

# 01

## 알고리즘과 문제 해결

- + 학습 목표**
  - 알고리즘이 무엇인지 알고 문제를 해결하기 위한 절차를 설명할 수 있다.
  - 추상화, 알고리즘 설계, 자동화를 통해 문제를 해결할 수 있다.
- + 학습 요소**
  - 문제, 문제 해결 절차, 추상화, 알고리즘, 자동화

### 생각 깨우기

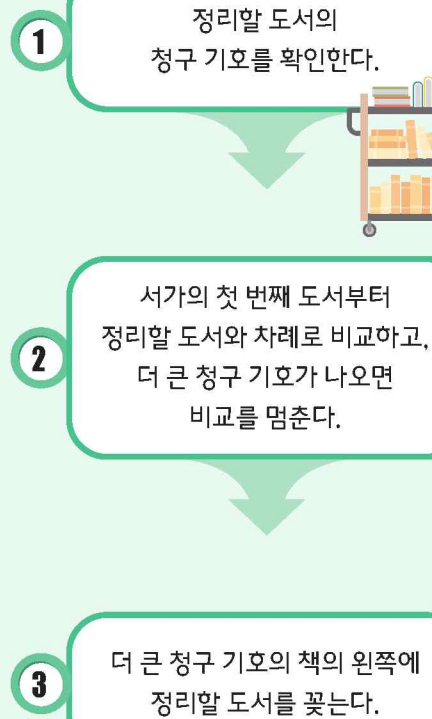
일상생활 속의 문제를 해결하기 위한 방법을 생각해 보자.

우리는 살면서 다양한 문제 상황에 직면하게 된다. 예를 들어, 라면 끓이기, 제목 순서대로 책 정리하기, 최소한의 역을 이동하여 목적지에 도착하기 등의 문제들이 있다.

#### 라면 끓이기



#### 도서 정리하기



문제는 어떤 절차로 해결할 수 있을까?

# 1 | 알고리즘의 이해

양치질을 처음 하는 어린 아이는 ‘양치질하기’ 문제를 어떻게 해결할 수 있을까? 양치질을 하는 과정을 순서대로 차근차근 설명해 주면 양치질을 쉽게 수행할 수 있을 것이다. 이처럼 문제를 해결하기 위해 필요한 구체적인 방법과 절차를 알고리즘이라고 한다. 우리 생활 속의 다양한 문제를 해결할 때는 문제 해결을 위한 알고리즘을 생각해 보는 것이 중요하다.

❗ 알고리즘(Algorithm)은 9세기 페르시아의 수학자 무함마드 알카리즈미(Muhammad ibn Mūsā al-Khwarīzmī)의 이름에서 유래되었다.

## 양치질 알고리즘

- 1 칫솔에 치약을 짜다.
- 2 칫솔을 사용하여 어금니 안쪽 면과 바깥 면을 잇몸에서부터 치아 쪽으로 10회씩 닦는다.
- 3 앞니의 바깥 면을 잇몸에서부터 치아 쪽으로 10회 닦는다.
- 4 앞니의 안쪽 면은 칫솔을 45도 각도로 세워 10회 닦는다.
- 5 어금니의 씹는 면을 앞뒤로 왕복하며 10회 닦는다.
- 6 혀를 닦는다.
- 7 ②~⑥을 3분 이상 반복한다.
- 8 물로 입 안을 헹군다.

우리는 이미 일상생활 속에서 다양한 종류의 알고리즘을 사용하고 있다. 길 찾기 알고리즘의 도움을 받아 가장 빠른 경로를 검색하기도 하고, 인터넷 사이트에서 게시글이나 댓글을 특정 기준에 따라 정렬해서 보기 위해 정렬 알고리즘을 사용한다. 과거의 영상 시청 이력으로 동영상 추천을 주고, 구매 이력을 바탕으로 상품을 추천해 주는 것도 알고리즘을 사용하는 것이다.



## 해 보기 1 일상생활 속 알고리즘 찾아보기

우리의 일상생활 속에서 찾아볼 수 있는 알고리즘에는 어떤 것들이 있는지 생각나는 대로 적고, 친구와 이야기해 보자.

## 2 | 문제와 문제 해결 절차

### 문제의 사전적 정보

- 해답을 요구하는 물음
- 논쟁, 논의, 연구 등의 대상이 되는 것
- 해결하기 어렵거나 난처한 대상 또는 그런 일
- 귀찮은 일이나 말썽
- 어떤 사물과 관련되는 일

### 계산 문제(Computational Problem)

계산 문제란 컴퓨터의 계산 능력을 이용하여 해결 가능한 문제로, 산술 연산이나 논리 연산의 조합을 통해 해결할 수 있는 문제를 말하며 연산 문제, 경우의 수 문제, 결정 문제, 최적화 문제 등이 있다.

우리는 살면서 매 순간 다양한 문제 상황에 놓이게 된다. ‘문제’란 목표로 하는 상태와 현재의 상태가 일치하지 않는 상황을 뜻하며, 문제를 해결하는 과정은 초기 상태에서 시작하여 목표 상태까지 도달하는 과정을 의미한다. 초기 상태와 목표 상태의 차이를 줄여 나가면서 문제를 해결할 수 있으며, 결국 문제가 해결되었다는 것은 현재 상태와 목표 상태가 일치되었다는 것을 뜻한다.

다양한 문제 중 계산 문제는 컴퓨터를 이용하여 절차적으로 해결할 수 있는데, 컴퓨터로 문제를 해결하기 위해서는 추상화, 알고리즘 설계, 자동화의 절차가 필요하다.

#### 추상화

주어진 문제를 이해하고 분석하여 꼭 필요한 핵심 요소를 추출하고, 문제를 해결하기 쉬운 형태로 표현한다.

#### 알고리즘 설계

글이나 도형, 흐름선 등으로 문제를 해결하기 위한 작업을 절차에 맞춰 나열한다.

#### 자동화

알고리즘을 컴퓨팅 시스템이 수행할 수 있도록 프로그래밍 언어로 작성하고 공유하여 성능을 평가한다.

### 예제

다음 상황을 예로 추상화, 알고리즘 설계, 자동화의 문제 해결 절차를 살펴보자.

‘왕의 열쇠’를 보기 위해 하루에 4만 명의 관람객이 찾는 박물관에서 도난 사건이 발생했다. 황금 열쇠가 은색 열쇠로 바뀌어 버렸다.



#### <입장 안내>

- 관람객은 1번부터 입장 번호를 받습니다.
- 전시장에는 입장 번호대로 1명의 관람객만 입장할 수 있습니다.

경찰 여러 명이 동시에 출발하여 입장했던 관람객을 한 명씩 찾아가 열쇠의 색을 질문하려고 한다.



범인만이 거짓으로 이야기할 때, 필요한 경찰의 최소 인원은 몇 명일까?



### 풀이

#### Step1 추상화

추상화 단계에서는 문제의 복잡성을 제거하여 해결하기 쉬운 형태로 문제를 재표현한다. 문제를 해결하기 위해서는 먼저 주어진 문제를 이해하고 분석해야 한다.

문제의 초기 상태와 목표 상태를 설정하고, 문제 해결을 위해 꼭 필요한 핵심 요소를 추출한 후, 목표 상태에 이르기 위해 수행해야 하는 작업들을 정의해야 한다. 이 과정에서 문제를 해결 가능한 작은 문제로 분해하고 모델링한다면 복잡한 문제를 쉽게 해결할 수 있다.

### 문제를 추상화한 결과

초기 상태	범인을 찾기 위해 최소 몇 명의 경찰이 필요한지 모르는 상태
목표 상태	범인을 찾기 위해 최소 몇 명의 경찰이 필요한지 아는 상태
핵심 요소	<ul style="list-style-type: none"> <li>• 총 관람객의 수, 관람객의 입장 순서</li> <li>• 범인만 거짓을 이야기할 것이라는 사실</li> <li>• 은색 열쇠를 처음 본 관람객의 다음 순서부터는 범인 후보에서 제외된다는 사실</li> </ul>
수행 작업	<ul style="list-style-type: none"> <li>• 관람객이 1명만 남을 때까지 관람객 수를 절반으로 나눌 수 있는 입장 번호 찾기</li> <li>• 입장 번호를 절반으로 나눈 횟수를 세기</li> <li>• 남은 1명의 관람객까지 조사하기 위해 필요한 경찰의 최소 인원수 구하기</li> </ul>

### Step2 알고리즘 설계

알고리즘이란 문제를 해결하기 위한 구체적인 방법이나 절차를 의미한다. 추상화를 통해 문제를 재표현한 이후에는 순차, 선택, 반복 구조 등을 이용하여 알고리즘을 설계할 수 있다. 알고리즘은 자연어, 의사 코드, 순서도, 프로그래밍 언어를 사용하여 표현할 수 있으며 다음 5가지 조건을 만족할 수 있도록 작성해야 한다.

- **입력** 필요한 데이터를 입력받을 수 있어야 한다.
- **출력** 알고리즘이 수행되면 적어도 1개 이상의 결과가 출력되어야 한다.
- **명확성** 알고리즘을 수행할 때 모호함이 없이 의미가 분명해야 한다.
- **유한성** 알고리즘은 반드시 종료되어야 한다.
- **수행 가능성** 모든 명령은 수행 가능해야 한다.

하나의 문제를 해결하기 위한 알고리즘은 다양할 수 있으며, 이를 비교·분석하여 더 좋은 알고리즘을 설계하도록 해야 한다.

다음은 문제를 해결하기 위한 알고리즘을 자연어, 의사 코드, 순서도로 표현한 것이다.

자연어	의사 코드	순서도
1 visitors를 40000, police를 0, count를 0으로 정한다.	1 visitors ← 40000 2 police ← 0 3 count ← 0	<pre> graph TD     Start([시작]) --&gt; Init[visitors ← 40000 police ← 0 count ← 0]     Init --&gt; Decision{visitors &gt; 1}     Decision -- 아니요 --&gt; End([끝])     Decision -- 예 --&gt; Loop[visitors ← visitors // 2 count ← count + 1]     Loop --&gt; Decision     Loop --&gt; Police[police ← count + 1]     Police --&gt; Output[police]     Output --&gt; End         </pre>
2 만약 visitors가 1보다 크다면 단계 3으로 이동하고 그렇지 않으면 단계 6으로 이동한다.	4 만약 visitors > 1이라면 단계 3으로 이동하고, 그렇지 않으면 단계 6으로 이동한다.	
3 visitors에 visitors의 값을 2로 나누었을 때의 몫을 저장한다.	5 visitors ← visitors // 2	
4 count의 값을 1 증가시킨다.	6 count ← count + 1	
5 단계 2로 이동한다.	7 go to 2	
6 police를 count에 1을 더한 값으로 정한다.	8 police ← count + 1	
7 범인을 잡기 위해 필요한 경찰의 최소 인원수 police를 출력한다.	9 police 출력	
8 알고리즘을 종료한다.	10 종료	

### 핵심 요소

문제를 해결하기 위해 반드시 고려해야 하는 요소로, 수행 작업에 영향을 미치는 요소이다.

### 수행 작업

현재 상태에서 목표 상태에 도달하기 위해 수행해야 하는 작업을 말한다.

### 선택 구조

조건에 따라 다른 동작을 하도록 표현한 구조를 의미한다.

### 반복 구조

같은 명령을 여러 번 반복하여 실행하도록 표현한 구조다.

### 순서도

마리 약속한 도형과 흐름선을 사용하여 알고리즘을 표현하는 방법이다.

### 의사 코드(Pseudo Code)

가상의 코드라는 의미이며, 프로그래밍 언어와 유사한 형태로 알고리즘을 표현하는 방식이다. 실제 프로그래밍 언어가 아니기 때문에 컴퓨터에서 실행할 수는 없다.

### 프로그래밍 언어

컴퓨터 프로그램을 만들기 위해 사용하는 언어로, 텍스트 기반 프로그래밍 언어로는 파이썬, C 언어, C++ 등이 있다.

## 🔧 프로그래밍

알고리즘을 컴퓨터를 통해 구현하기 위해 다양한 종류의 프로그래밍 언어를 사용하여 작성하는 것을 의미한다.

🔧 본문에서 사용한 프로그래밍 언어는 파이썬(Python)이다.

🔗 228~229쪽 <부록 1>의 구글 코랩(Google Colab) 활용하기를 참고하여 프로그램을 작성해 보자.

### Step3 자동화

자동화 단계는 설계한 알고리즘을 컴퓨팅 기기로 구현하는 프로그래밍 과정과 프로그램의 실행 결과를 분석하고 평가하는 성능 평가 과정을 포함한다. 구현된 프로그램이 원래 해결하고자 했던 문제를 올바르게 해결할 수 있는지, 더 효율적인 해결 방법이 있는지를 확인하며 성능 평가를 수행할 수 있다.

다음은 문제를 해결하기 위한 알고리즘을 파이썬 언어로 구현한 프로그램이다.

#### 📄 프로그램

```

1 visitors = 40000 # 전체 입장객의 수를 변수 visitors에 저장
2 count = 0 # 절반으로 나누는 횟수를 세기 위한 변수 count를 0으로 설정
3 while visitors > 1 : # 입장객이 1명만 남을 때까지 반복
4     visitors = visitors // 2 # 입장객의 수를 반으로 나누었을 때의 몫을 저장
5     count = count + 1 # 변수 count의 값 1 증가
6 police = count + 1 # 경찰의 수를 저장할 변수 police를 count + 1로 설정
7 print(police) # 경찰의 수를 출력
    
```

#### 📊 성능 평가

- 필요한 경찰의 최소 인원수가 올바르게 구해지는가?
- 더 효율적으로 해결할 수 있는 방법이 있는가?

이와 같이 컴퓨팅 시스템을 활용하여 복잡한 문제를 해결하기 위해서는 문제를 추상화하고 알고리즘을 설계하여 프로그램을 작성한 후 성능을 평가하는 과정이 필요하다. 이러한 문제 해결 과정을 거친다면 복잡한 문제도 빠르고 정확하게 해결할 수 있다.

## 🤖 해 보기 2 자동화한 프로그램 실행해 보기

🗨️ 예제를 해결하기 위한 프로그램을 작성하고 실행하여 결과를 확인해 보자.

### 소단원 1 요약

- 1 문제를 해결하기 위해서는 문제 해결을 위한 방법과 절차인 알고리즘을 설계하는 것이 중요하다.
- 2 '문제란' 목표로 하는 상태와 현재의 상태가 일치하지 않는 상황을 뜻하며, 문제를 해결하는 과정은 초기 상태부터 시작하여 목표 상태까지 도달하는 과정을 의미한다.
- 3 컴퓨터로 문제를 해결하기 위해서는 추상화, 알고리즘 설계, 자동화의 절차가 필요하다.
- 4 추상화는 문제의 복잡성을 제거하여 해결하기 용이한 형태로 문제를 재표현하는 것이고, 알고리즘은 문제를 해결하기 위한 구체적인 방법이나 절차를 의미하며, 자동화는 설계한 알고리즘을 컴퓨팅 기기로 구현하는 프로그래밍 과정과 프로그램의 실행 결과를 분석하고 평가하는 성능 평가 과정을 포함한다.

### 소단원 자기 평가

평가 항목	평가 기준		
	잘함	보통	노력
1. [지식이해] 알고리즘이 무엇인지 설명할 수 있다.			
2. [과정기능] 문제 해결 절차에 따라 문제를 해결할 수 있다.			
3. [가치태도] 알고리즘 효율의 가치와 영향력을 인식하고 적극적으로 탐구하려는 태도를 함양할 수 있다.			



다음 문제 상황을 보고, 문제 해결 절차에 따라 해결해 보자.

문제 상황

8퍼즐이란 9개의 칸으로 구성된 판에 '주어진 숫자판처럼 1부터 8까지의 숫자와 한 개의 빈칸으로 구성된 후, 빈칸을 이용하여 주변의 숫자들을 옮겨서 '완성된 숫자판'과 같이 배치하는 게임이다.

주어진 숫자판			완성된 숫자판		
1	2	3	1	2	3
7	4	5	4	5	6
	8	6	7	8	

1 문제의 초기 상태, 목표 상태를 설정하고 수행 작업을 정의한 뒤 핵심 요소를 추출하여 추상화해 보자.

초기 상태	예 주어진 숫자판
목표 상태	
핵심 요소	
수행 작업	

2 문제를 해결하기 위한 알고리즘을 설계하여 작성해 보자.

예 목표 상태의 숫자판과 비교하여 '1'이 올바른 위치에 있는지 확인한다.

3 작성한 2번의 알고리즘을 프로그램으로 만들면 어떤 장점이 있을지 생각해 보자.

탐구 활동 자기 평가	평가 항목	평가 기준		
		잘함	보통	노력
	1. 문제를 이해하고 분석하여 추상화할 수 있는가?			
	2. 알고리즘이 무엇인지 알고 문제를 해결하기 위한 알고리즘을 작성할 수 있는가?			

• 잘함 (내용을 이해하고 설명함) • 보통 (내용을 이해함) • 노력 (내용을 부분적으로 이해함)