

자료구조 1 (연습)

최백준 choi@startlink.io

단어 뒤집기 2

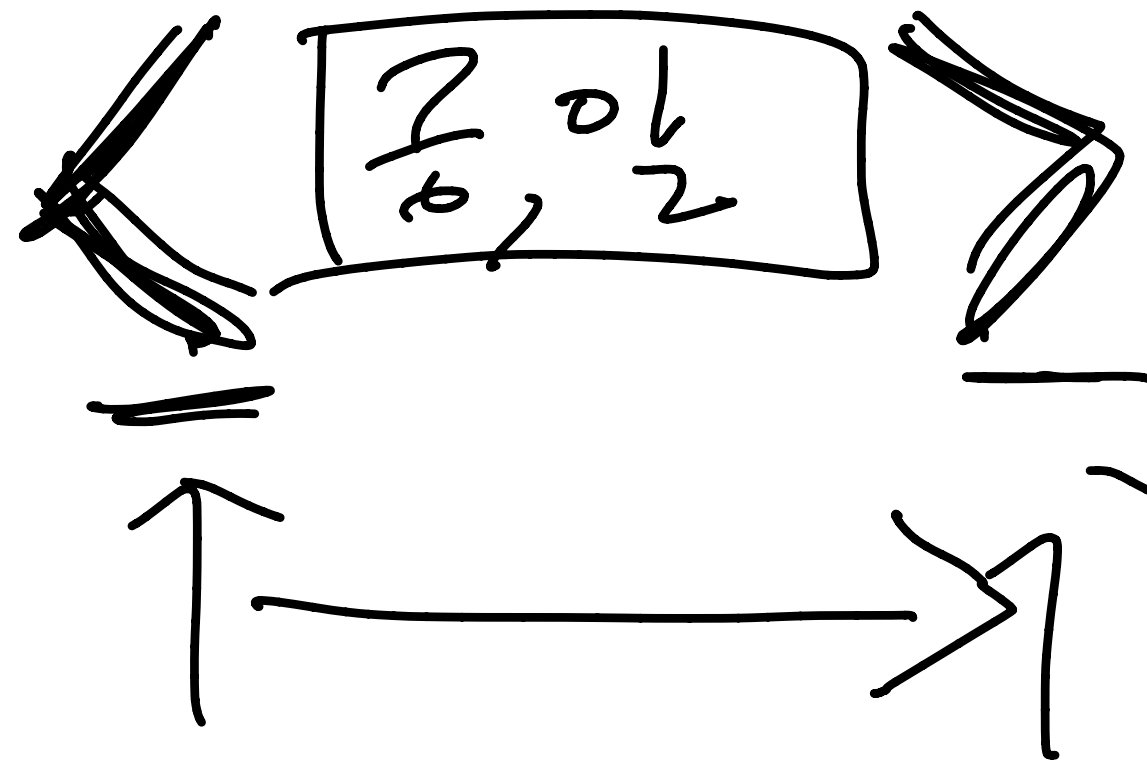
<https://www.acmicpc.net/problem/17413>

- 알파벳 소문자, 숫자, 공백, 특수 문자(<, >)로 이루어진 문자열 S에서 단어를 뒤집는 문제
- 태그(<로 시작해서 >로 끝나는 문자열, 사이에는 알파벳 소문자와 공백만 있음)
- 단어 알파벳 소문자와 숫자로만 이루어짐
- 단어와 단어 사이에는 공백 한 칸, 단어와 태그 사이에는 공백이 없음
- <open>tag<close> → <open>gat<close>
- <ab cd>ef gh<ij kl> → <ab cd>fe hg<ij kl>

단어 뒤집기 2

<https://www.acmicpc.net/problem/17413>

- 단어 뒤집기와 비슷하게 풀 수 있다.
- 태그 안인지 밖인지 구분해서 뒤집을지 말지 결정해야 한다.

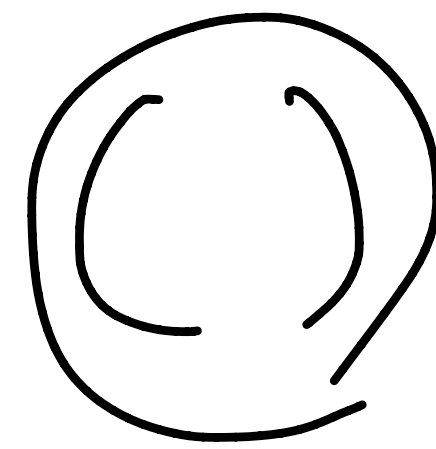


단어 뒤집기 2

<https://www.acmicpc.net/problem/17413>

- 소스: <http://codeplus.codes/b9fd7cb42aa346b2b12669c955839368>

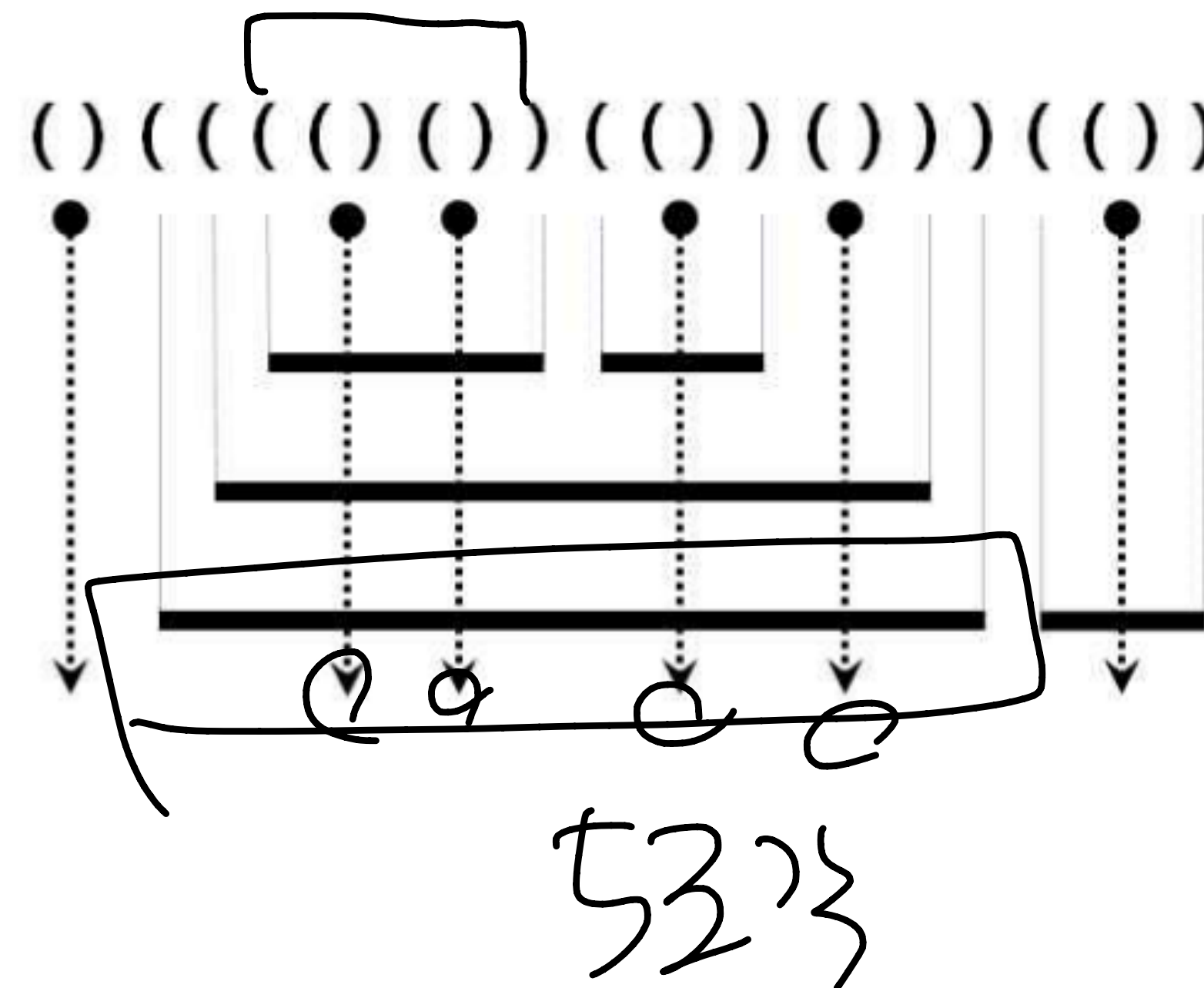
쇠막대기



5

<https://www.acmicpc.net/problem/10799>

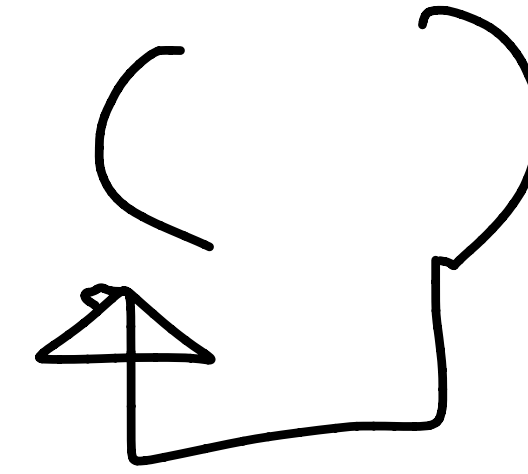
- 레이저는 여는 괄호와 닫는 괄호의 인접한 쌍 ‘()’ 으로 표현된다. 또한, 모든 ‘()’는 반드시 레이저를 표현한다.
- 쇠막대기의 왼쪽 끝은 여는 괄호 ‘(’ 로, 오른쪽 끝은 닫힌 괄호 ‘)’ 로 표현된다



1개의 쇠막대기
레이저 (37)
4743 43

쇠막대기

<https://www.acmicpc.net/problem/10799>



- 올바른 괄호 문자열과 비슷하게 풀 수 있다
- ()가 나올 때 마다 스택에 들어있는 (의 개수를 세어준다
- 그런데,)가 나왔을 때, 이것이 레이저인지 쇠막대기인지 구분을 해줘야 한다.
- 레이저는 항상 ()와 같이 붙어진 상태로 나온다
- 스택에 (의 인덱스를 넣어서 인덱스가 1차이 나는지 확인해야 한다

쇠막대기

7

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1]

쇠막대기

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2]

쇠막대기

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 3]

쇠막대기

10

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 3, 4]

쇠막대기

11

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 3, ~~4~~]
- 스택의 top과 인덱스 차이가 1 나기 때문에 레이저
- 스택: [1, 2, 3]
- 스택의 크기는 레이저가 자르는 쇠막대기의 개수와 같다.
- 정답 = 3

쇠막대기

12

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 3, 6]

쇠막대기

13

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 3, ~~6~~]
- 스택의 top과 인덱스 차이가 1 나기 때문에 레이저
- 스택: [1, 2, 3]
- 스택의 크기는 레이저가 자르는 쇠막대기의 개수와 같다.
- 정답 = 3 + 3

쇠막대기

14

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 3]
- 스택의 top과 인덱스 차이가 1 나지 않기 때문에 쇠막대기의 끝
- 스택: [1, 2]
- 쇠막대기 1개를 레이저가 3번 자르면, 쇠막대기는 4개의 조각으로 나누어진다.
- 레이저가 자르는 횟수를 정답에 더했기 때문에, 쇠막대기의 끝에서 정답을 1 더해야 한다.
- 정답 = $3 + 3 + 1$

쇠막대기

15

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 9]

쇠막대기

16

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 9, 10]

쇠막대기

17

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

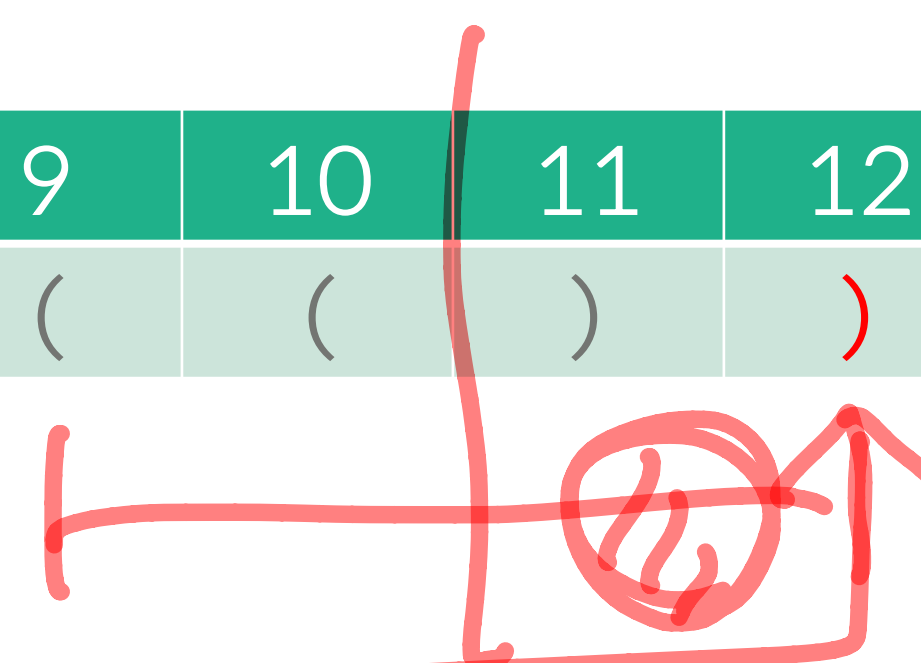
- 스택: [1, 2, 9, 10]
- 스택의 top과 인덱스 차이가 1 나기 때문에 레이저
- 스택: [1, 2, 9]
- 스택의 크기는 레이저가 자르는 쇠막대기의 개수와 같다.
- 정답 = 3 + 3 + 1 + 3

쇠막대기

18

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))



- 스택: [1, 2, 9]
- 스택의 top과 인덱스 차이가 1 나지 않기 때문에 쇠막대기의 끝
- 스택: [1, 2]
- 정답 = $3 + 3 + 1 + 3 + 1$

쇠막대기

19

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 13]

쇠막대기

20

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2, 13]
- 스택의 top과 인덱스 차이가 1 나기 때문에 레이저
- 스택: [1, 2]
- 스택의 크기는 레이저가 자르는 쇠막대기의 개수와 같다.
- 정답 = $3 + 3 + 1 + 3 + 1 + 2$

쇠막대기

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1, 2]
- 스택의 top과 인덱스 차이가 1 나지 않기 때문에 쇠막대기의 끝
- 스택: [1]
- 정답 = 3+ 3 + 1 + 3 + 1 + 2 + 1

쇠막대기

22

<https://www.acmicpc.net/problem/10799>

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
(((()	())	(())	()))

- 스택: [1]
- 스택의 top과 인덱스 차이가 1 나지 않기 때문에 쇠막대기의 끝
- 스택: []
- 정답 = 3 + 3 + 1 + 3 + 1 + 2 + 1 + 1

$O(N)$

최막대기

<https://www.acmicpc.net/problem/10799>

- 소스: <http://codeplus.codes/dc40a92c327543c6958c28ca4cde6daa>

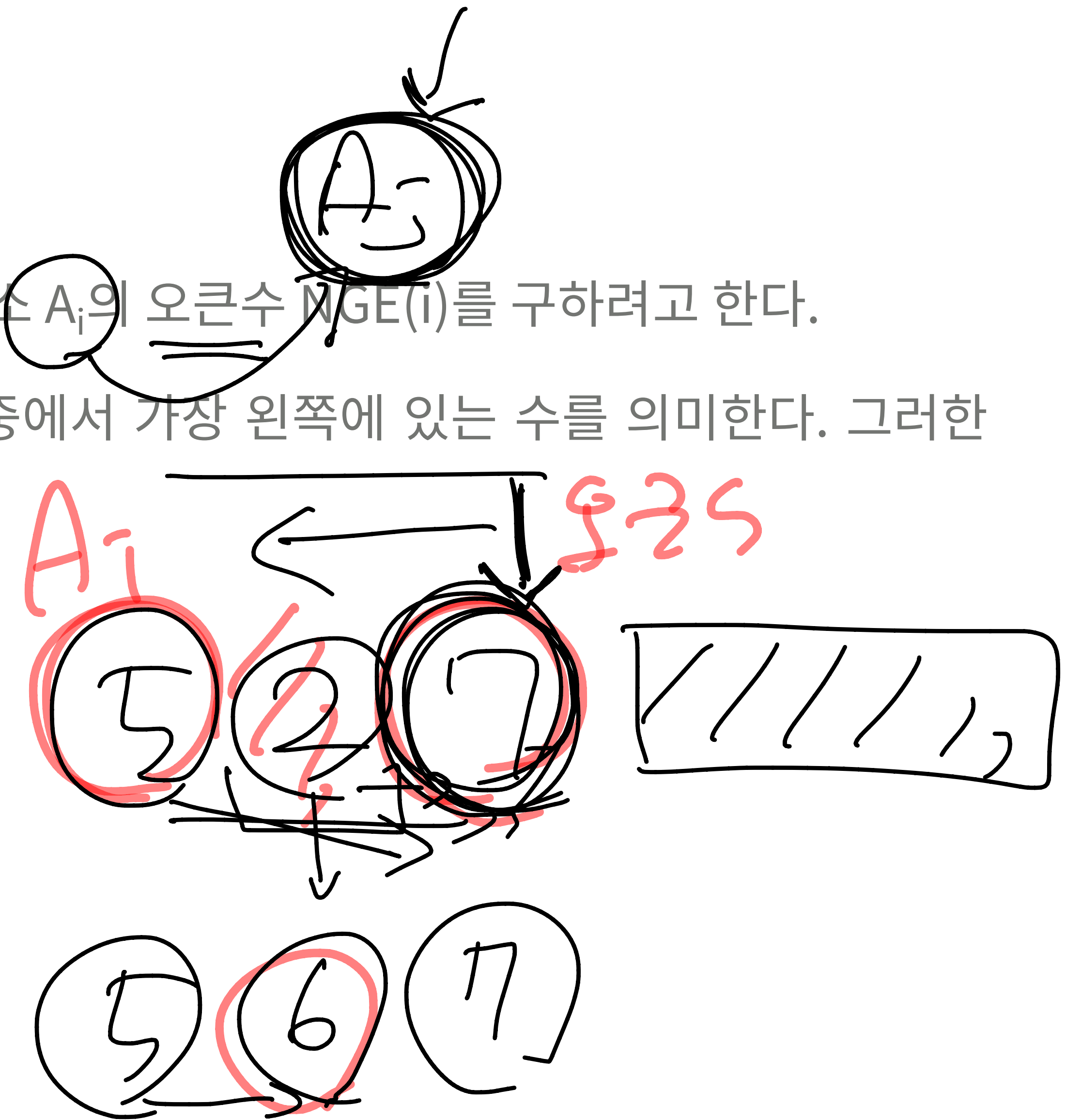
오큰수

<https://www.acmicpc.net/problem/17298>

- 크기가 N 인 수열 $A = A_1, A_2, \dots, A_N$ 이 있고, 각 원소 A_i 의 오큰수 $NGE(i)$ 를 구하려고 한다.
- A_i 의 오큰수는 오른쪽에 있으면서 A_i 보다 큰 수 중에서 가장 왼쪽에 있는 수를 의미한다. 그러한 수가 없는 경우에 오큰수는 -1 이다.

$A = [3, 5, 2, 7] \rightarrow NGE = [5, 7, 7, -1]$

$A = [9, 5, 4, 8] \rightarrow NGE = [-1, 8, 8, -1]$



오큰수

<https://www.acmicpc.net/problem/17298>

- 각각의 수 A_j 에 대해서, A_j 가 오큰수인 A_i 를 찾는 방법이 있다. ($i < j$)
- 오큰수를 찾지 못한 수를 스택에 넣는다면, 스택에는 top에 있는 수가 가장 작은 내림차순처럼 저장되게 된다.

오큰수

<https://www.acmicpc.net/problem/17298>

i	1	2	3	4
A[i]	3	5	2	7
NGE(i)				

26

- $A[1] = 3$ 인 경우 아직 오큰수를 찾을 수 없다. 따라서, 스택에 위치 1을 넣는다.
- 스택 = [1]

오큰수

<https://www.acmicpc.net/problem/17298>

i	1	2	3	4
A[i]	3	5	2	7
NGE(i)	5			

27

- $A[2] = 5$, 스택 = [1]
- 스택의 가장 위에 있는 수 $A[1] = 3$ 이고, 현재 수는 $A[2] = 5$ 이다.
- $A[1] < A[2]$ 이기 때문에, $NGE(1) = 5$ 인 것을 알 수 있다.
- 스택 = []
- 더 이상 스택에 수가 없기 때문에, 위치 2를 스택에 넣는다.
- 스택 = [2]

오큰수

<https://www.acmicpc.net/problem/17298>

i	1	2	3	4
A[i]	3	5	2	7
NGE(i)	5			

28

- $A[3] = 2$, 스택 = [2]
- 스택의 가장 위에 있는 수 $A[2] = 5$ 이고, 현재 수는 $A[3] = 2$ 이다.
- $A[2] > A[3]$ 이기 때문에, 현재 스택에 있는 수의 오큰수는 $A[3]$ 이 아니다.
- 스택에 들어있는 수의 오큰수는 3보다 큰 위치에 있기 때문에, 그냥 넘어간다.
- 위치 3을 스택에 넣는다.
- 스택 = [2, 3]

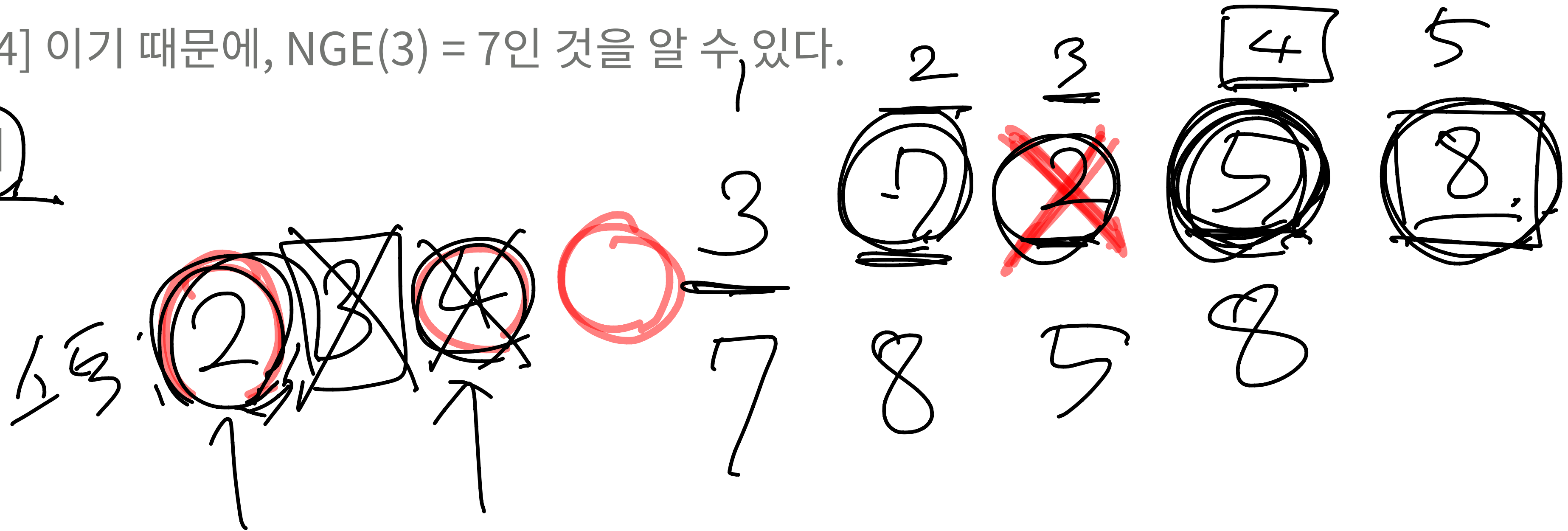
오큰수

<https://www.acmicpc.net/problem/17298>

29

i	1	2	3	4
A[i]	3	5	2	7
NGE(i)	5	7	7	

- $A[4] = 7$, 스택 = $[2, 3]$
- 스택의 가장 위에 있는 수 $A[3] = 2$ 이고, 현재 수는 $A[4] = 7$ 이다.
- $A[3] < A[4]$ 이기 때문에, $NGE(3) = 7$ 인 것을 알 수 있다.
- 스택 = $[2]$



오큰수

<https://www.acmicpc.net/problem/17298>

i	1	2	3	4
A[i]	3	5	2	7
NGE(i)	5	7	7	

- $A[4] = 7$, 스택 = [2]
- 스택의 가장 위에 있는 수 $A[2] = 5$ 이고, 현재 수는 $A[4] = 7$ 이다.
- $A[2] < A[4]$ 이기 때문에, $NGE(2) = 7$ 인 것을 알 수 있다.
- 스택 = []
- 위치 4를 스택에 넣는다.

오큰수

<https://www.acmicpc.net/problem/17298>

- $A[4] = 7$, 스택 = []
- 모든 것이 끝났고, 스택이 비어있지 않다.
- 스택에 들어있는 수는 오큰수가 -1이다.

i	1	2	3	4
A[i]	3	5	2	7
NGE(i)	5	7	7	-1

2 4 4

오큰수

<https://www.acmicpc.net/problem/17298>

- 소스: <http://codeplus.codes/71cfbac1762c472da9d1bff4568a3611>

오등큰수

<https://www.acmicpc.net/problem/17299>

- 크기가 N 인 수열 $A = A_1, A_2, \dots, A_N$ 이 있고, 각 원소 A_i 의 오등큰수 $NGF(i)$ 를 구하려고 한다.
- $F(A_i) = A_i$ 가 수열 A 에서 등장한 횟수
- A_i 의 오등큰수는 오른쪽에 있으면서 $F(A_i)$ 보다 등장한 횟수가 큰 수 중에서 가장 왼쪽에 있는 수를 의미한다. 그러한 수가 없는 경우에 오등큰수는 -1 이다.
- $A = [1, 1, 2, 3, 4, 2, 1] \rightarrow NGF = [-1, -1, 1, 2, 2, 1, -1]$

오등큰수

<https://www.acmicpc.net/problem/17299>

- 오큰수와 같은 방법으로 풀 수 있다.
- 수의 크기 대신 등장 횟수를 비교하는 방식으로 구현하면 된다.

오등큰수

35

<https://www.acmicpc.net/problem/17299>

- 소스: <http://codeplus.codes/cd9b2c3491e24caf83ecff84abf00757>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.