

A03. 알고리즘에 익숙해지기

알고리즘



- 원광대학교 컴퓨터소프트웨어공학과
- 2019학년도 2학기 화6수78
- 알고리즘 / 374015-01

목차

- 알고리즘에 익숙해지기
 - 01. 삼각형의 면적을 계산하는 알고리즘
 - 02. 두 데이터의 대소를 판단하는 알고리즘
 - 03. 두 변수의 데이터를 교환하는 알고리즘
 - 04. 합계값을 계산하는 알고리즘
 - 05. 최댓값을 찾는 알고리즘



01. 삼각형의 면적을 계산하는 알고리즘

- Point

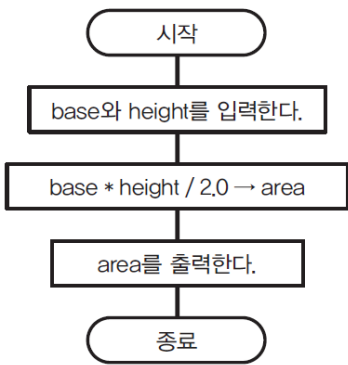
- 아는 것부터 처리로 분해하여 절차를 생각한다.
- 사칙연산 처리에는 산술 연산자를 사용한다.
- 나눗셈을 나타내는 '/'와 '%'의 차이에 주의한다.

01. 삼각형의 면적을 계산하는 알고리즘

- 아는 것부터 처리 단위로 분해한 후 각각의 절차를 생각한다
 - 프로그램의 실행 과정을 디스플레이에 표시하는 것을 프로그래밍 언어에서는 '표준 출력으로 출력한다.'라고 말한다.
- 표준 출력
 - 표준 출력이란, 컴퓨터에서 가장 보편적으로 사용하는 출력 장치를 의미한다.
 - 대부분의 컴퓨터에는 디스플레이가 연결되어 있으므로 표준 출력이라 할 때는 대개 디스플레이를 가리킨다.

01. 삼각형의 면적을 계산하는 알고리즘

- 알고리즘을 나타내는 순서도 만들기
 - 표준 입력
 - 여기서 입력은 키보드 등의 입력 장치를 사용하여 컴퓨터에 데이터를 입력하는 것을 가리킨다.
 - 특히 키보드로 입력하는 것을 '표준 입력'이라고 한다.
 - 여기서 표준 입력은 컴퓨터에서 당연하게 사용하는 데이터 장치라는 의미다.



삼각형의 면적을 계산하는 알고리즘(순서도)

01. 삼각형의 면적을 계산하는 알고리즘

- 사칙 연산을 나타내는 산술 연산자
 - 사칙 연산이란, 더하기(덧셈), 빼기(뺄셈), 곱하기(곱셈), 나누기(나눗셈)를 말한다.

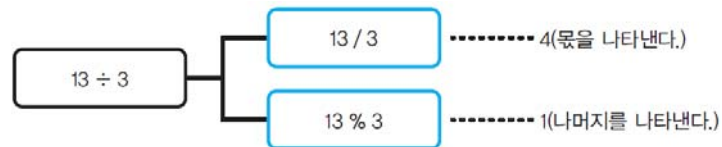
산술 연산자

연산자	의미	일반적인 기호	C언어	Java
+	덧셈으로 합을 구한다.	+	+	+
-	뺄셈으로 차를 구한다.	-	-	-
*	곱셈으로 곱한 값을 구한다.	×	*	*
/	나눗셈으로 몫을 구한다.	÷	/	/
%	나눗셈으로 나머지 값을 구한다.	나머지(...)	%	%

※ 나눗셈에서 나머지를 구할 때 '%'가 아닌 'Mod'를 사용하는 프로그래밍 언어도 있다.

01. 삼각형의 면적을 계산하는 알고리즘

- 사칙 연산을 나타내는 산술 연산자



01. 삼각형의 면적을 계산하는 알고리즘

- 알고리즘을 의사 언어로 작성하기

- 실수형: base, height, area
 - base와 height를 입력한다.
 - $area \leftarrow base * height / 2.0$
 - area를 출력한다.

- 삼각형의 면적을 계산하는 알고리즘(의사 언어)

02. 두 데이터의 대소를 판단하는 알고리즘

- Point

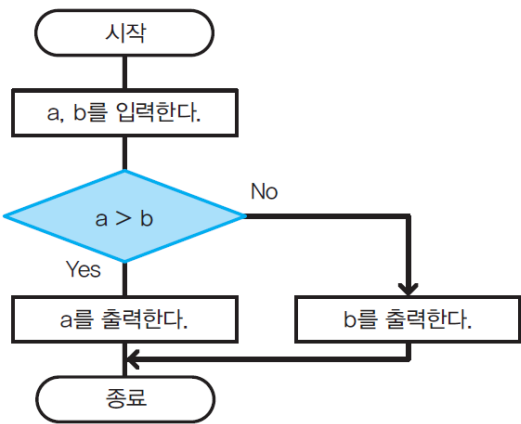
- 2개의 데이터를 비교하려면 선택 구조를 사용한다.
- 조건식에서는 관계 연산자를 사용한다.

02. 두 데이터의 대소를 판단하는 알고리즘

- 2개의 데이터 중에서 큰 것은 어느 쪽일까?

02. 두 데이터의 대소를 판단하는 알고리즘

- 알고리즘을 나타내는 흐름 만들기



- 두 데이터의 대소를 판정하는 알고리즘(순서도)

02. 두 데이터의 대소를 판단하는 알고리즘

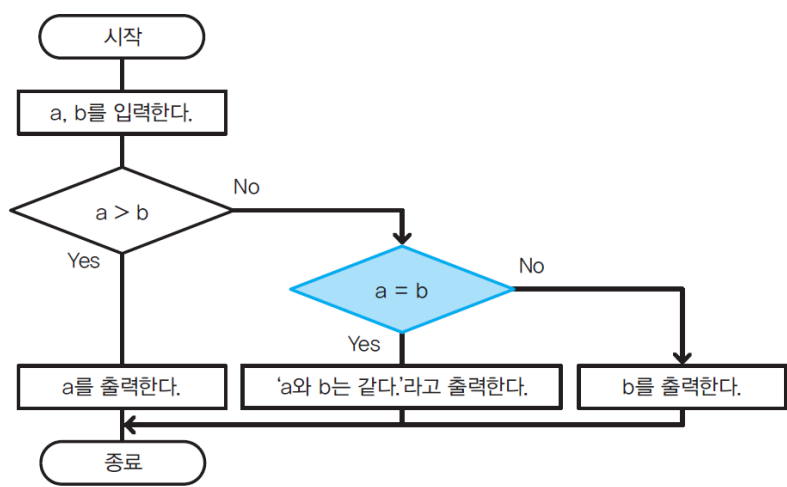
- 데이터를 비교하는 관계 연산자

- 관계 연산자

연산자	의미	일반적인 기호	C언어	Java
=	a와 b는 같은가?	a = b	a == b	a == b
≠	a와 b는 같지 않은가?	a ≠ b	a != b	a != b
>	a는 b보다 큰가?	a > b	a > b	a > b
<	a는 b보다 작은가?	a < b	a < b	a < b
≥	a는 b보다 크거나 같은가?	a ≥ b	a >= b	a >= b
≤	a는 b보다 작거나 같은가?	a ≤ b	a <= b	a <= b

02. 두 데이터의 대소를 판단하는 알고리즘

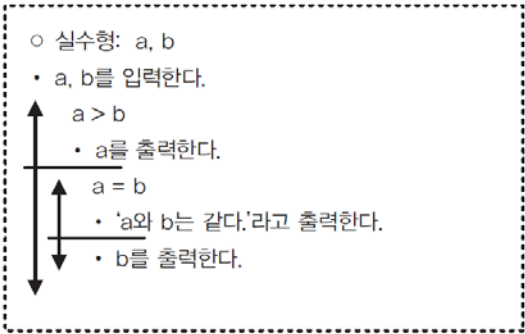
- 데이터를 비교하는 관계 연산자



- 두 데이터의 대소를 판정하는 알고리즘(순서도)

02. 두 데이터의 대소를 판단하는 알고리즘

- 알고리즘을 의사 언어로 작성하기



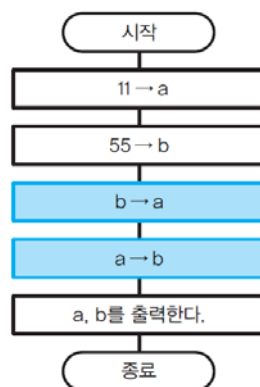
- 두 데이터의 대소를 판정하는 알고리즘(의사 언어)

03. 두 변수의 데이터를 교환하는 알고리즘

- Point
 - 두 변수의 데이터를 직접 교환할 수 없다.
 - 교환용의 변수를 사용한다.

03. 두 변수의 데이터를 교환하는 알고리즘

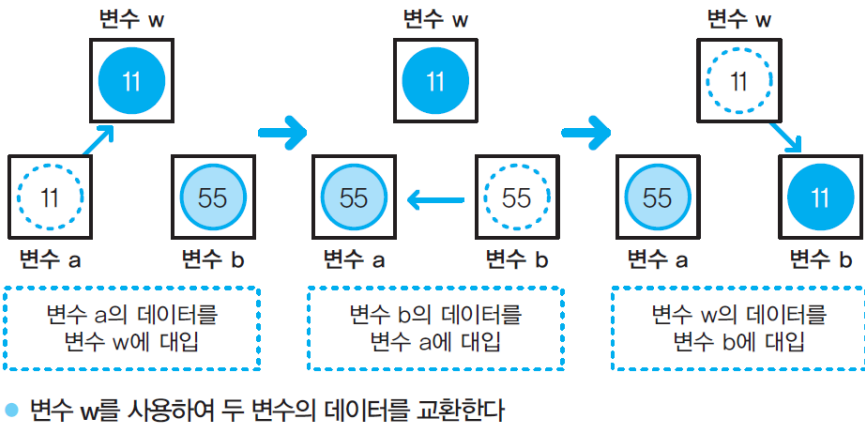
- 두 데이터를 교환하는 것은 간단한가?



- 2개의 데이터를 직접 서로 대입한다.

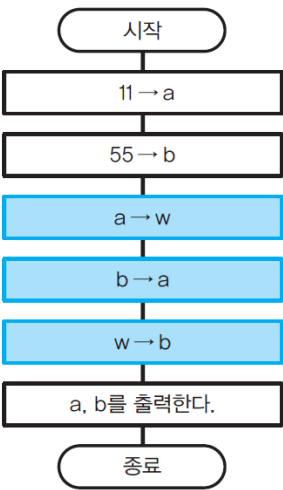
03. 두 변수의 데이터를 교환하는 알고리즘

- 데이터 교환용 변수를 사용한다



03. 두 변수의 데이터를 교환하는 알고리즘

- 데이터 교환용 변수를 사용한다



- 두 변수의 데이터를 교환하는 알고리즘(순서도)

03. 두 변수의 데이터를 교환하는 알고리즘

- 알고리즘을 의사 언어로 작성하기

```
○ 정수형: a, b, w
• a ← 11
• b ← 55
• w ← a
• a ← b
• b ← w
• a, b를 출력한다.
```

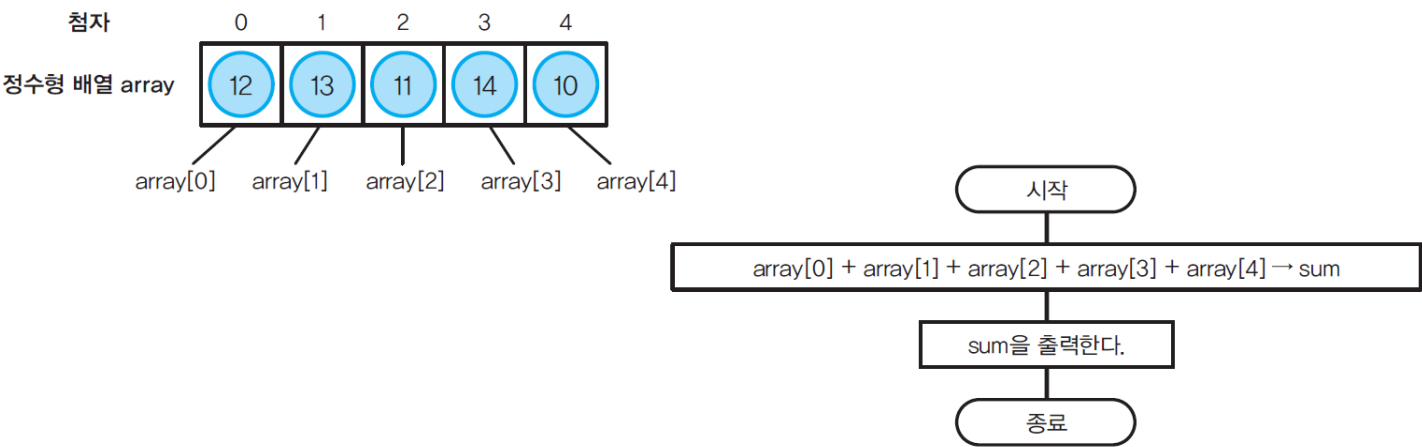
- 두 변수의 데이터를 교환하는 알고리즘(의사 언어)

04. 합계값을 계산하는 알고리즘

- Point
 - 배열 요소의 합계를 계산하려면 반복 구조를 사용해야 한다.
 - 합계값을 대입할 변수 sum은 초기화해 둔다.
 - 변화하는 첨자는 변수 i에 대입한다.
 - 반복 구조에서는 무한 루프하지 않는 처리를 잊지 말고 넣어 둔다.

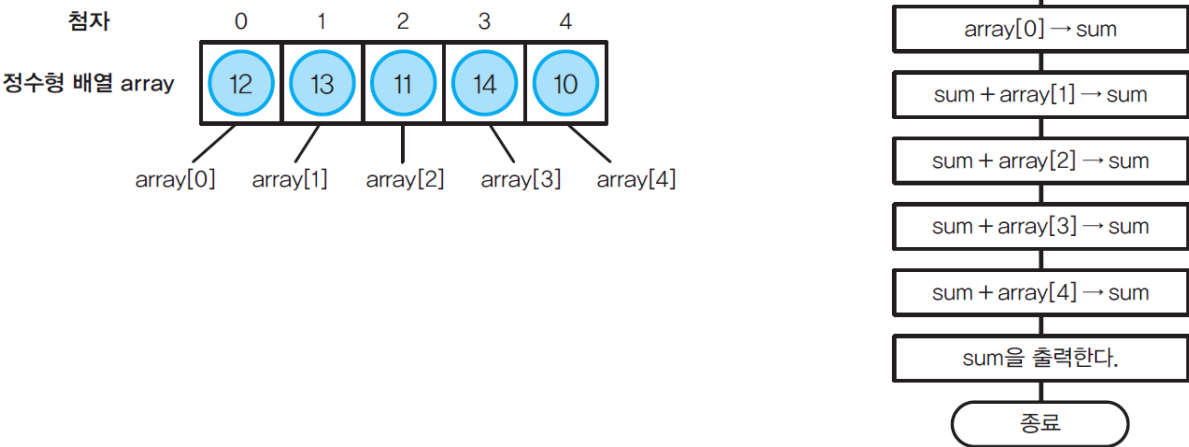
04. 합계값을 계산하는 알고리즘

- 단순히 합계를 구하는 알고리즘도 있다



04. 합계값을 계산하는 알고리즘

- 반복 구조를 사용한 알고리즘 생각해 보기

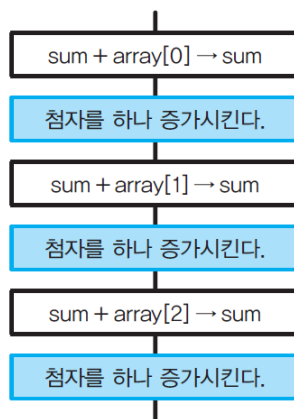


04. 합계값을 계산하는 알고리즘

- 반복 구조를 사용한 알고리즘 생각해 보기
 - 변수의 초기화
 - 왜 초기화가 필요한가?
 - 변수는 메모리에서 사용하고 있지 않은 장소를 자동으로 찾아주는 기능이다.
 - 그러나 그 장소에는 전에 사용한 데이터가 지워지지 않은 상태로 남아있을 가능성이 있다.
 - 즉, 새롭게 선언한 변수 안에는 이미 어떤 데이터가 들어가 있을지도 모른다.
 - 변수 `sum`의 값이 0일 것이라는 생각만 가지고 `array[0]`을 더하면 혹시 그 안에 남아 있는 데이터가 있을 경우 합계값이 이상해진다.
 - 이러한 일이 발생하지 않도록 초기화를 실시해야 한다.

04. 합계값을 계산하는 알고리즘

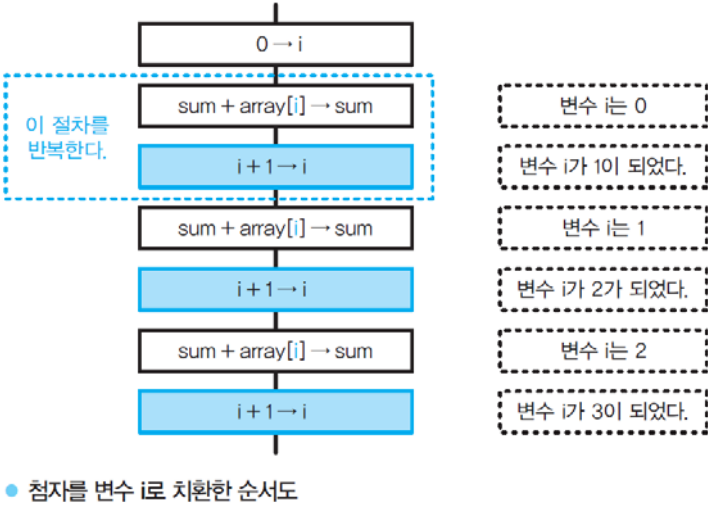
- 반복 구조를 사용한 알고리즘 생각해 보기
 - 똑같은 절차를 반복 구조로 한다



- 변화하는 첨자가 핵심

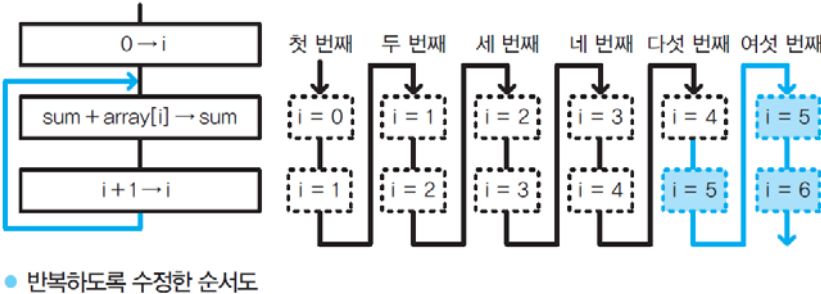
04. 합계값을 계산하는 알고리즘

- 반복 구조를 사용한 알고리즘 생각해 보기
 - 배열의 첨자를 변수 i 로 치환한다



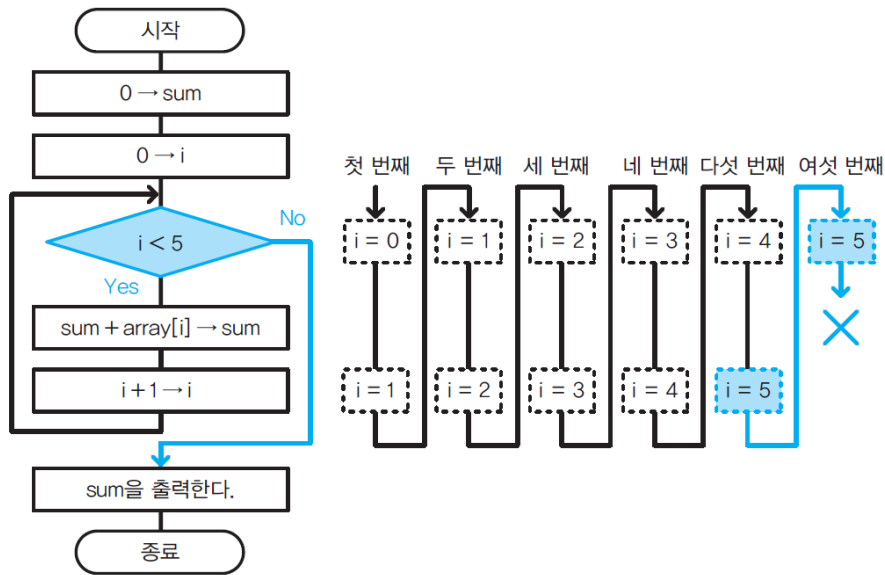
04. 합계값을 계산하는 알고리즘

- 반복 구조를 사용한 알고리즘 생각해 보기
 - 배열의 첨자를 변수 i 로 치환한다



04. 합계값을 계산하는 알고리즘

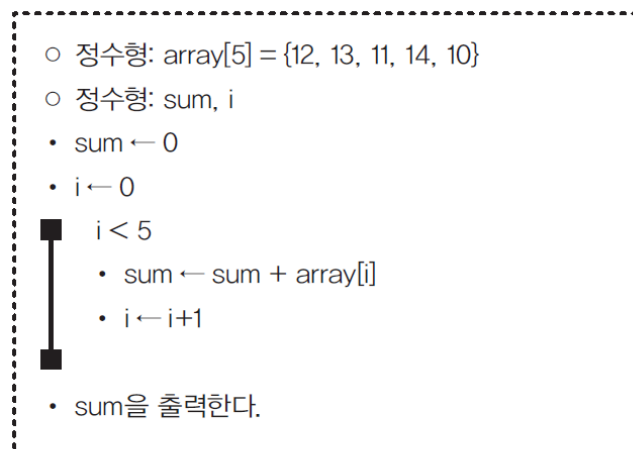
- 반복 구조를 사용한 알고리즘 생각해 보기
 - 무한 루프를 방지하는 처리를 추가한다



- 합계를 계산하는 알고리즘(순서도)

04. 합계값을 계산하는 알고리즘

- 알고리즘을 의사 언어로 작성하기



- 합계를 계산하는 알고리즘(의사 언어)

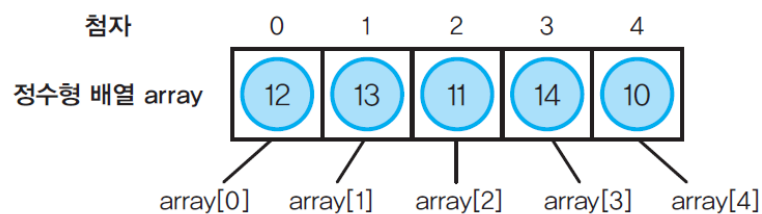
05. 최댓값을 찾는 알고리즘

- Point

- 복잡하다고 느낄 경우에는 순서도를 그리면서 알고리즘을 생각한다.
- 그때마다 최댓값을 보관하는 변수 `max`를 사용한다.
- 변화하는 첨자를 보관하는 변수 `i`를 사용한다.
- 반복 처리가 무한 루프가 되지 않도록 처리를 넣는다.

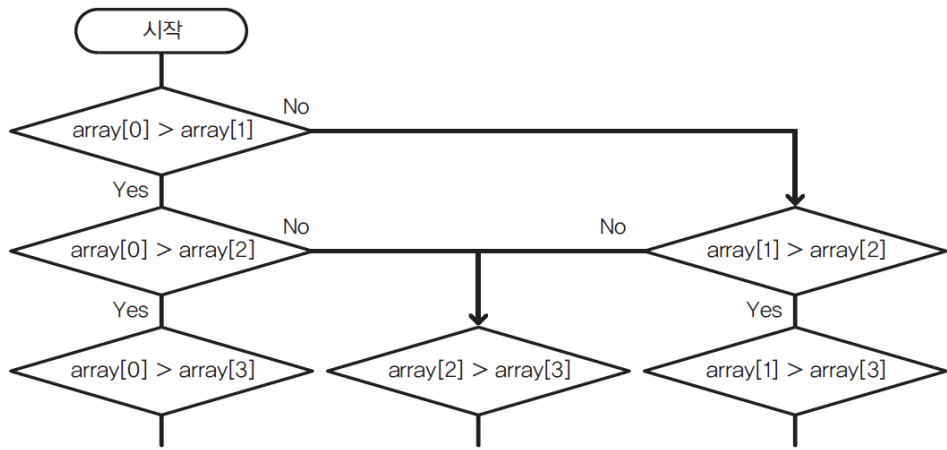
05. 최댓값을 찾는 알고리즘

- 5개의 데이터 중에서 가장 큰 값은 무엇인가?



05. 최댓값을 찾는 알고리즘

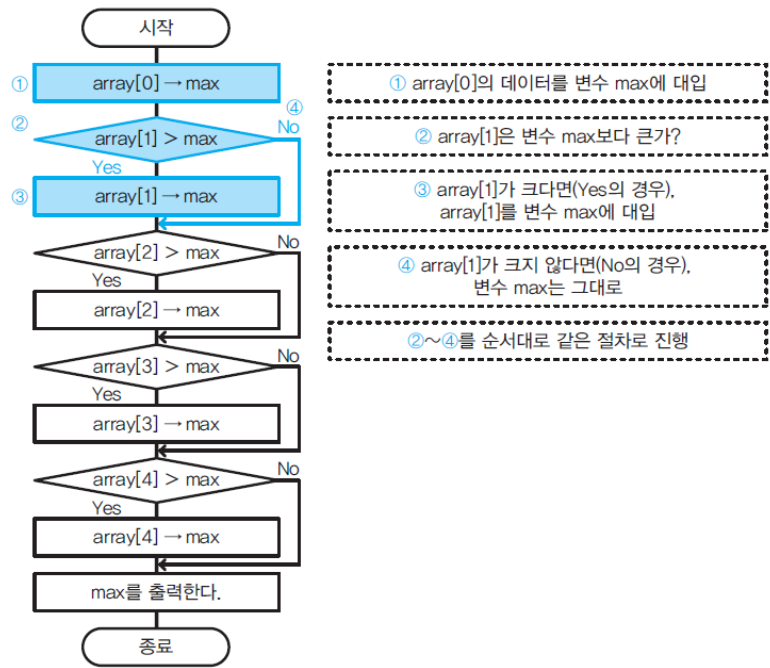
- 순서도를 만들면서 알고리즘 고려하기



• 구조가 복잡할 것 같은 순서도

05. 최댓값을 찾는 알고리즘

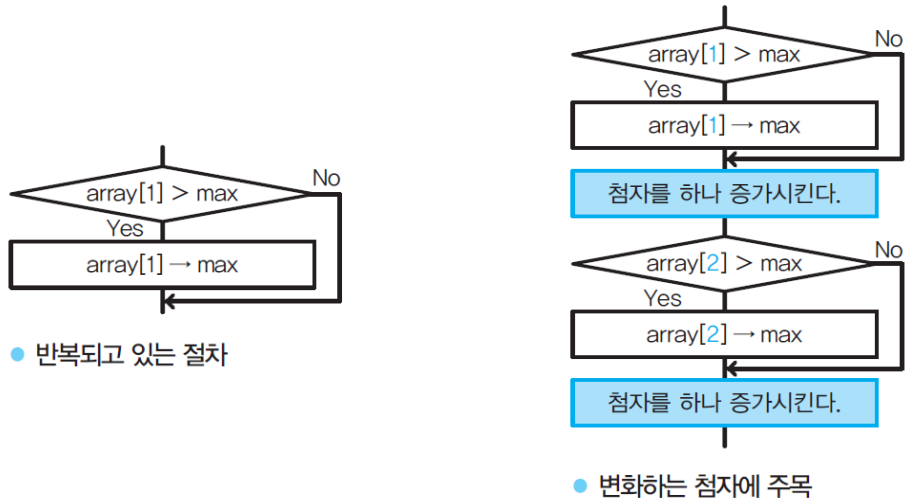
- 잠정적으로 최댓값을 보관하는 변수 max 준비하기



• 최댓값을 보관하는 변수 max를 사용한 알고리즘

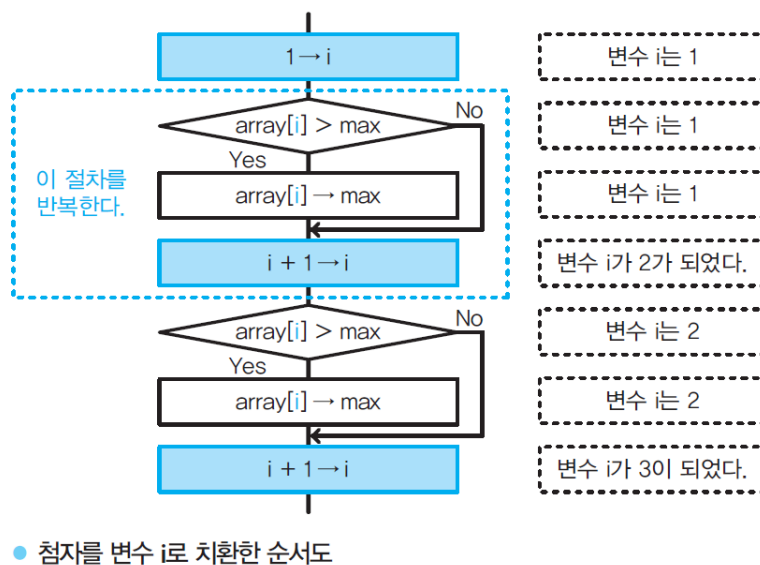
05. 최댓값을 찾는 알고리즘

- 같은 절차를 반복 구조로 하기



05. 최댓값을 찾는 알고리즘

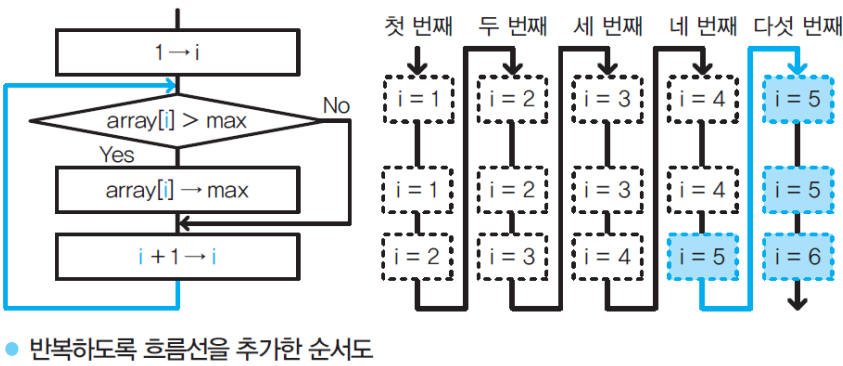
- 배열 첨자를 변수 i로 치환한다



- 첨자를 변수 i로 치환한 순서도

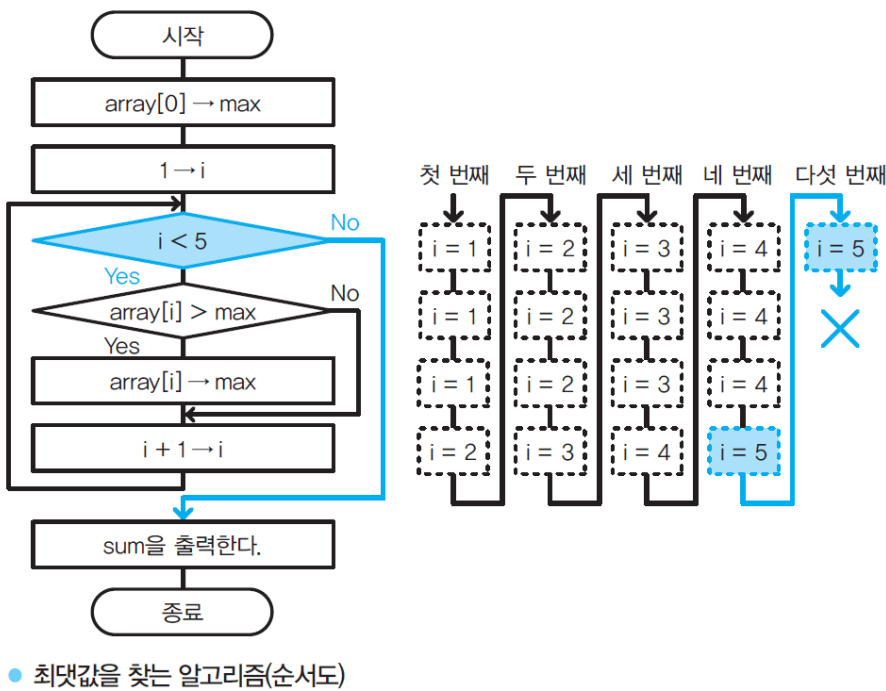
05. 최댓값을 찾는 알고리즘

- 배열 첨자를 변수 i로 치환한다



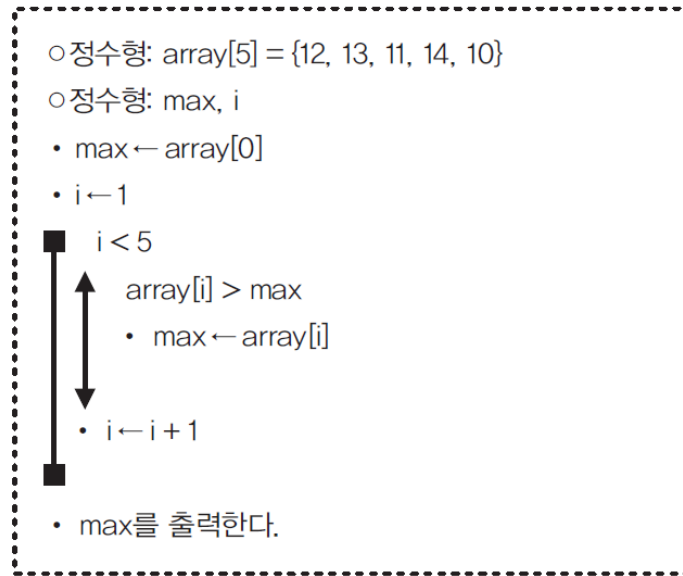
05. 최댓값을 찾는 알고리즘

- 무한 루프를 방지하는 처리를 추가한다



05. 최댓값을 찾는 알고리즘

- 알고리즘을 의사 언어로 작성한다



- 최댓값을 찾는 알고리즘(의사 언어)