

# IT 세상을 만나는 컴퓨터 개론 2판

인공지능, 빅데이터, 확장현실까지

## Chapter 07. 컴퓨팅 사고와 알고리즘

### 목차

1. 컴퓨팅 사고
2. 알고리즘
3. 자료구조와 알고리즘

## 학습목표

- 컴퓨팅 사고를 이해하고 컴퓨팅 사고의 구성 요소를 파악한다.
- 알고리즘의 표현 방식과 특성을 이해한다.
- 자료구조와 다양한 정렬 알고리즘을 알아본다.

# 01

## 컴퓨팅 사고

## 01. 컴퓨팅 사고

### 1. 컴퓨팅 사고의 개념

- 컴퓨터는 인간이 만든 알고리즘에 따라 동작함
  - 컴퓨터가 문제를 해결하려면 먼저 해결 방법을 컴퓨터에 알려줘야 함



그림 7-1 내비게이션

5 / 66

## 01. 컴퓨팅 사고

### 1. 컴퓨팅 사고의 개념

- 컴퓨팅 사고 Computational Thinking(CT)
  - 컴퓨터를 이용하여 문제를 해결할 수 있는 논리적이고 창의적인 생각 방식
  - 지네트 윈 박사가 처음으로 언급함



그림 7-2 지네트 윈

6 / 66

## 01. 컴퓨팅 사고

### 2. 컴퓨팅 사고의 예

- 일상생활에서 문제를 해결하는 다양한 방식이 컴퓨팅 사고와 닮아있음

#### ▪ 메뉴가 2가지인 식당: 2진수

- 메뉴가 2개인 식당은 메뉴가 10개인 식당보다 조리 속도가 빠름
  - 컴퓨터는 계산을 빠르게 하려고 2진수를 사용함

7 / 66

## 01. 컴퓨팅 사고

### 2. 컴퓨팅 사고의 예

#### ▪ 구내 식당에서 줄 서기: 입출력 채널 분리

- 반찬을 담아주는 한식 줄과 조리를 해야 하는 분식 줄을 분리하면  
한식을 먹을 사람이 분식 조리 시간 동안 기다리지 않아도 됨
  - 입출력 채널을 느린 채널과 빠른 채널로 분리하면 전체 입출력 효율이 올라감



그림 7-3 한식 줄과 분식 줄을 분리한 구내식당

8 / 66

## 01. 컴퓨팅 사고

### 2. 컴퓨팅 사고의 예

#### ▪ 사과 주스 공장: 버퍼

- 공장에서 매우 빠르게 사과를 가는 기계의 속도에 맞추기 위해 사과를 큰 바구니에 담아 통째로 옮김
  - 버퍼는 두 장치 사이에서 속도 차이를 완화

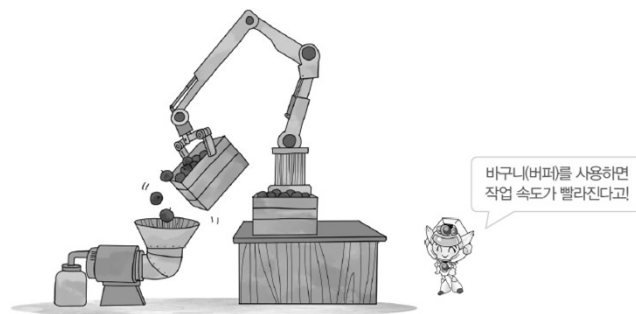


그림 7-4 사과 주스 공장

9 / 66

## 01. 컴퓨팅 사고

### 2. 컴퓨팅 사고의 예

#### ▪ 조미료통 사용: 캐시

- 대용량 포장으로 구매한 조미료를 조금씩 조미료통에 덜어놓음
  - 캐시에 앞으로 사용할 것으로 예상되는 데이터를 미리 가져다 놓으면 작업 효율이 향상됨



그림 7-5 조미료통

10 / 66

## 01. 컴퓨팅 사고

### 2. 컴퓨팅 사고의 예

#### ▪ 소문난 맛집: 멀티프로세싱

- 요리사를 채용하고 주방도 하나 더 만들어 조리 속도를 2배로 함
  - 멀티코어 시스템에서는 하나의 칩에 코어가 여러 개
  - 명령어 병렬처리는 하나의 코어에서 2개의 명령어를 거의 동시에 처리



그림 7-6 소문난 맛집의 주방



주방에 요리사가 두 명  
있으면 동시에 두 가지  
요리를 만들 수 있어.

11 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

- 추상화: 문제에서 중요하지 않은 부분을 제거하고  
중요한 특징만으로 문제를 구성
- 분해: 문제를 보다 해결하기 쉬운 작은 단위로 나눔
- 패턴 인식: 문제를 특징별로 나누어 의미 있는 패턴을 파악하고,  
유사한 문제 해결 방식이 있는지 찾아봄
- 알고리즘: 문제에 적합한 해결 방법을 공식화하여 표현

12 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

#### ▪ 추상화 abstraction

- 불필요한 세부 사항을 제거하여 간결하게 만들
- 중요한 특징만 남기 때문에 문제의 본질을 쉽게 파악
- 다각형으로만 코뿔소를 표현했지만 특징을 잘 묘사하여 알아볼 수 있음

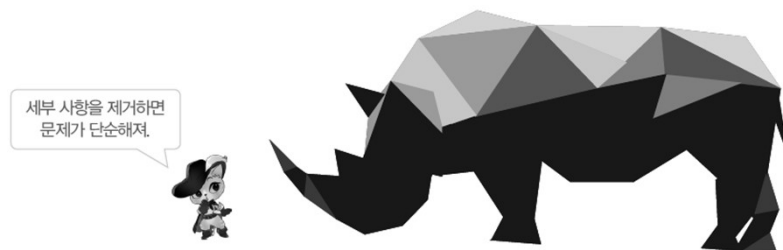


그림 7-8 코뿔소를 추상화한 그림

13 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

#### ▪ 추상화 abstraction

- 네온사인이나 간판에 음식을 추상화한 그림을 넣으면 멀리서도 인지



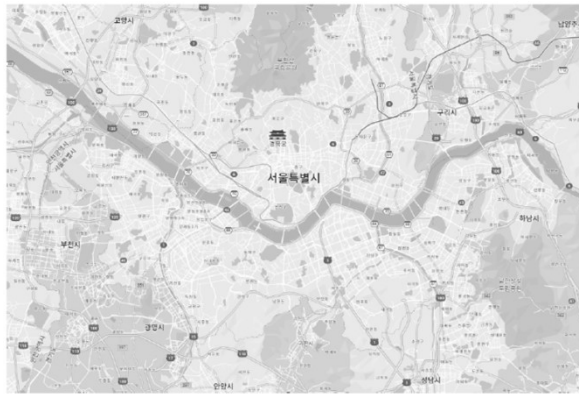
그림 7-9 음식을 추상화한 간판

14 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

- 추상화 abstraction
  - 지도 → 지하철 노선도



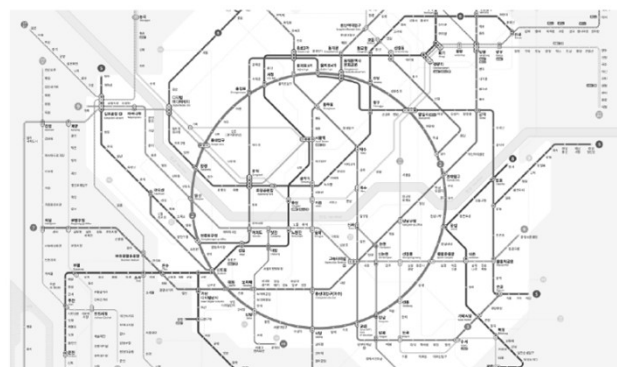
(a) 서울시 구글 지도

15 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

- 추상화 abstraction
  - 지도 → 지하철 노선도



(b) 지하철 노선도 © 서울메트로

그림 7-10 지도를 추상화한 지하철 노선도

16 / 66



## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

#### ▪ 추상화 abstraction

- 일반화: 추상화로 공통의 특성을 추려내는 과정과 추려낸 결과

1, 2, 3, 5, 8, 13, 21, 34, 55  
 $\rightarrow X_n = X_{n-1} + X_{n-2}$

그림 7-11 수열의 일반화

17 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

#### ▪ 분해 decomposition

- 복잡한 문제를 풀기 쉬운 간단한 문제 여러 개로 나누는 것
- 짜장면을 만드는 문제  $\rightarrow$  짜장 만들기, 면 삶기
  - 짜장 만들기  $\rightarrow$  고기 볶기, 양파 볶기, 춘장 넣기
  - 면 삶기  $\rightarrow$  물 끓이기, 면 넣기

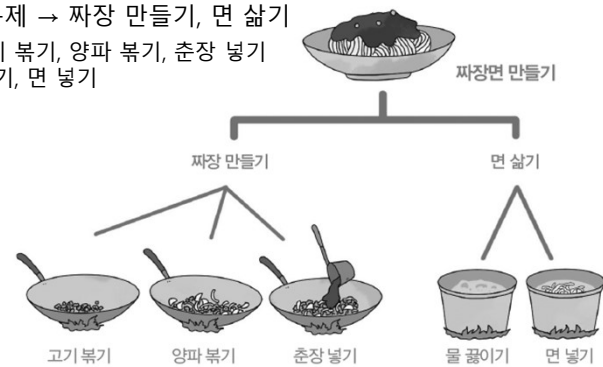


그림 7-12 짜장면 만들기의 분해

18 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

#### ▪ 분해 decomposition

- 가로가 3m, 세로가 2m인 벽에 가로세로가 각각 10cm인 타일을 붙일 때 타일 한 장을 붙이는 데 1g의 접착제가 필요하다면, 전체 타일은 몇 장? 접착제는 얼마나 필요?



그림 7-13 타일 붙이기의 분해

19 / 66

## 01. 컴퓨팅 사고

### 3. 컴퓨팅 사고의 구성

#### ▪ 패턴 인식 pattern recognition

- 문제에서 패턴을 찾고 패턴이 유사한 문제를 해결했던 방법을 찾음
- 걷는 사람 모습의 패턴
  - '오른발 앞으로 & 오른팔 뒤로', '왼발 앞으로 & 왼팔 뒤로'의 반복



그림 7-14 사람이 걷는 패턴

20 / 66

01. 컴퓨팅 사고

3. 컴퓨팅 사고의 구성

- **패턴 인식** pattern recognition
  - 디지털 이미지에서 빛의 삼원색이라는 패턴을 이용하여 다른 색을 만듦
  - 수열에서 패턴을 찾아 문제를 해결

1, 2, 3, 5, 8, 13, 21, 34, X

그림 7-15 수열

01. 컴퓨팅 사고

3. 컴퓨팅 사고의 구성

- **패턴 인식** pattern recognition
  - 데이터에서 패턴을 찾기 어려울 때  
어떤 데이터를 추가하면 패턴이 드러나기도 함

17 14 19 26 22 16 55 16 15 20 21 21 18 60  
12 24 20 19 23 19 61 12 18 19 17 12 18 49

(a) 패턴을 찾기 어려운 데이터

월 화 수 목 금 토 일 월 화 수 목 금 토 일  
17 14 19 26 22 16 55 16 15 20 21 21 18 60  
12 24 20 19 23 19 61 12 18 19 17 12 18 49

(b) 패턴이 드러나는 데이터

그림 7-16 식당의 하루 방문자 수

## 02 알고리즘

### 02. 알고리즘

#### 1. 알고리즘의 개념

- 알고리즘 algorithm : 주어진 문제를 해결하는 방법과 절차를 기술한 것
- 프로그래밍 언어: 알고리즘을 컴퓨터가 이해하는 언어로 변환한 것
- 요리의 레시피는 알고리즘과 같음
  - 음식을 만드는 문제의 해결 방법을 모아놓음
  - 재료의 종류와 양, 작업 순서



그림 7-17 라면 끓이기 알고리즘

## 02. 알고리즘

### 2. 알고리즘 표현 방법



그림 7-18 알고리즘 표현 방법

25 / 66

## 02. 알고리즘

### 2. 알고리즘 표현 방법

- **자연어** natural language
  - 인간이 일상생활에서 쓰는 언어
    - ↔ 컴퓨터가 사용하는 프로그래밍 언어
  - 두 수 바꾸기 알고리즘을 자연어로 표현

시작

X에 3, Y에 5를 대입한다.  
 X 값을 Z에 대입한다.  
 Y 값을 X에 대입한다.  
 Z 값을 Y에 대입한다.  
 X와 Y 값을 출력한다.

끝

그림 7-19 자연어

26 / 66

## 02. 알고리즘

### 2. 알고리즘 표현 방법

#### ■ 순서도 flow chart

- 기호와 선을 사용하여 문제 해결 과정을 표현
- 알고리즘의 구조나 흐름을 파악하는 데 유용하나 복잡한 프로그램을 작성하기 까다로움

순서도는 약속된 기호와 선으로 문제 해결 과정을 표현한 것!

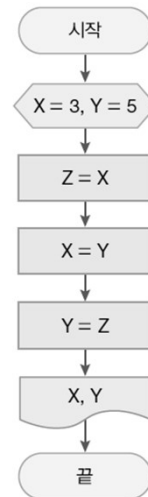


그림 7-20 순서도

27 / 66

## 02. 알고리즘

### 2. 알고리즘 표현 방법

#### ■ 순서도 flow chart

표 7-1 순서도의 기호

기호	명칭	의미
	단말	순서도의 시작과 끝
	흐름선	각 기호를 연결하여 순서도의 흐름을 나타냄
	처리	계산 등 데이터의 연산 또는 처리
	준비	변수의 초기값, 기억 장소의 설정 등 작업의 준비 과정
	판단	조건을 판단하여 '예' 또는 '아니요'로 이동
	입출력	데이터의 입력과 출력
	출력	출력 장치를 통한 출력

28 / 66

## 02. 알고리즘

### 2. 알고리즘 표현 방법

#### ▪ 의사 코드 pseudo code

- 프로그래밍 언어를 흉내 내어  
프로그래밍 언어와 유사한 방식으로 알고리즘을 표현

```
START
    X=3, Y=5
    Z=X
    X=Y
    Y=Z
    PRINT X, Y
END
```

그림 7-21 의사 코드

29 / 66

## 02. 알고리즘

### 3. 알고리즘의 조건

#### ▪ 의사 코드 pseudo code

- 입력
- 출력
- 유한성
- 명확성
- 수행 가능성

30 / 66

## 02. 알고리즘

### 4. 알고리즘 설계

- 알고리즘 설계: 문제를 해결하는 가장 효율적인 방법을 찾아내는 과정
  - 먼저 문제의 현재 상태와 목표 상태를 명확히 정의
- 제어 구조: 알고리즘의 명령이 실행되는 순서를 결정함
  - 순차 구조
  - 선택 구조: if (조건)
  - 반복 구조: while (조건)

31 / 66

## 02. 알고리즘

### 4. 알고리즘 설계

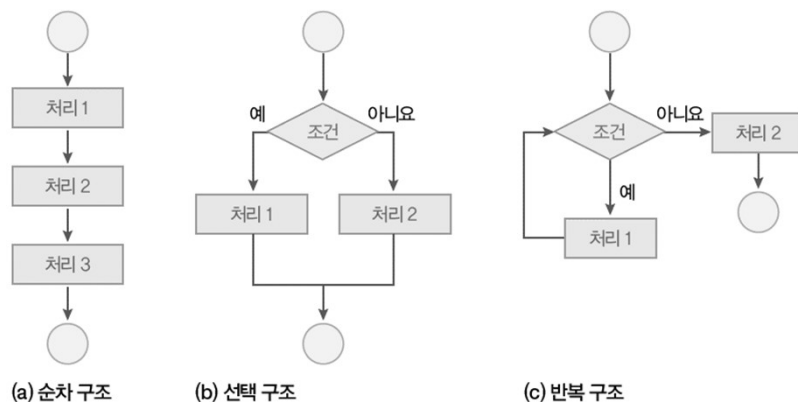


그림 7-22 제어 구조를 활용한 알고리즘

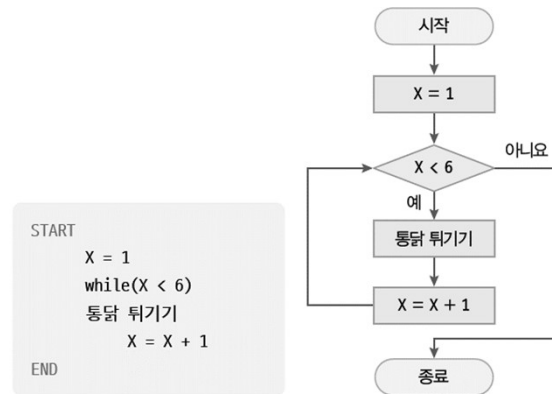
32 / 66



## 02. 알고리즘

### 4. 알고리즘 설계

- 통닭 다섯 마리를 차례로 튀기는 알고리즘의 의사 코드와 순서도



(a) 의사 코드

(b) 순서도

그림 7-23 통닭 5마리 튀기기의 의사 코드와 순서도

33 / 66

## 02. 알고리즘

### 5. 알고리즘 분석

- 문제를 해결하는 알고리즘이 다양할 때는 알고리즘을 비교·분석하여 가장 효율적인 것을 선택

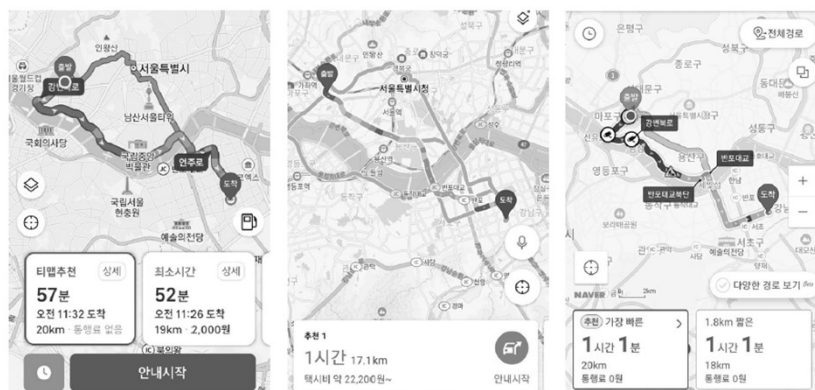


그림 7-24 세 가지 종류의 내비게이션 앱

34 / 66

## 02. 알고리즘

### 5. 알고리즘 분석

- 알고리즘의 효율을 평가하는 기준
  - 시간 복잡도
  - 공간 복잡도
- 성능 분석 방법
  - 프로그램을 직접 실행
  - 복잡도 분석: 알고리즘의 실행 횟수를 분석(시간 복잡도, 공간 복잡도)

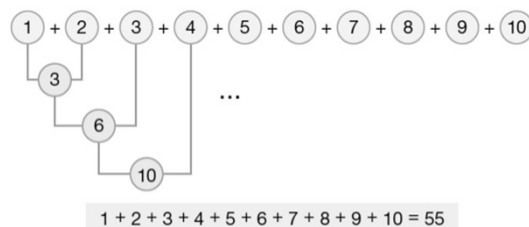
35 / 66

## 02. 알고리즘

### 5. 알고리즘 분석

#### ▪ 시간 복잡도

- 알고리즘이 실행되어 종료될 때까지 시간이 얼마나 걸리는지
  - 알고리즘의 주요 코드가 몇 번 실행되는지를 분석하여 간접적으로 알아냄
- 1부터 10까지 정수의 합을 구하는 두 가지 알고리즘 - 방법 1



(a) 방법 1

덧셈을 열 번이나  
해야 해.



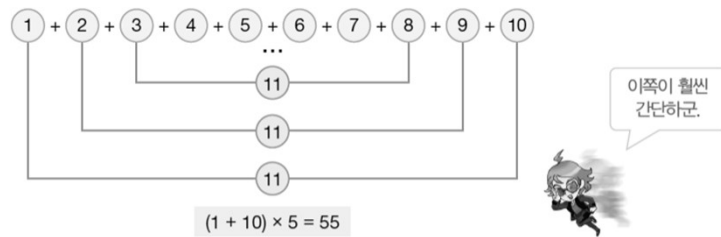
36 / 66

## 02. 알고리즘

### 5. 알고리즘 분석

#### ▪ 시간 복잡도

- 1부터 10까지 정수의 합을 구하는 두 가지 알고리즘 - 방법 2



(b) 방법 2

그림 7-25 1부터 10까지 정수의 합을 구하는 알고리즘

- 방법 2의 시간 복잡도가 더 낮음

37 / 66

## 02. 알고리즘

### 5. 알고리즘 분석

#### ▪ 공간 복잡도

- 알고리즘이 문제를 해결하는 데 저장 공간이 얼마나 필요한지
  - 데이터가 차지하는 공간
  - + 알고리즘이 사용하는 공간
- 1부터 10까지 정수의 합을 구하는 알고리즘 - 방법 3



(a) 방법 3

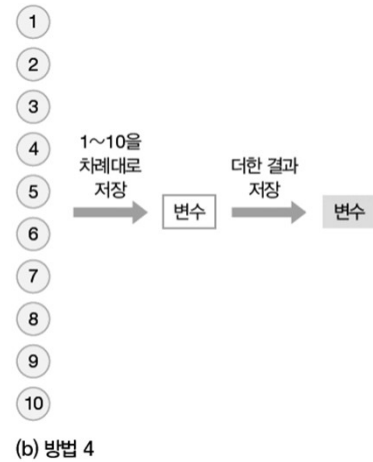
38 / 66

## 02. 알고리즘

### 5. 알고리즘 분석

#### ▪ 공간 복잡도

- 1부터 10까지 정수의 합을 구하는 알고리즘 - 방법 4
- 방법 4의 공간 복잡도가 더 낮음



39 / 66

## 03

## 자료구조와 알고리즘

### 03. 자료구조와 알고리즘

#### 1. 자료구조

- 자료구조: 여러 데이터를 구조적으로 표현하고 접근하는 방법

##### ▪ 큐와 스택

- 큐 queue : 먼저 들어온 데이터가 먼저 처리되는 자료구조
  - First In First Out(FIFO)
  - 햄버거를 주문하기 위해 줄을 서서 기다림



(a) 먼저 줄을 선 사람이 먼저 주문

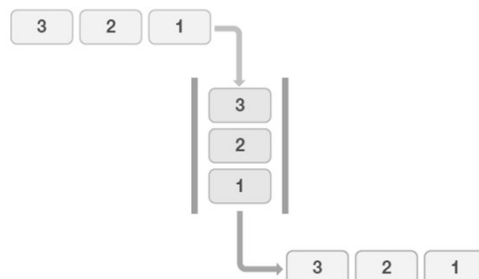
41 / 66

### 03. 자료구조와 알고리즘

#### 1. 자료구조

##### ▪ 큐와 스택

- 햄버거를 주문하기 위해 줄을 서서 기다림
- 파이프의 한쪽 구멍으로 데이터를 넣으면 반대쪽 구멍으로 나옴



(b) 큐의 구조  
그림 7-27 큐

42 / 66

### 03. 자료구조와 알고리즘

#### 1. 자료구조

##### ▪ 큐와 스택

- 스택 stack : 가장 나중에 들어온 데이터가 먼저 처리되는 자료구조
  - Last In First Out(LIFO)
  - 햄버거 여러 개를 한 봉투에 담으면 꺼낼 때는 위쪽에 있는 것부터 나눔



(a) 봉투에 마지막에 담은 햄버거를 먼저 꺼냄

(b) 스택의 구조

그림 7-28 스택

43 / 66

### 03. 자료구조와 알고리즘

#### 1. 자료구조

##### ▪ 배열 array

- 같은 종류의 데이터를 나열한 자료구조
- 배열을 이용하여 테이블을 간단히 구현할 수 있음
- 공간 관리와 삽입·삭제할 때의 순서에 주의
  - 공간 관리: 배열의 크기를 너무 크게 정하면 메모리에 낭비되는 공간이 생기고, 너무 적게 정하면 모든 손님을 입력할 수 없음
  - 데이터 삽입과 삭제: 데이터를 중간에 삽입하려 해도 빈칸이 순서대로 채워짐

44 / 66

### 03. 자료구조와 알고리즘

#### 1. 자료구조

##### ▪ 배열 array

0	13시 10분	김영철 2
1	13시 15분	홍길동 3
2	13시 20분	박범순 4
3	13시 22분	배영준 4
4	13시 40분	윤희철 3
5	13시 45분	박기범 3
6	13시 30분(예약)	아리수 3
7		
8		
9		
10		

순서대로  
정렬하기가  
어렵군.



도대체 명단이  
얼마나 긴 거야?



그림 7-29 배열

45 / 66

### 03. 자료구조와 알고리즘

#### 1. 자료구조

##### ▪ 연결 리스트 linked list

- 데이터가 포인터로 연결됨
  - 포인터: 데이터의 위치를 저장하는 자료구조, 별표( \*)로 표시함
- 연결 리스트 테이블의 크기는 동적으로 변함
  - 낭비되는 공간이 없음
- 배열은 크기를 예상할 수 있는 메모리 또는 파일 관리에 사용
- 연결 리스트는 데이터를 빈번히 삽입하거나 삭제하는 테이블에 사용

46 / 66

### 03. 자료구조와 알고리즘

#### 1. 자료구조

##### ▪ 연결 리스트 linked list

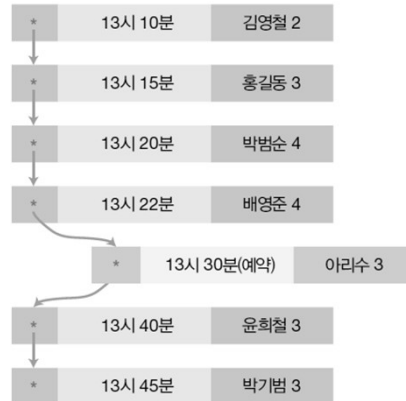


그림 7-30 연결 리스트

47 / 66

### 03. 자료구조와 알고리즘

#### 2. 2진 탐색 알고리즘

##### • 스무고개 게임 구현

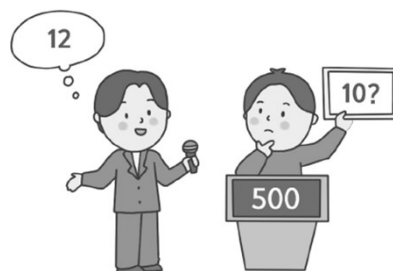


그림 7-31 스무고개

48 / 66



### 03. 자료구조와 알고리즘

#### 2. 2진 탐색 알고리즘

- 분해: 문제에서 출제자와 참가자의 역할을 나누기
- 패턴: 참가자는 특정 숫자를 말하고  
출제자는 숫자가 맞을 때까지 '크다' 혹은 '작다'를 반복
- 알고리즘을 의사 코드로 표현

다음은 무한 반복

IF(참가자 숫자 = 출제자 숫자) : 종료

ELSE IF(참가자 숫자 > 출제자 숫자) : '크다' 출력

ELSE : '작다' 출력

출제자는 특정 숫자를 생각한다. 그리고 참가자 숫자가 출제자 숫자와 같으면 프로그램이 종료된다.

같지 않으면, 참가자 숫자가 출제자 숫자보다 클 때 '크다'를 출력한다.

같지도 크지도 않으면, 참가자 숫자가 작은 경우만 남는다. 그러므로 '작다'를 출력한다.

이러한 작업을 반복한다.

그림 7-32 스무고개의 의사 코드

49 / 66

### 03. 자료구조와 알고리즘

#### 2. 2진 탐색 알고리즘

- 컴퓨팅 사고가 아닌 경우
  - 참가자가 1부터 차례대로 말하기

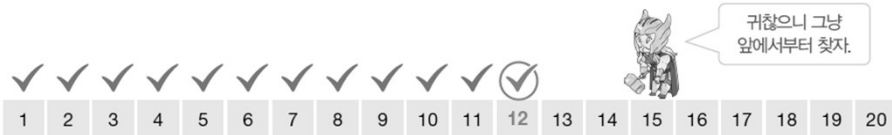


그림 7-33 1부터 차례대로 말하는 문제 해결 방식

50 / 66

### 03. 자료구조와 알고리즘

#### 2. 2진 탐색 알고리즘

- 분해 정복 divide and conquer
  - 문제를 작은 문제로 분해하여 이를 해결하고, 해결된 작은 문제를 결합하여 큰 문제를 해결하는 방식
- 2진 탐색 binary search
  - 전체 숫자를 계속 반으로 나누어 비교해가는 방식
  - 분해 정복 중 단순한 방식

51 / 66

### 03. 자료구조와 알고리즘

#### 2. 2진 탐색 알고리즘



그림 7-34 2진 탐색 알고리즘의 문제 해결 방식

52 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

- 정렬 알고리즘 sort algorithm
  - 주어진 데이터를 순서대로 정리하는 알고리즘

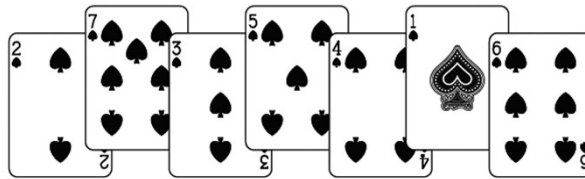


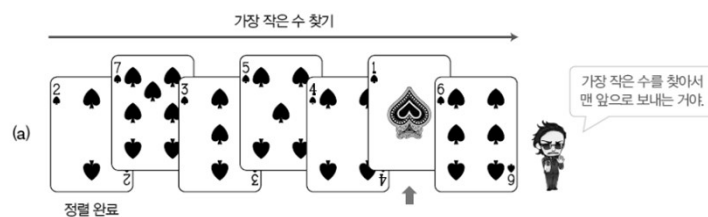
그림 7-35 숫자가 섞여 있는 카드

53 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

- 선택 정렬 selection sort
  - 카드를 뒤져 가장 작은 수를 선택하고 빼놓는 과정을 반복



54 / 66

## 03. 자료구조와 알고리즘

### 3. 정렬 알고리즘

#### ▪ 선택 정렬 selection sort

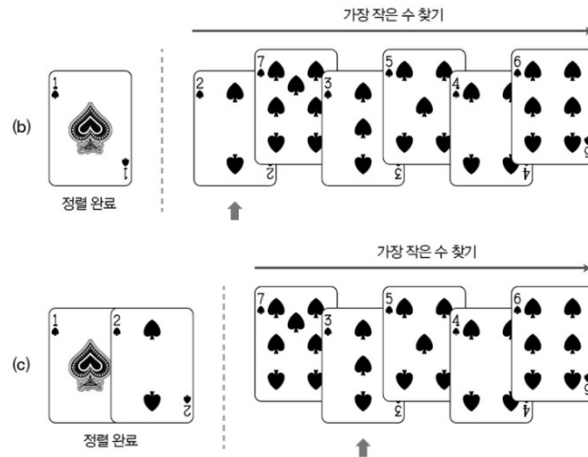


그림 7-36 선택 정렬

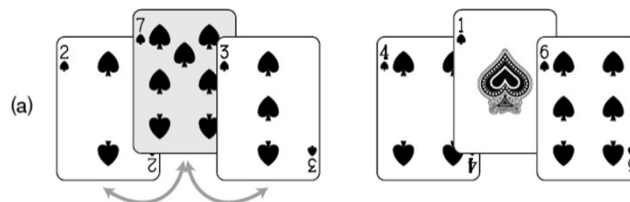
55 / 66

## 03. 자료구조와 알고리즘

### 3. 정렬 알고리즘

#### ▪ 버블 정렬 bubble sort

- 이웃한 카드 2개의 수를 비교하여  
뒤의 수가 더 작으면 카드의 순서를 바꾸는 과정을 반복



56 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

##### ▪ 버블 정렬 bubble sort

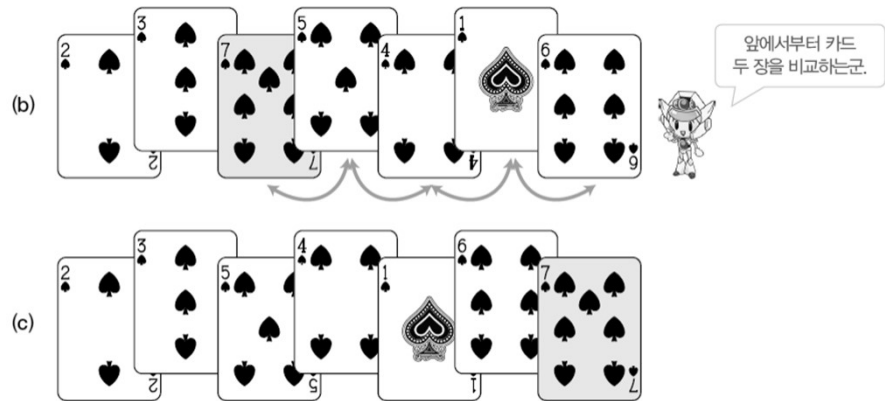


그림 7-37 버블 정렬

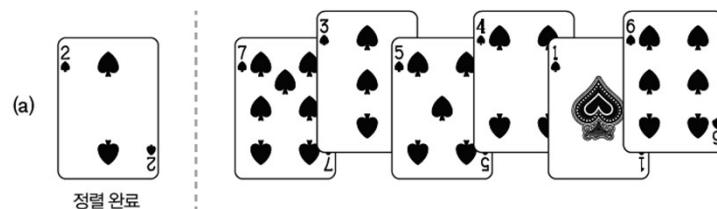
57 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

##### ▪ 삽입 정렬 insertion sort

- 정렬을 완료한 카드와 아직 정렬하지 않은 카드를 나누어 새로운 카드를 정렬을 완료한 카드 사이에 삽입



58 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

##### ▪ 삽입 정렬 insertion sort

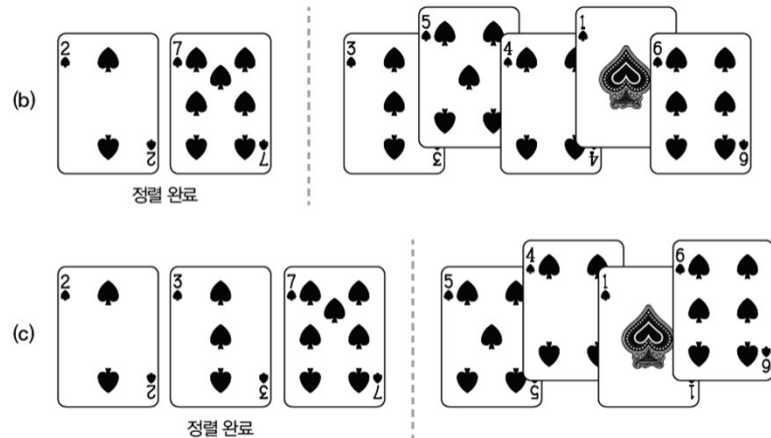


그림 7-38 삽입 정렬

59 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

##### ▪ 버킷 정렬 bucket sort

- 정렬할 데이터를 버킷에 넣었다가 빼기만 하면 정렬이 끝남
- 기존 정렬 알고리즘과 달리 비교 연산이 없음

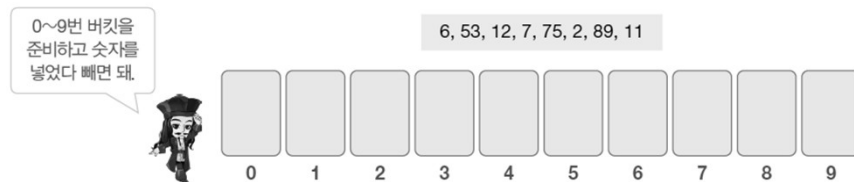


그림 7-39 버킷 정렬의 초기 상태

60 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

##### ▪ 버킷 정렬 bucket sort

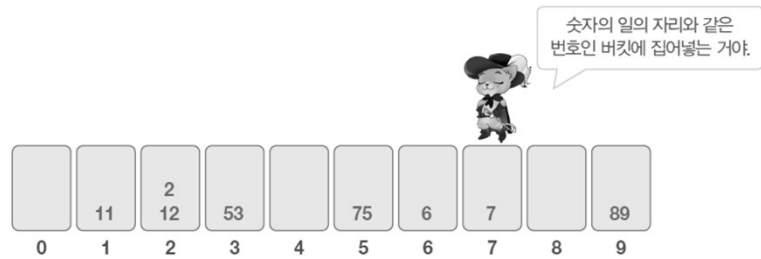


그림 7-40 일의 자리를 보고 버킷에 집어넣은 상태

61 / 66

### 03. 자료구조와 알고리즘

#### 3. 정렬 알고리즘

##### ▪ 버킷 정렬 bucket sort

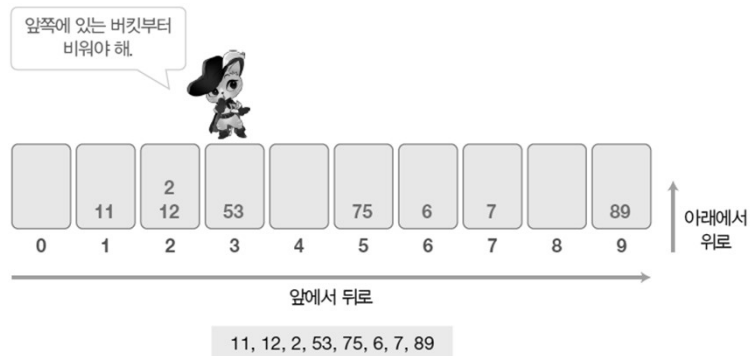


그림 7-41 버킷에 들어 있는 숫자를 빼내는 과정

62 / 66

## 03. 자료구조와 알고리즘

### 3. 정렬 알고리즘

#### ▪ 버킷 정렬 bucket sort



그림 7-42 십의 자리와 같은 번호인 버킷에 수를 넣고 빼내어 정렬 완료

63 / 66

## 03. 자료구조와 알고리즘

### 3. 정렬 알고리즘

#### ▪ 정렬 알고리즘과 시간 복잡도

- 빅오 표기법 big-O notation
  - 알고리즘의 시간 복잡도를 표현
  - 입력 데이터의 크기에 따라 알고리즘의 중심 연산이 얼마나 증가하는지 나타냄
- 선택 정렬, 버블 정렬, 삽입 정렬은  $O(n^2)$  알고리즘
  - 전체 데이터가  $n$ 개일 때 계산할 양이  $n^2$
- 버킷 정렬은  $O(n)$  알고리즘

64 / 66

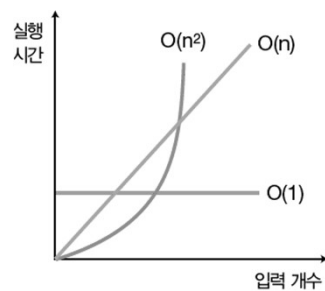


## 03. 자료구조와 알고리즘

### 3. 정렬 알고리즘

#### ▪ 정렬 알고리즘과 시간 복잡도

- $O(n)$  알고리즘과 비교하여  $O(n^2)$  알고리즘의 실행 시간이 급격히 증가



- $O(n)$ : 알고리즘의 수행 횟수가  $n$ 만큼 커진다.
- $O(n^2)$ : 알고리즘의 수행 횟수가  $n^2$ 만큼 커진다.

그림 7-43 입력 개수에 대한 실행 시간의 변화

65 / 66

# Thank You!