/* elice */

수포자를 위한 프로그래밍 수학

알고리즘 기초 다지기

조웅오 선생님

목차

개요

2장을 배우고 나면!

1. 수열의 기본적인 형태를 알고 식으로 나타낼 수 있습니다.
2. 우리에게 필요한 수열의 식을 스스로 세울 수 있습니다.
3. 자신이 세운 식이 정확한지 판단할 수 있는 증명법을 알게 됩니다.
4. 어려운 문제를 여러 개의 쉬운 문제로 나누어 해결하는 방법을 알게 됩니다.

수열이란?

숫자의 나열

1 2 3 4 5 6 7 8…

1 1 2 3 5 8 13 21…

2 4 6 8 10 12 14…

2 3 5 7 11 13 17…

수열이란?

표기법

$$\textbf{수열 } a_n = \mathbf{2\ 3\ 5\ 7\ 11\ 13\ 17\cdots}$$

a_1 a_2 a_3 a_4 a_5

등차수열

각 항 사이에 **일정한 수가 더해지는 수열**

공차

매 항 더해지는 수

+3 +3 +3 +3 +3 +3

공차 d

3 6 9 12 15 18 21 ⋯

초항 a

수열의 가장 첫 항

등차수열

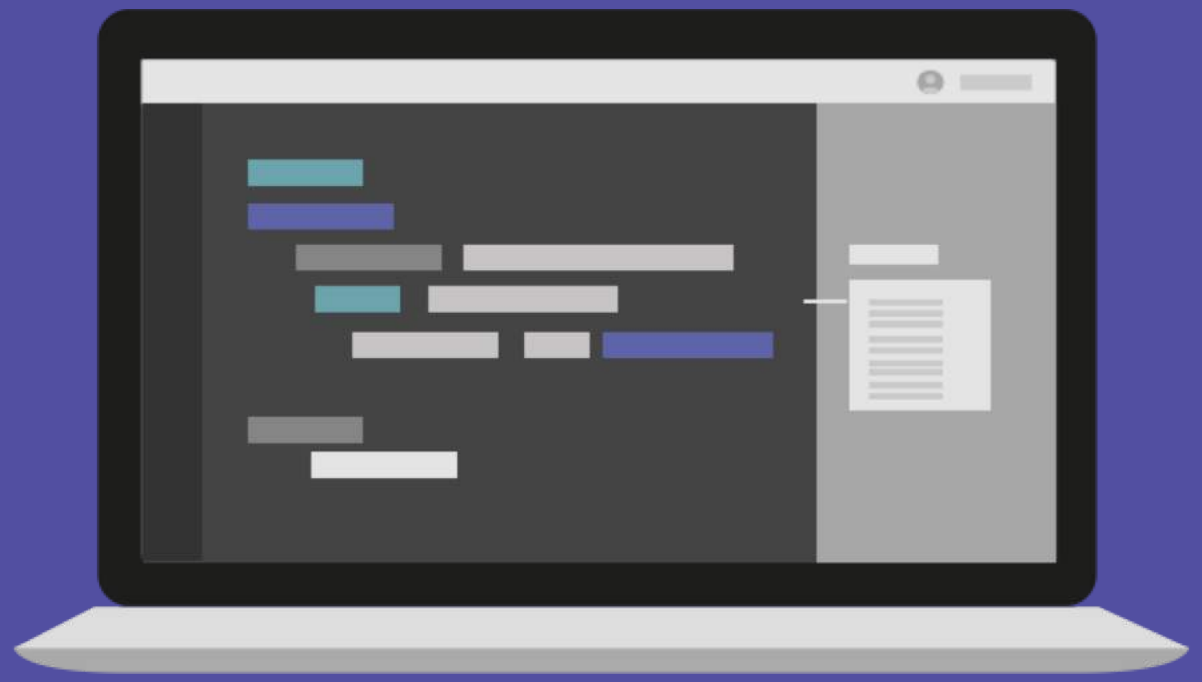
등차수열의 일반항

공차 d

$$a_n = a + (n-1)d$$

초항 a

[실습1] 등차수열



등비수열

각 항 사이에 **일정한 수가 곱해지는 수열**

공비

매 항 곱해지는 수

공비 r

x2 x2 x2 x2 x2 x2

2 4 8 16 32 64 128 ···

초항 a

수열의 가장 첫 항

등비수열

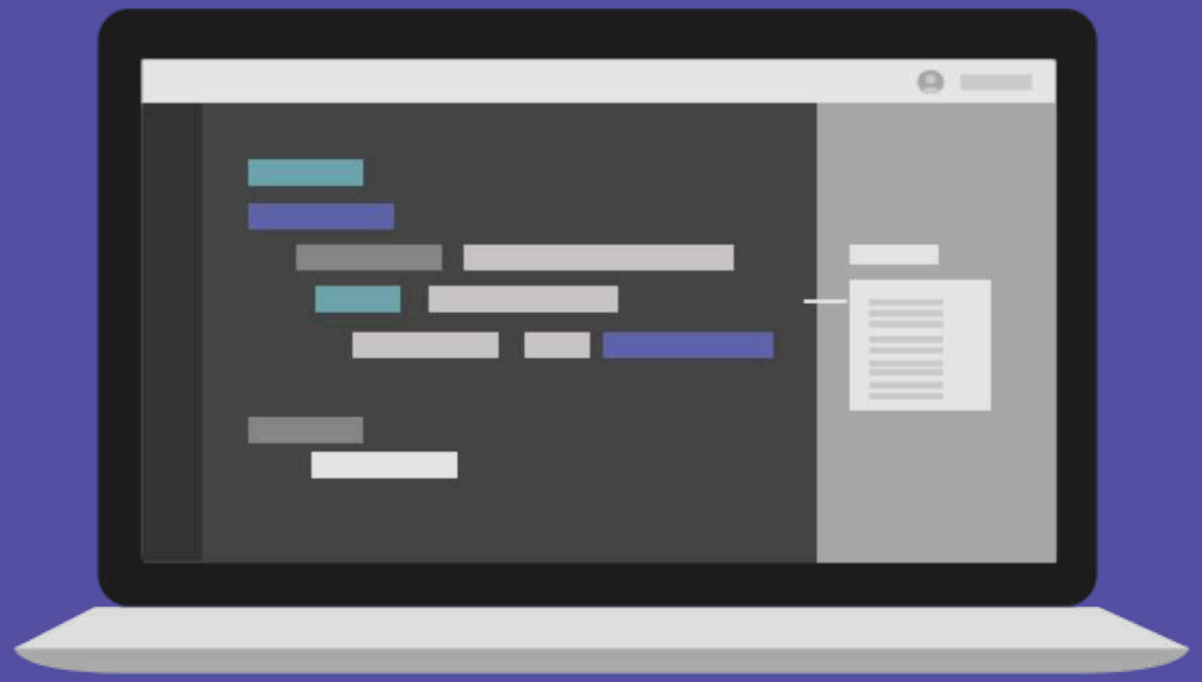
등비수열의 일반항

공비 r

$$a_n = a \times r^{n-1}$$

초항 a

[실습2] 등비수열



계차수열

각 항 사이에 **특정한 수열이 더해지는 수열**

+2 +2 +2 +2 +2

+2 +4 +6 +8 +10 +12

1 3 7 13 21 31 43 ⋯

계차수열

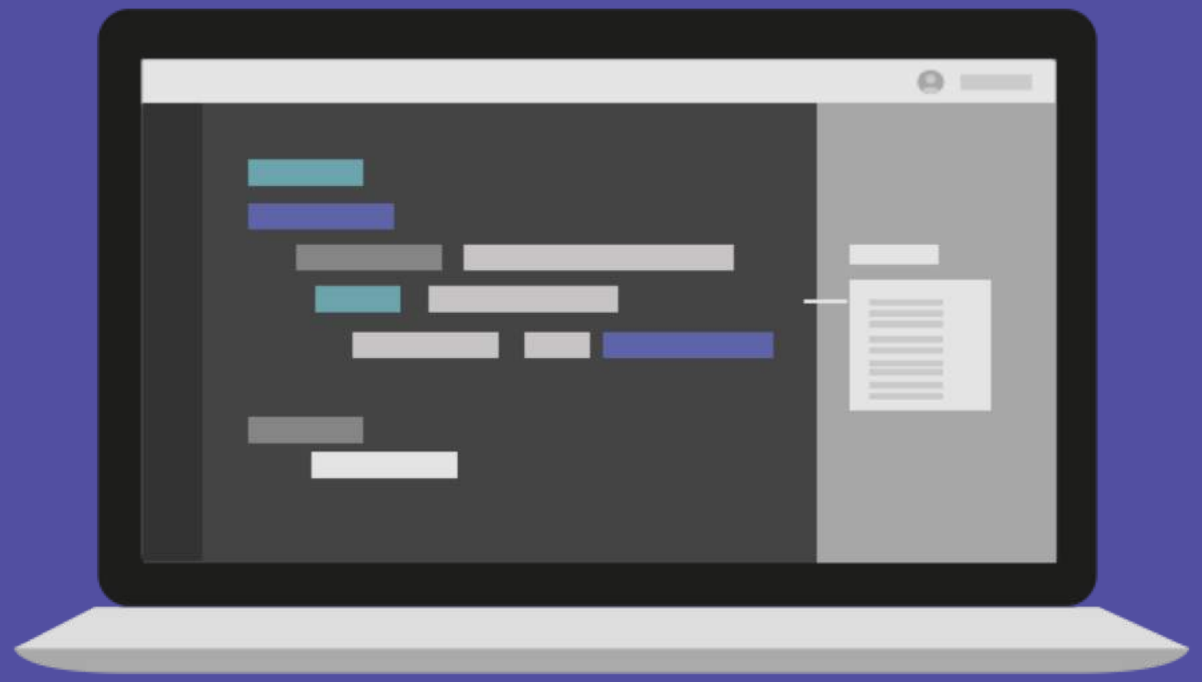
초항 a

계차수열

		b_1	b_2	b_3	b_4	b_5	b_6	
b_n		+2	+4	+6	+8	+10	+12	
a_n	1	3	7	13	21	31	43	$\cdots$
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	

$$a_n = a_1 + b_1 + b_2 + \cdots + b_{n-1}$$

[실습3] 계차수열



급수

수열의 합

$$a_n = a_1\ a_2\ a_3\ a_4\ \ \cdots$$

$$S_1 = a_1$$
$$S_2 = a_1 + a_2$$
$$S_3 = a_1 + a_2 + a_3$$
$$S_4 = a_1 + a_2 + a_3 + a_4$$

급수

시그마 표기법

$$S_n = a_1 + a_2 + \cdots + a_n$$

$$= \sum_{k=1}^{n} a_k$$

n까지

1부터

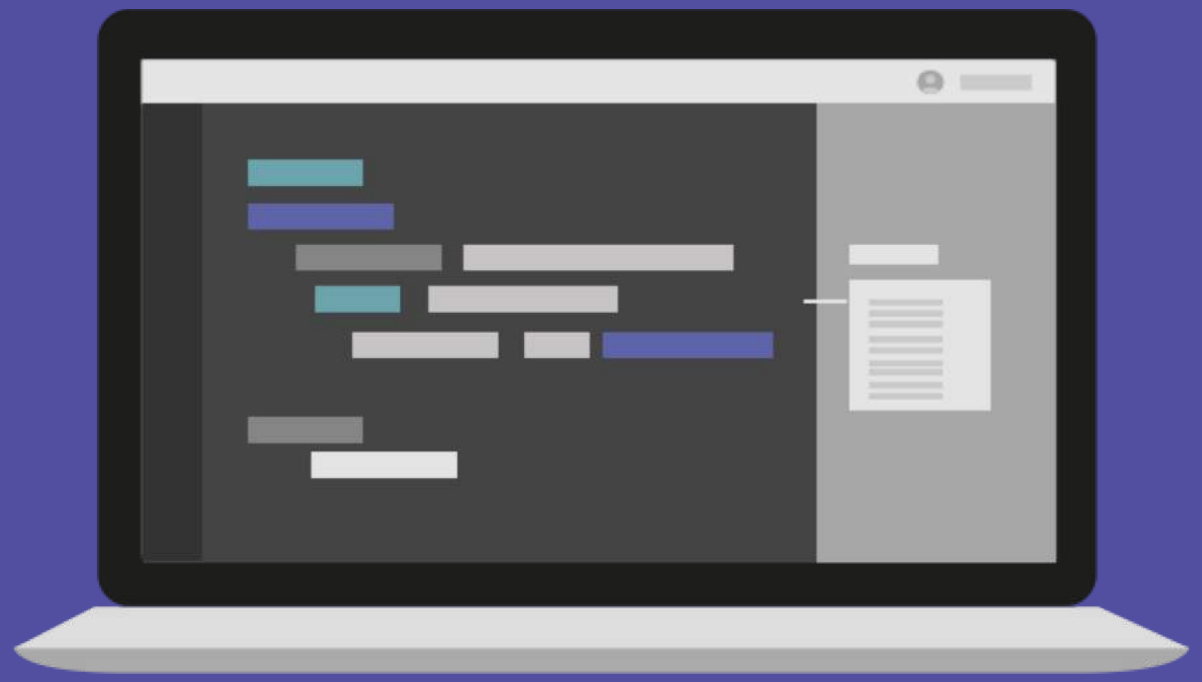
급수

		b_1		b_2		b_3		b_4		b_5		b_6		
b_n		+2		+4		+6		+8		+10		+12		
a_n	1		3		7		13		21		31		43	…
	a_1		a_2		a_3		a_4		a_5		a_6		a_7	

$$a_n = a_1 + b_1 + b_2 + \cdots + b_{n-1}$$

$$= a_1 + \sum_{k=1}^{n-1} b_k$$

[실습4] 급수



점화식

항들 간의 관계를 나타낸 식

$$a_{n+1} = f(a_n)$$

점화식

$$\underset{a_1}{1} \xrightarrow{f} \underset{a_2}{5} \xrightarrow{f} \underset{a_3}{13} \xrightarrow{f} \underset{a_4}{29} \xrightarrow{f} \underset{a_5}{61} \xrightarrow{f} \underset{a_6}{125} \quad \cdots$$

$$a_{n+1} = f(a_n) = 2a_n + 3$$

점화식

피보나치 수열

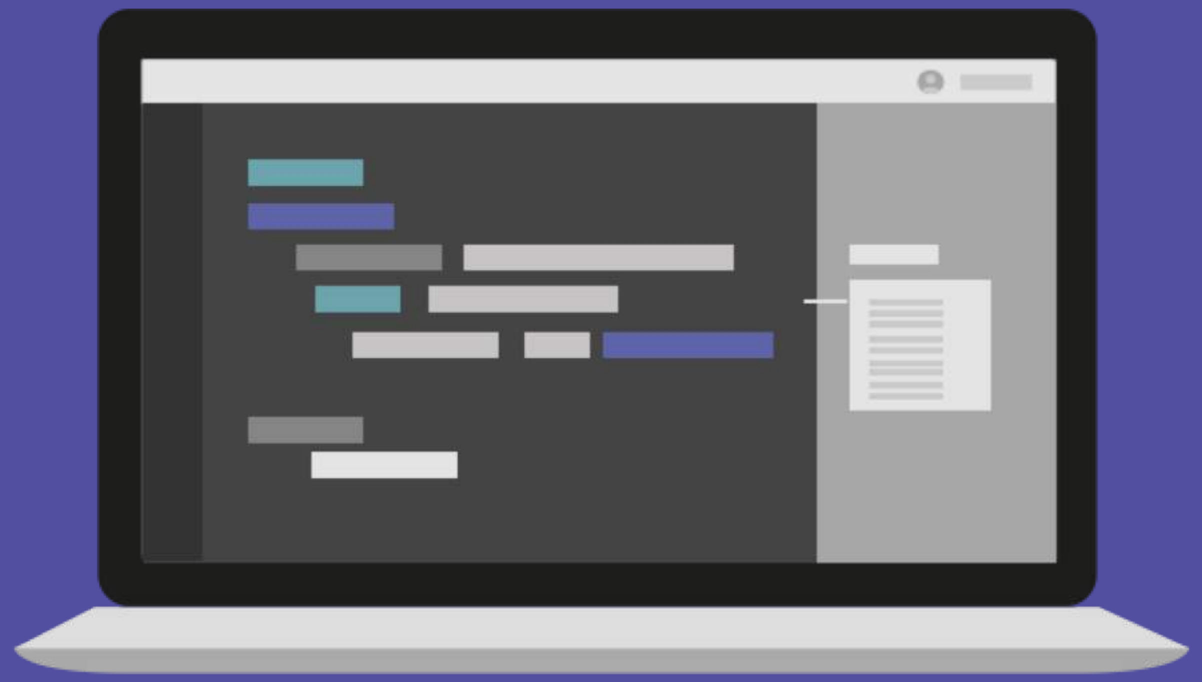
$$1 \;\; 1 \;\; 2 \;\; 3 \;\; 5 \;\; 8 \;\; 13 \;\; 21 \cdots$$

$$a_1 = 1$$

$$a_2 = 1$$

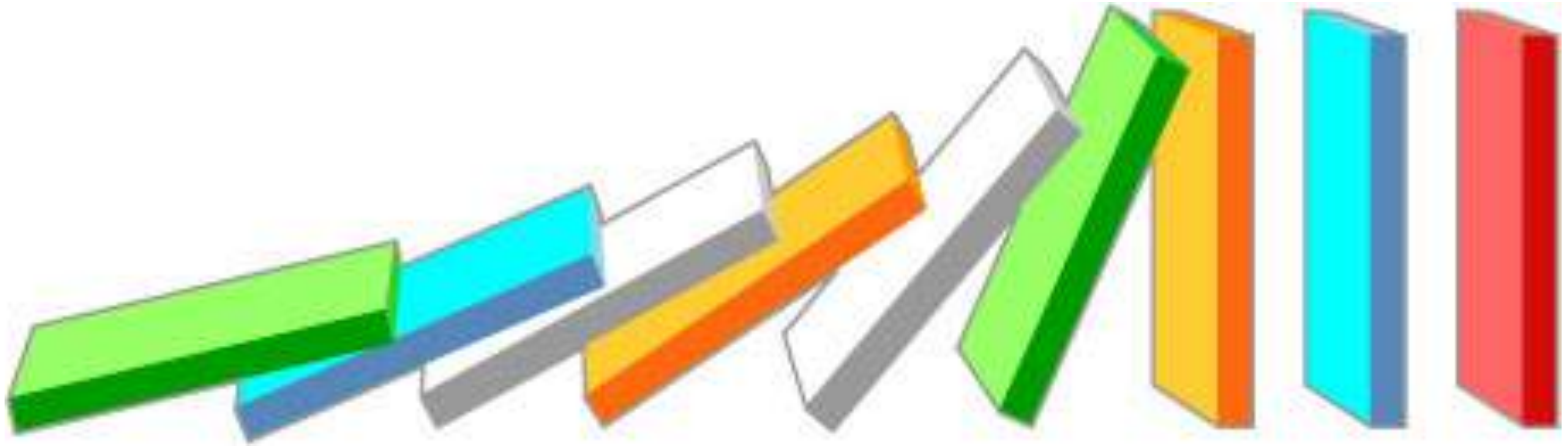
$$a_{n+2} = a_{n+1} + a_n \; (n \geq 3)$$

[실습5] 점화식



수학적 귀납법

연쇄 반응을 이용한 등식의 증명법



수학적 귀납법

1) n=1일 때 주어진 등식이 성립함을 보인다.

주어진 등식

$$a_{n+1} = f(a_n)$$

n=1

$$a_2 = f(a_1)$$

OK!

수학적 귀납법

2) n=k일 때 주어진 등식이 성립한다고
가정하고, 이때 n=k+1도 성립함을 보인다.

n=k 가 된다면…

$$a_{k+1} = f(a_k)$$

↓

n=k+1 도!

$$a_{k+2} = f(a_{k+1})$$ OK!

수학적 귀납법

3) 2)에 의해, n=1일 때 성립하면
n=2 일 때도 성립한다.

n=1 가 되므로

$$a_1 = f(a_2)$$ **OK?**

↓

n=1+1 도 성립!

$$a_2 = f(a_3)$$ **OK!**

수학적 귀납법

4) 위 과정을 반복하며 범위를 확장한다.

n=1 가 되므로

$$a_1 = f(a_2)$$ **OK!**

$\downarrow$

n=2 도 성립!

$$a_2 = f(a_3)$$ **OK!**

$\downarrow$

n=3 도 성립!

$$a_3 = f(a_4)$$ **OK!**

수학적 귀납법

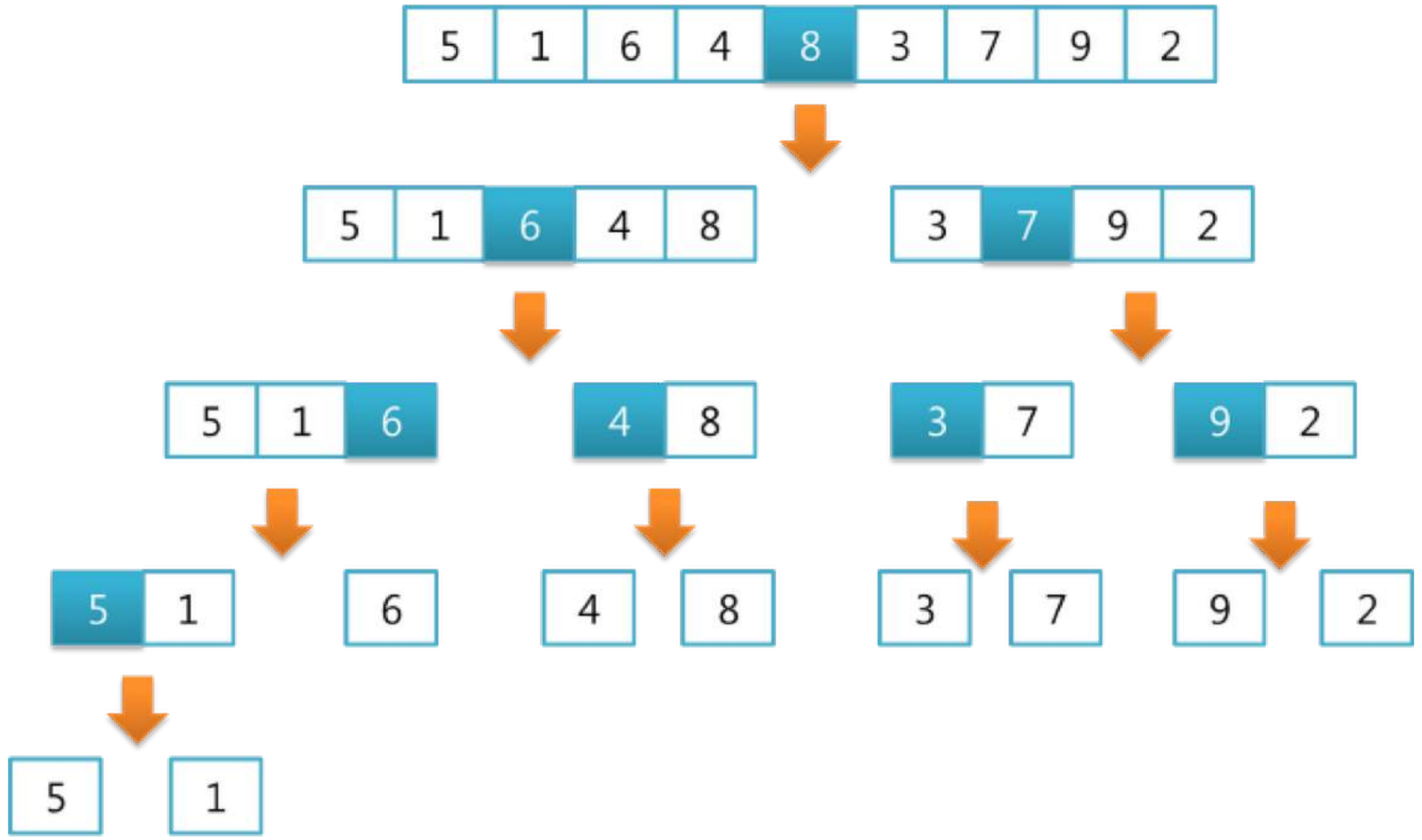
5) 따라서 수학적 귀납법에 의해 모든 n에 대해 등식이 성립한다.

모든 n에 대해

$$a_{n+1} = f(a_n)$$ **OK!**

분할 정복

어려운 하나의 문제를
여러 개의 쉬운 문제로



분할 정복

분할 정복을 안 써도 되지만

주어진 리스트의 최댓값을

굳이 분할정복으로 구해보자

Ex) MAX([6, 2, 9, 8, 1, 4, 17, 5]) = 17

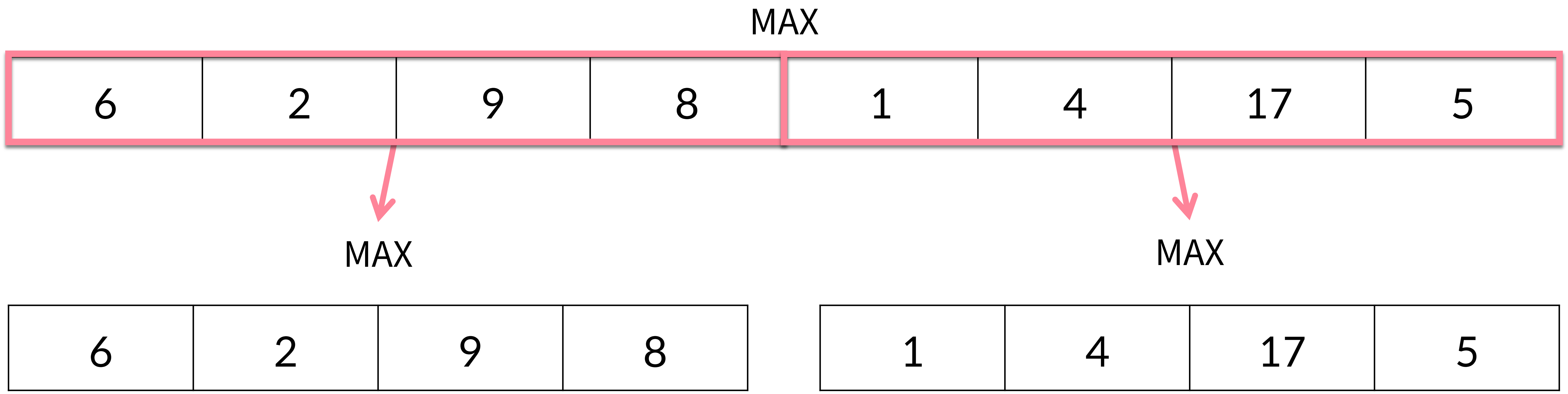
분할 정복

MAX

6	2	9	8	1	4	17	5

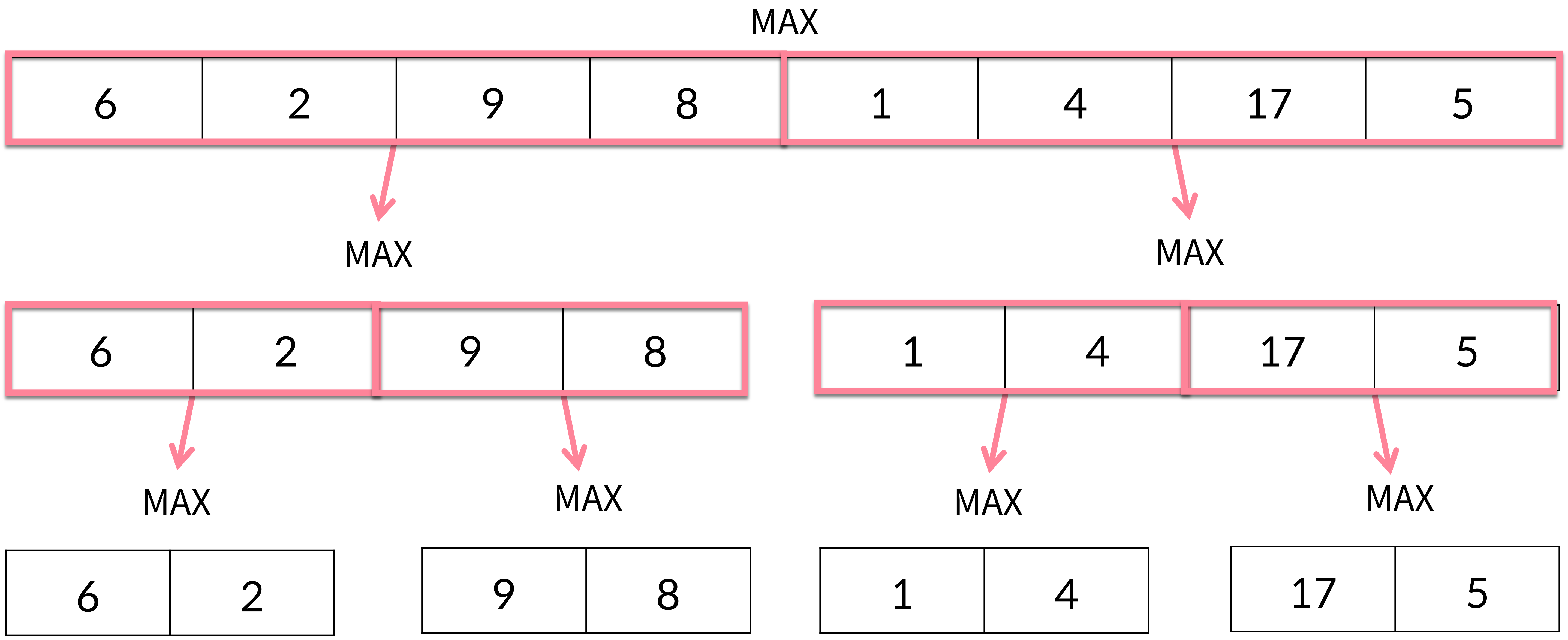
1) 리스트가 주어진다.

분할 정복



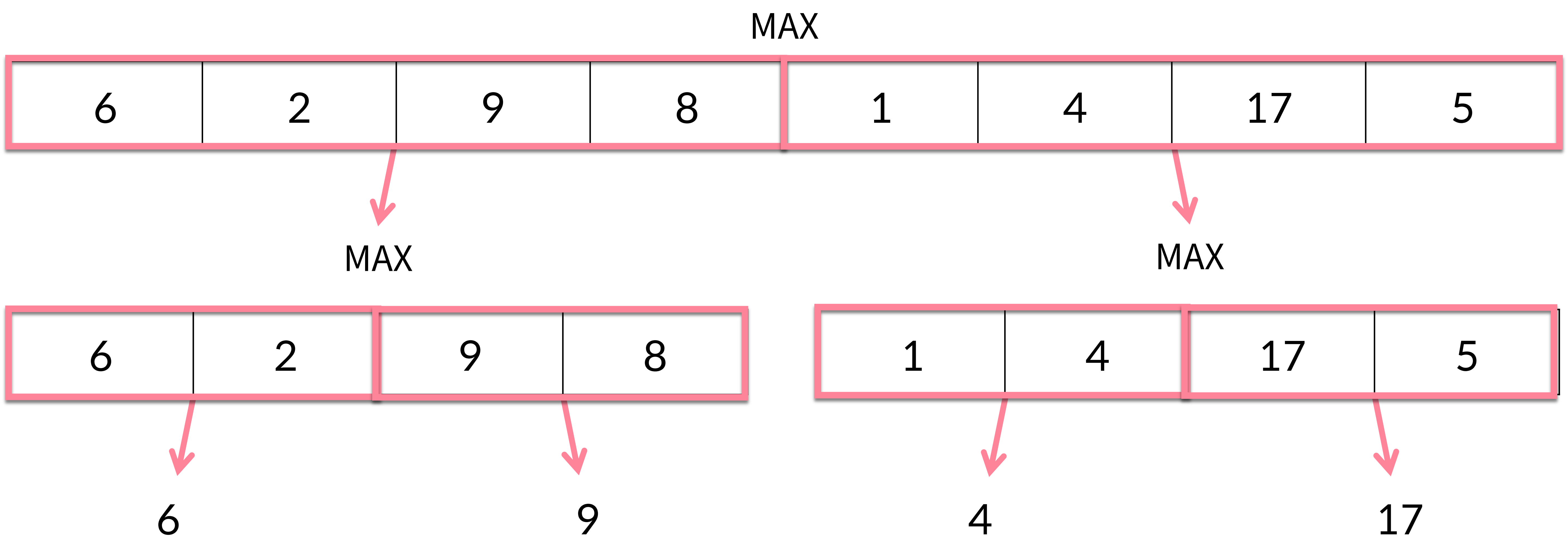
2) 리스트를 2개로 분리하여 다시 MAX

분할 정복



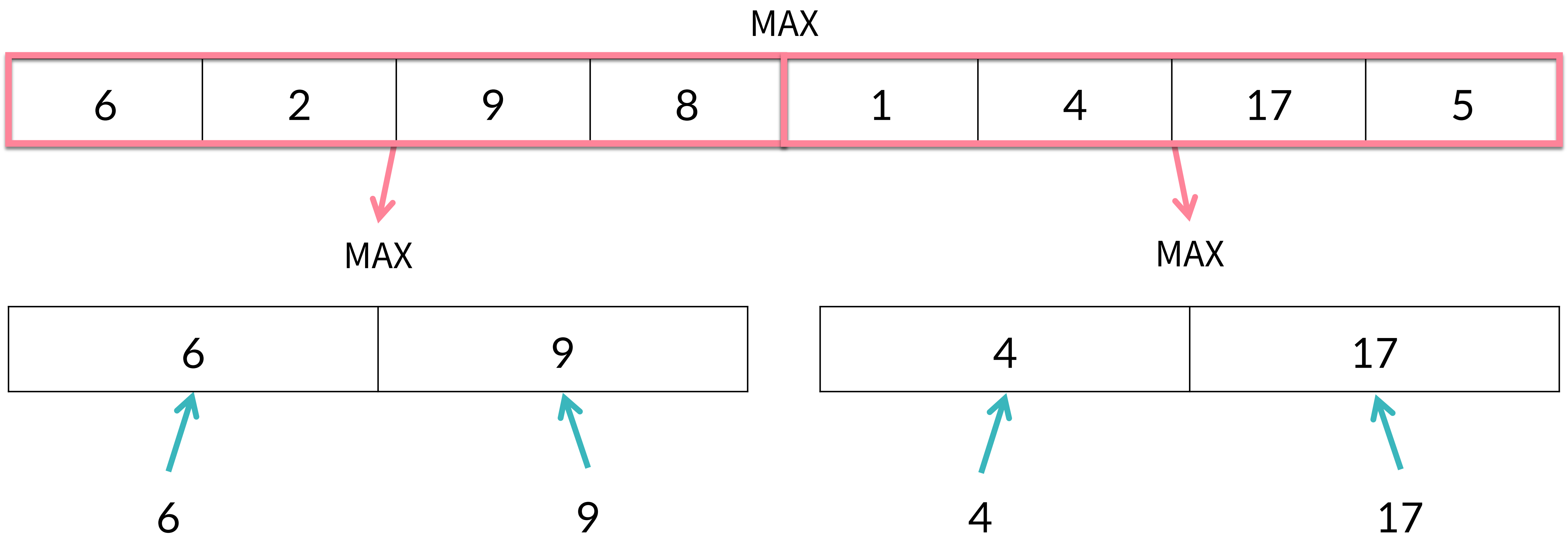
3) 충분히 작아질 때까지 쪼개기를 반복

분할 정복



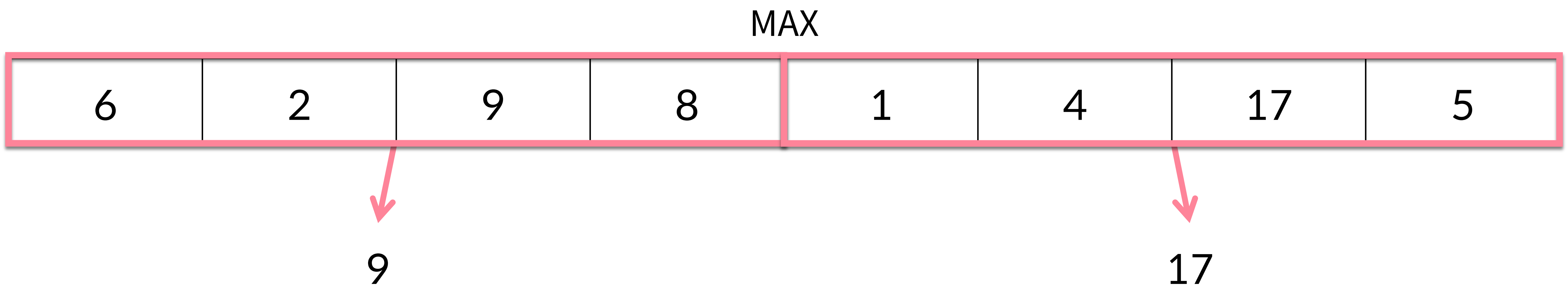
4) 숫자 2개를 비교하여 더 큰 수를 구한다.

분할 정복



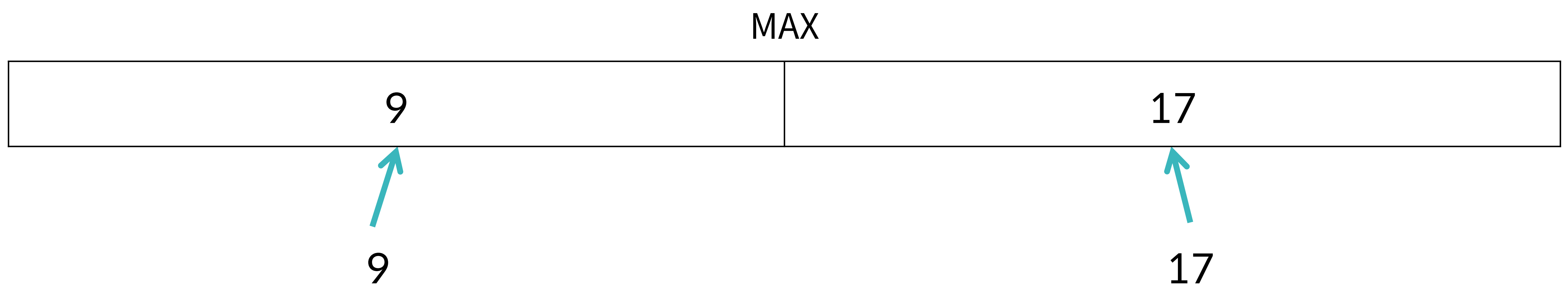
5) 기존의 문제가 쉬운 문제로 바뀐다.

분할 정복



6) 반복하며 올라간다.

분할 정복



6) 반복하며 올라간다.

분할 정복

7) 완료

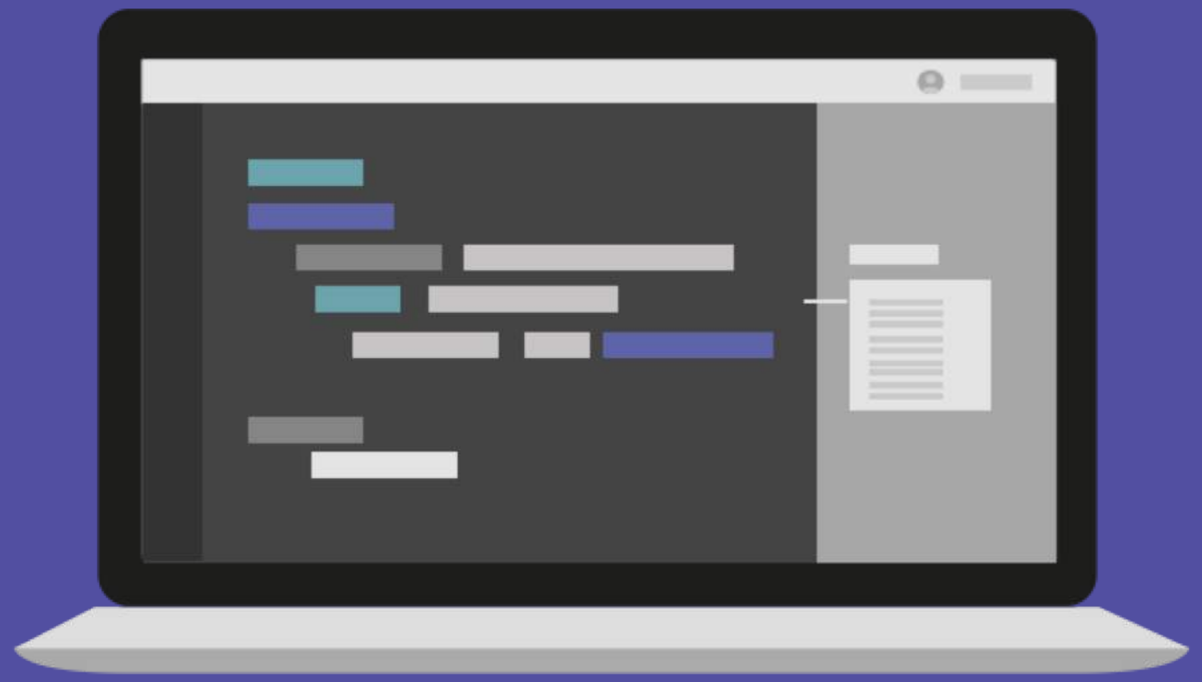
분할 정복

```
def div_max(a):
    # a의 크기가 1개인 경우 그대로 리턴합니다
    if len(a) == 1:
        return a[0]
    # a의 크기가 2개인 경우 바로 계산합니다
    if len(a) == 2:
        if a[0] > a[1]:
            return a[0]
        else:
            return a[1]
```

분할 정복

```
# a의 크기가 2보다 큰 경우 2개로 분할하여 풉니다
else:
    l = len(a)
    m1 = div_max(a[0:l//2])
    m2 = div_max(a[l//2:l])
    if m1 > m2:
        return m1
    else:
        return m2
```

[실습6] 분할정복



이번 장에서는!

1. 등차수열에서 시작해 등비수열, 계차수열에 대해 알아가며
기본적인 수열의 형태를 익혔습니다.
2. 급수와 점화식을 통해 한 단계 나아가 우리가 필요한 식을
세우는 법을 배웠습니다.
3. 수학적 귀납법을 통해 스스로 세운 식이 정확한지를
판단하는 법을 배웠습니다.
4. 분할정복법이라는 어려운 문제를 여러개의 쉬운 문제로 만들어
해결하는 귀납적 알고리즘에 대해서 알 수 있었습니다.

/* elice */

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice