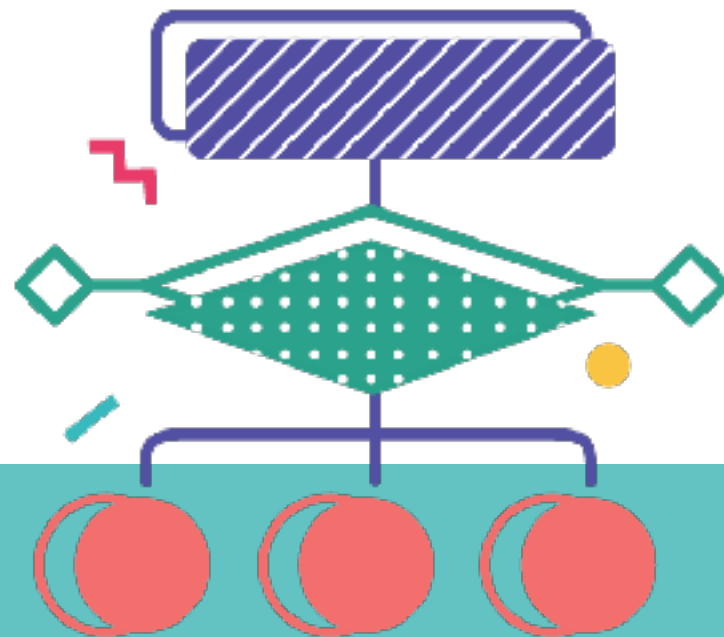


/* 알고리즘 II */

신현규 강사, 일 14:00

8월 20일 ~ 9월 16일



/* elice */

목차

- 01 지난 시간 요약
- 02 실습 문제 풀이
- 03 동적계획법 (2)
- 04 동적계획법 요약

01 지난 시간 요약

온라인 강의가 풀어야 할 숙제

강의의 지속적 참여의 어려움

1주 밀리면 곤란해짐 (강의 약 3시간 + 코딩 과제 약 2개)

직접적 상호작용의 어려움

엘리스가 열심히 해결하고 있는 숙제

강의를 따라가는 것의 어려움

강의에 투자할 수 있는 물리적인 시간이 제한되어 있음

학생의 지식 분포가 매우 넓음

활동 1 : 오픈 카톡방 참여

<http://goo.gl/ExW6iv>

지금까지의 이야기

컴퓨터공학적 사고

COMPUTATIONAL THINKING



DS & AL



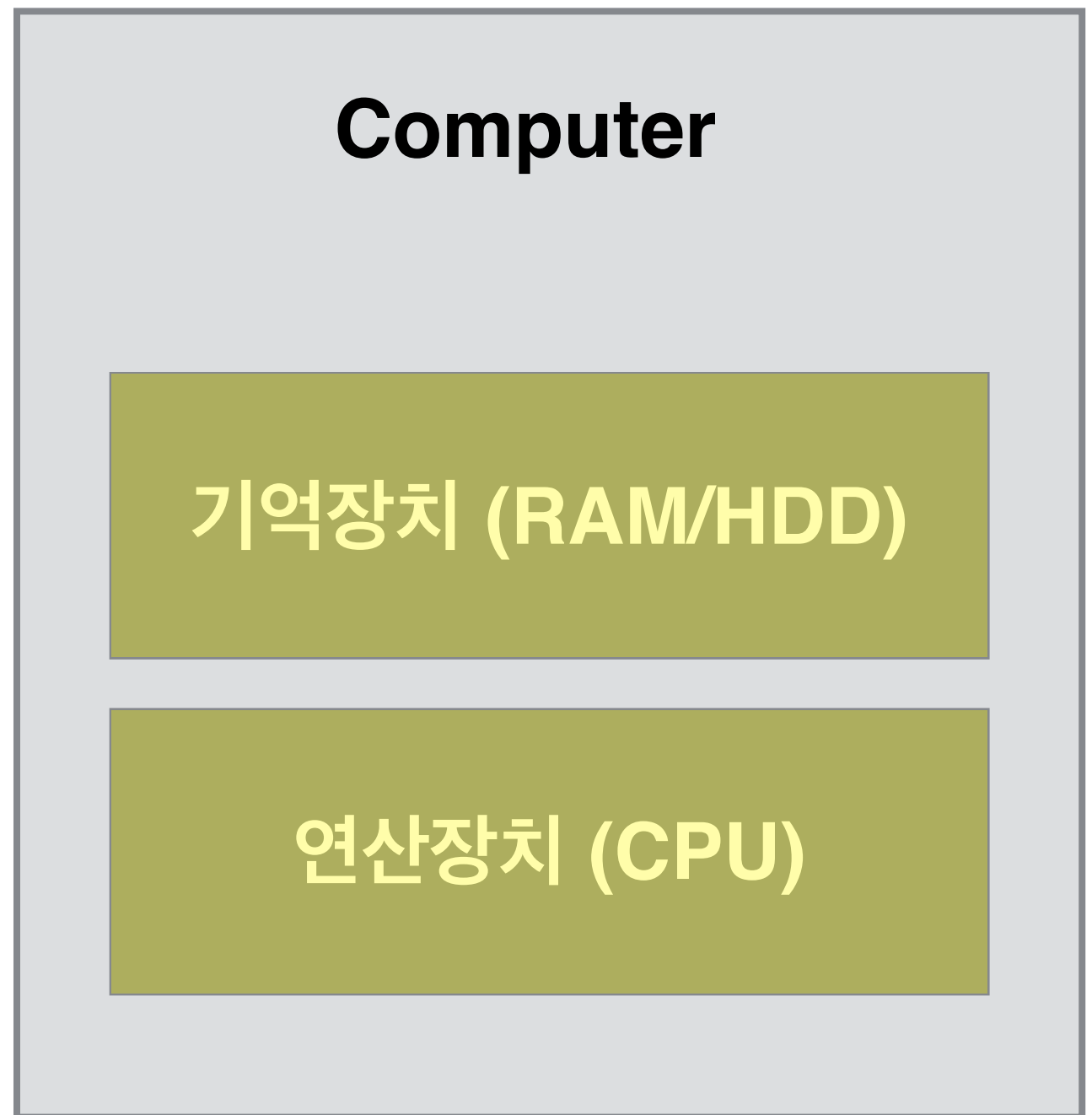
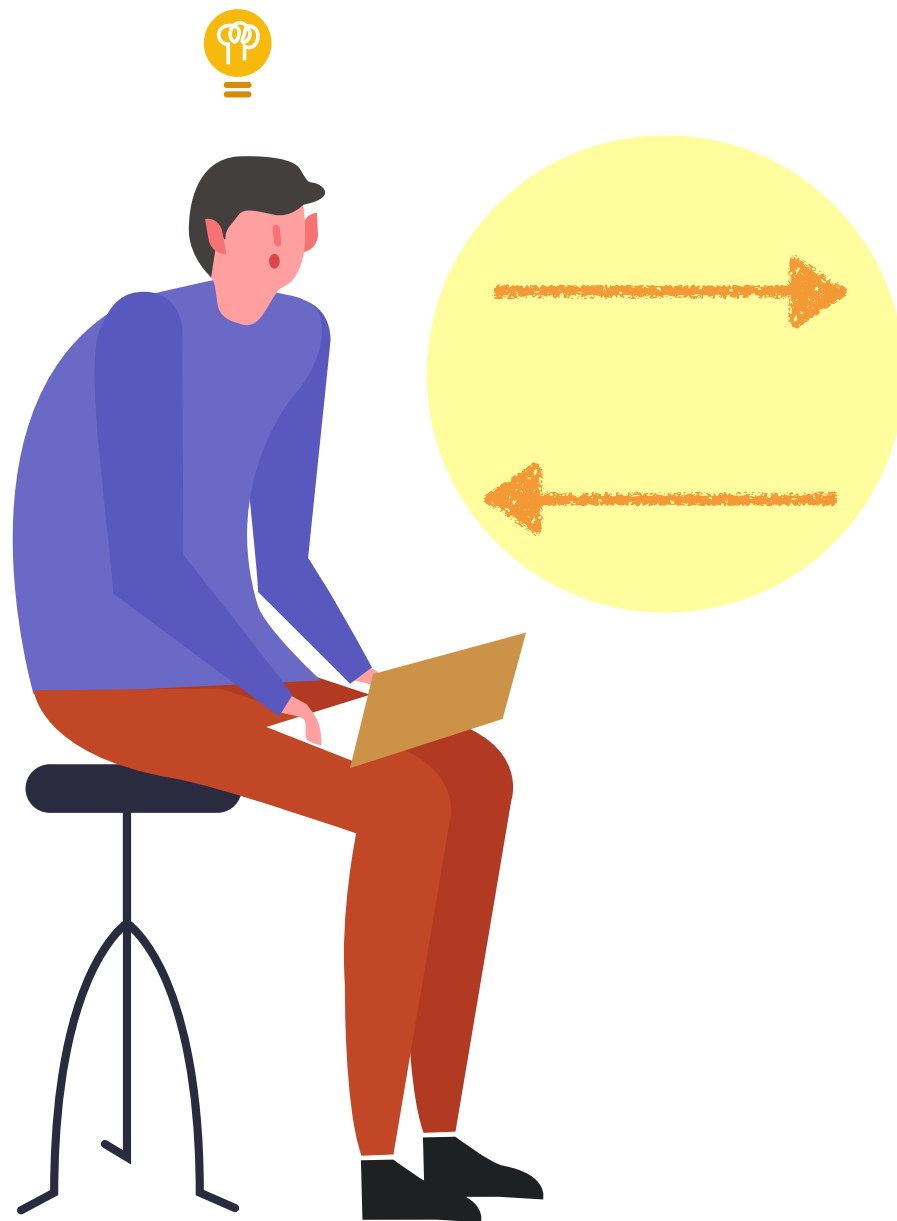
AL I



AL II

/* elice */

계산기로써의 컴퓨터 (Compute + er)



컴퓨터 공학의 기본 커리큘럼



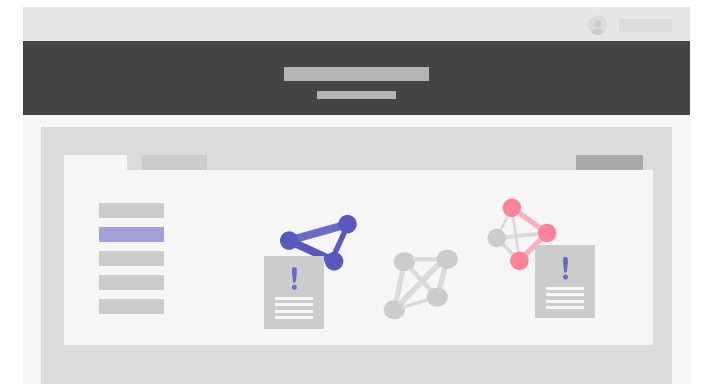
1. 프로그래밍 언어

C / C++
Python
Matlab



2. 자료구조

Stack
Queue
Tree



3. 알고리즘

Brute-Force
Divide & Conquer
Dynamic Programming

데이터 구조 및 알고리즘 커리큘럼

1주차 과정 소개, 배열, 연결리스트, 클래스

2주차 스택, 큐, 해싱

3주차 시간복잡도

4주차 트리, 트리순회, 재귀호출

5주차 힙

6주차 그래프 소개, DFS

7주차 그래프 심화, BFS

8주차 강의 요약, 알고리즘 과정 소개

알고리즘 커리큘럼

1주차	과정 소개, 재귀호출
2주차	문제 해결의 절차, 완전탐색, 시간복잡도
3주차	분할정복법
4주차	탐욕적 기법
5주차	동적 계획법 I
6주차	동적 계획법 II
7주차	그래프 알고리즘 I
8주차	그래프 알고리즘 II, 강의 요약

동적 계획법 (Dynamic Programming)

부분 문제를 해결한 결과를 이용하여 전체 문제를 해결

“나”를 해결함으로써 “나”를 해결

재귀호출

퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬

4	7	4	2	10	19	2	4	5	3	1	5
---	---	---	---	----	----	---	---	---	---	---	---

퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬

4	7	4	2	10	19	2	4	5	3	1	5
---	---	---	---	----	----	---	---	---	---	---	---

pivot

퀵정렬 (Quick Sort)

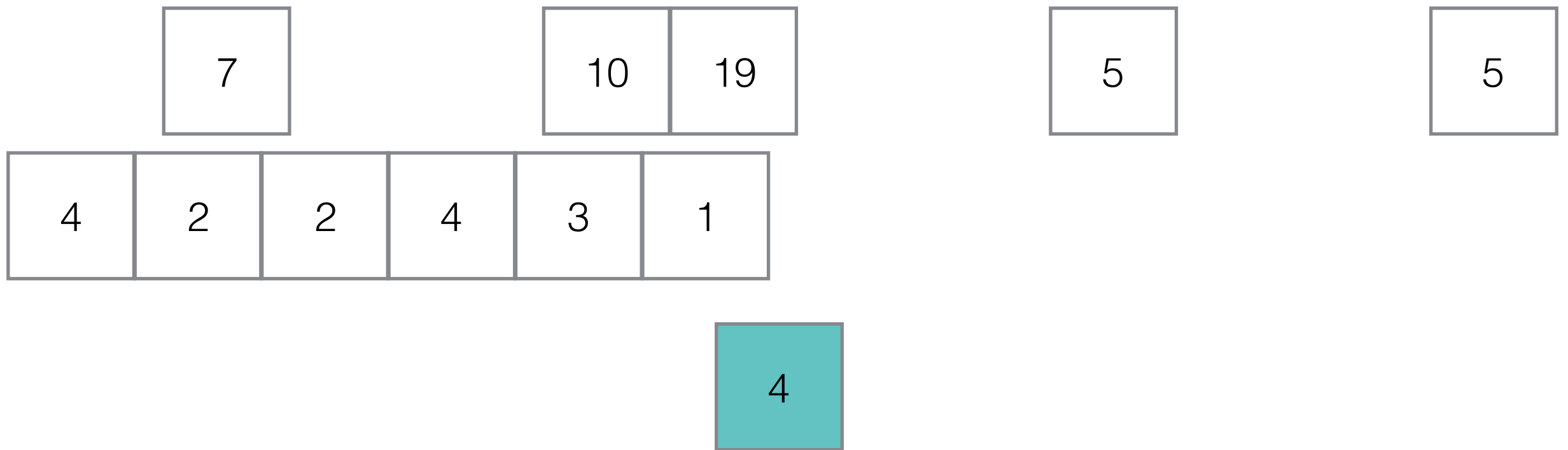
재귀호출을 이용한 대표적인 정렬

7	4	2	10	19	2	4	5	3	1	5
---	---	---	----	----	---	---	---	---	---	---



퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬



퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬

4	2	2	4	3	1
---	---	---	---	---	---

7	10	19	5	5
---	----	----	---	---

4

퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬

4	2	2	4	3	1	4	7	10	19	5	5
---	---	---	---	---	---	---	---	----	----	---	---

퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬

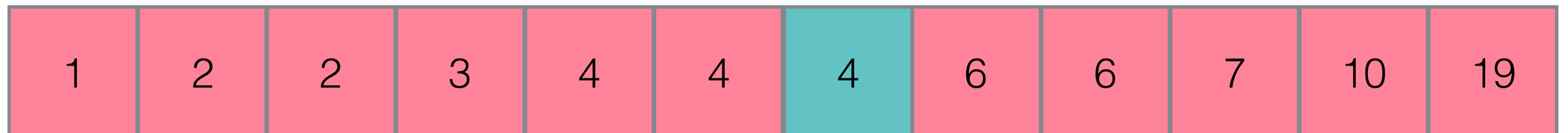
4	2	2	4	3	1	4	7	10	19	5	5
---	---	---	---	---	---	---	---	----	----	---	---

Quicksort!

Quicksort!

퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬



Quicksort!

Quicksort!

퀵정렬 (Quick Sort)

재귀호출을 이용한 대표적인 정렬

1	2	2	3	4	4	4	6	6	7	10	19
---	---	---	---	---	---	---	---	---	---	----	----

Quicksort!

Quicksort!

피보나치 수 구하기

$$F_n = F_{n-1} + F_{n-2}$$

피보나치 수 구하기

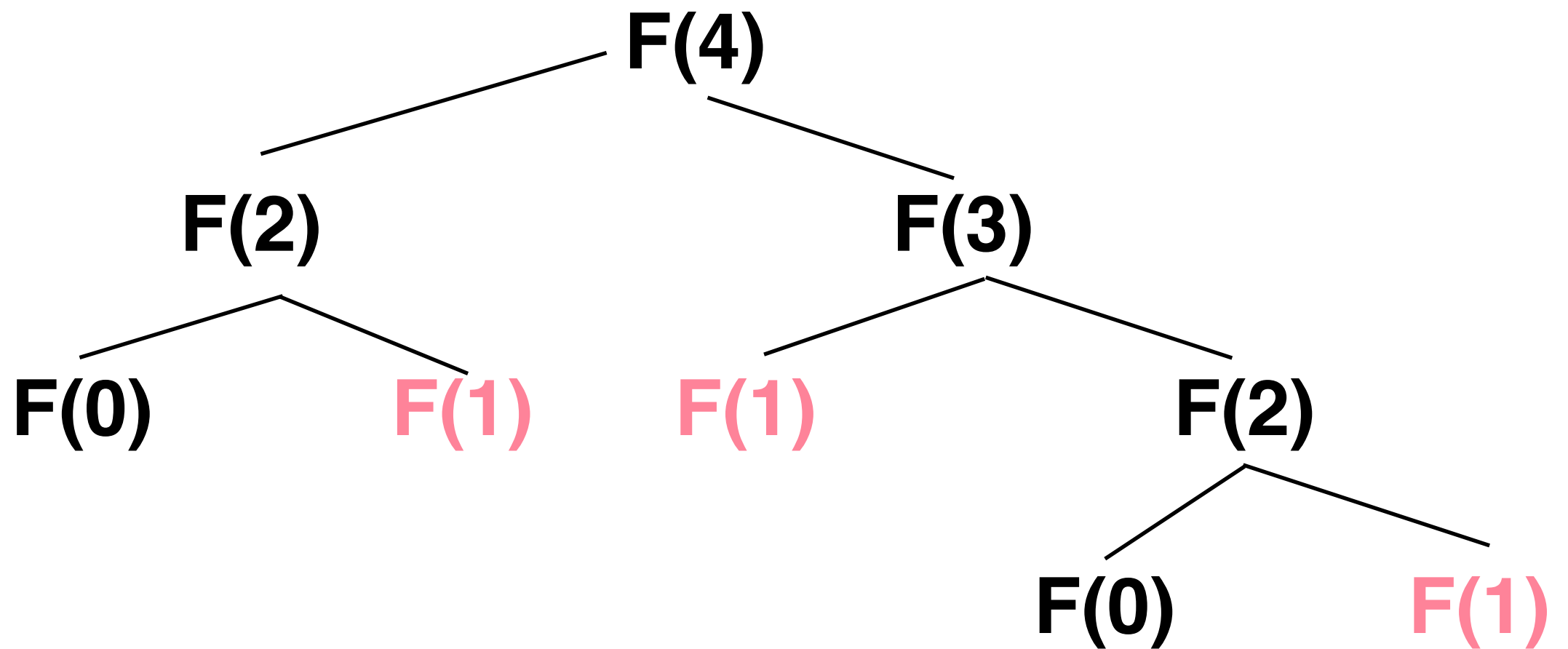
$$F_n = F_{n-1} + F_{n-2}$$

```
def getFibo(n) :  
    if n == 0 or n == 1 :  
        return 1  
    else :  
        return getFibo(n-1) + getFibo(n-2)
```

피보나치 수 구하기

F(4)

피보나치 수 구하기



피보나치 수 구하기

분할정복법을 이용한 풀이가 아님
문제를 독립적으로 나누지 않았기 때문

작은 “나”를 해결한 결과를 계속해서 써먹는다
 $F(n)$ 을 알기 위해서는 $F(1) \sim F(n-1)$ 을 모두 알아야 함

“나” 보다 작은 모든 풀이를 먼저 기억하자
 $F(n)$ 을 구하기 전에 $F(1) \sim F(n-1)$ 을 모두 구하자

동적 계획법의 문제 풀이 순서

1. 부분 문제를 정의한다.
2. 점화식을 구한다
3. 문제를 해결한다

동적 계획법의 문제 풀이 순서

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다
2. 점화식을 구한다
그 값을 어떻게 구할지에 대한 식을 세운다
3. 문제를 해결한다
값을 직접 구하는 코드를 작성한다

피보나치 수 구하기

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다

$$F(i) = F_i \text{의 값}$$

피보나치 수 구하기

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$$F(i) = F_i \text{의 값}$$

$$F(i) = F(i-1) + F(i-2)$$

피보나치 수 구하기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$F(i) = F_i$ 의 값

$F(i) = F(i-1) + F(i-2)$

	0	1	2	3	4	5	6	7
F								

피보나치 수 구하기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$F(i) = F_i$ 의 값

$$F(i) = F(i-1) + F(i-2)$$

	0	1	2	3	4	5	6	7
F	1	1						

피보나치 수 구하기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$F(i) = F_i$ 의 값

$F(i) = F(i-1) + F(i-2)$

	0	1	2	3	4	5	6	7
F	1	1	2					

피보나치 수 구하기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$F(i) = F_i$ 의 값

$F(i) = F(i-1) + F(i-2)$

	0	1	2	3	4	5	6	7
F	1	1	2	3				

피보나치 수 구하기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$F(i) = F_i$ 의 값

$$F(i) = F(i-1) + F(i-2)$$

	0	1	2	3	4	5	6	7
F	1	1	2	3	5	8	13	21

피보나치 수 구하기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

기저조건 (Base condition)

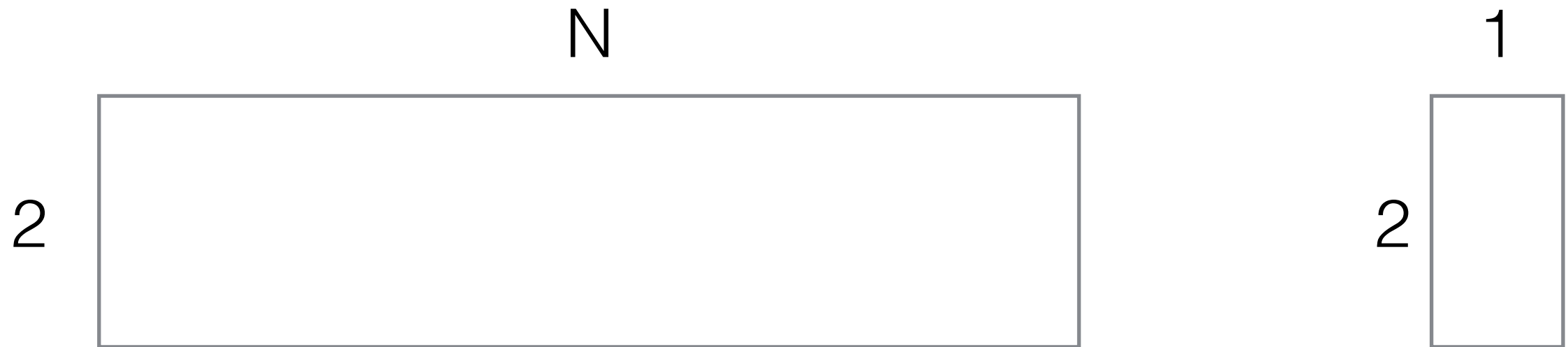
$F(i) = F_i$ 의 값

$F(i) = F(i-1) + F(i-2)$

	0	1	2	3	4	5	6	7
F	1	1	2	3	5	8	13	21

[예제 1] 블록 채우기

$2 \times N$ 의 상자를 2×1 의 블록으로 채우는 경우의 수는 ?



입력의 예



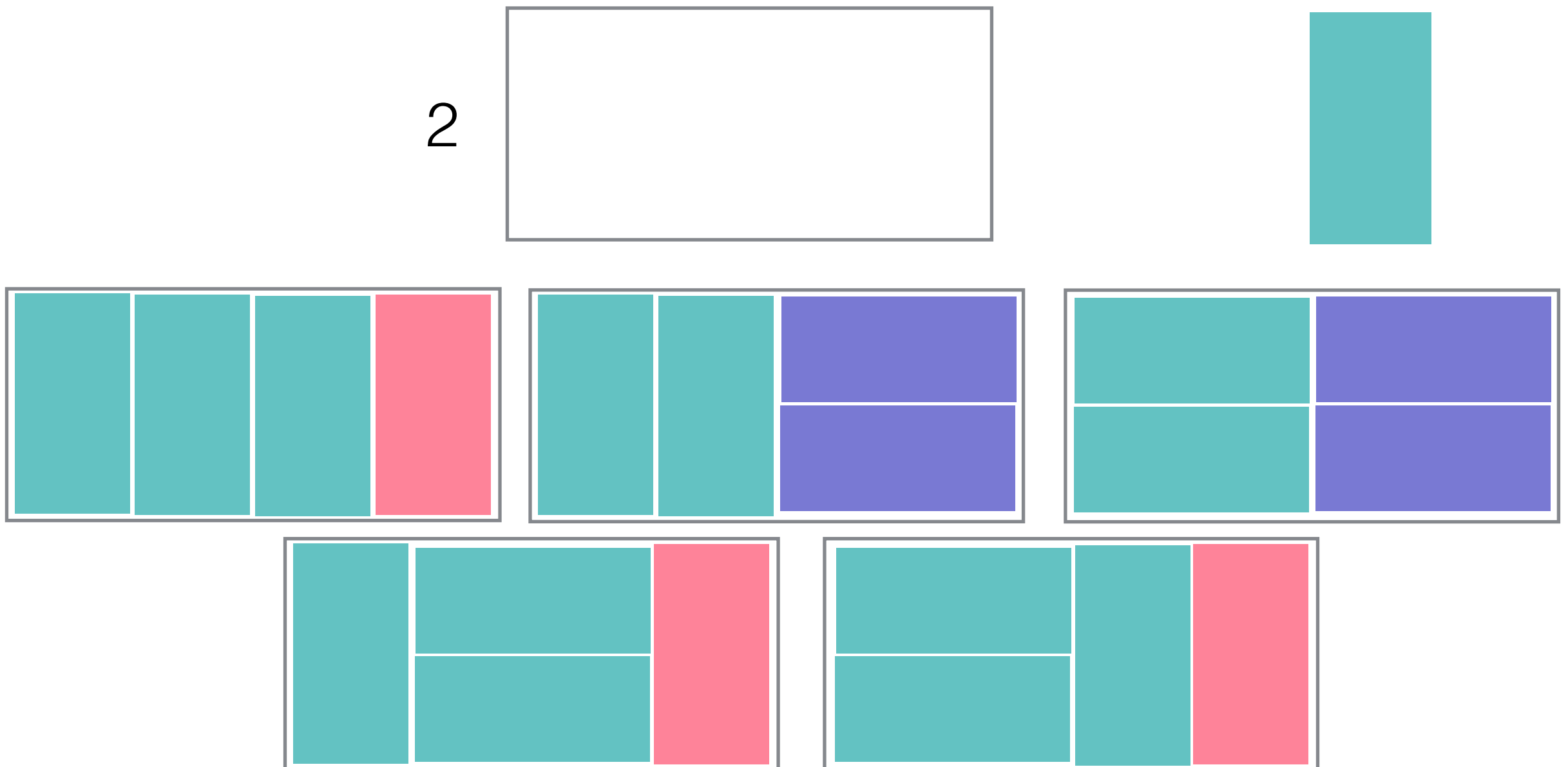
출력의 예



[예제 1] 블록 채우기

4

2



[예제 1] 블록 채우기

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다
2. 점화식을 구한다
그 값을 어떻게 구할지에 대한 식을 세운다
3. 문제를 해결한다
값을 직접 구하는 코드를 작성한다

[예제 1] 블록 채우기

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다

[예제 1] 블록 채우기

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

[예제 1] 블록 채우기

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

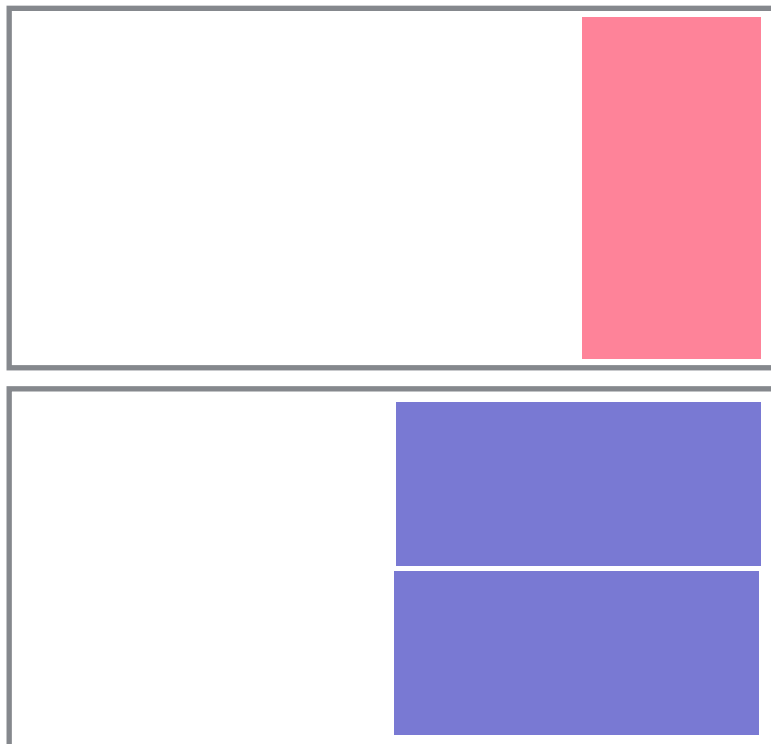
$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

[예제 1] 블록 채우기

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수



[예제 1] 블록 채우기

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수



$2 \times (i-1)$ 의 블록을 채우는 경우의 수

$2 \times (i-2)$ 의 블록을 채우는 경우의 수

[예제 1] 블록 채우기

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수



$2 \times (i-1)$ 의 블록을 채우는 경우의 수
 $T(i-1)$

$2 \times (i-2)$ 의 블록을 채우는 경우의 수
 $T(i-2)$

[예제 1] 블록 채우기

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

	0	1	2	3	4	5	6	7
T								

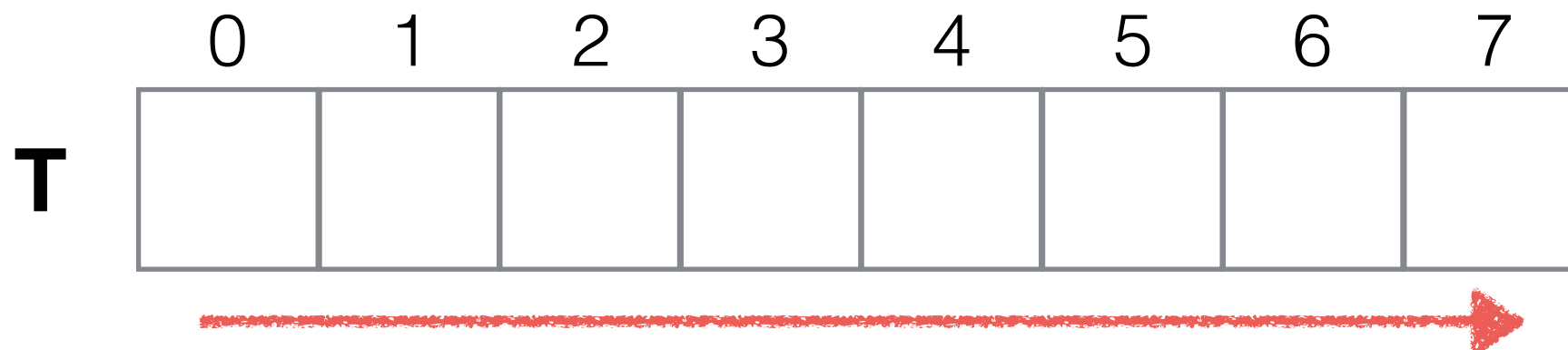
[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$



[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

	0	1	2	3	4	5	6	7
T	x							

[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

	0	1	2	3	4	5	6	7
T	X	1						

[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

	0	1	2	3	4	5	6	7
T	X	1	2					

[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

	0	1	2	3	4	5	6	7
T	X	1	2	3				

[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

	0	1	2	3	4	5	6	7
T	X	1	2	3	5			

[예제 1] 블록 채우기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i) = 2 \times i$ 의 상자를 채우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

	0	1	2	3	4	5	6	7
T	X	1	2	3	5	8	13	21

동적 계획법 (Dynamic Programming)

부분 문제를 정의하는 것이 가장 어렵다
어떻게 정의하느냐에 따라 풀리기도 하고 안풀리기도 함

문제가 “재귀적으로 해결되는지”를 볼 줄 알아야 함
규칙 찾는 것으로 보일 때도 있고, 실제로 틀린 말도 아님

무조건 많은 예제를 풀어야 한다
단언컨데 재귀호출, 동적계획법 기초 실력 향상은 양으로 승부봐야함

[예제 2] 숫자 만들기

1 ~ M 까지의 숫자를 더하여 N을 만드는 경우의 수는 ?

입력의 예

5 3

출력의 예

13

[예제 2] 숫자 만들기

5

1 2 3

$$1 + 1 + 1 + 1 + 1$$

$$1 + 1 + 1 + 2$$

$$1 + 1 + 3$$

$$1 + 1 + 2 + 1$$

$$1 + 2 + 2$$

$$2 + 3$$

$$1 + 2 + 1 + 1$$

$$2 + 1 + 2$$

$$2 + 1 + 1 + 1$$

$$3 + 2$$

$$2 + 2 + 1$$

$$1 + 3 + 1$$

$$3 + 1 + 1$$

동적 계획법의 문제 풀이 순서

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다
2. 점화식을 구한다
그 값을 어떻게 구할지에 대한 식을 세운다
3. 문제를 해결한다
값을 직접 구하는 코드를 작성한다

[예제 2] 숫자 만들기

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

[예제 2] 숫자 만들기

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

[예제 2] 숫자 만들기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

M=3

	0	1	2	3	4	5	6	7
T	x							

[예제 2] 숫자 만들기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

M=3

	0	1	2	3	4	5	6	7
T	X	1						

[예제 2] 숫자 만들기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

M=3

	0	1	2	3	4	5	6	7
T	X	1	2					

[예제 2] 숫자 만들기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

M=3

	0	1	2	3	4	5	6	7
T	X	1	2	4				

[예제 2] 숫자 만들기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

M=3

	0	1	2	3	4	5	6	7
T	X	1	2	4	7			

[예제 2] 숫자 만들기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

M=3

	0	1	2	3	4	5	6	7
T	X	1	2	4	7	13		

[예제 2] 숫자 만들기

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = 1 ~ M 의 수를 이용하여 i 를 만드는 경우의 수

$$T(i) = T(i-1) + T(i-2) + \dots + T(i-M)$$

M=3

	0	1	2	3	4	5	6	7
T	X	1	2	4	7	13	24	44

[예제 2] 숫자 만들기



```
/* elice */
```

[예제 3] 연속 부분 최대합

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

[예제 3] 연속 부분 최대합

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

[예제 3] 연속 부분 최대합

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

[예제 3] 연속 부분 최대합

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

완전탐색법

$O(n^3)$

분할정복법

$O(n \log n)$

[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

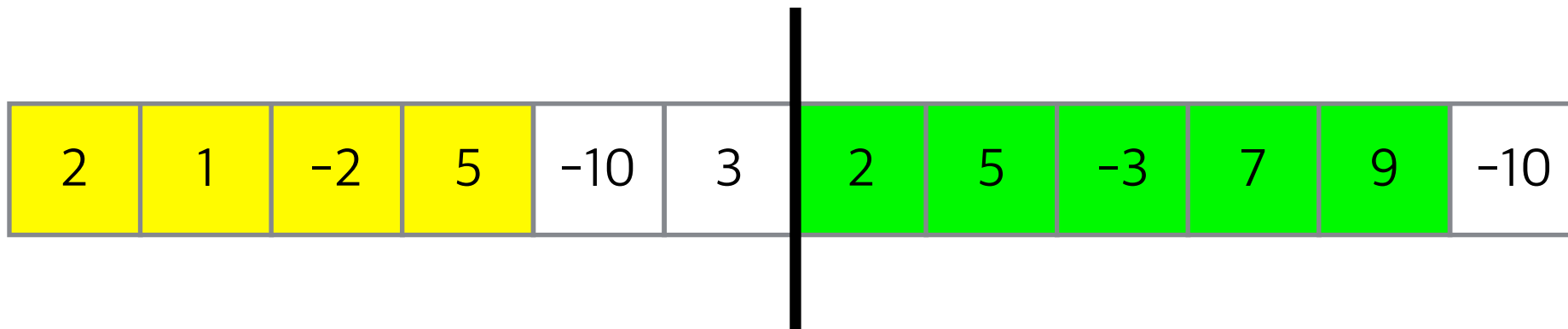
[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

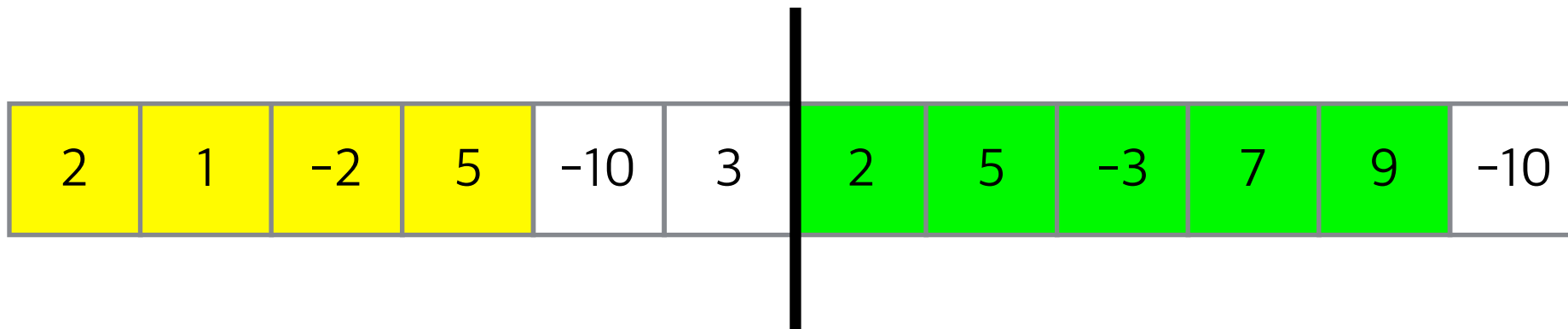
[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자



[예제 3] 연속부분 최대합 (D&C)

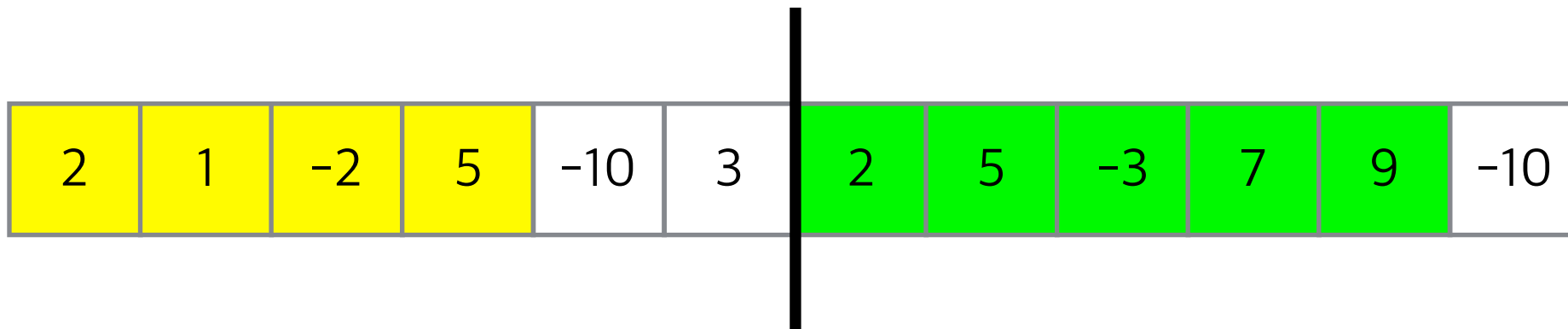
우선 절반으로 나누어 각각을 구해보자



고려하지 않은 경우는 ?

[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자

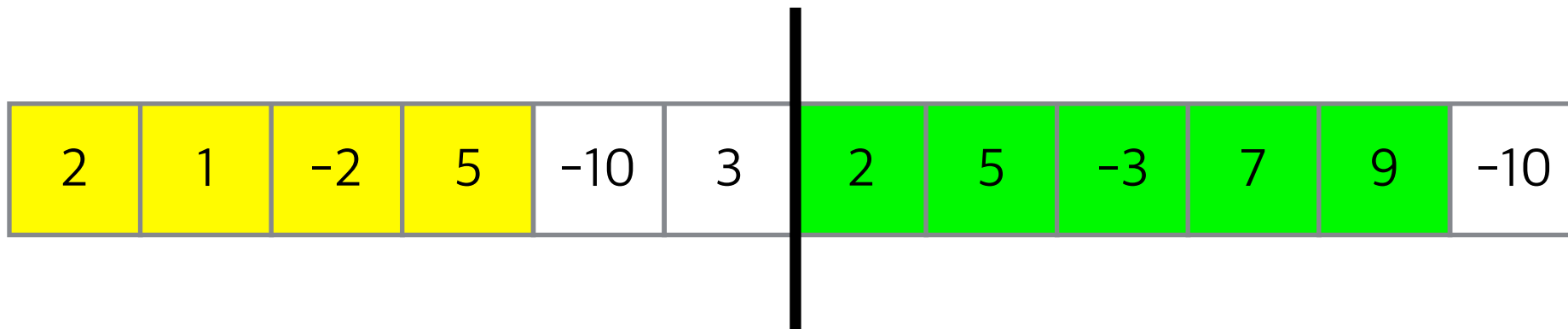


고려하지 않은 경우는 ?

자른 지점을 포함하는 연속 부분!

[예제 3] 연속부분 최대합 (D&C)

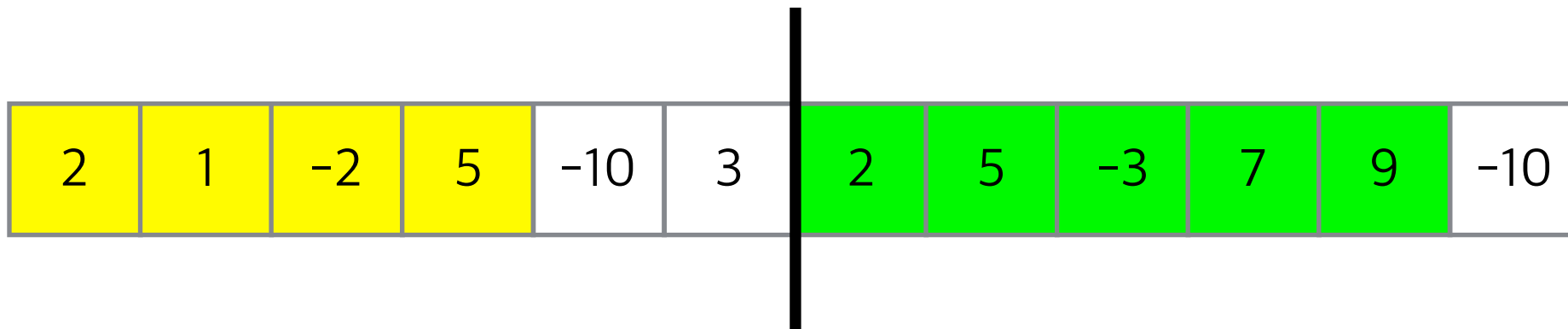
우선 절반으로 나누어 각각을 구해보자



자른 지점을 포함하는 연속 부분의 최대 합은 어떻게 구할까?

[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자

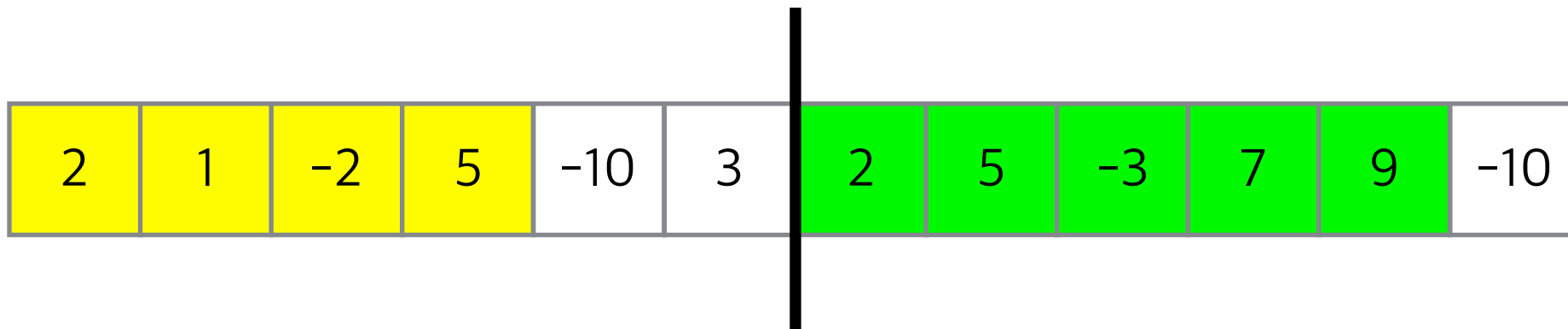


자른 지점을 포함하는 연속 부분의 최대 합은 어떻게 구할까?

Idea : 왼쪽과 오른쪽을 독립적으로 생각하자

[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자



[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

					3
				-10	3
			5	-10	3
		-2	5	-10	3
	1	-2	5	-10	3
2	1	-2	5	-10	3

/* elice */

[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자

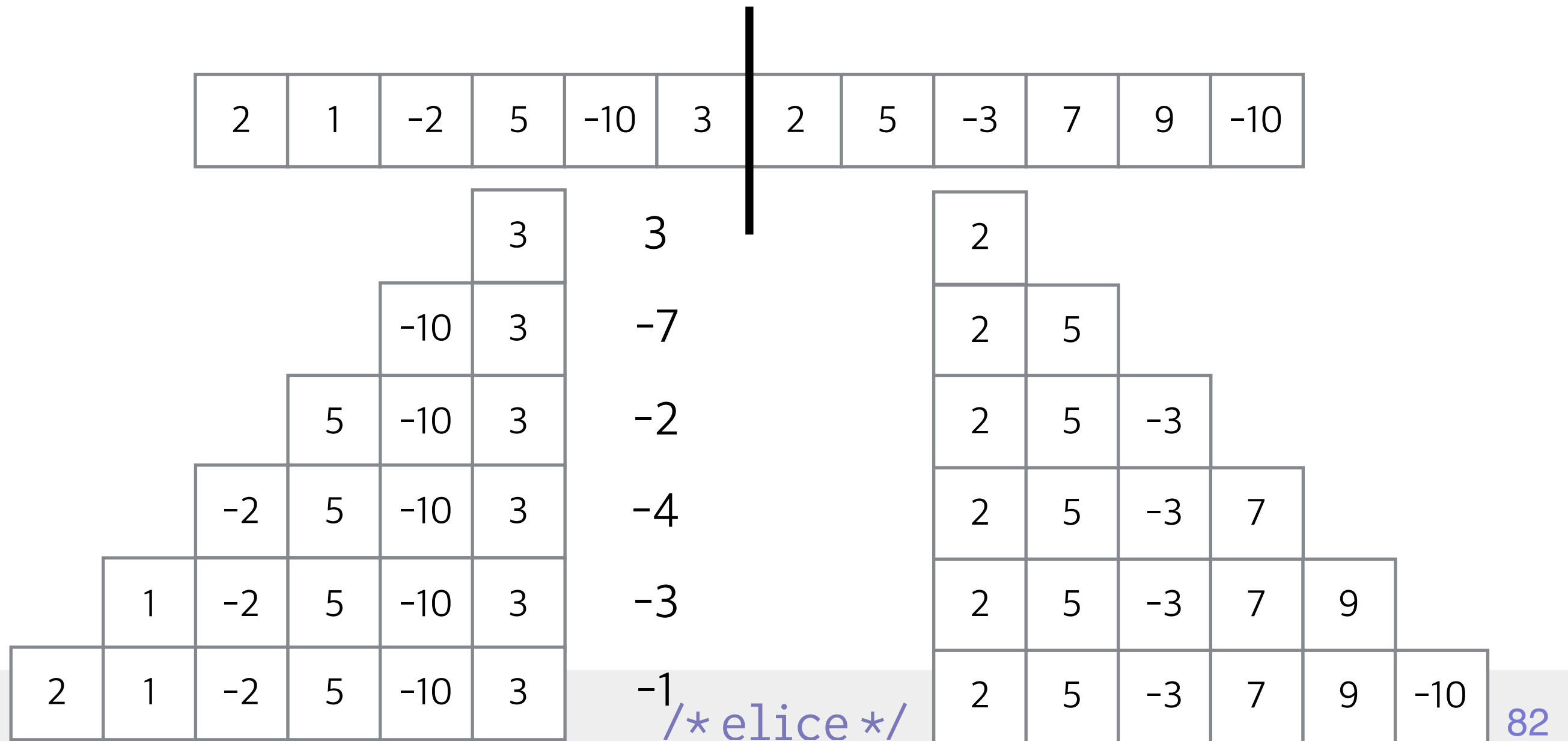
2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

					3	3
				-10	3	-7
			5	-10	3	-2
		-2	5	-10	3	-4
	1	-2	5	-10	3	-3
2	1	-2	5	-10	3	-1

/* elice */

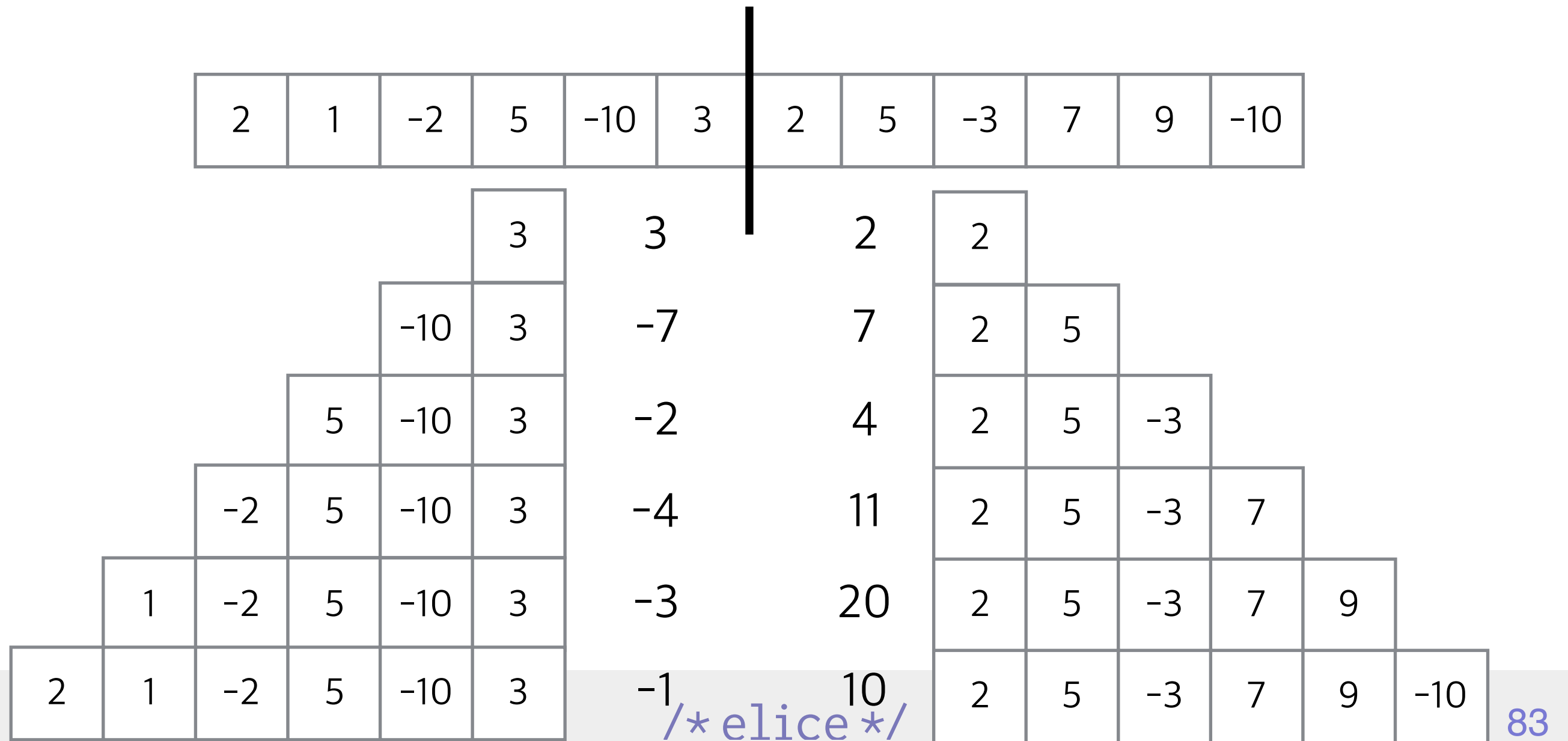
[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자



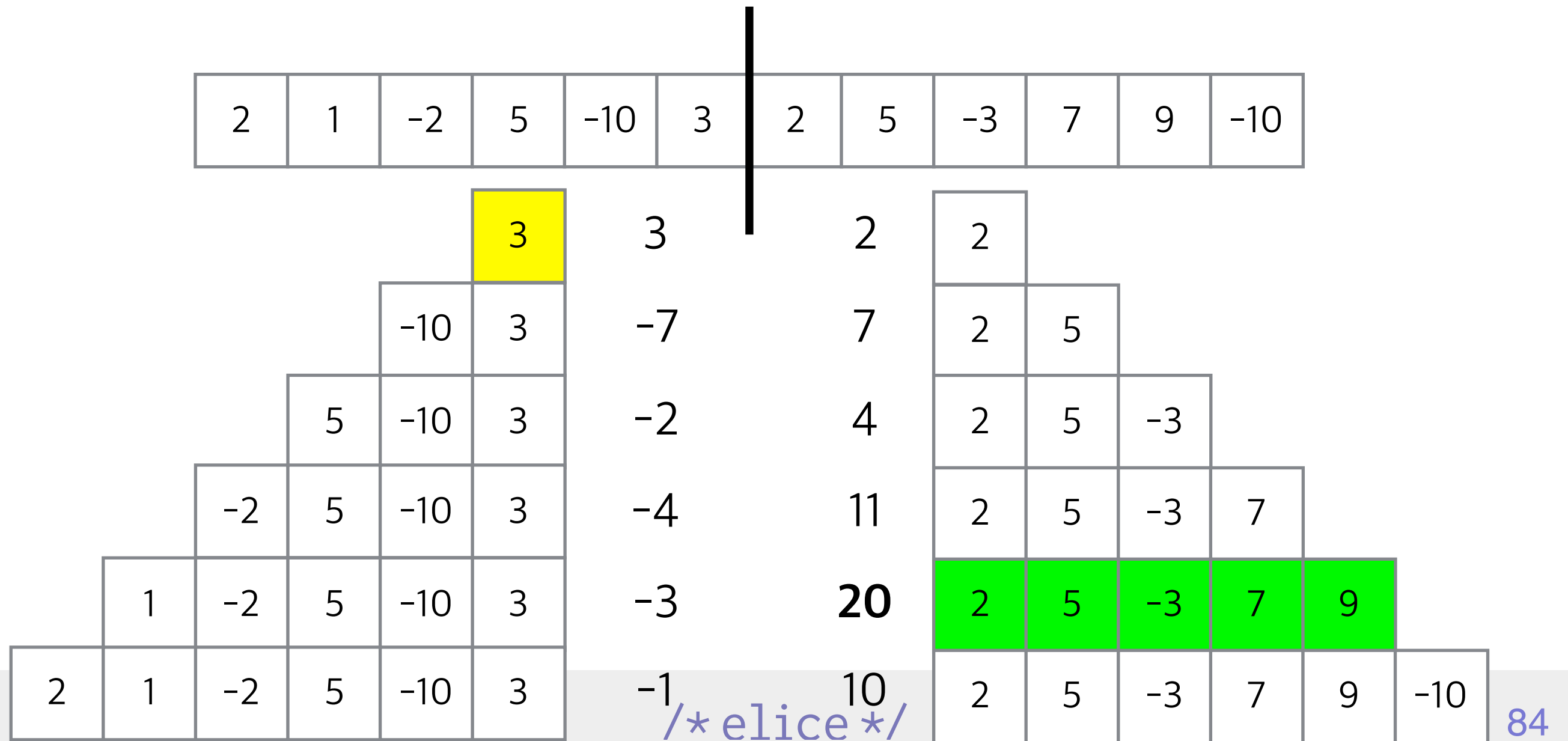
[예제 3] 연속부분 최대합 (D&C)

우선 절반으로 나누어 각각을 구해보자



[예제 3] 연속부분 최대합 (D&C)

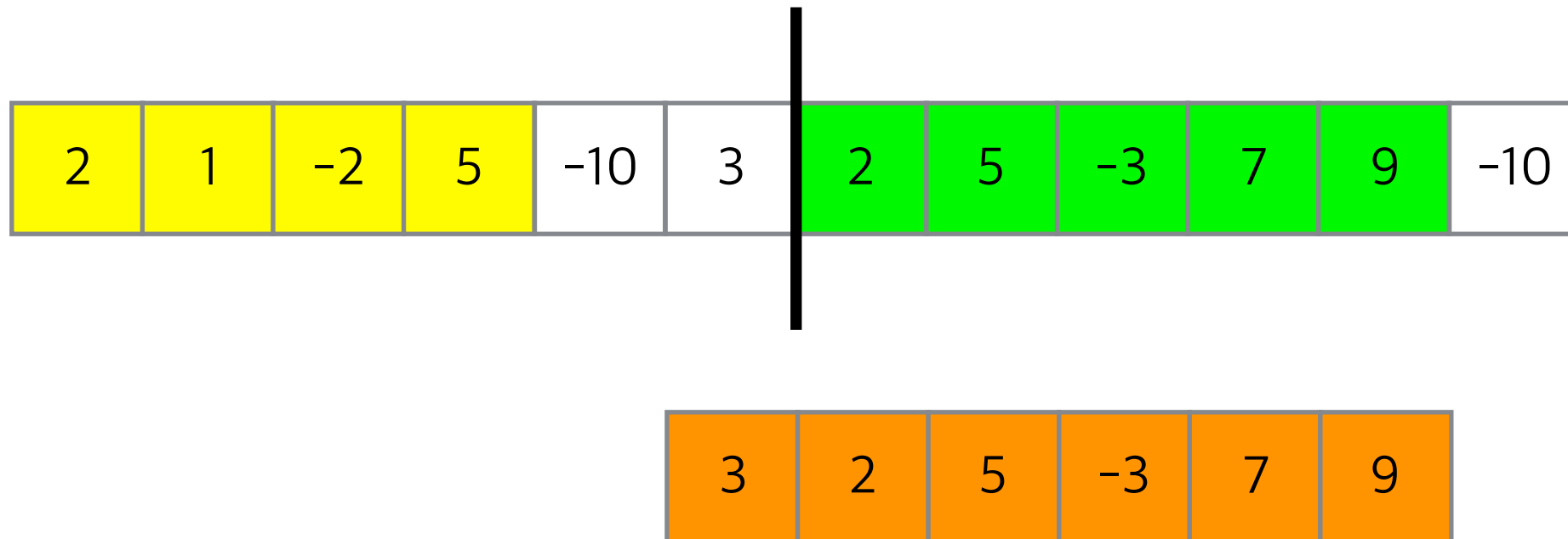
우선 절반으로 나누어 각각을 구해보자



[예제 3] 연속부분 최대합 (D&C)

모든 경우를 고려했음

1. 왼쪽만 포함하는 경우,
2. 오른쪽만 포함하는 경우,
3. 자른 자리를 포함하는 경우



[예제 3] 연속부분 최대합 (D&C)

최댓값!

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

[예제 3] 연속부분 최대합 (D&C)

시간복잡도

[예제 3] 연속부분 최대합 (D&C)

시간복잡도

$$T(N) = 2 * T(N/2) + O(N)$$

$$O(N \log N)$$

[예제 3] 연속 부분 최대합

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

완전탐색법

$O(n^3)$

분할정복법

$O(n \log n)$

[예제 3] 연속 부분 최대합 (DP)

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

[예제 3] 연속 부분 최대합 (DP)

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

[예제 3] 연속 부분 최대합 (DP)

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
			5				
		-4	5				
	2	-4	5				
1	2	-4	5				

/* elice */

[예제 3] 연속 부분 최대합 (DP)

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
			5	5			
		-4	5	1			
	2	-4	5	3			
1	2	-4	5	4			

/* elice */

[예제 3] 연속 부분 최대합 (DP)

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
			5	5			
		-4	5	1			
	2	-4	5	3			
1	2	-4	5	4			

/* elice */

[예제 3] 연속 부분 최대합 (DP)

연속된 부분을 선택하였을 때, 그 최대 합을 출력

1	2	-4	5	3	-2	9	10
			5	5			
		-1	5	4			

동적 계획법의 문제 풀이 순서

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다
2. 점화식을 구한다
그 값을 어떻게 구할지에 대한 식을 세운다
3. 문제를 해결한다
값을 직접 구하는 코드를 작성한다

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

1. 부분 문제를 정의한다.

무슨 값을 구할지를 정의한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T								

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----


3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T								



동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1							

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3						

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1					

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1	6				

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1	6	9			

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1	6	9	7		

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1	6	9	7	16	

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

3. 문제를 해결한다

값을 직접 구하는 코드를 작성한다

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1	6	9	7	16	26

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

그래서 답은 어디있나 ?

$T(i)$ = i 번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1	6	9	7	16	26

동적 계획법의 문제 풀이 순서

1	2	-4	5	3	-2	9	10
---	---	----	---	---	----	---	----

그래서 답은 어디있나 ?

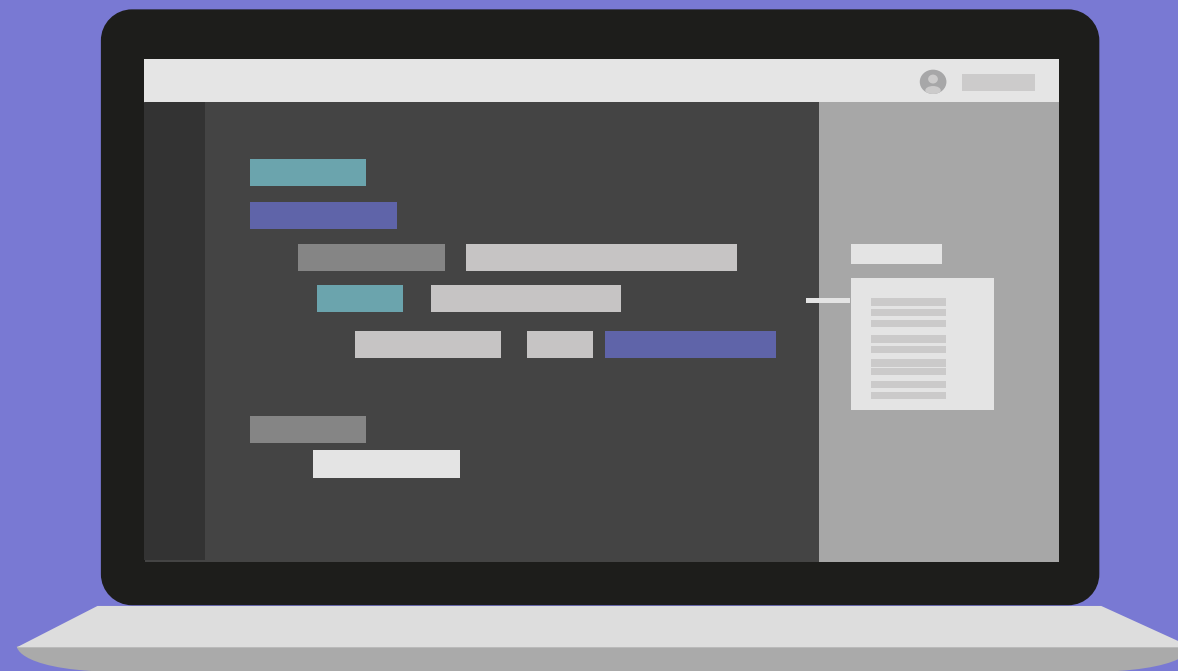
T의 최댓값

$T(i)$ = i번째 숫자를 오른쪽 끝으로 하는 연속 부분 최대합

$$T(i) = \max(T(i-1) + \text{Data}(i), \text{Data}(i))$$

	0	1	2	3	4	5	6	7
T	1	3	1	6	9	7	16	26

[예제 3] 연속 부분 최대합



`/* elice */`

[문제 1] 줄 세우기

남녀를 줄 세우는 경우의 수는 ?
단, 남자는 서로 인접해서는 안된다

입력의 예

5

출력의 예

13

[문제 1] 줄 세우기

5

Q Q Q Q Q

Q Q Q Q P

Q Q Q P Q

Q Q P Q Q

Q P Q Q Q

P Q Q Q Q

Q Q P Q P

Q P Q Q P

P Q Q Q P

Q P Q P Q

P Q Q P Q

P Q P Q Q

P Q P Q P

[문제 1] 줄 세우기

5

Q Q P Q P
Q P Q Q P
P Q Q Q P
Q Q Q Q P
P Q P Q P

Q Q Q Q Q
Q Q Q P Q
Q Q P Q Q
Q P Q Q Q
P Q Q Q Q
Q P Q P Q
P Q Q P Q
P Q P Q Q

[문제 1] 줄 세우기

5

... P

... Q

[문제 1] 줄 세우기

5

... P : P의 앞에는 무조건 Q를 세워야 함

... Q : Q 앞에는 아무거나 세워도 됨

동적 계획법의 문제 풀이 순서

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다
2. 점화식을 구한다
그 값을 어떻게 구할지에 대한 식을 세운다
3. 문제를 해결한다
값을 직접 구하는 코드를 작성한다

[문제 1] 줄 세우기

5

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다

[문제 1] 줄 세우기

5

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다

$T(i)$ = i 명의 학생을 일렬로 세우는 경우의 수

[문제 1] 줄 세우기

5

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

$T(i)$ = i 명의 학생을 일렬로 세우는 경우의 수

[문제 1] 줄 세우기

5

2. 점화식을 구한다

그 값을 어떻게 구할지에 대한 식을 세운다

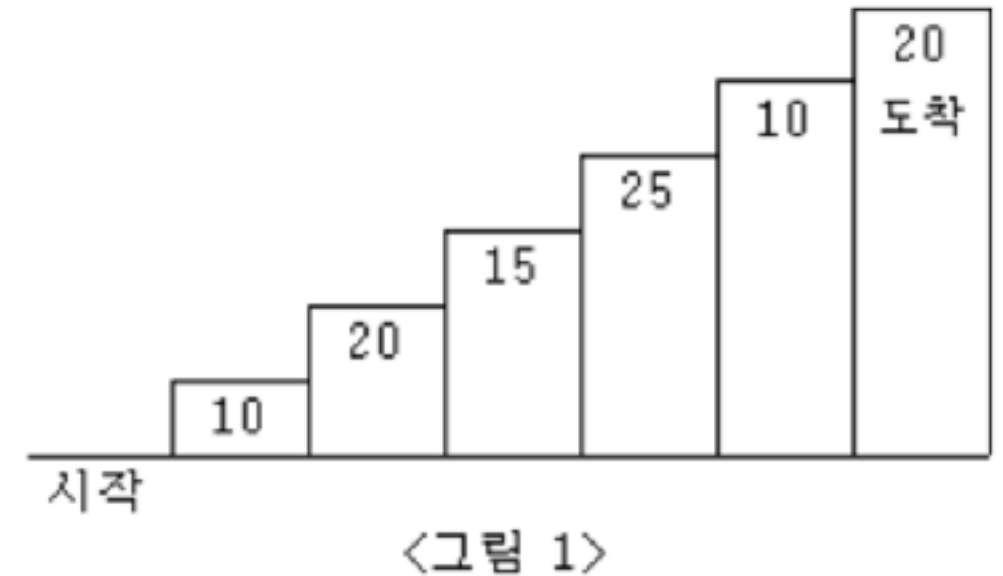
$T(i)$ = i 명의 학생을 일렬로 세우는 경우의 수

$$T(i) = T(i-1) + T(i-2)$$

... Q ... P

[문제 2] 계단 오르기

연속된 세 개의 계단을 밟지 않고,
마지막 계단을 밟을 때
얻을 수 있는 점수의 최댓값은 ?



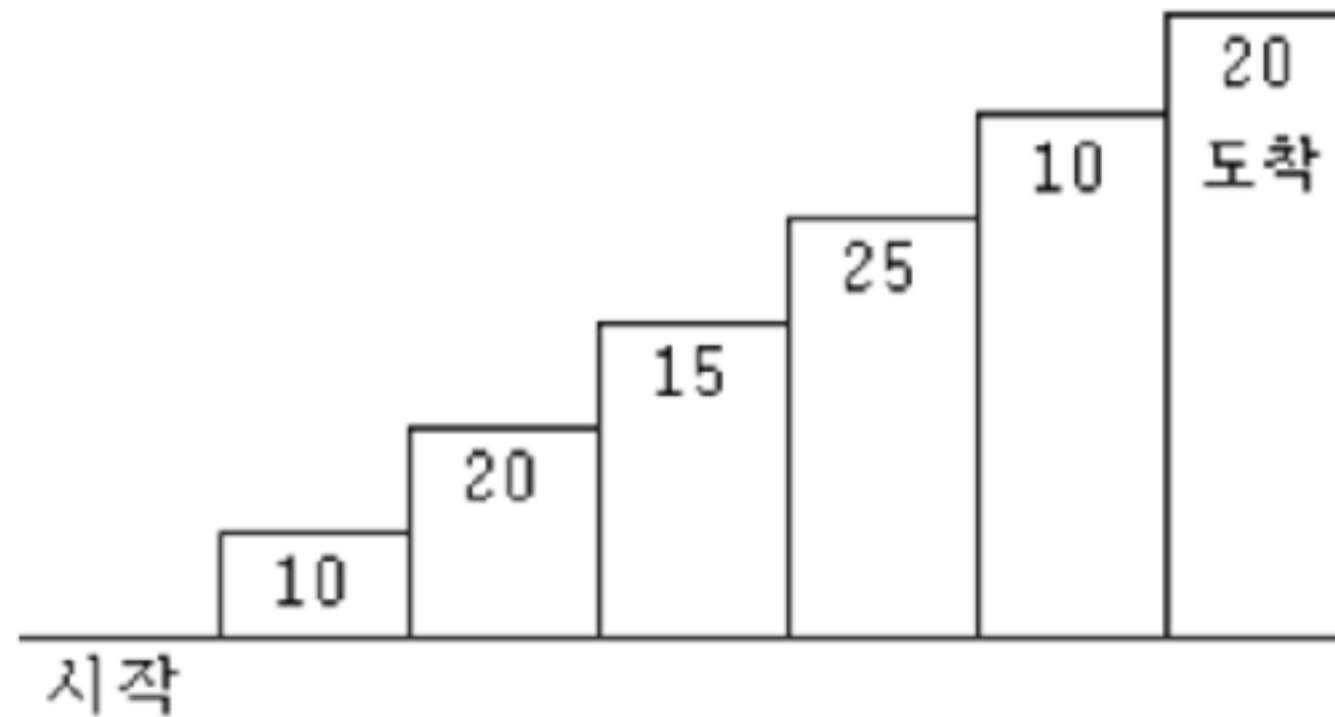
입력의 예

10 20 15 25 10 20

출력의 예

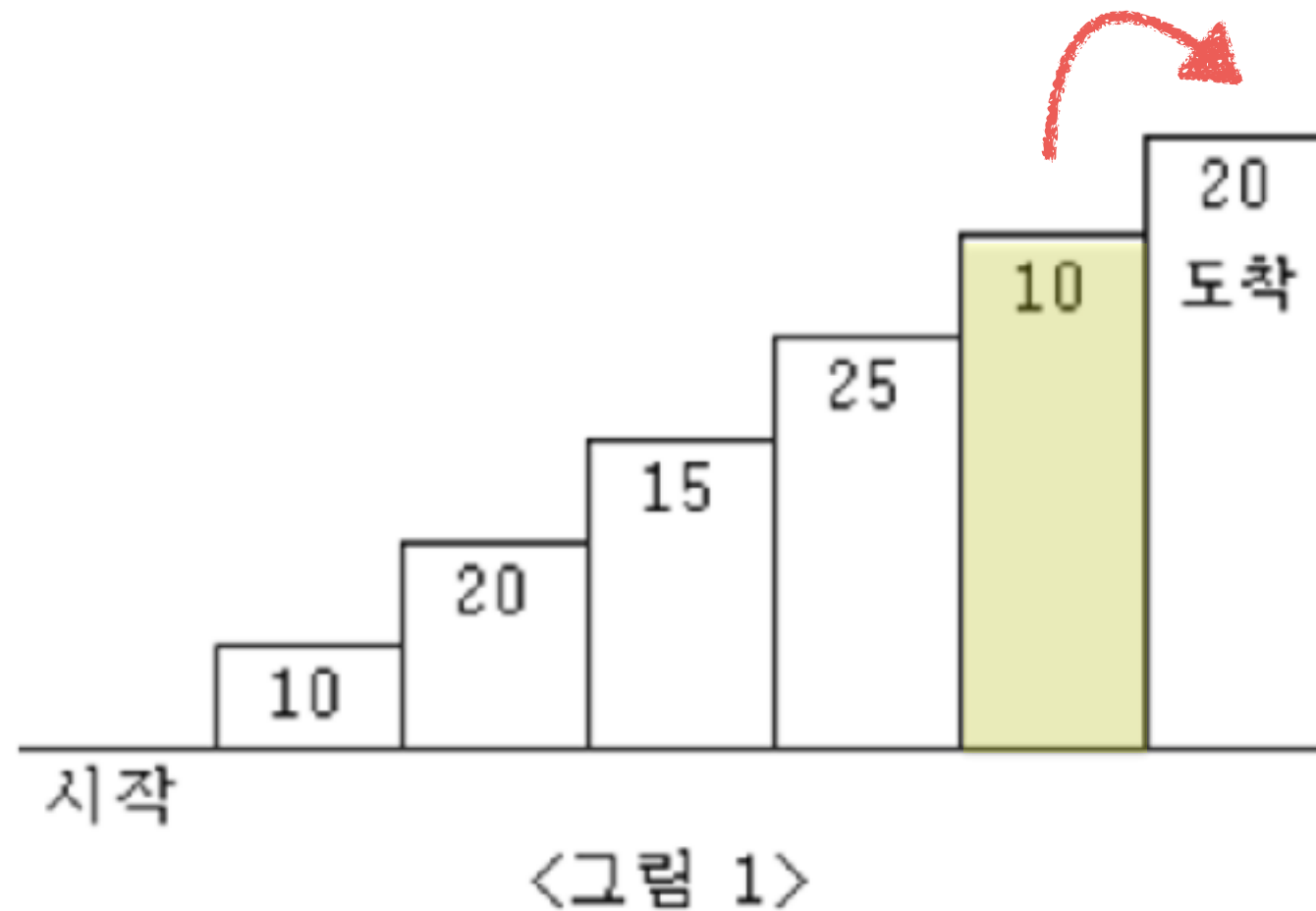
75

[문제 2] 계단 오르기

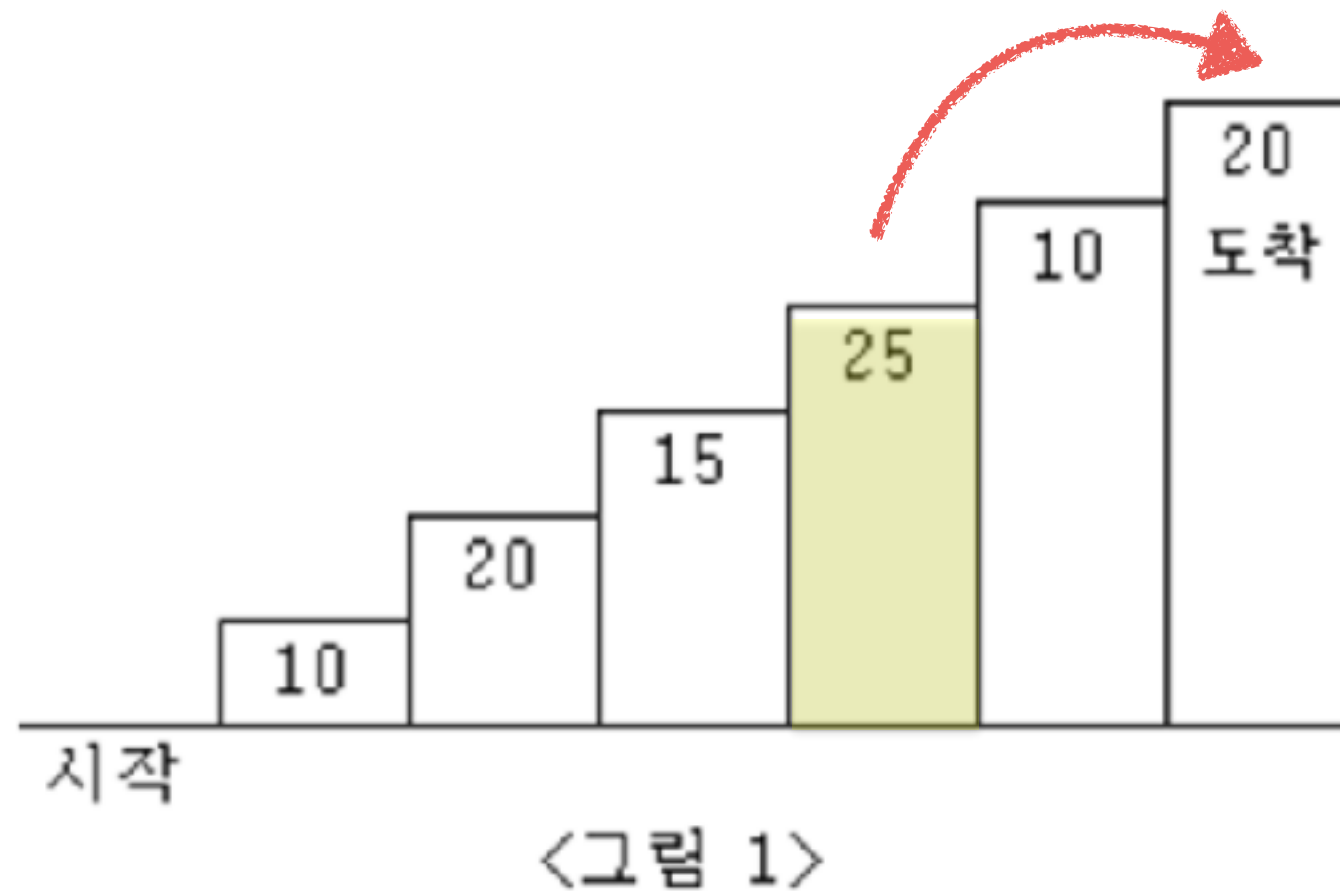


<그림 1>

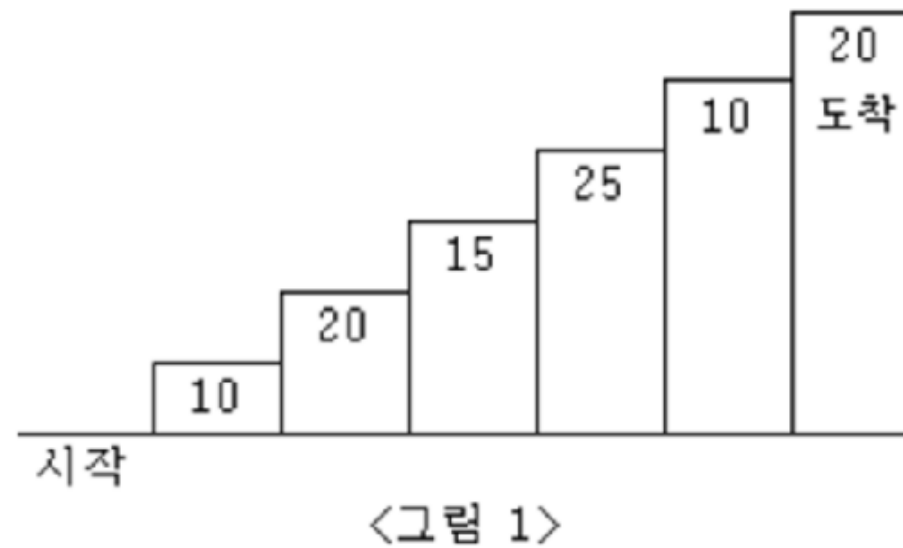
[문제 2] 계단 오르기



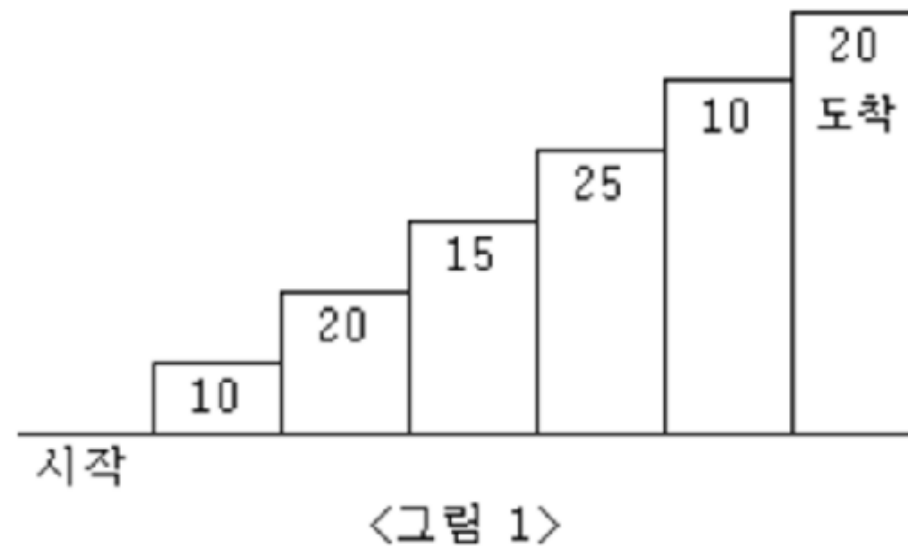
[문제 2] 계단 오르기



[문제 2] 계단 오르기

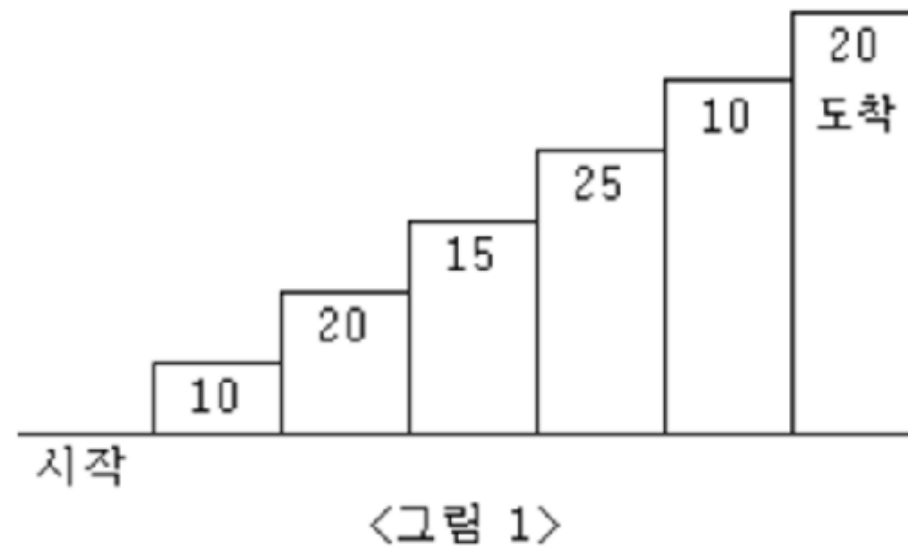


[문제 2] 계단 오르기



$T(i)$ = i 번째 칸을 도착이라고 할 때, 얻을 수 있는 점수의 최댓값

[문제 2] 계단 오르기



$T(i)$ = i 번째 칸을 도착이라고 할 때, 얻을 수 있는 점수의 최댓값

$$T(i) = \max(T(i-1), T(i-2)) + \text{Data}(i)$$

[문제 3] 짜장, 짬뽕, 볶음밥

각 날짜에 대하여 음식의 선호도가 주어질 때,
얻을 수 있는 만족감의 최댓값은 ?

입력의 예

```
3
27 8 35
18 36 10
7 22 45
```

출력의 예

```
116
```

[문제 3] 짜장, 짬뽕, 볶음밥

27 8 35

18 36 10

7 22 45

[문제 3] 짜장, 짬뽕, 볶음밥

27 8 35

18 36 10

7 22 45

[문제 3] 짜장, 짬뽕, 볶음밥

27 8 35

18 36 10

7 22 45

[문제 3] 짜장, 짬뽕, 볶음밥

27	8	35
18	36	10
7	22	45

[문제 3] 짜장, 짬뽕, 볶음밥

27 8 35

18 36 10

7 22 45

$T(i, 0)$ = i 번째 날까지 모든 음식을 먹으며, i 번째에는 짜장을 먹을 때,
얻을 수 있는 만족도의 최댓값

[문제 3] 짜장, 짬뽕, 볶음밥

27 8 35

18 36 10

7 22 45

$T(i, 0)$ = i 번째 날까지 모든 음식을 먹으며, i 번째에는 **짜장**을 먹을 때,
얻을 수 있는 만족도의 최댓값

$T(i, 1)$ = i 번째 날까지 모든 음식을 먹으며, i 번째에는 **짬뽕**을 먹을 때,
얻을 수 있는 만족도의 최댓값

$T(i, 2)$ = i 번째 날까지 모든 음식을 먹으며, i 번째에는 **볶음밥**을 먹을 때,
얻을 수 있는 만족도의 최댓값

[문제 3] 짜장, 짬뽕, 볶음밥

27	8	35
18	36	10
7	22	45

$T(i, 0) =$

[문제 3] 짜장, 짬뽕, 볶음밥

27	8	35
18	36	10
7	22	45

$$T(i, 0) = \max(T(i-1, 1), T(i-1, 2)) + \text{Data}(i, 0)$$

[문제 3] 짜장, 짬뽕, 볶음밥

27	8	35
18	36	10
7	22	45

$$T(i, 0) = \max(T(i-1, 1), T(i-1, 2)) + \text{Data}(i, 0)$$

$$T(i, 1) = \max(T(i-1, 0), T(i-1, 2)) + \text{Data}(i, 1)$$

$$T(i, 2) = \max(T(i-1, 0), T(i-1, 1)) + \text{Data}(i, 2)$$

[문제 3] 짜장, 짬뽕, 볶음밥



`/* elice */`

[예제 1] 최장 증가 부분 수열

최장 증가 부분 수열의 길이를 구하라

입력의 예

```
5
1 4 2 3 5
```

출력의 예

```
4
```

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

5를 포함하는 경우 :

5를 포함하지 않는 경우 :

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

5를 포함하는 경우 :

5를 포함하지 않는 경우 : (1 4 2 3 의 최장 증가 부분 수열)

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

5를 포함하는 경우 : ???

5를 포함하지 않는 경우 : (1 4 2 3 의 최장 증가 부분 수열)

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

... 1 5

... 4 5

5를 포함하는 경우 :

... 2 5

... 3 5

5를 포함하지 않는 경우 : (1 4 2 3 의 최장 증가 부분 수열)

[예제 1] 최장 증가 부분 수열

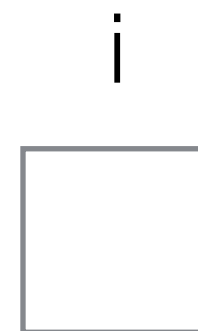
1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이



[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이



[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이



$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

	0	1	2	3	4
T					

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

	0	1	2	3	4
T	1				

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

	0	1	2	3	4
T	1	2			

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

	0	1	2	3	4
T	1	2	2		

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

	0	1	2	3	4
T	1	2	2	3	

[예제 1] 최장 증가 부분 수열

1 4 2 3 5

$T(i)$ = i 번째 숫자를 끝으로 하는 최장 증가 부분 수열의 길이

$$T(i) = \max(T(j) + 1) \quad \text{if } \text{Data}(j) < \text{Data}(i)$$

	0	1	2	3	4
T	1	2	2	3	4

[예제 1] 최장 증가 부분 수열



`/* elice */`

[예제 2] 최대 공통 부분 수열

최대 공통 부분 수열의 길이를 구하라

입력의 예

Telephone
Television

출력의 예

6

[예제 2] 최대 공통 부분 수열

T e l e p h o n e

T e l e v i s i o n

[예제 2] 최대 공통 부분 수열

T e l e p h o n e

T e l e v i s i o n

[예제 2] 최대 공통 부분 수열

Telephone

Television

MAX(Telephone, Telephon
Television)

[예제 2] 최대 공통 부분 수열

T e l e p h o n e

T e l e v i s i o n

$T(i, j)$ = $s_1[1..i]$, $s_2[1..j]$ 의 최대 공통 부분 수열의 길이

[예제 2] 최대 공통 부분 수열

T e l e p h o n e

T e l e v i s i o n

$T(i, j) = s_1[1..i], s_2[1..j]$ 의 최대 공통 부분 수열의 길이

s_1

s_2

j

i

[예제 2] 최대 공통 부분 수열

Telephone

Television

$T(i, j) = s_1[1..i], s_2[1..j]$ 의 최대 공통 부분 수열의 길이



[예제 2] 최대 공통 부분 수열

Telephone

Television

$T(i, j) = s_1[1..i], s_2[1..j]$ 의 최대 공통 부분 수열의 길이



$$T(i, j) = \max(T(i, j-1), T(i-1, j))$$

[예제 2] 최대 공통 부분 수열

T e l e p h o n e

T e l e v i s i o n

$T(i, j) = s_1[1..i], s_2[1..j]$ 의 최대 공통 부분 수열의 길이



[예제 2] 최대 공통 부분 수열

Telephone

Television

$T(i, j) = s_1[1..i], s_2[1..j]$ 의 최대 공통 부분 수열의 길이



$$T(i, j) = T(i-1, j-1) + 1$$

/* elice */

[예제 2] 최대 공통 부분 수열

T e l e p h o n e

T e l e v i s i o n

$T(i, j)$ = $s_1[1..i]$, $s_2[1..j]$ 의 최대 공통 부분 수열의 길이

$$T(i, j) = \begin{cases} \max(T(i-1, j), T(i, j-1)) & \text{if } s_1[i] \neq s_2[j] \\ T(i-1, j-1) + 1 & \text{if } s_1[i] == s_2[j] \end{cases}$$

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T										
e										
l										
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1									
e										
l										
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1								
e										
l										
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e										
l										
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1									
l										
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2								
l										
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l										
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1									
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2								
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3							
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e										
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1									
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2								
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3							
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4						
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p										
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p	1									
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p	1	2	3	4	4	4	4	4	4	4
h										
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p	1	2	3	4	4	4	4	4	4	4
h	1	2	3	4	4	4	4	4	4	4
o										
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p	1	2	3	4	4	4	4	4	4	4
h	1	2	3	4	4	4	4	4	4	4
o	1	2	3	4	4	4	4	4	5	5
n										
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p	1	2	3	4	4	4	4	4	4	4
h	1	2	3	4	4	4	4	4	4	4
o	1	2	3	4	4	4	4	4	5	5
n	1	2	3	4	4	4	4	4	5	6
e										

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p	1	2	3	4	4	4	4	4	4	4
h	1	2	3	4	4	4	4	4	4	4
o	1	2	3	4	4	4	4	4	5	5
n	1	2	3	4	4	4	4	4	5	6
e	1	2	3	4	4	4	4	4	5	6

[예제 2] 최대 공통 부분 수열

	T	e	l	e	v	i	s	i	o	n
T	1	1	1	1	1	1	1	1	1	1
e	1	2	2	2	2	2	2	2	2	2
l	1	2	3	3	3	3	3	3	3	3
e	1	2	3	4	4	4	4	4	4	4
p	1	2	3	4	4	4	4	4	4	4
h	1	2	3	4	4	4	4	4	4	4
o	1	2	3	4	4	4	4	4	5	5
n	1	2	3	4	4	4	4	4	5	6
e	1	2	3	4	4	4	4	4	5	6

[예제 2] 최대 공통 부분 수열



`/* elice */`

동적 계획법 (Dynamic Programming)

부분 문제를 해결한 결과를 이용하여 전체 문제를 해결

“나”를 해결함으로써 “나”를 해결

재귀호출

동적 계획법의 문제 풀이 순서

1. 부분 문제를 정의한다.
무슨 값을 구할지를 정의한다
2. 점화식을 구한다
그 값을 어떻게 구할지에 대한 식을 세운다
3. 문제를 해결한다
값을 직접 구하는 코드를 작성한다

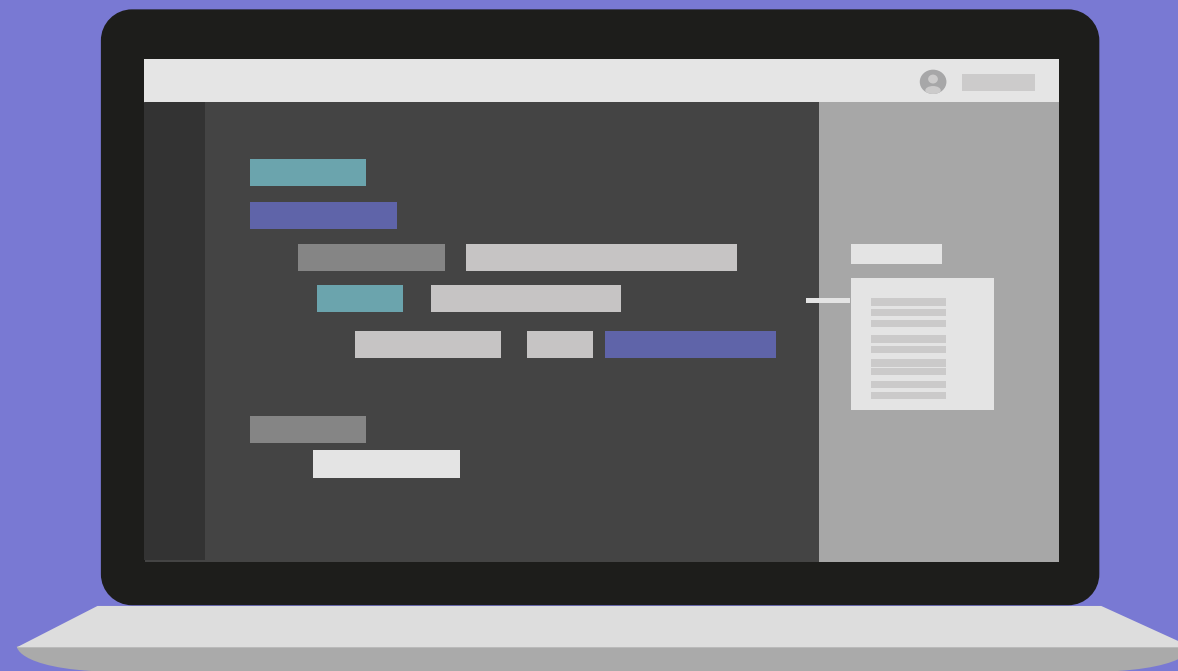
동적 계획법 (Dynamic Programming)

부분 문제를 정의하는 것이 가장 어렵다
어떻게 정의하느냐에 따라 풀리기도 하고 안풀리기도 함

문제가 “재귀적으로 해결되는지”를 볼 줄 알아야 함
규칙 찾는 것으로 보일 때도 있고, 실제로 틀린 말도 아님

무조건 많은 예제를 풀어보아야 한다
단언컨데 재귀호출, 동적계획법 기초 실력 향상은 양으로 승부봐야함

퀴즈 및 설문



/* elice */

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

blog.naver.com/elicer