보행자 신호등이 초록색 불이면 건널목을 건넌다.	IF 보행자 신호등이 초록색 불이다. THEN 건널목을 건넌다.

지도 방법

- 논리와 의미망, 프레임에 이용하여 지식 표현을 할 수 있도록 지도한다.
- 교과서 이외의 예시를 제시하고 이를 논리와 의미망, 프레임으로 표현해 볼 수 있도록 지도한다.
- 학생이 지식을 지식 표현 방법에 따라 표현해 볼 수 있도록 지도한다.
- 의미망에서 각 각의 호에 대한 의미를 작성해 볼 수 있도록 지도한다.

논리

- 1 명제 논리: 주어와 술어를 구분하지 않고, 전체를 하나의 명제로 처리하여 '참' 또는 '거짓'을 판별하는 논리
- 2 술어 논리: 하나의 명제를 술어와 그 술어의 수식을 받는 객체로 분리하여 '술어'의 형태로 표현

의미망

- 방향 그래프를 활용하여 개념 간의 관계를 표현한 지식 표현 방법이다.
- 인과관계를 표현하기에 좋지만, 관계가 많아지면 유지 보수가 복잡해진다.
- 표준 지침이 없어 보편화된 표현 방법이 없다.

58

2 논리

논리는 수학 논리학에서 사용하는 명제 논리나 술어 논리를 사용한다. 명제 논리는 명제를 가지고 논리를 판별하는 것이고, 술어 논리는 주어와 술어 간의 관계를 다루는 것이다. 명제 논리에서 나타나는 단점은 술어 논리로 보완될 수 있다. '개는 동물이다'라는 평서문을 술어 논리로 표현하면, 'animal(개)'처럼 표현할 수 있다. 가장 명확한 방법이지만 복잡한 상황에서의 적용이 어렵다는 특징이 있다.

평서문	술어 논리 표현
개는 동물이다.	animal(개)
개는 사람을 좋아한다.	like(개, 사람)

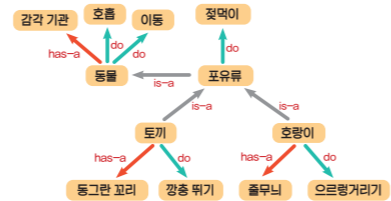
▲ [표 1-5] 논리 표현의 예

호는 에지(edge)나 링크(link)로 표현되기도 한다.

3 의미망

의미망(semantic net)은 지식이나 기억, 실세계를 노드(node)와 호(arc)를 이용하여 네트워크 구조로 표현한 것이다. 노드는 객체, 개념, 사건 등을 표현하고, 호는 노드 간의 관계를 화살표로 나타낸다.

아래 의미망을 보면 노드는 동물, 포유류, 토끼, 호랑이, 감각 기관, 호흡, 이동 등이 있고, 호는 has-a(~이 있다), is-a(~이다), do(~을 한다)가 있다. 동물과 감각 기관이 has-a로 연결된 것은 '동물은 감각 기관이 있다'를 의미하고, 포유류와 동물이 is-a로 연결된 것은 '포유류는 동물이다'를 의미한다.



▲ [그림 1-18] 의미망 표현의 예

[출처] http://ai4school.org/?page_id=2522



알고 가기 지식 베이스와 전문가 시스템

지식 베이스는 전문가 시스템의 구성 요소 중 하나로 특정 분야 전문가의 지식이나 문제 해결에 필요한 사실, 규칙 등이 저장되어 있는 데이터베이스다. 전문 지식을 표현하기 위해서 IF~THEN 형식의 생성 규칙을 사용한다. 전문가 시스템은 인간의 지적 활동과 경험을 통해 축적된 전문가의 지식과 전문가에 의해 정의된 추론 규칙을 활용하여 결정을 내리거나 문제를 해결하는 시스템이다. 필수 구성 요소로 지식 베이스와 추론 엔진이 포함되어 있어야 한다.

[출처] 한국정보통신기술협회의 정보통신 용어사전

7 명제 논리 vs 술어 논리

- 1 명제 논리에서의 추론: 가지고 있는 지식과 알고 있는 사실로부터 새로운 사실을 유추하는 것이다.

- 예 • 지식: 우리 집 고양이는 안방에 있거나 캣타워에 있다.
- 사실: 고양이가 안방에 없다.
- 추론: 고양이는 캣타워에 있을 것이다.

2 술어 논리에서의 추론

- 술어 논리를 명제 논리로 변환한 뒤 명제 논리의 추론 기법을 적용한다.
- 어느 문장으로부터 다른 문장을 이끌어 내는 도출로 추론한다. 도출은 리터럴과 부정 리터럴을 가지고 있는 2개의 절을 조합하여 새로운 절을 생성하는 방법이다.

8 지식과 전문가 시스템

지식과 추론 기능이 합쳐서 전문가 시스템이 만들어지고 할 만큼 지식이란 전문가 시스템의 가장 중요한 요소이다. 획득한 지식을 어떻게 효율적이면서 효과적으로 표현·저장하는가 하는 것은 전문가 시스템의 성능과 직결되는 문제이다. 따라서 이러한 지식이 과연 무엇인지 우선적으로 규명할 필요가 있다.

9 프레임의 구성 요소

- 1 슬롯 • 객체의 속성을 기술하는 것
- 슬롯 이름과 슬롯값으로 구성
- 2 패킷 • '측면' 또는 '양상'을 의미

앞의 의미망이 나타내는 지식은 다음 표와 같다.

호(arc)	지식	호(arc)	지식
is-a	포유류는 동물이다.	do	포유류는 컷을 먹인다.
	토끼는 포유류다.		동물은 호흡을 한다.
	호랑이는 포유류다.		동물은 이동을 한다.
has-a	동물은 감각 기관이 있다.		토끼는 강종 뛰기를 한다.
	토끼는 동그란 꼬리가 있다.		호랑이는 오르막거리기를 한다.
	호랑이는 줄무늬가 있다.		

▲ [표 1-6] 의미망이 나타내는 지식

4 프레임

프레임(frame)은 특정한 객체나 개념에 대한 지식을 슬롯(slot)으로 표현한 것이다. 하나의 프레임은 여러 개의 슬롯으로 구성되고, 슬롯은 슬롯 이름과 슬롯 값으로 구성된다. 제목이 '인공지능'인 책을 예로 들어 살펴보면, 책에는 작가가 있고, 출판 연도가 있다. 또한 책의 분량을 페이지로 나타낼 수 있다. 프레임은 관련된 지식을 한군데 모을 수 있다는 것과 표현력이 우수하다는 장점이 있지만, 슬롯에 대한 표준이 없고 지식을 생성하는 것이 매우 힘들다는 단점이 있다.

슬롯 이름	슬롯 값
제목	인공지능
작가	홍길동
출판 연도	2023
페이지	500

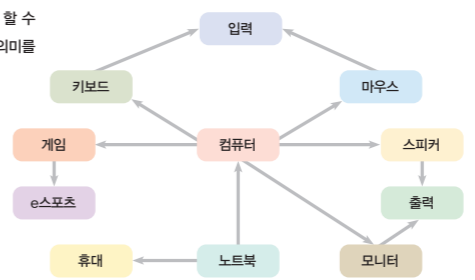
▲ [표 1-7] 책을 프레임으로 표현한 예



해 보기 3 의미망을 이용하여 지식 표현하기

표현하기

- '~은(는) ~이(가) 있다', '~은(는) ~을(를) 할 수 있다', '~은(는) ~이다'를 이용하여 호의 의미를 작성해 보자.
- 마우스는 입력을 할 수 있다.



59

프레임

- 마빈 민스키(Marvin Minsky)가 제안한 방식이다.
- 특정 객체 또는 개념에 대한 전형적인 지식을 슬롯(slot)의 집합으로 표현한 것이다.
- '객체(object)'의 인공지능식 표현으로 특정 객체에 속하는 속성을 표현하는 방법이다.



해 보기 3 지도 방법

각각의 호에 '~은(는) ~이(가) 있다', '~은(는) ~을(를) 할 수 있다', '~은(는) ~이다' 중 알맞은 것을 먼저 작성하도록 한 뒤, 노드에 있는 용어와 연결하여 지식을 작성할 수 있도록 지도한다.

예시 답안

- 컴퓨터는 키보드가 있다.
- 키보드는 입력을 할 수 있다.
- 컴퓨터는 마우스가 있다.
- 마우스는 입력을 할 수 있다.
- 컴퓨터는 스피커가 있다.
- 스피커는 출력을 할 수 있다.
- 컴퓨터는 모니터가 있다.
- 모니터는 출력을 할 수 있다.
- 컴퓨터는 게임을 할 수 있다.
- 게임은 e스포츠이다.
- 노트북은 컴퓨터이다.
- 노트북은 휴대를 할 수 있다.

지도 방법

인공지능을 활용한 문제 해결 사례를 다양하게 제시해 준다. 단, 인공지능으로 모든 문제를 해결할 수 있다는 그릇된 인식이 생성되지 않도록 주의한다.

* 용어 해설

- **추론, 전제, 결론:** 이미 알고 있는 것으로부터 결론을 도출하는 행위 또는 과정이다. 엄밀하게 말하면 특정한 명제에서 다른 명제를 이끌어 내는 사고 과정을 말한다. 여기서 추론을 시작하는 명제를 전제, 추론을 통해 이끌어 내는 명제를 결론이라고 한다. 즉 추론이란 이미 알고 있는 지식인 전제에서 새로운 지식인 결론을 이끌어 내는 과정으로, 이는 인류의 지식량을 획기적으로 늘릴 수 있는 방법 중의 하나로 널리 이용되어 왔다.

60

2 추론

인공지능에서는 주로 연역 추론과 귀납 추론 중 하나의 방식을 사용하거나 이 둘을 조합하여 추론한다.

01 연역 추론

연역 추론이란 일반적인 전제를 바탕으로 새로운 결론을 논리적으로 도출하는 추론 방법이다. 대표적인 방법으로 삼단 논법이 있다. 삼단 논법은 두 개의 전제와 하나의 결론으로 된 형식을 가지고 있다. 최초의 대전제가 결론을 이끌어 내는 중요한 근거가 되며 논리의 일관성이 있다.

- 대전제: 모든 인간은 죽는다.
- 소전제: 소크라테스는 인간이다.
- 결론: 그러므로 소크라테스는 죽는다.

▲ 연역 추론의 예(삼단 논법)

- 소크라테스는 죽었다.
- 공자는 죽었다.
- 맹자는 죽었다.
- 세종대왕도 죽었다.
- 소크라테스, 공자, 맹자, 세종대왕은 모두 인간이다. 그러므로 모든 인간은 죽는다.

▲ 귀납 추론의 예

02 귀납 추론

귀납 추론이란 구체적인 사례에서 일반적인 결론을 이끌어 내는 추론으로 이미 알고 있는 사실로부터 모르고 있던 사실을 유추해 내는 것이다. 그렇기 때문에 사례의 다양성이나 정확성이 결론에 큰 영향을 끼친다.

03 추론의 오류

연역 추론이나 귀납 추론으로 언제나 옳은 결론에 도달하는 것은 아니다. 연역 추론에서 오류는 전제 조건을 생략하거나 잘못 설정했을 때 발생할 수 있고, 귀납 추론에서 오류는 관찰된 표본이 전체를 대표할 수 있는 표본이 아닌 경우나 전체에 대한 표본의 해석이 잘못되는 경우에 발생할 수 있다.

연역 추론의 오류

- 대전제: 네 발이 있으면 포유류다.
- 소전제: 거북이는 네 발이 있다.
- 결론: 거북이는 포유류다.

• 오류: '네 발이 있으면 포유류다'라는 대전제가 오류다. 네 발이 있어도 포유류가 아닌 경우는 많다.

귀납 추론의 오류

- 사례1: 백조1은 희다.
- 사례2: 백조2는 희다.
- 사례3: 백조3은 희다.
- 결론: 그러므로 모든 백조는 희다.

• 오류: 모든 백조를 본 것이 아니다. 실제 오스트레일리아에 검은 백조가 살고 있다.

04 추론을 통한 지식 생성

알고 있는 사실을 바탕으로 추론을 통해 새로운 지식을 생성할 수 있다. 주어진 상황과 조건을 보고 추론을 통해 새로운 지식을 생성하는 과정을 살펴보자.

인스턴스 표현

frame-name	Ultra-Slim-Notebook	
frame-type	instance(class Computer)	
CPU	value	ARM
OS	value	Android
memory	value	4G
warranty	value	3years
HDD	value	512GB
price	if-needed	look-up-the-list
stock	if-needed	ask-for-vendor

HDD	value	512GB
price	if-needed	look-up-the-list
stock	if-needed	ask-for-vendor

데몬 실행 조건 데몬 이름

- if-needed: 슬롯값을 사용할 때
- if-added: 슬롯값을 추가할 때
- if-removed: 삭제할 때
- if-modified: 수정할 때

```
(frame
  (frame-name Ultra-Slim_Notebook)
  (frame-type instance(class Computer)
    (CPU(value ARM)
      (OS(value Android)
        (memory(value 4G))
          (HDD(value 512GB))
            (price(if-needed look-up-the-list))
              (stock(if-needed ask-for-vendor))))))
```

- 클래스(부류에 대한 표현), 인스턴스(객체에 대한 표현) 상하위 프레임이 있는데 클래스로부터 상속을 받은 것이 인스턴스 프레임이다.

출처 <https://anjaekwang.github.io/KnowledgeExpressionAndInference/>

예제 아인슈타인 문제

다음 문제 상황과 조건을 확인하고, 물을 마시는 사람과 얼룩말을 키우는 사람은 어느 나라 사람인지 맞춰 보자.

문제 상황 집의 색, 국적, 마시는 음료, 먹는 음식, 키우는 동물이 모두 다르다.

조건

- ① 모두 다섯 채의 집이 있다.
- ② 호주인은 빨간색 집에 산다.
- ③ 이탈리아인은 개를 기른다.
- ④ 초록색 집에 사는 사람은 커피를 마신다.
- ⑤ 우크라이나인은 차를 마신다.
- ⑥ 초록색 집은 아이보리색 집의 바로 왼쪽이다.
- ⑦ 버섯을 먹는 사람은 달팽이를 기른다.
- ⑧ 노란색 집에 사는 사람은 사과를 먹는다.
- ⑨ 한가운데 있는 집에 사는 사람은 우유를 마신다.
- ⑩ 노르웨이인은 왼쪽에서 첫 번째 집에 산다.
- ⑪ 양파를 먹는 사람은 여우를 기르는 사람 옆 집에 산다.
- ⑫ 사과를 먹는 사람은 말을 기르는 사람 옆집에 산다.
- ⑬ 케이크를 먹는 사람은 오렌지 주스를 마신다.
- ⑭ 일본인은 바나나를 먹는다.
- ⑮ 노르웨이인은 파란색 집 옆집에 산다.

풀이

- ① 조건 10번과 15번을 보면 노르웨이인은 첫 번째 집에 살고, 오른쪽 옆집은 파란색이다. 또한 조건 9번을 통해 3번 집은 우유를 마신다는 것을 알 수 있다.

생성된 지식 16. 두 번째 집은 파란색이다.

구분	집1	집2	집3	집4	집5
색		파란색			
국적	노르웨이				
음료			우유		
음식					
동물					

- ② 조건 6번을 보면 초록색 집이 아이보리색 왼쪽 집이라고 하였다. 이 경우 초록색이 가능한 경우는 세 번째 집과 네 번째 집이지만, 조건 4번을 통해 초록색 집은 네 번째 집이고, 다섯 번째 집이 아이보리색 집인 것을 알 수 있다.

생성된 지식 17. 네 번째 집은 초록색이다. 18. 다섯 번째 집은 아이보리색 집이다.

구분	집1	집2	집3	집4	집5
색		파란색		초록색	아이보리색
국적	노르웨이				
음료			우유	커피	
음식					
동물					

61

지도 방법

- 조건을 확인한 후 주어진 사실을 바탕으로 새로운 사실을 하나씩 유추해 낼 수 있도록 지도한다.
- 문제를 해결한 후, 해결 과정을 설명할 수 있도록 지도한다.
- 추측을 통한 해결이 아니라 새로운 사실 도출을 통해 문제를 해결할 수 있도록 한다.
- 제시된 조건의 번호에 상관없이 새로운 사실을 찾을 수 있는 조건부터 확인해서 문제를 풀 수 있도록 안내한다.
- 문제를 해결하는 과정을 통해 지식 표현에 따른 추론 과정 및 원리를 이해할 수 있도록 지도한다.

1단계

- ① 조건 10번: 노르웨이인은 왼쪽 첫 번째 집에 산다.
- ② 조건 15번: 노르웨이인은 파란 집 옆집에 산다. → 두 번째 집은 파란색이다. 두 조건에서 노르웨이인이 사는 집은 다섯 집 중, 왼쪽에서 첫 번째이기 때문에, 옆집은 왼쪽이 아니라 오른쪽에 있는 집이 되고, 그러므로 두 번째 집이 파란색인 것을 알 수 있다.
- ③ 조건 9번: 한가운데 있는 집에 사는 사람은 우유를 마신다. → 다섯 집 중에서 한 가운데는 세 번째이다.

2단계

- ① 조건 6번: 초록색 집은 아이보리색 집의 바로 왼쪽이다. → 두 번째 집이 파란색이므로, 초록색 집은 세 번째나 네 번째만 가능하다. 초록색 집의 오른쪽이 아이보리색인데 첫 번째 오른쪽은 파란색이고, 다섯 번째 오른쪽은 없기 때문이다.
- ① 조건 4번: 초록색 집에 사는 사람은 커피를 마신다. → 초록색 집은 네 번째 집인 것을 알 수 있다. 세 번째 집은 우유를 마시기 때문이다. 이를 통해 다섯 번째 집이 아이보리색인 것을 알 수 있다.

- 규칙과 다른 경우가 발견되면 결론을 사용하지 못한다.

12 기호주의

- 인간의 지식을 기호화하여 다루는 것은 인공지능에서 '기호주의'의 기원이 된다.

10 연역 추론

- '지식 기반 인공지능' 또는 '기호 기반 인공지능'이라고 부르는 인공지능 구현 방법이다.
- 지식을 명제로 만들고 명제들 사이의 연역적 추론을 통해 새로운 지식이나 사실을 만들어 내는 방식이다.
- 세상의 모든 것을 명제로 만들어야 하는 문제가 있다.
- 명제가 틀릴 수 있어서 정의한 명제를 증명하여야 한다.

11 귀납 추론

- 관찰 및 탐색으로 규칙을 발견하여 결론을 도출하는 추론 방식이다.
- 경험에 의존하는 방식으로 경험하지 않은 경우의 고려가 불가능하다.

3단계

1 조건 2번: 호주인은 빨간색 집에 산다. → 첫 번째 집과 세 번째 집의 색을 모르는데 첫 번째 집은 노르웨이인이 살기 때문에 세 번째 집이 호주인이 사는 집이고 빨간색인 것을 알 수 있다. 첫 번째 집은 마지막으로 하나 남은 노란색이다.

4단계

1 조건 8번: 노란색 집에 사는 사람은 사과를 먹는다. → 첫 번째 집에서 사과를 먹는다.
2 조건 12번: 사과를 먹는 사람은 말을 기르는 사람 옆집에 산다. → 사과를 먹는 사람은 첫 번째 집이고, 첫 번째 집은 왼쪽 옆집이 없으므로, 오른쪽인 두 번째 집이 말을 기르는 집이다.

5단계

1 조건 5번: 우크라이나인은 차를 마신다. → 세 번째 집은 우크라이나인이 사는 집이 아니다. 두 번째나 다섯 번째 집이다.
2 조건 13번: 케이크를 먹는 사람은 오렌지 주스를 마신다. → 첫 번째, 세 번째, 네 번째 집은 아니다. 두 번째나 다섯 번째 집이다. 두 번째 집이 차를 마시면, 다섯 번째 집은 오렌지 주스를 마시고 케이크를 먹으며, 다섯 번째 집이 차를 마시면 두 번째 집이 오렌지 주스를 마시고 케이크를 먹는다.
2 조건 14번: 일본인은 바나나를 먹는다. → 국적과 음식을 함께 입력할 수 있는 집은 네 번째 집만 가능하다.

3 조건 2번의 호주인은 빨간색 집에 산다고 되어 있는데 아직 색을 모르는 집은 1번과 3번 집이고, 1번 집은 노르웨이인이 살기 때문에 3번 집에 호주인이 사는 것을 알 수 있다. 따라서, 3번 집은 빨간색이고 1번 집은 마지막 남은 색인 노란색인 것을 알 수 있다.

생성된 지식 19. 세 번째 집은 빨간색이고, 호주인이 산다. 20. 첫 번째 집은 노란색이다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이		호주		
음료			우유	커피	
음식					
동물					

4 조건 8번을 보면 1번 집이 사과를 먹는 것을 알 수 있고 조건 12번을 통해 1번 집의 옆집인 2번 집이 말을 기르는 것을 알 수 있다.

생성된 지식 21. 두 번째 집은 말을 기른다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이		호주		
음료			우유	커피	
음식	사과				
동물		말			

5 조건 5번과 13번은 모두 2번 집과 5번 집만 입력이 가능하다. 따라서 조건 14번에서 제시한 조건이 가능한 집은 4번 집이다.

생성된 지식 22. 네 번째 집에 일본인이 산다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이		호주	일본	
음료			우유	커피	
음식	사과			바나나	
동물		말			

- 기초가 되는 지식을 바탕으로 모든 지식을 기호화하여 새로운 문제를 풀기 위해 이미 존재하는 지식을 학습 과정에서 사용하는 방법이다.
- MYCIN(Mycode in Inhibition): 1960년대 후반과 1970년대에 카네기 멜론 대학교에서 개발된 기호주의 인공지능을 기반으로 한 의료 처방 전문가 시스템이다. 이 시스템은 박테리아 감염의 진단과 치료에 사용되었다. 500개의 규칙을 사용하여 당시 의료 전문가 수준의 의료 진단 기능을 수행하였다.

13 연결주의

- 인간의 두뇌에서 어떻게 생각하는지 모방하여 컴퓨터를 학습시키는 방법이다.

- 하나의 개념이 여러 곳에 흩어져서 표현된다는 인지 과학 분야의 접근법이다.

추가 예제 세 사람의 취미 알아맞히기

➔ A, B, C의 취미는 뜨개질, 독서, 마라톤 중 서로 다른 하나이다. 다음 <조건>을 보고 세 사람의 취미는 각각 무엇인지 알아보자.

조건

1. A, B, C의 취미는 뜨개질, 독서, 마라톤 중 서로 다른 하나이다.
2. 좋아하는 색은 빨간색 또는 파란색이다.
3. A의 취미는 몸 전체를 움직이는 활동이다.
4. B는 빨간색을 좋아한다.
5. C는 파란색을 좋아한다.
6. A, B, C 중 빨간색을 좋아하는 사람은 손으로 복잡한 일을 하는 것을 싫어한다.

6 3번 조건의 이탈리아인은 개를 기른다고 되어 있는데 국적이 비어 있는 것은 2번, 5번 집이지만, 2번 집은 말을 기르기 때문에 이탈리아인은 5번 집에 사는 것을 알 수 있다.

생성된 지식 23. 다섯 번째 집에 이탈리아인이 산다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이		호주	일본	이탈리아
음료			우유	커피	
음식	사과			바나나	
동물		말			개

7 다시 5번 조건을 통해 2번 집에 우크라이나인이 사는 것을 알 수 있고, 이를 통해 13번 조건에 해당하는 집은 5번 집이다.

생성된 지식 24. 두 번째 집에 우크라이나인이 산다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이	우크라이나	호주	일본	이탈리아
음료		차	우유	커피	주스
음식	사과			바나나	케이크
동물		말			개

8 1번 집의 음료가 비어 있는 것을 통해 1번 집의 음료는 물인 것을 알 수 있다.

생성된 지식 25. 첫 번째 집은 물을 마신다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이	우크라이나	호주	일본	이탈리아
음료	물	차	우유	커피	주스
음식	사과			바나나	케이크
동물		말			개

<조건>을 표로 나타내면 아래와 같다.

구분	취미	좋아하는 색
A		
B		빨간색
C		파란색

이 사실을 바탕으로 A, B, C의 취미를 추론하는 과정을 단계별로 살펴보면 다음과 같다.

1단계

1번과 3번을 통해 A의 취미는 마라톤인 것을 알 수 있다.

구분	취미	좋아하는 색
A	마라톤	
B		빨간색
C		파란색

사실

1. A, B, C의 취미는 뜨개질, 독서, 마라톤 중 서로 다른 하나이다.
2. 좋아하는 색은 빨간색 또는 파란색이다.
3. A의 취미는 몸 전체를 움직이는 활동이다.
4. B는 빨간색을 좋아한다.
5. C는 파란색을 좋아한다.
6. A, B, C 중 빨간색을 좋아하는 사람은 손으로 복잡한 일을 하는 것을 싫어한다.
7. A의 취미는 마라톤이다.

9단계

1 조건 7번: 버섯을 먹는 사람은 달팽이를 기른다. → 두 번째 집은 말을 기르기 때문에 세 번째 집이 버섯을 먹고 달팽이를 기른다.

10단계

1 두 번째 집의 음식이 비어 있어 두 번째 집이 양파를 먹는 것을 알 수 있다.
2 조건 11번: 양파를 먹는 사람은 여우를 기르는 사람 옆집에 산다. → 두 번째 집이 양파를 먹어 첫 번째 집이나 세 번째 집이 여우를 기르는데, 세 번째 집은 달팽이를 기르기 때문에 첫 번째 집이 여우를 기르는 것을 알 수 있다.

11단계

네 번째 집의 동물이 비어 있다. 따라서 얼룩말을 기르는 집은 네 번째 집이다.

7번 조건에서 버섯을 먹는 사람은 달팽이를 기르는데, 2번 집은 말을 키우기 때문에 3번 집이 버섯을 먹는 것을 알 수 있다.

생성된 지식 26. 세 번째 집은 버섯을 먹는다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이	우크라이나	호주	일본	이탈리아
음료	물	차	우유	커피	주스
음식	사과		버섯	바나나	케이크
동물		말	달팽이		개

음식이 비어 있는 2번 집은 양파를 먹고, 11번 조건을 통해 양파를 먹는 사람의 옆집인 1번 집이 여우를 기르는 것을 알 수 있다.

생성된 지식 27. 두 번째 집은 양파를 먹는다. 28. 첫 번째 집은 여우를 기른다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이	우크라이나	호주	일본	이탈리아
음료	물	차	우유	커피	주스
음식	사과	양파	버섯	바나나	케이크
동물	여우	말	달팽이		개

마지막으로 4번 집의 동물만 비어 있다. 이를 통해 4번 집은 얼룩말을 키우는 것을 알 수 있다.

생성된 지식 29. 네 번째 집은 얼룩말을 키운다.

구분	집1	집2	집3	집4	집5
색	노란색	파란색	빨간색	초록색	아이보리색
국적	노르웨이	우크라이나	호주	일본	이탈리아
음료	물	차	우유	커피	주스
음식	사과	양파	버섯	바나나	케이크
동물	여우	말	달팽이	얼룩말	개

소단원 1번 요약

- 1 데이터를 가공하면 정보, 정보를 가공하면 지식이 된다.
- 2 IF~THEN의 형태로 규칙을 이용하여 지식을 표현할 수 있다.
- 3 알고 있는 사실을 바탕으로 추론하여 새로운 지식을 생성할 수 있다. 추론에는 연역 추론과 귀납 추론이 있다.



기관사의 성은 무엇일까?

다음 문제 상황과 조건을 읽고, 직원의 직업과 승객이 사는 지역, 기관사의 성을 추론해 보자.

문제 상황 어떤 기차에 남궁, 독고, 제갈의 성(姓)을 가진 기관사, 승무원, 엔지니어가 타고 있다. 성의 순서대로 해당 직업을 가진 것인지는 알 수 없다. 어느 날 이들과 똑같은 성을 가진 승객 3명이 기차에 탑승했다. 성 앞에 승객을 붙여 같은 성을 구분했다.

- 조건
- 1 승객 제갈 씨는 부산에 산다.
 - 2 엔지니어는 부산과 대전의 중간 지역에 산다.
 - 3 승객 독고 씨는 연봉 2,000만 원을 번다.
 - 4 남궁 씨는 당구로 승무원을 이긴다.
 - 5 승객 중 한 명은 엔지니어의 이웃이고, 엔지니어 연봉의 정확히 3배가 되는 연봉을 번다.
 - 6 대전에 사는 승객은 엔지니어와 이름이 같다.

[1단계] 조건 1번과 4번을 통해 승객 제갈 씨가 부산에 산다는 것과 직원 남궁 씨가 승무원이 아니라는 것을 알 수 있다.

직원	기관사	승무원	엔지니어	승객	대전	중간 지역	부산
남궁 씨		X		남궁 씨			X
독고 씨				독고 씨			X
제갈 씨				제갈 씨			○

[2단계] 조건 5번과 3번을 통해 엔지니어의 이웃이 승객 독고 씨는 아닌 것을 알 수 있다. 2,000만 원은 3으로 나누어서 떨어지지 않기 때문이다. 또한, 2번을 통해 승객은 승객 독고 씨는 대전에 사는 것을 알 수 있다. 따라서 나머지 승객인 남궁 씨가 중간에 사는 것을 알 수 있다.

직원	기관사	승무원	엔지니어	직원	대전	중간	부산
남궁		X		남궁	X	○	X
독고				독고	○	X	X
제갈				제갈	X	X	○

[3단계] 승객 독고 씨가 대전에 사는 것과 6번 조건을 통해 직원 독고 씨의 직업은 엔지니어인 것을 알 수 있다.

직원	기관사	승무원	엔지니어	직원	대전	중간	부산
남궁		X	X	남궁	X	○	X
독고	X	X	○	독고	○	X	X
제갈			X	제갈	X	X	○

[4단계] 독고 씨는 엔지니어이고, 남궁 씨는 승무원이 아니기 때문에 승무원은 제갈 씨이고, 기관사는 남궁 씨인 것을 알 수 있다.

직원	기관사	승무원	엔지니어	직원	대전	중간	부산
남궁	○	X	X	남궁	X	○	X
독고	X	X	○	독고	○	X	X
제갈	X	○	X	제갈	X	X	○

따라서 최종적으로 기관사의 성은 남궁인 것을 알 수 있다.



제시된 조건을 확인하고 주어진 사실을 바탕으로 새로운 사실을 도출하며 문제를 풀 수 있도록 지도한다.

사실을 바탕으로 한 단계씩 표의 빈칸을 채우며 문제를 풀면 어렵지 않게 풀 수 있음을 안내한다.

TIP 추론 관련 활동으로 제시된 사실을 바탕으로 새로운 사실을 추론하며 최종 질문에 대한 답을 도출하는 문제이다. 머릿속으로 생각만 하였을 때, 헛갈리는 것을 방지하기 위해 표를 그리고 사실에 해당하는 부분을 표에 작성하면 쉽게 문제를 해결할 수 있다.

2단계

4번과 6번을 통해 B의 취미는 뜨개질이 아닌 것을 알 수 있다. 마라톤은 A의 취미이기 때문에 B의 취미는 독서이다.

구분	취미	좋아하는 색
A	마라톤	
B	독서	빨간색
C		파란색

사실

1. A, B, C의 취미는 뜨개질, 독서, 마라톤 중 서로 다른 하나이다.
2. 좋아하는 색은 빨간색 또는 파란색이다.

3. A의 취미는 몸 전체를 움직이는 활동이다.
4. B는 빨간색을 좋아한다.
5. C는 파란색을 좋아한다.
6. A, B, C 중 빨간색을 좋아하는 사람은 손으로 복잡한 일을 하는 것을 싫어한다.
7. A의 취미는 마라톤이다.
8. B의 취미는 독서이다.

3단계

7번과 8번을 통해 C의 취미는 뜨개질인 것을 알 수 있다.

구분	취미	좋아하는 색
A	마라톤	
B	독서	빨간색
C	뜨개질	파란색

사실

1. A, B, C의 취미는 뜨개질, 독서, 마라톤 중 서로 다른 하나이다.
2. 좋아하는 색은 빨간색 또는 파란색이다.
3. A의 취미는 몸 전체를 움직이는 활동이다.
4. B는 빨간색을 좋아한다.

5. C는 파란색을 좋아한다.
6. A, B, C 중 빨간색을 좋아하는 사람은 손으로 복잡한 일을 하는 것을 싫어한다.
7. A의 취미는 마라톤이다.
8. B의 취미는 독서이다.
9. C의 취미는 뜨개질이다.

추론을 통해 세 사람의 취미를 알 수 있다.

추론 결과

- A의 취미는 마라톤이다.
B의 취미는 독서이다.
C의 취미는 뜨개질이다.



오픈AI·메타, 사람처럼 '추론하는 AI' 내놓는다 GPT5·라마3 출시 예고(기사)

인간과 유사한 사고를 하도록 진화한 인공지능(AI)이 이르면 올해부터 현실화될 전망이다. AI 기술을 선도하는 오픈AI와 메타가 앞다퉀 '추론과 계획(reasoning and planning)'을 수행할 수 있는 차세대 AI 모델 출시를 예고하고 나섰다. 현재 AI는 기존에 학습한 데이터 가운데서 최적의 답을 찾아 제시하지만, 앞으로는 학습하지 않은 질문에도 기존 데이터를 이용한 추론을 통해 적합한 답을 내놓는 식이다.

9일(현지 시각) 페이스북 등을 운영하는 메타는 "차세대 대규모 언어 모델(LLM)인 라마(Llama)3를 다음 달에 출시할 것"이라고 밝혔다. 지난해 7월 전작인 라마2를 내놓은 후 1년도 되지 않아 차세대 모델을 내놓겠다는 것이다. 지난달 말에는 오픈AI가 이르면 올해 여름에 차세대 LLM인 'GPT-5'를 출시할 계획이라는 보도가 나왔다.

오픈 AI와 메타의 'AI 진화'		
	출시 시기	AI 모델
오픈 AI	2020년 6월	GPT-3
	2023년 3월	GPT-4
	2024년 여름	GPT-5(예정)
메타	2023년 2월	라마1
	2023년 7월	라마2
	2024년 5월	라마3(예정)

전문가들은 GPT-5와 라마3는 과거 모델과 비교할 때 추론 및 계획 능력에서 근본적인 진화가 이뤄질 것으로 보고 있다. 이전에는 데이터 학습량과 속도가 빨라지는 방식의 개선이었다면, 이번에는 성능 측면에서 차원이 다른 발전이 있을 것으로 전망하고 있다.

지금까지의 AI 모델은 주로 대규모 데이터를 학습하고, 그 안에서 질문에 대한 답을 제공했다. 학습되지 않은 분야에 대해선 '정보가 없다'고 답하거나, 아예 가짜 정보를 짜깁기하는 '환각' 현상이 벌어지기도 한다. 하지만 추론 능력을 갖춘 AI는 학습한 데이터 중에서 질문과 유사한 내용을 참고해, 자체

적으로 논리적인 답변을 만들어 낼 수 있다. 상황 판단이 가능해진 AI는 사람처럼 사안의 경중을 따져 우선순위를 계획할 수도 있게 된다. 실리콘밸리의 한 AI 엔지니어는 "추론과 계획 능력은 AI가 인간 고유의 창의성, 상상력, 도덕관까지 흉내낼 수 있게 하는 결정적인 변곡점"이라고 했다.

최근 인간과 유사한 범용 인공지능(AGI)의 등장을 예고하는 목소리가 커지고 있다. 일론 머스크(53) 테슬라 최고경영자(CEO)는 최근 "범용 인공지능을 가장 똑똑한 사람보다 똑똑한 AI라고 정의한다면, 아마도 내년쯤에는 가능할 것"이라고 했다. AI 분야 4대 석학 중 한 명으로 꼽히는 요수아 벤지오 몬트리올대 교수도 올 초 본지 인터뷰에서 "빅테크들은 우리가 인지하고 있는 것보다 훨씬 거대하고 정교한 AI를 개발하고 있으며, 올해 그 실체가 드러날 것"이라고 예고했다.

오픈AI와 메타는 모두 실제 이에 근접한 AI 모델 공개를 눈앞에 두고 있다. 메타의 AI 연구 부사장인 조엘 피누는 "라마3는 단지 말을 만들어 내는 것뿐 아니라, 실제 추론하고 계획할 능력을 갖추 수 있게 노력 중"이라고 말했다. 샘 올트먼 오픈AI 최고경영자(CEO) 역시 "차세대 GPT는 추론 같은 어려운 문제를 해결하는 데 획기적인 진전을 보일 것"이라고 말했다.

추론과 계획 능력을 갖춘 AI의 출현은 AI 산업의 발전 양상을 변화시킬 전망이다. 예컨대 사람은 약 20시간을 학습하면 자동차를 운전할 수 있는 '상식'을 갖추게 되지만, 현재 AI는 운전 중 발생할 수 있는 모든 경우의 수를 학습하지 않고선 운전을 제대로 할 수 없다. 완전 자율 주행 자동차가 어려운 이유다. AI가 추론과 계획 능력을 갖추게 되면, 사람처럼 학습하지 않고도 우선순위를 정해 합리적 판단을 내릴 수 있게 된다. 테크 업계 관계자는 "지금까지의 AI가 직선적인 사고를 해왔다면, GPT-5와 라마3부터는 사람처럼 복잡하고 다층적인 생각을 하는 AI가 나오는 것"이라고 말했다.

출처 조선일보(2024. 4. 11.), https://www.chosun.com/economy/tech_it/2024/04/11/OHHRB3AVA5EXDIWM35FY5WQNXI/

추가 활동지 1

인공지능을 활용한 문제 해결 사례 비교 분석하기

☑ 우리 주변에서 발생한 다양한 문제 중에서 인공지능을 활용하여 해결한 사례를 찾아보고 인공지능을 활용하기 전과 비교·분석해 보자.

1 인공지능을 활용하여 해결한 문제 사례의 문제 상황을 작성해 보자.

예시 답안 식당에서 인건비가 많이 발생하여 운영이 어려울 때

2 인공지능을 활용하기 전과 후를 비교해 보자.

	인공지능 활용 전	인공지능 활용 후
사람 또는 인공지능의 역할	• 인간의 역할: 예시 답안 서빙	• 인공지능이 적용된 로봇의 역할: 서빙 • 인간의 역할: 서빙 로봇 제어
인공지능 활용으로 얻는 이익	인건비 감소, 빠른 서빙, 가게 홍보 효과	
앞으로의 발전 가능성	로봇의 속도 및 안정성 증가로 빠른 서빙	

제시한 사례에서 인공지능의 활용으로 달라진 변화를 구체적으로 작성해 보자.

음식점에서 서빙하는 직원의 수가 줄어들고 인공지능이 탑재된 로봇이 서빙을 대신하며 인건비가 줄어 음식의 맛과 양이 개선되고 음식점의 수익이 늘어날 수 있다.

3 새롭게 인공지능이 적용될 수 있는 분야를 생각해 보고 인공지능의 적용으로 얻을 수 있는 가치를 작성해 보자.

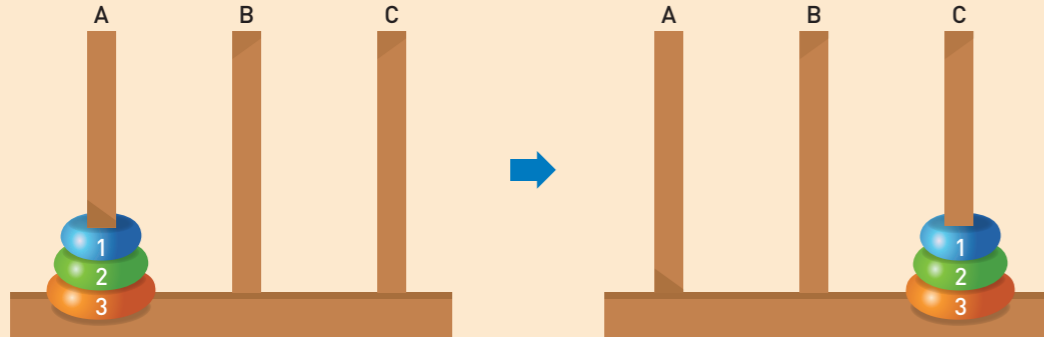
인공지능이 적용될 수 있는 분야	예시 답안 교육
인공지능의 역할	학생의 학습 습관 분석
인공지능의 적용으로 발생하는 가치	학생의 학습 능력 향상

하노이 탑 문제 해결하기

다음 문제 상황을 살펴보고 탐색을 통해 문제를 해결해 보자.

문제 상황

하노이의 탑은 세 개의 막대와 여러 개의 원판으로 구성된 퍼즐이다. 이 퍼즐의 목표는 한 막대에 쌓여 있는 원판들을 다른 막대로 모두 옮기는 것이다. (단, 한 번에 하나의 원판만 이동할 수 있으며, 큰 원판이 작은 원판 위에 놓일 수 없다.)



1 주어진 문제의 상태를 정의해 보자.

• 상태: 예시 답안 각 막대에 쌓인 원판의 순서

2 초기 상태와 목표 상태 설정

(1) 문제의 초기 상태와 목표 상태를 설정해 보자.

• 초기 상태: 모든 원판이 A 막대에 쌓여 있는 상태
• 목표 상태: 모든 원판이 C 막대에 쌓여 있는 상태

(2) 간단하게 표현해 보자.

• 초기 상태: ([1, 2, 3], [], [])
• 목표 상태: ([], [], [1, 2, 3])

3 수행 작업 파악하기

(1) 문제 해결을 위한 첫 번째 수행 작업과 다음 상태를 파악해 보자.

• A 막대의 1번 원판을 B로 옮긴다. ([2, 3], [1], [])
• A 막대의 1번 원판을 C로 옮긴다. (C, A, A)

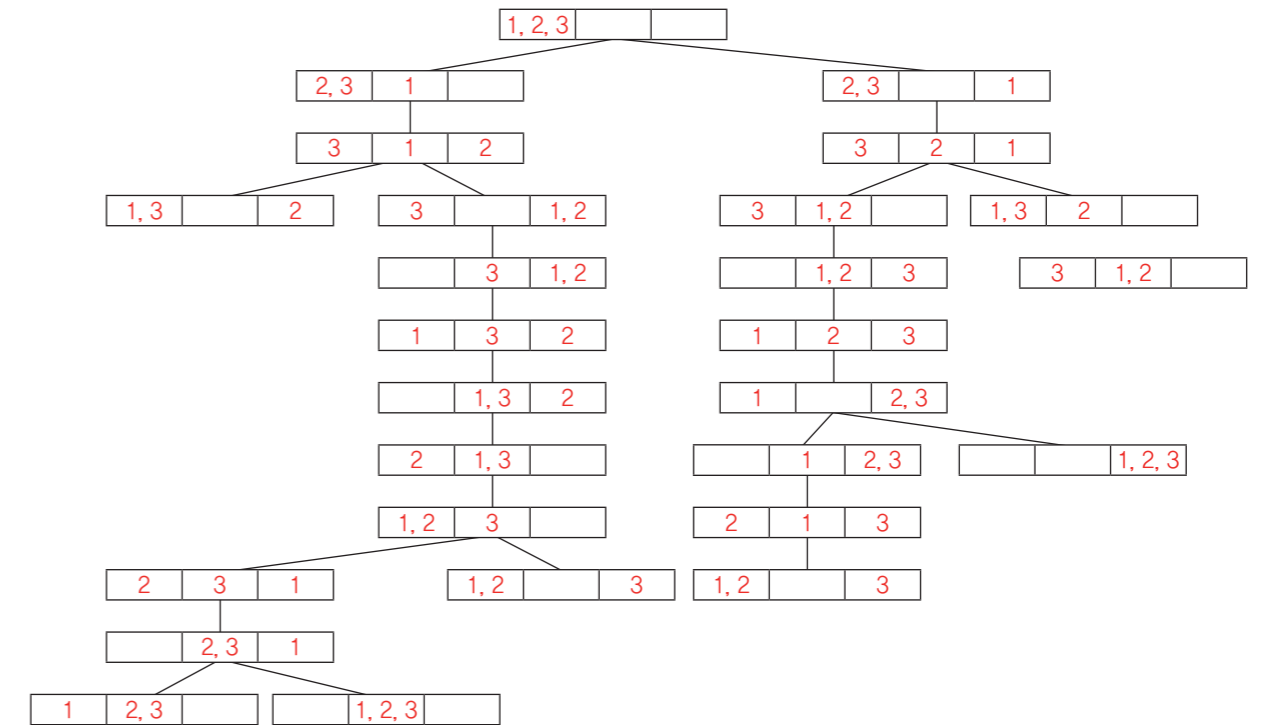
(2) 가능한 모든 수행 작업을 파악해 보자.

예 A에서 B로 원판 이동 (단, $A[1] < B[1]$, [1]은 가장 상위의 원판을 뜻한다.)

A에서 C로 원판 이동, B에서 A로 원판 이동, B에서 C로 원판 이동, C에서 A로 원판 이동, 에서 B로 원판 이동

4 탐색 전략 세우기

(1) 문제의 상태 공간 트리를 그려 보자.



(2) 상태 공간 트리를 보고 목표 상태에 가장 빨리 도달하는 방법을 적어 보자.

$([1, 2, 3], [], []) \rightarrow ([2, 3], [], [1]) \rightarrow ([3], [2], [1]) \rightarrow ([3], [1, 2], []) \rightarrow ([], [1, 2], [3]) \rightarrow ([1], [2], [3]) \rightarrow ([], [1], [2, 3]) \rightarrow ([], [], [1, 2, 3])$

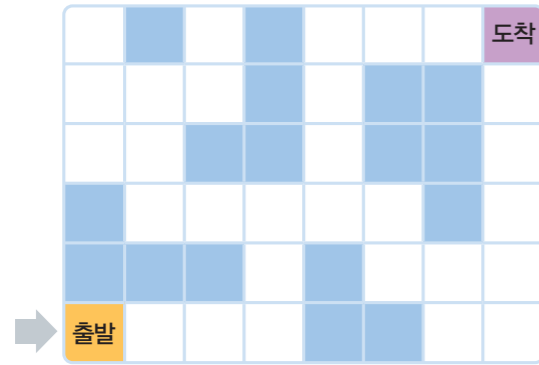
지도 방법

실제 하노이 퍼즐 실물을 사용하여 직접 수행해 보면 학습자의 흥미를 돕고, 이해를 높일 수 있다.

해설 하노이 탑 문제는 탐색과 상태 공간 트리를 배우기에 좋은 예제이다. 간단하게 숫자로 표현하거나 트리로 구조화하는 다양한 방법이 있지만, 리스트를 직관적으로 표현하여 학생들의 이해를 쉽게 할 수 있도록 막대와 원판 모두를 표현하는 방법을 선택하였다. 한 번에 하나의 원판만 이동할 수 있고, 큰 원판이 작은 원판 위에 놓일 수 없다는 제약 조건을 잘 파악하며 상태 공간 트리를 채울 수 있도록 지도한다.

다양한 알고리즘으로 미로 찾기

☑ 교과서 21쪽에 제시된 미로를 깊이 우선 탐색과 A* 탐색으로 탐색해 보자.

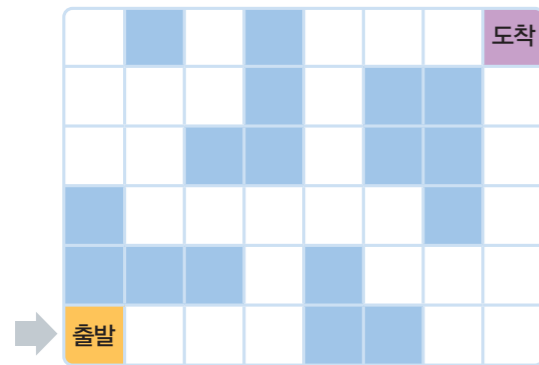


1 친구와 문제의 조건을 설정해 보자.

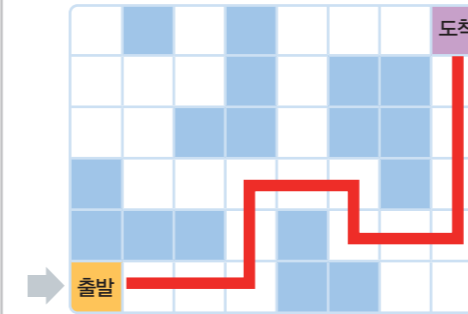
예시 답안

- 이동 순서는 오른쪽, 위, 아래, 오른쪽 순으로 한다.
- 한 번에 한 칸씩 이동한다. 파란 벽은 통과하지 못한다.
- 휴리스틱 값은 도착지와의 거리로 정한다.

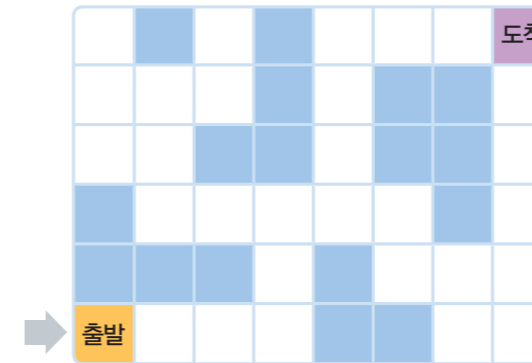
2 정해진 조건 대로 탐색하는 깊이 우선 탐색의 경로를 미로 위에 표시해 보자.



예시 답안



3 정해진 조건 대로 탐색하는 A* 탐색의 휴리스틱 값과 경로를 미로 위에 표시해 보자.



예시 답안

휴리스틱 값, 탐색 경로

	7	6	5	4	3	2	1	도착
	8	7	6	5	4	3	2	1
	9	8	7	6	5	4	3	2
	10	9	8	7	6	5	4	3
	11	10	9	8	7	6	5	4
출발	11	10	9	8	7	6	5	4

지도 방법

언플러그드 활동(unplugged activity)으로 탐색의 차이를 쉽게 이해할 수 있도록 지도한다.

해설 DFS는 탐색 순서에 따라 탐색 경로가 달라진다. 또한 최단 거리를 보장하지 않는다. A* 탐색은 탐색 순서와 상관없이 같은 경로를 예측한다. 또한 같은 휴리스틱 값을 가졌을 때는 모두 확장하고 최단 경로를 결정한다.

추가
활동지 4

철이, 돌이, 민이의 직업 추론하기

철이, 돌이, 민이의 직업은 축구 선수, 야구 선수, 농구 선수 중 서로 다른 하나다. 다음 [조건]을 읽고 물음에 답해 보자.

조건

1. 철이는 야구 선수가 아니다.
2. 민이는 농구 선수보다 나이가 많다.
3. 야구 선수는 돌이보다 나이가 적다.

1 주어진 문제의 상태를 정의해 보자.

- 철이: 예시 답안 농구 선수
- 돌이: 축구 선수
- 민이: 야구 선수

지도 방법

학생 스스로 추론을 통해 새로운 사실을 도출할 수 있도록 한다.

해설

1 조건 1, 2, 3을 표로 나타내면 아래와 같다.

구분	축구 선수	야구 선수	농구 선수
철이		X	
돌이		X	
민이			X

☞ 민이가 야구 선수인 것을 알 수 있다.

2 민이는 야구 선수이므로 축구 선수가 아니다.

구분	축구 선수	야구 선수	농구 선수
철이		X	
돌이		X	
민이	X	O	X

3 야구 선수인 민이는 돌이보다 나이가 많다.

- 나이 비교: 농구 선수 < 민이 < 돌이
☞ 돌이는 농구 선수가 아니다.

구분	축구 선수	야구 선수	농구 선수
철이		X	
돌이		X	X
민이	X	O	X

4 돌이가 축구 선수인 것을 알 수 있다.

구분	축구 선수	야구 선수	농구 선수
철이		X	
돌이	O	X	X
민이	X	O	X

5 철이는 축구 선수가 아니고 농구 선수인 것을 알 수 있다.

구분	축구 선수	야구 선수	농구 선수
철이	X	X	O
돌이	O	X	X
민이	X	O	X

추가
활동지 5

<마틴 가드너>의 옷 색깔 추론하기

다음 내용을 읽고, 빨간색 옷을 입은 여학생과 함께 춤을 추는 남학생의 옷 색깔을 맞춰 보자.

어느 학교 댄스 동아리에 서로 다른 성별로 구성된 세 팀이 모였다. 세 팀의 여학생은 각각 빨간색, 초록색, 파란색 옷을 입고 있고, 남학생도 같은 세 가지 색의 옷을 입고 있다. 이 세 쌍의 팀이 춤을 추었는데, 빨간색 옷을 입은 남학생이 옆에 다른 남학생과 춤추고 있던 초록색 옷을 입은 여학생에게 이렇게 말하였다.
“우리 중에 같은 색의 옷을 입고 춤을 추는 팀이 한 쌍도 없어.”

예시 답안 빨간색 옷을 입은 여학생의 파트너는 초록색 옷을 입은 남학생이다.

지도 방법

표를 그려서 결론을 도출할 수 있도록 지도한다.

해설

1 제시된 조건을 통해 알 수 있는 사실은 다음과 같다.

- 1 파트너는 다른 색 옷을 입고 있다.
- 2 빨간색 옷을 입은 남학생과 초록색 옷을 입은 여학생은 파트너가 아니다.

1번 사실에 의해 아래와 같은 표를 그릴 수 있다.

남학생 \ 여학생	빨간색	초록색	파란색
빨간색	X		
초록색		X	
파란색			X

☞ 파트너는 O로 표시, 파트너가 아니면 X로 표시

2번 사실에 의해 아래와 같은 표를 그릴 수 있다.

남학생 \ 여학생	빨간색	초록색	파란색
빨간색	X	X	
초록색		X	
파란색			X

☞ 위의 표에서 빨간색 옷을 입은 남학생의 파트너는 파란색 옷을 입은 여학생이고, 초록색 옷을 입은 여학생의 파트너는 파란색 옷을 입은 남학생인 것을 알 수 있다.

남학생 \ 여학생	빨간색	초록색	파란색
빨간색	X	X	O
초록색		X	
파란색		O	X

☞ 파란색 옷을 입은 남학생과 빨간색 옷을 입은 여학생, 초록색 옷을 입은 남학생과 파란색 옷을 입은 여학생은 파트너가 아니다.

남학생 \ 여학생	빨간색	초록색	파란색
빨간색	X	X	O
초록색		X	X
파란색	X	O	X

☞ 빨간색 옷을 입은 여학생의 파트너는 초록색 옷을 입은 남학생이다.

남학생 \ 여학생	빨간색	초록색	파란색
빨간색	X	X	O
초록색	O	X	X
파란색	X	O	X



선택형

01 인공지능 학습을 위한 데이터 수집으로 볼 수 없는 것은?

- ① 설문 조사 ② 웨어러블 기기 ③ 독서
- ④ SNS ⑤ 각종 감지 센서

해설 독서는 데이터를 수집하는 행위가 아니다.

02 로봇 청소기에서 인공지능의 판단에 따라 외부 환경에 영향을 미치는 것은?

- ① 라이더 ② 근접 센서 ③ 레이더
- ④ 바퀴 ⑤ 카메라

해설 지능 에이전트로 볼 수 있는 로봇 청소기는 센서를 통해 인식한 데이터를 인공지능으로 판단한 뒤, 액추에이터로 외부 환경에 영향을 미친다. 로봇 청소기에서 액추에이터는 바퀴와 청소 슴을 돌리는 모터이다.

03 인공지능의 특성에 대한 설명으로 옳바른 것은?

- ① 인식은 사람의 눈과 귀 등의 감각 기관처럼 센서를 통해 사물을 감지하는 것을 말한다.
- ② 추론은 데이터에서 일정한 패턴이나 규칙을 찾아 모델을 생성하는 것이다.
- ③ 탐색은 이미 알고 있는 사실이나 지식으로부터 새로운 사실을 예측하는 것이다.
- ④ 학습은 문제를 해결하기 위한 최선의 방법을 더 쉽게 찾는 것이다.
- ⑤ 인공지능의 여러 특성은 인간의 지능을 모방할 수 없다는 것을 전제로 한다.

해설 데이터에서 일정한 패턴이나 규칙을 학습하여 모델을 생성할 수 있다. 추론은 이미 알고 있는 지식으로부터 새로운 사실을 예측하는 것이다. 인공지능은 탐색을 통해 문제 해결을 위한 최선의 방법을 더 쉽게 찾을 수 있다. 인공지능의 여러 특성은 인간의 지능을 모방하려 한다.

04 인공지능이 적용된 소프트웨어의 특징을 모두 고른 것은?

가. 추론이나 예측이 가능하다.
나. 입력 데이터가 같으면 이를 처리하는 시간과 결과가 같다.
다. 데이터 기반의 기계학습을 구현한 인공지능의 경우 학습을 통해 성능이 개선될 수 있다.

- ① 가 ② 가, 나 ③ 가, 다
- ④ 나, 다 ⑤ 가, 나, 다

해설 인공지능이 적용된 소프트웨어는 추론이나 예측이 가능한 작업이 가능하고, 데이터 기반의 기계학습을 구현한 경우에는 학습을 통해 성능이 개선되기도 한다. 입력 데이터가 같으면 이를 처리하는 시간과 결과가 같은 것은 인공지능이 적용되지 않은 소프트웨어이다.

05 다음 중 탐색(Search)의 정의로 가장 적절한 것은?

- ① 데이터를 암호화하는 과정
- ② 정보를 저장하고 관리하는 과정
- ③ 데이터의 구조를 분석하는 과정
- ④ 네트워크의 노드를 생성하는 과정
- ⑤ 문제 해결을 위한 경로를 찾는 과정

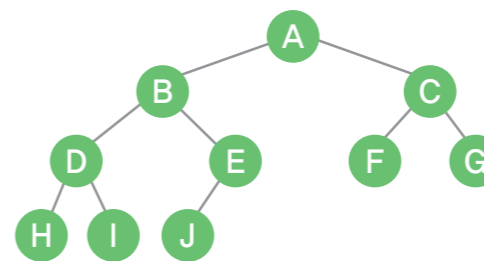
해설 탐색(Search)이란, 문제를 해결하기 위해 가능한 상태 공간(State Space)을 체계적으로 조사하여 목표 상태에 도달하는 경로를 찾는 과정을 의미한다.

06 다음 중 휴리스틱(Heuristic)의 특성에 관한 설명으로 가장 알맞은 것은?

- ① 경험적 지식을 기반으로 탐색 효율성을 향상시킨다.
- ② 모든 상태 공간을 탐색하는 것을 목표로 한다.
- ③ 탐색 중 발견한 노드를 무조건 우선 탐색한다.
- ④ 탐색 과정에서 비용을 고려하지 않는다.
- ⑤ 항상 최적의 결과를 보장한다.

해설 휴리스틱은 특정 문제를 더 빠르게 해결하거나 비용을 줄이기 위해 경험적 지식이나 추정치를 활용하는 탐색 기법이다. 이는 최적의 답을 반드시 보장하지는 않지만, 문제의 복잡도를 줄여 효율적인 탐색을 가능하게 한다.
② 휴리스틱 탐색은 전체 상태 공간을 탐색하기보다 효율성을 중시한다.
③ 노드를 탐색하는 순서는 휴리스틱 함수에 의해 결정된다.
④ 휴리스틱 탐색에서는 비용을 고려하기 위해 $f(n)=g(n)+h(n)$ 을 사용한다.
⑤ 휴리스틱은 항상 최적의 결과를 보장하지는 않는다.

07 다음 트리를 깊이 우선으로 탐색했을 때, 1의 바로 다음에 방문할 노드로 옳은 것은?



- ① B ② C ③ E
- ④ F ⑤ J

해설 그림의 트리를 깊이 우선으로 탐색하는 순서는 A-B-D-H-I-E-J-C-F-G이다. 따라서 1 노드 다음에 방문할 노드는 ③ E 노드이다.

08 정보에 관한 설명으로 바른 것을 모두 고른 것은?

가. 관찰이나 실험을 기록한 것이다.
나. 풍속, 풍향, 온도, 습도를 근거로 비 올 확률을 80%로 예보하였다.
다. 비 올 확률이 80% 이상이면 외출할 때 우산을 가지고 가는 것이 좋다.

- ① 가 ② 나 ③ 다
- ④ 가, 나 ⑤ 가, 다

해설 가. 관찰이나 실험을 기록한 것은 데이터이다.
나. 풍속, 풍향, 온도, 습도 등의 데이터를 근거로 비 올 확률을 예보한 것은 정보이다.
다. 비 올 확률이 높으면 우산을 가지고 다닌 것이 좋다고 하는 것은 지식이다.

09 지식을 IF-THEN을 활용한 지식 표현으로 바르게 표현한 것이 아닌 것은?

지식	지식 표현
① 밤을 새고 공부하면, 다음날 피곤하다.	IF 밤을 새고 공부했다. THEN 다음날 피곤하다.
② 봄이 되면, 제비가 돌아온다.	IF 봄이 된다. THEN 제비가 돌아온다.
③ 기온이 영하로 내려가면 방한이 잘 되는 옷을 입어라.	IF 방한이 잘 되는 옷을 준비하면 THEN 기온이 영하일 수 있다.
④ 송곳니가 있고 포유류이면 육식동물이다.	IF 송곳니가 있다. AND 포유류이다. THEN 포유류이다.
⑤ 등교 시간까지 잠을 자면, 지각을 한다.	IF 등교 시간까지 잠을 잤다. THEN 지각을 한다.

해설 IF 기온이 영하로 내려갔다.
THEN 방한이 잘 되는 옷을 입어라

10 귀납 추론에 관한 설명으로 바른 것을 모두 고른 것은?

가. 대표적으로 삼단논법이 있다.
나. 이미 알고 있는 사실로부터 모르고 있던 사실을 유추해 낸다.
다. 최초의 대전제가 결론을 이끌어 내는 중요한 근거가 된다.

- ① 가 ② 나 ③ 다
- ④ 가, 나 ⑤ 가, 다

해설 삼단논법은 연역 추론의 대표적인 방법으로, 최초의 대전제가 결론을 이끌어 내는 중요한 근거가 된다.



단답형

11~13 다음은 강 건너기 문제 상황이다. 물음에 답하시오.

어느 평화로운 마을에 농부가 살고 있었습니다. 농부는 강 건너편에 있는 농장에 가기 위해 강을 건너야 했습니다. 그런데 농부는 양배추, 양, 그리고 여우를 함께 데리고 가야 했습니다. 문제는 농부의 배는 너무 작아서 한 번에 두 가지만 태울 수 있었고, 농부만 배를 몰 수 있었어요. 그리고 농부가 없다면 양은 양배추를 먹어 버리고, 여우는 양을 몰 수도 있습니다. 과연 어떻게 하면 안전하게 강을 건널 수 있을까요? 농부는 깊은 고민에 빠졌습니다.



11 현재 농부, 양배추, 양, 여우의 위치를 이진수로 표현하고자 한다. 이동 전 상태를 00이라고 하고, 강을 건넨 후의 상태를 1이라 할 때, 초기 상태와 목표 상태를 숫자로 나타내시오. (단, 농부, 양배추, 양, 여우의 순서대로 표시한다.)

- 초기 상태: 예시 답안 0000
- 목표 상태: 1111

해설

- 초기 상태: 농부, 양배추, 양, 여우가 모두 강의 왼편에 있다. 왼편은 0으로 표현되므로, 모두 0으로 표시 된다. (0000)
- 목표 상태: 농부, 양배추, 양, 여우가 모두 강의 오른편으로 이동한 상태이다. 오른편은 1로 표현되므로, 모두 1로 표시된다. (1111)

12 다음 중 가능하지 않은 상태를 모두 골라 쓰시오.

0001, 0010, 0011, 0100, 1001, 1010, 1100

예시 답안 0001, 0011, 0100, 1001, 1100

해설

- 0001: 농부가 여우만 데리고 강을 건넜다가 다시 되돌아온 상태, 양배추와 양은 농부 없이 남아 있었어야 하므로 불가능한 상태이다.
- 0010: 농부가 양을 데리고 강을 건너고 농부만 돌아온 상태이므로 가능한 상태
- 0011: 양과 여우가 강을 건너 있기 때문에 불가능한 상태
- 0100: 농부가 양배추를 가지고 강을 건넜다가 다시 돌아온 상태로 농부 없이 양과 여우가 남아 있었어야 하므로 불가능한 상태
- 1001: 농부가 여우만 강을 건넨 상태로 양과 양배추가 농부 없이 남아 있었으므로 불가능한 상태
- 1010: 농부가 양을 데리고 건넨 상태로 가능한 상태
- 1100: 농부가 양배추만 데리고 강을 건넨 상태로, 양과 여우만 남아 있어서 불가능한 상태

13 가장 빨리 문제를 해결할 수 있는 방법을 이진수 기호로 표시하고, 글로 설명하시오.

예시 답안

0000 → 1010 → 0010 → 1110 → 0100 → 1101 → 0101 → 1111

1. 농부와 양이 강을 건넌
2. 농부가 다시 돌아옴
3. 농부가 여우와 강을 건넌
4. 농부가 양과 다시 돌아옴
5. 농부가 양배추와 강을 건넌
6. 농부가 다시 돌아옴
7. 농부가 양과 강을 건넌

해설 탐색 트리에서 농부, 여우, 양, 양배추 객체의 위치 정보가 출발지 즉 강을 건너기 전의 위치에 있다면 0으로 표현하고 강 건너편에 도착했다면 1로 표현한다. 최단 거리를 구할 때 양과 양배추, 양과 여우 둘만 남지 않도록 해야 한다는 점에 유의하도록 한다.

서술형

14 맹목적 탐색과 정보 이용 탐색의 주요 특징을 서술하고, 각각 어떤 문제를 해결하는 데 쓰는 것이 유리할지 논하시오.

예시 답안

맹목적 탐색은 기존의 정보를 사용하지 않고 상태 공간을 순서대로 탐색하여 목표 상태에 도달하는 방법으로, 모든 상태를 탐색하므로 목표 상태가 존재하면 반드시 찾아낼 수 있다. 예를 들어, 서랍장에서 물건을 찾을 때, 물건의 위치에 대한 추가 정보가 없을 경우 단순히 하나씩 서랍을 열어 확인하는 것이다. 서랍장이 작고 물건이 반드시 있는 경우, 빨리 목표에 다다를 수 있다는 점이 장점이다. 하지만 서랍장이 많거나 물건이 없을 경우, 맹목적 탐색은 많은 시간과 노력을 요구하는 비효율적인 방법이 될 수 있다. 반면, 정보 이용 탐색은 목표 상태에 가까워질 가능성을 평가하기 위해 휴리스틱 정보를 활용하는 방식으로, 상태 공간이 클 때 효율적이며 좋은 결과를 빨리 얻을 가능성이 높다. 예를 들어, 물류 배송 경로를 정한다면 배송 차량이 여러 도시에 상품을 배달해야 할 때 각 도시 간 거리를 고려하여 가장 짧은 경로를 선택하는 것이 중요하다. 이 경우, 휴리스틱 정보로 두 도시 간의 직선 거리를 활용할 수 있다. A* 알고리즘은 누적된 경로 비용과 도착지까지의 직선 거리를 조합하여 가장 효율적인 경로를 선택하며, 물류 비용과 시간을 절감할 수 있다. 단, 휴리스틱 정보가 부정확하거나 잘못 설정된 경우에는 탐색 성능이 저하될 위험도 있다.

따라서 맹목적 탐색은 상태 공간이 작거나 추가 정보를 얻기 어려운 문제에 적합하며, 간단한 퍼즐 해결과 같은 상황에 유리하다. 반면, 정보 이용 탐색은 상태 공간이 크거나 추가 정보가 있는 문제에 적합하며, 물류 경로 최적화, 내비게이션 시스템, 또는 로봇 경로 계획과 같은 실생활 문제에서 특히 유용하다.

해설 이 답안에서는 실생활과 관련된 물건 찾기와 물류 경로 최적화를 구체적인 사례로 제시하여 탐색 방식의 차이를 설명했다. 맹목적 탐색은 추가 정보를 활용할 수 없는 상황에서도 신뢰할 수 있는 결과를 제공하지만, 상태 공간이 크면 비효율적이다. 반면, 정보 이용 탐색은 휴리스틱 정보를 활용하여 효율성을 높이며, 실시간 경로 탐색이나 대규모 물류 문제를 해결하는 데 적합하다.