

정보

Sangwook Lee
Duil Middle School



저작자를 밝히면 이용이 가능하지만, 영리 목적으로 이용할 수 없으며, 내용을 수정해서도 안 됩니다

III 알고리즘과 프로그래밍

1 알고리즘

2 프로그래밍

1 알고리즘

01 문제의 이해와 추상화

02 알고리즘 기초

03 알고리즘 설계

02 알고리즘 기초 (86쪽)

- 알고리즘(algorithm)
 - 문제를 해결하는 절차나 방법
- 생활 속의 알고리즘
 - 음식을 만드는 과정
 - 잊어버린 비밀번호를 찾는 과정



02 알고리즘 기초 (87쪽)

- 알고리즘이 갖추어야 할 조건
 - 입력: 입력 과정이 있어야 한다
 - 출력: 출력 과정이 있어야 한다
 - 명확성: 과정이 명확해야 한다
 - 수행 가능성: 과정이 실행 가능해야 한다
 - 유한성: 종료해야 한다

Q. 자판기 작동 과정에서 입력과 출력은?

입력: 화폐 투입, 버튼 입력

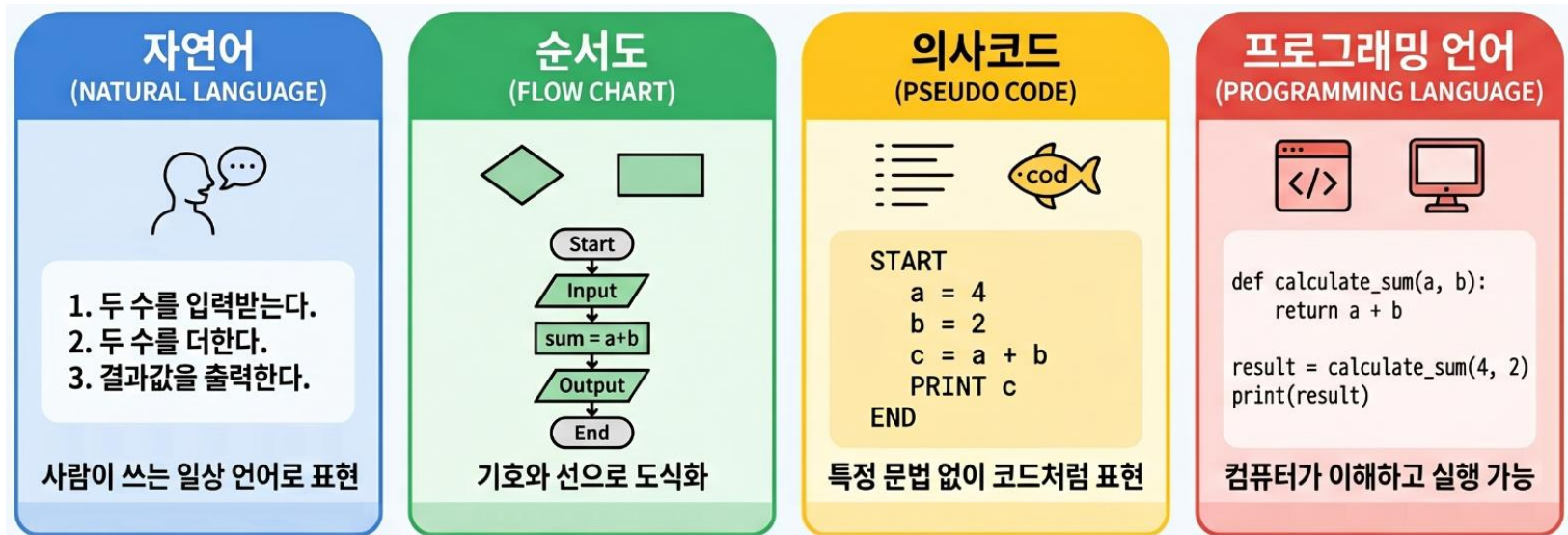
출력: 상품 배출, 거스름돈 반환



02 알고리즘 기초

• 알고리즘 표현 방법

- 자연어
- 순서도
- 의사코드
- 프로그래밍 언어



02 알고리즘 기초

- 자연어

- 사람들이 사용하는 언어

<라면 끓이는 알고리즘>

냄비에 물을 넣고

냄비를 가스레인지에 올리고

가스레인지 불을 켜고

물이 끓으면 면과 스프를 넣고

다 익을 때까지 기다리고

가스레인지 불을 끈다

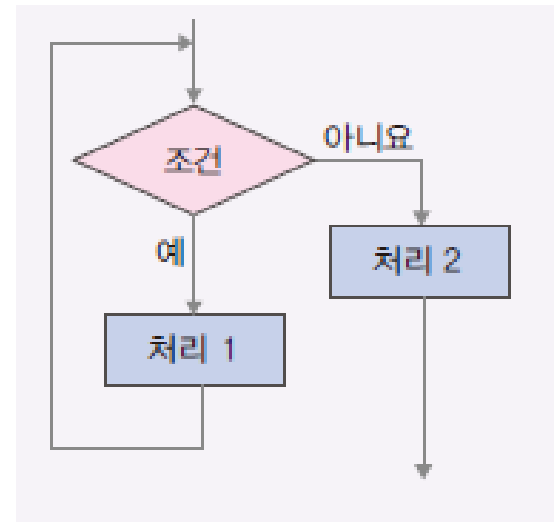
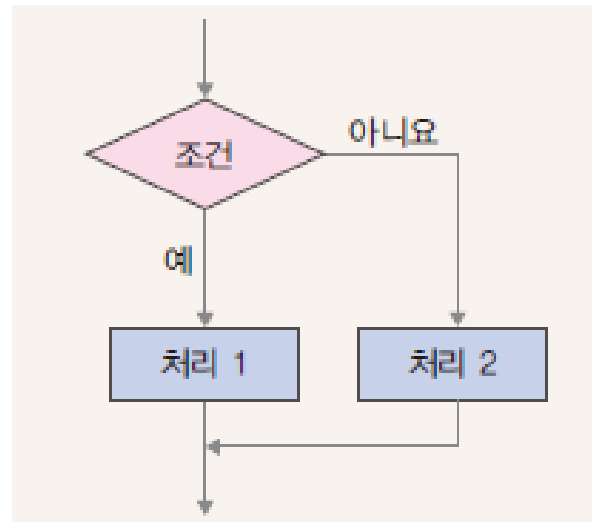
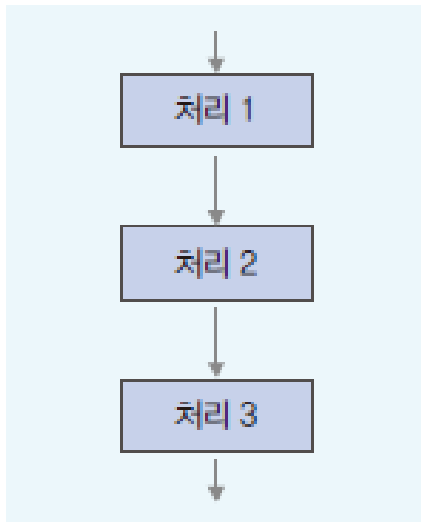


02 알고리즘 기초

- 순서도

- 의미를 가진 여러 가지 도형을 화살표로 연결한 것

도형 안에는 자연어나 프로그래밍 언어에서 사용하는 단어나 문장이 올 수 있음



02 알고리즘 기초

- 의사코드

- 단어나 문장을 프로그래밍 언어의 형태로 작성한 것



Derp Johnson
@DerpyJohn

```
while (!morning){  
  alcohol++  
  dance++  
} #mylife #partyhard
```

아침이 올 때까지 술 마시고 춤 추는 과정을 작성한 의사코드

02 알고리즘 기초

- 프로그래밍 언어
 - 컴퓨터가 이해하고 처리할 수 있는 언어



다양한 프로그래밍 언어

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int arr[5];
7     printf("%d\n", arr[5]);
8 }
```

C 프로그래밍 언어 예

02 알고리즘 기초 (88쪽)

- 효율적인 알고리즘 (= 좋은 알고리즘)
 - 수행 시간이 짧은 알고리즘
 - 자원(인력, 장비, 저장 공간 등)을 적게 사용하는 알고리즘

알고리즘의 수행 시간은

중요한 작업이나 연산의 최대 실행 횟수를 계산하여 구함

결과가 동일한 두 알고리즘 중에서

중요한 작업이나 연산의 최대 실행 횟수가 적은 알고리즘이
더 좋은 알고리즘이다

알고리즘 비교 예제 (88쪽)

- [문제] 7개의 제품 중 불량으로 무게가 가벼운 1개의 제품을 양팔 저울을 사용하여 찾으려고 한다. 아래 두 알고리즘 중 더 빠른 알고리즘은?

알고리즘 1: 양팔 저울에 순서대로 두 개씩 올려 비교하면서 찾기

알고리즘 2: 양팔 저울에 반씩 나누어 올려 비교하면서 찾기



알고리즘 비교 예제 (88쪽)

• 알고리즘 1

• 1단계 1번, 2번 제품 비교



⇒
무게가 다름.

1 2
[Green Cube] 혹은 [Yellow Cube]
양팔 저울이 올라간 쪽이 가벼운 제품임.

1번 만에
찾기 완료

• 2단계 3번, 4번 제품 비교

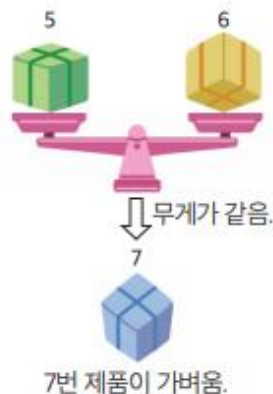


⇒
무게가 다름.

3 4
[Blue Cube] 혹은 [Red Cube]
양팔 저울이 올라간 쪽이 가벼운 제품임.

2번 만에
찾기 완료

• 3단계 5번, 6번 제품 비교



⇒
무게가 다름.

5 6
[Green Cube] 혹은 [Yellow Cube]
양팔 저울이 올라간 쪽이 가벼운 제품임.

3번 만에
찾기 완료

↑
최대 비교
횟수 3회

알고리즘 비교 예제 (88쪽)

• 알고리즘 2

• 1단계 세 개씩 나누어 비교(1~3, 4~6번 제품 비교)



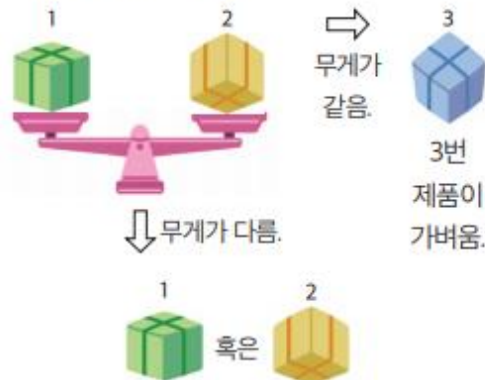
⇒
무게가 같음.



1번 만에
찾기 완료

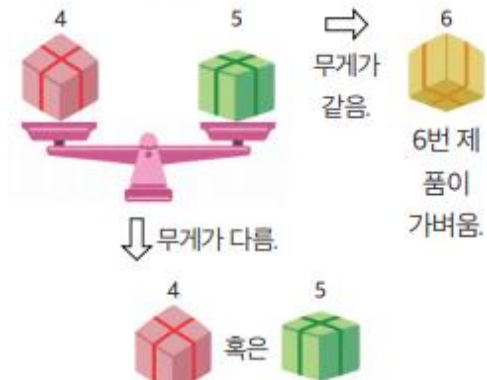
• 2단계 1단계에서 가벼운 쪽 제품 중 두 개의 제품 무게 비교

• 1~3번 제품이 가볍다면



양팔 저울이 올라간 쪽이 가벼운 제품임.

• 4~6번 제품이 가볍다면



양팔 저울이 올라간 쪽이 가벼운 제품임.

2번 만에
찾기 완료

↑
최대 비교
횟수 2회

알고리즘 비교 예제 (88쪽)

- [정답] 알고리즘 2

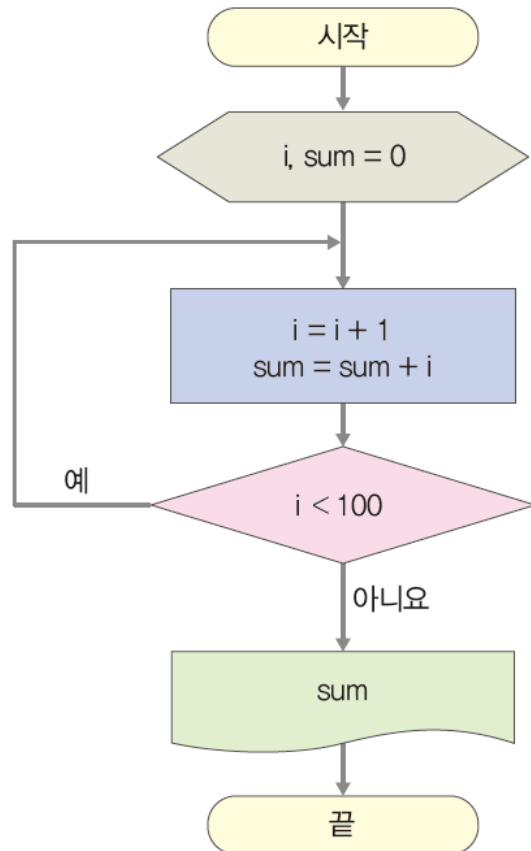
알고리즘 1과 2에서 중요한 작업은
저울을 사용하여 비교하는 작업이며

알고리즘 1보다 알고리즘 2가
저울을 사용하여 비교하는 작업의 최대 횟수가 더 적기 때문에
알고리즘 2가 더 좋은 알고리즘이다

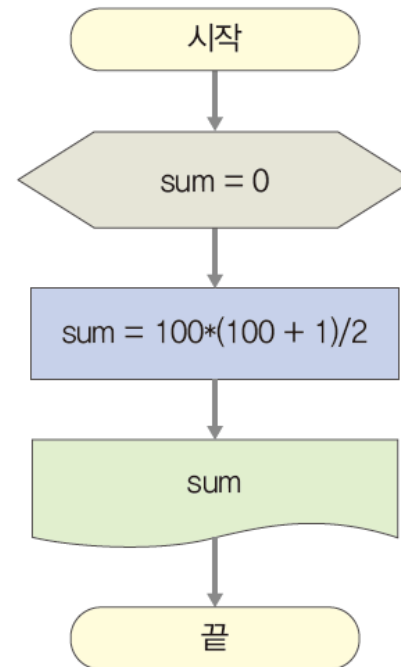
프로그래밍으로 알고리즘의 효율성 살펴보기 (90쪽)

- 1부터 100까지 자연수의 합을 구하는 알고리즘

알고리즘 1



알고리즘 2



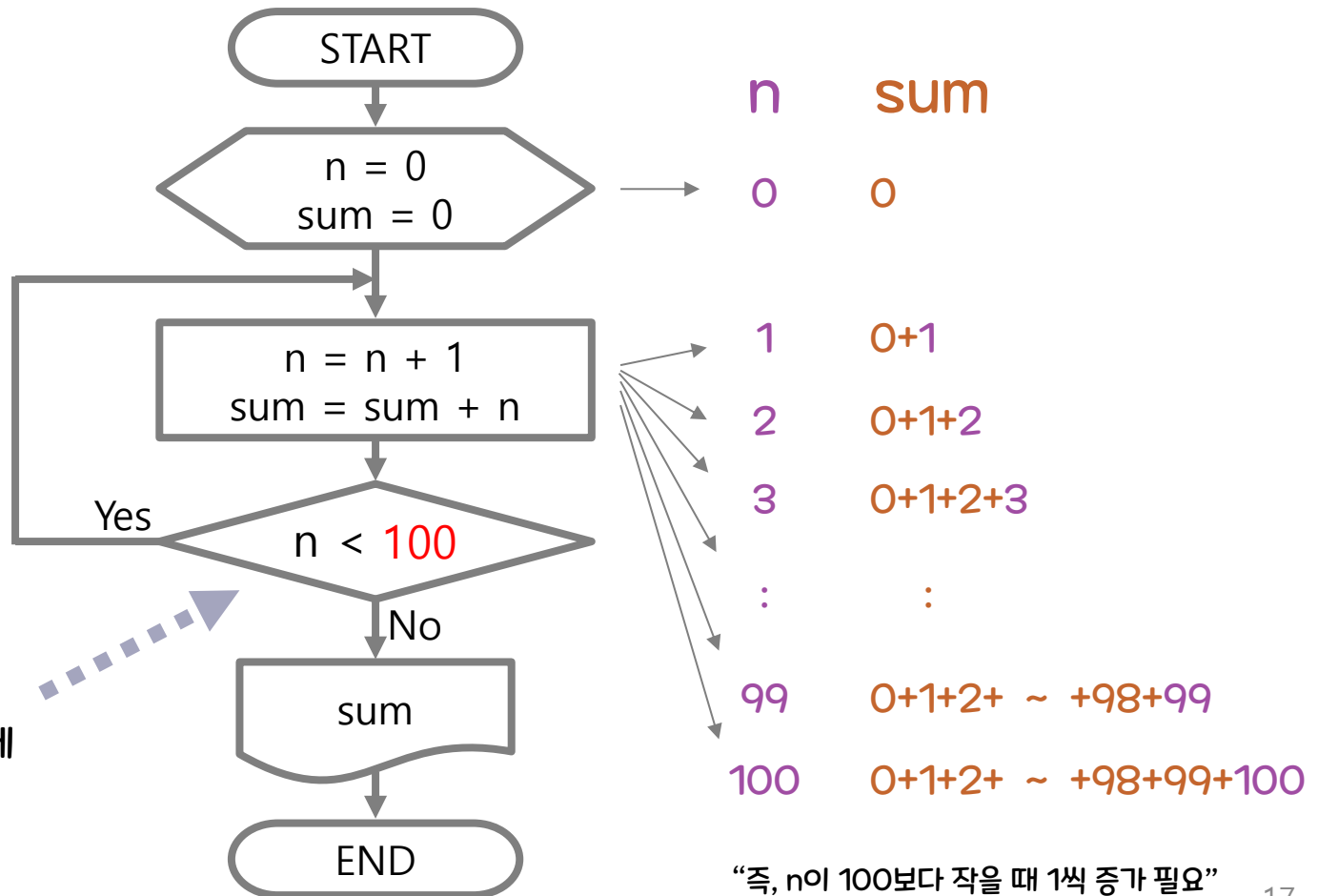
프로그래밍으로 알고리즘의 효율성 살펴보기 (90쪽)

• 알고리즘 1

==	값이 같은지 비교
!=	값이 다른지 비교
<	미만
>	초과
<=	이하
>=	이상

[비교 연산자]

판단 기호 속
미만(<) 오른쪽에
들어갈 내용은?



프로그래밍으로 알고리즘의 효율성 살펴보기 (90쪽)

알고리즘 1	알고리즘 2
덧셈을 반복적으로 수행	덧셈, 곱셈, 나눗셈 연산을 한 번만 수행

알고리즘 2는
가우스 덧셈 공식을 이용하여
1부터 100까지 정수의 합을
구함



가우스의 덧셈 공식

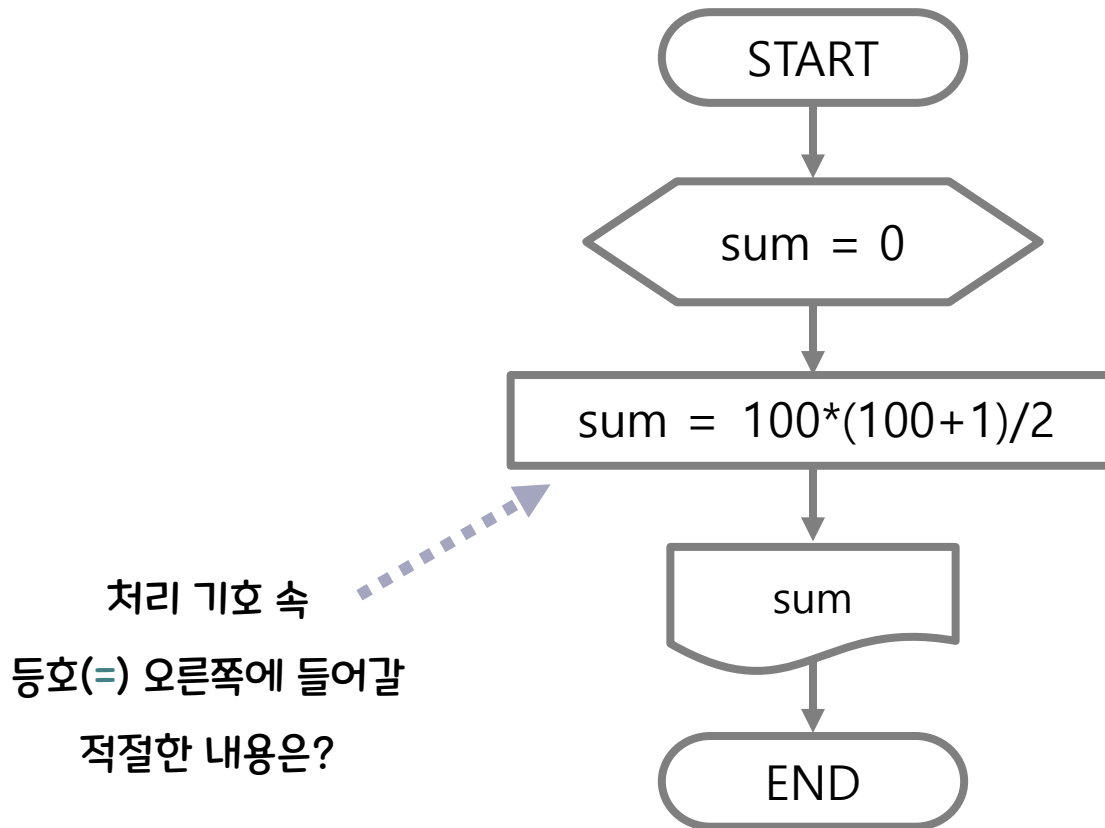
1부터 100까지의 합을 구하는
가우스의 계산 방식을 살펴보면,

$$\begin{array}{r} 1 + 2 + 3 \cdots \cdots 99 + 100 \\ 100 + 99 + 98 \cdots \cdots 2 + 1 \\ \hline 101 + 101 + 101 \cdots \cdots 101 + 101 \end{array}$$

이므로 $101 \times 100 = 10100$ 이고,
1부터 100까지 두 번씩 더했기
때문에 10100을 2로 나누면 답
은 5050이다. 따라서, 가우스의
덧셈 공식은 일반식 $\frac{n(n+1)}{2}$ 로
나타낼 수 있다.

프로그래밍으로 알고리즘의 효율성 살펴보기 (90쪽)

• 알고리즘 2



1 알고리즘

01 문제의 이해와 추상화

02 알고리즘 기초

03 알고리즘 설계