

세그먼트 트리 활용하기

최백준 choi@startlink.io

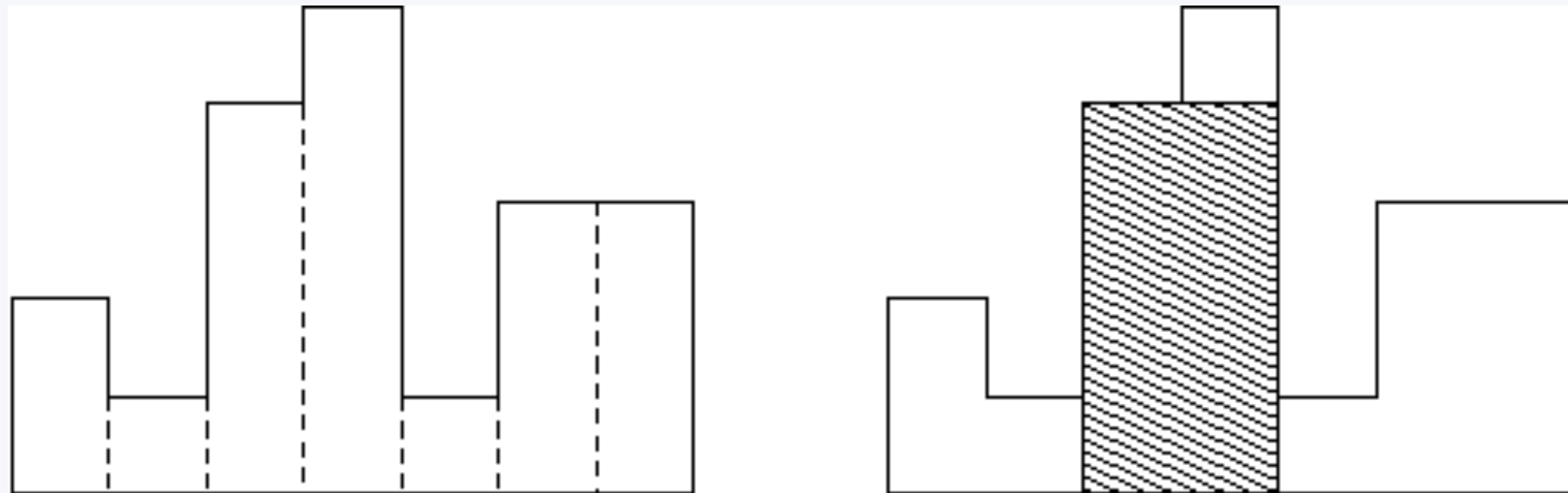
최소값 찾기

히스토그램에서 가장 큰 직사각형

3

<https://www.acmicpc.net/problem/6549>

- 히스토그램이 주어졌을 때, 가장 큰 직사각형을 찾는 문제

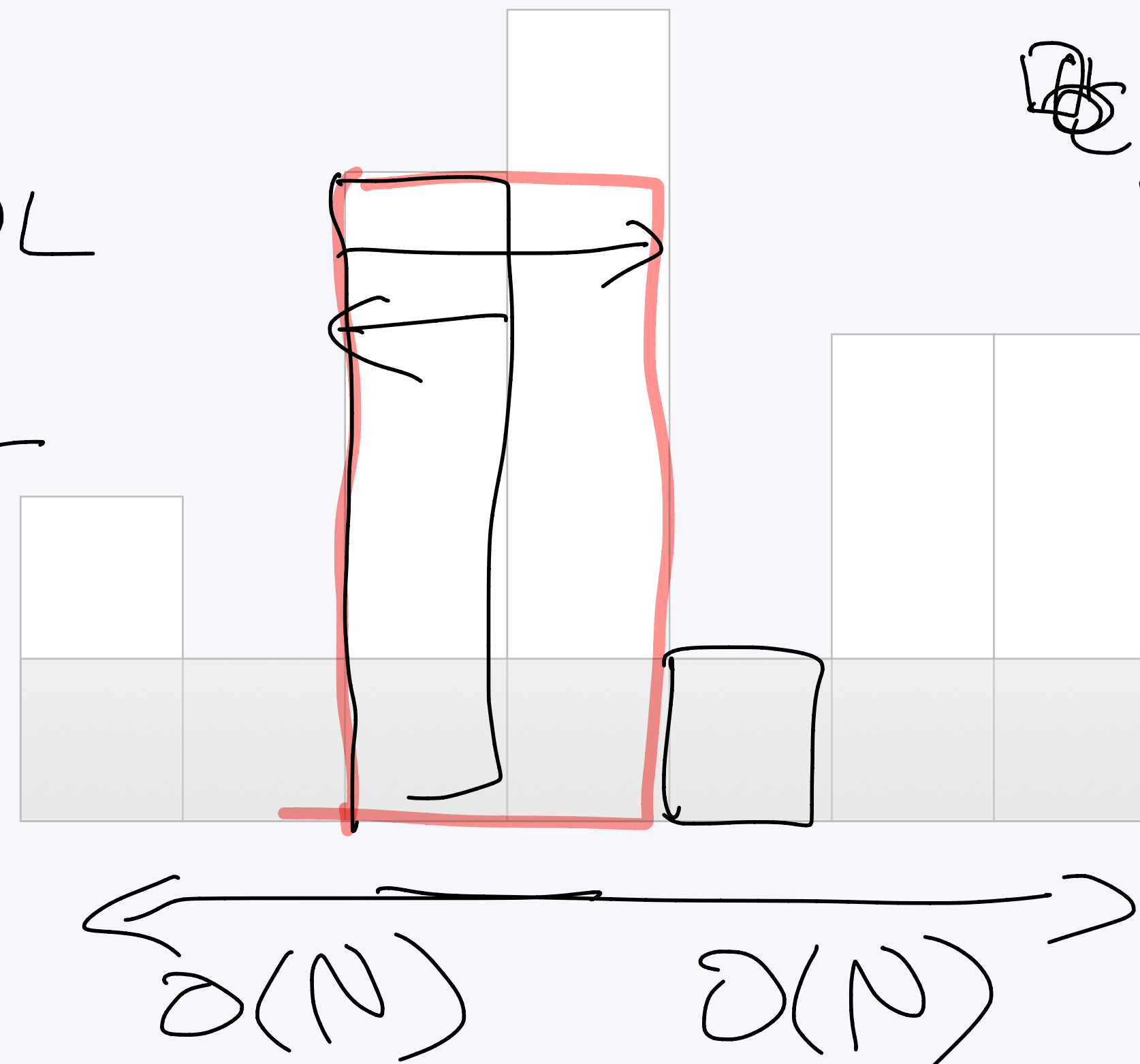


히스토그램에서 가장 큰 직사각형

<https://www.acmicpc.net/problem/6549>

- 가장 왼쪽 끝과 오른쪽 끝을 변으로 하는 가장 큰 직사각형의 높이는?
- 높이 : 히스토그램에서 가장 높이가 낮은 막대의 높이

↔
 막대 높이 높이기
 크기나 같게



각각의 막대씩 대해/n

그 막대를 높이로 하는

가장 큰 직사각형

$O(N)$

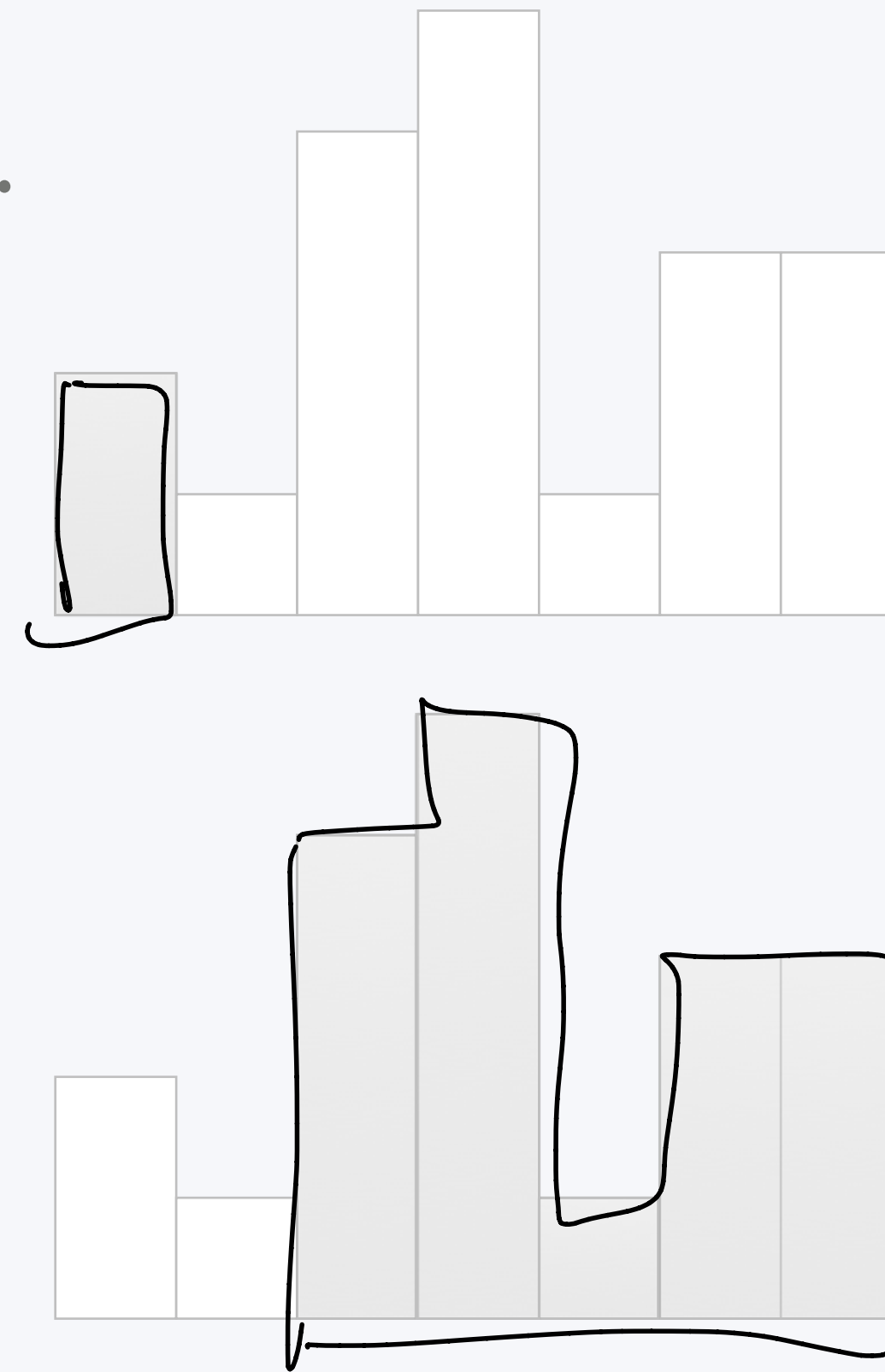
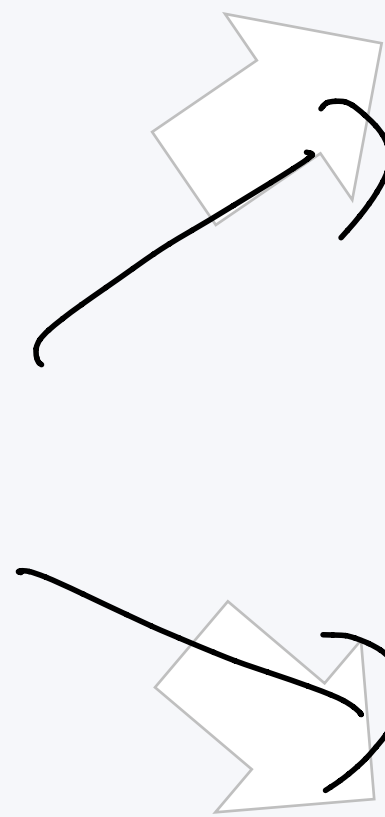
$O(N^2)$

히스토그램에서 가장 큰 직사각형

5

<https://www.acmicpc.net/problem/6549>

- 높이가 가장 낮은 막대의 번호를 m 이라고 했을 때, 이 직사각형은 높이가 $h[m]$ 이면서 만들 수 있는 가장 큰 직사각형이다.
- 이제 m 의 왼쪽과 오른쪽으로 나뉘어서 문제를 풀 수 있다.

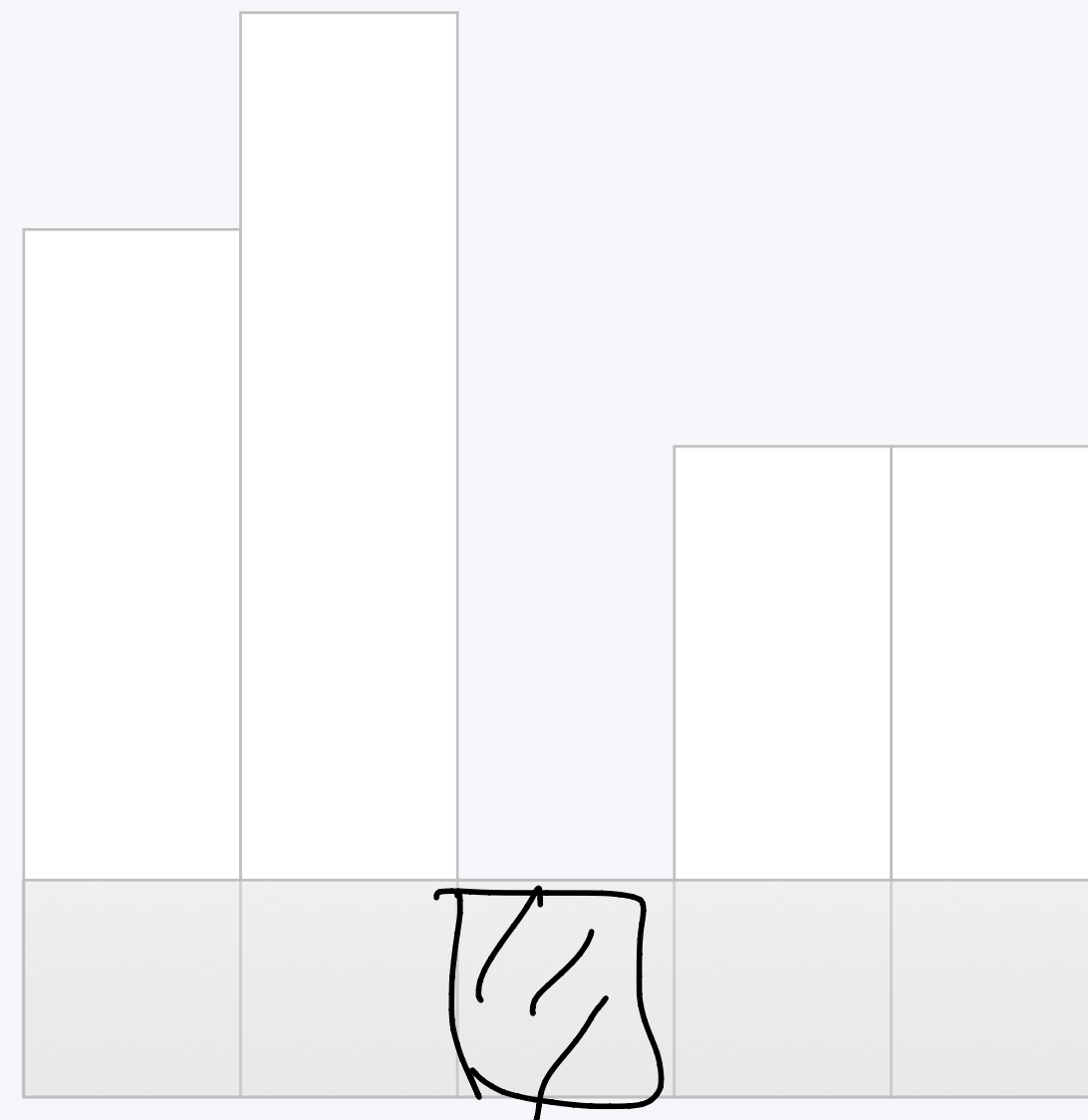


히스토그램에서 가장 큰 직사각형

6

<https://www.acmicpc.net/problem/6549>

- 오른쪽 히스토그램에서 만들 수 있는 가장 큰 직사각형은 아래 그림과 같다

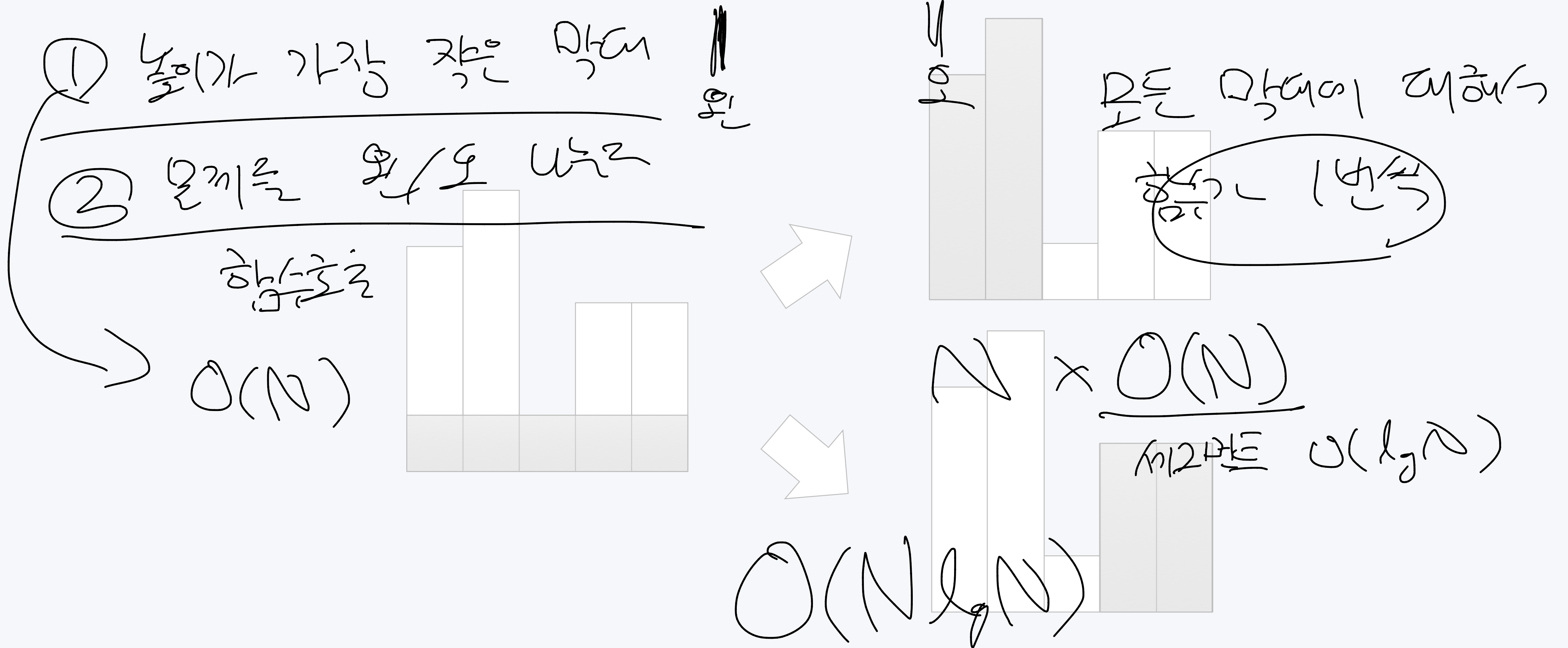


히스토그램에서 가장 큰 직사각형

7

<https://www.acmicpc.net/problem/6549>

- 여기서 다시 높이가 가장 낮은 막대 m 을 찾고 왼쪽과 오른쪽으로 나눠서 문제를 풀 수 있다



히스토그램에서 가장 큰 직사각형

8

<https://www.acmicpc.net/problem/6549>

- 세그먼트 트리에 최소값의 위치를 저장해서 풀 수 있다.
- C/C++: <https://gist.github.com/Baekjoon/570dfd7caaf5ba597bf2>

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 A가 주어졌을 때, 가장 긴 증가하는 부분 수열을 구하는 프로그램을 작성하시오
- 예를 들어, 수열 $A = \{10, 20, 10, 30, 20, 50\}$ 인 경우에 가장 긴 증가하는 부분 수열은 $A = \{10, 20, 10, 30, 20, 50\}$ 이고, 길이는 4이다

$N \leq 1,000,000$

$1 \leq A[i] \leq 1,000,000$

$$DP[i] = \max(DP[j]) + 1$$

① $j < i$
② $A[j] < A[i]$

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- $tree[i]$ = 수 i 를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

$tree[i] = \tau_2 \text{ 마지막}$

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- tree[i] = 수 i를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

↓

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	1	2	3	4	5
tree[i]	0	0	0	0	0

$$DP[i] = (DP[j]) + 1$$

max

11

$$j < i$$
$$A[j] < A[i]$$

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- tree[i] = 수 i를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

$1 \leq A[i] \leq 1000000$

$J < I$

$A[J] < A[I]$

$1 \sim 2$

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

$1 + 1 = 2$

i	1	2	3	4	5
tree[i]	1	0	2	0	0

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- tree[i] = 수 i를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이



가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- tree[i] = 수 i를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

1 ~ 1

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

H=2

i	1	2	3	4	5
tree[i]	1	2	2	0	0

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- tree[i] = 수 i를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

트리에
들어갔다

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

1 ~ 3

i	1	2	3	4	5
tree[i]	1	2	2	0	0

$2 + 1 = 3$

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- tree[i] = 수 i를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

1~2

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

2+1=3

i	1	2	3	4	5
tree[i]	1	2	2 3	3	0

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- $tree[i]$ = 수 i 를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

1 ~ 3

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	1	2	3	4	5
tree[i]	1	2	3	3 4	0

$$3 + 1 = 4$$

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- tree[i] = 수 i를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

가장 긴 수열 찾기
구간이 최대

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

$O(\lg N)$
 $N \lg N$

i	1	2	3	4	5
tree[i]	1	2	3	4	0

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- $tree[i]$ = 수 i 를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

4

i	1	2	3	4	5
tree[i]	1	2	3	4	0

가장 긴 증가하는 부분 수열 2

20

<https://www.acmicpc.net/problem/12015>

- C/C++: <https://gist.github.com/Baekjoon/ccb08adc7c49b77f80c13a76884366d0>
- Java: <https://gist.github.com/Baekjoon/5996f7ae2c9c19c730cc85e3808c33c1>

가장 긴 증가하는 부분 수열 3

<https://www.acmicpc.net/problem/12738>

②

- 수의 범위가 $-1,000,000,000 \leq A[i] \leq 1,000,000,000$ 이다
- 세그먼트 트리를 만들 수 없다

242 N ≤ 10⁵
-1 ≤ A[i] ≤ 1,000,000,000

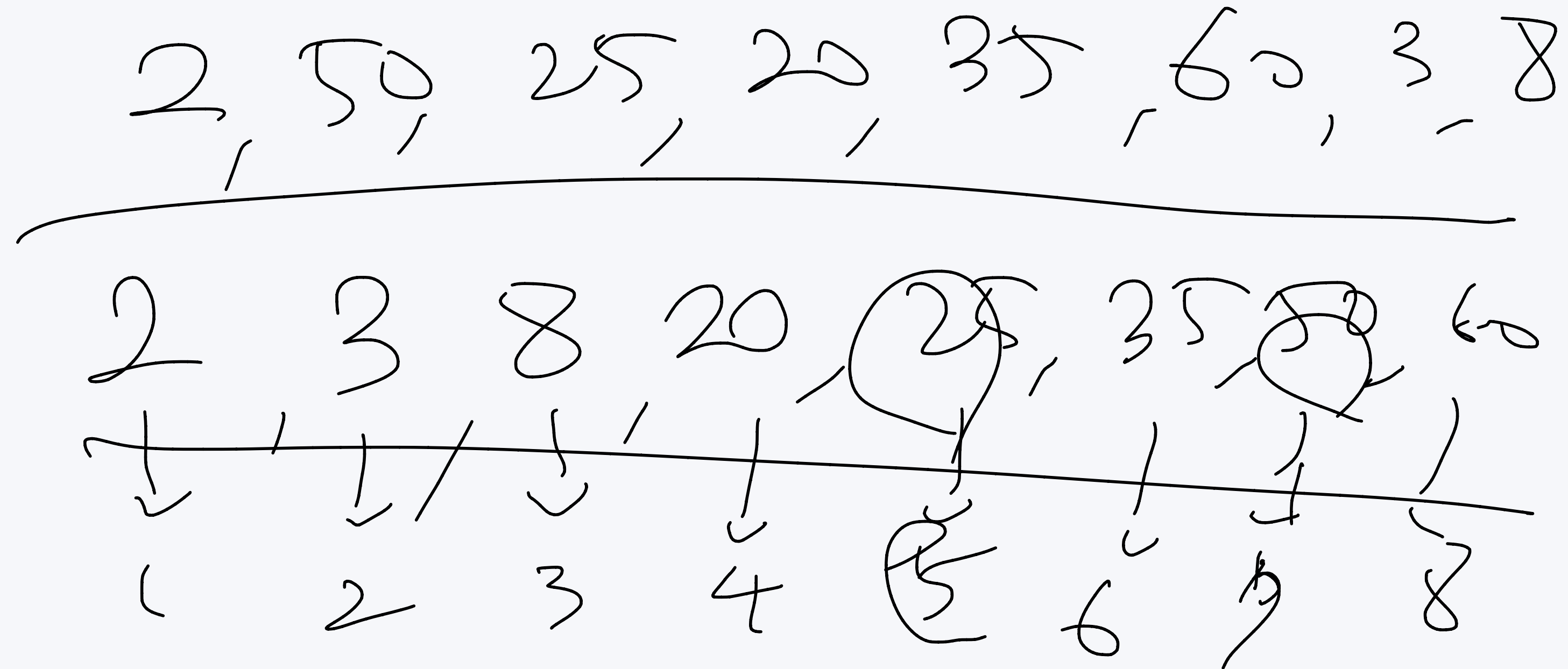
가장 긴 증가하는 부분 수열 3

22

<https://www.acmicpc.net/problem/12738>

- 2, 50, 25, 20, 35, 60, 20, 50, 35, 3, 8, 25
- 에서 정답은
- 1, 7, 5, 4, 6, 8, 4, 7, 6, 2, 3, 5
- 에서 정답과 같다

$$1 \leq A[i] \leq N$$



가장 긴 증가하는 부분 수열 3

23

<https://www.acmicpc.net/problem/12738>

- C/C++: <https://gist.github.com/Baekjoon/b8c2fa90b17a6db18e71d40ec65f95e2>

굉장한 학생

<https://www.acmicpc.net/problem/2336>

$$N \leq 5000$$

- A라는 학생이 B라는 학생보다 세 번의 시험에서 모두 성적이 좋다면, A가 B보다 '대단하다'고 한다.
- 또, C라는 학생보다 '대단한' 학생이 한 명도 없으면, C를 '굉장하다'고 한다.
- 세 번의 시험에서 각 학생의 성적이 주어졌을 때, '굉장한' 학생의 수를 구하는 문제

굉장한 학생

25

<https://www.acmicpc.net/problem/2336>

- $A1 < A2$: 위치
- $B1 < B2$: 넣는 순서
- $C1 < C2$: 넣는 값

굉장한 학생

<https://www.acmicpc.net/problem/2336>

$$tree[i] = \frac{\text{1번 시험을 [등호]한 학생의 수}}{\text{3번 시험 등수}}$$

- 학생의 등수
- 7 17 (굉장한 학생)
- 1 2 8
- 3 3 1
- 10 4 6
- 2 5 5
- 8 6 9
- 6 7 3
- 4 8 2
- 9 9 10
- 5 10 4

$$N = 10$$

$$N + 1$$

tree[1]	tree[2]	tree[3]	tree[4]	tree[5]	tree[6]	tree[7]	tree[8]	tree[9]	tree[10]
11	11	11	11	11	11	7	11	11	11
11									

굉장한 학생

<https://www.acmicpc.net/problem/2336>

• 학생의 등수

• 7 1 7

• 128 (굉장한 학생)

• 3 3 1

• 10 4 6

• 2 5 5

• 8 6 9

• 6 7 3

• 4 8 2

• 9 9 10

• 5 10 4

1 ~ 0

tree[1]	tree[2]	tree[3]	tree[4]	tree[5]	tree[6]	tree[7]	tree[8]	tree[9]	tree[10]
8	11	11	11	11	11	7	11	11	11

11

굉장한 학생

<https://www.acmicpc.net/problem/2336>

- 학생의 등수
- 7 1 7
- 1 2 8
- 3 3 1 (굉장한 학생)
- 10 4 6
- 2 5 5
- 8 6 9
- 6 7 3
- 4 8 2
- 9 9 10
- 5 10 4

1번 : 3등

141
1 ~ 2 등

1, 2 등

tree[1]	tree[2]	tree[3]	tree[4]	tree[5]	tree[6]	tree[7]	tree[8]	tree[9]	tree[10]
8	11	11	11	11	11	7	11	11	11
	8								

굉장한 학생

<https://www.acmicpc.net/problem/2336>

학생의 등수

• 7 1 7

• 1 2 8

• 3 3 1

• ~~10 4 6~~

• 2 5 5

• 8 6 9

• 6 7 3

• 4 8 2

• 9 9 10

• 5 10 4

1 ~ 9

tree[1]	tree[2]	tree[3]	tree[4]	tree[5]	tree[6]	tree[7]	tree[8]	tree[9]	tree[10]
8	11	1	11	11	11	7	11	11	11

1

굉장한 학생

<https://www.acmicpc.net/problem/2336>

- 학생의 등수
- 7 1 7
- 1 2 8
- 3 3 1
- 10 4 6
- 2 5 5 (굉장한 학생)
- 8 6 9
- 6 7 3
- 4 8 2
- 9 9 10
- 5 10 4

$$\frac{85}{10} > \frac{55}{6}$$

tree[1]	tree[2]	tree[3]	tree[4]	tree[5]	tree[6]	tree[7]	tree[8]	tree[9]	tree[10]
8	511	1	11	11	11	7	11	11	6

8

- 학생의 등수

- $5 \cdot 10^4$

tree[1]	tree[2]	tree[3]	tree[4]	tree[5]	tree[6]	tree[7]	tree[8]	tree[9]	tree[10]
8	5	1	11	11	11	7	11	11	6

1

- 학생의 등수

- 5 10 4

- 학생의 등수

- 5 10 4

- 학생의 등수

- 5 10 4

굉장한 학생

36

<https://www.acmicpc.net/problem/2336>

- 학생의 등수

• 7 1 7

• 1 2 8

• 3 3 1

• 10 4 6

• 2 5 5

• 8 6 9

• 6 7 3

• 4 8 2

• 9 9 10

• 5 10 4

tree[1]	tree[2]	tree[3]	tree[4]	tree[5]	tree[6]	tree[7]	tree[8]	tree[9]	tree[10]
8	5	1	2	4	3	7	9	10	6

굉장한 학생

<https://www.acmicpc.net/problem/2336>

- C/C++: <https://gist.github.com/Baekjoon/8c5fb26d5a0b2246ea2f>
- Java: <https://gist.github.com/Baekjoon/21e0a2a141fa54470c8f0a780926c27c>

합 구하기

5 4 3 2 1 2 6 4
1 5 4 3 2 6 3
1 5 4 3 2 6 3
1 2 5 4 3 6 2
1 2 3 5 4 6 1
1 2 3 4 5 6 1
1 3

터보소트

<https://www.acmicpc.net/problem/3006>

- 첫 번째 단계에서 숫자 1의 위치를 찾는다. 그 다음 바로 앞의 숫자와 위치를 바꾸어가면서, 1이 제일 앞에 오게 바꾼다.
- 두 번째 단계에서는 숫자 N의 위치를 찾는다. 그 다음 바로 뒤의 숫자와 위치를 바꾸어가면서, N이 제일 마지막에 오게 바꾼다.
- 세 번째 단계은 숫자 2의 위치를 찾은 후에, 바로 앞의 숫자와 위치를 바꾸어가면서, 두 번째 위치에 오게 바꾼다.
- 네 번째 단계은 숫자 N-1의 위치를 찾은 다음에, 바로 뒤의 숫자와 위치를 바꾸면서, 뒤에서 2번째 위치에 오게 바꾼다.
- 다섯 번째 단계도 위와 같은 식으로 하면 되고 이를 N번 반복하는 것이다.

터보소트

<https://www.acmicpc.net/problem/3006>

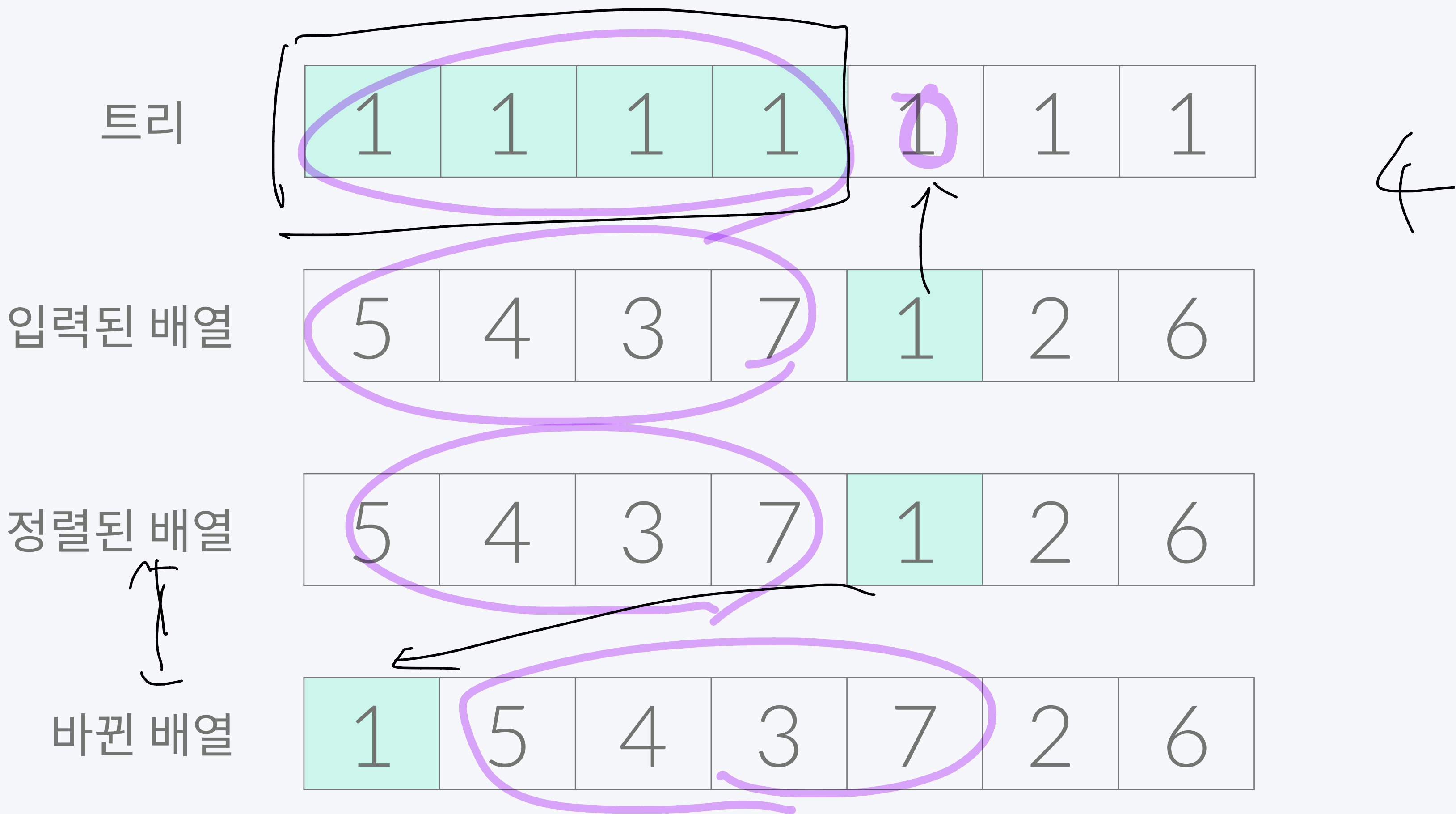
$tree[i] = \text{현재까지의 } AC[i] \text{가}$

40

이중(정렬되는지) \bigcirc

아니면 /

- 1단계

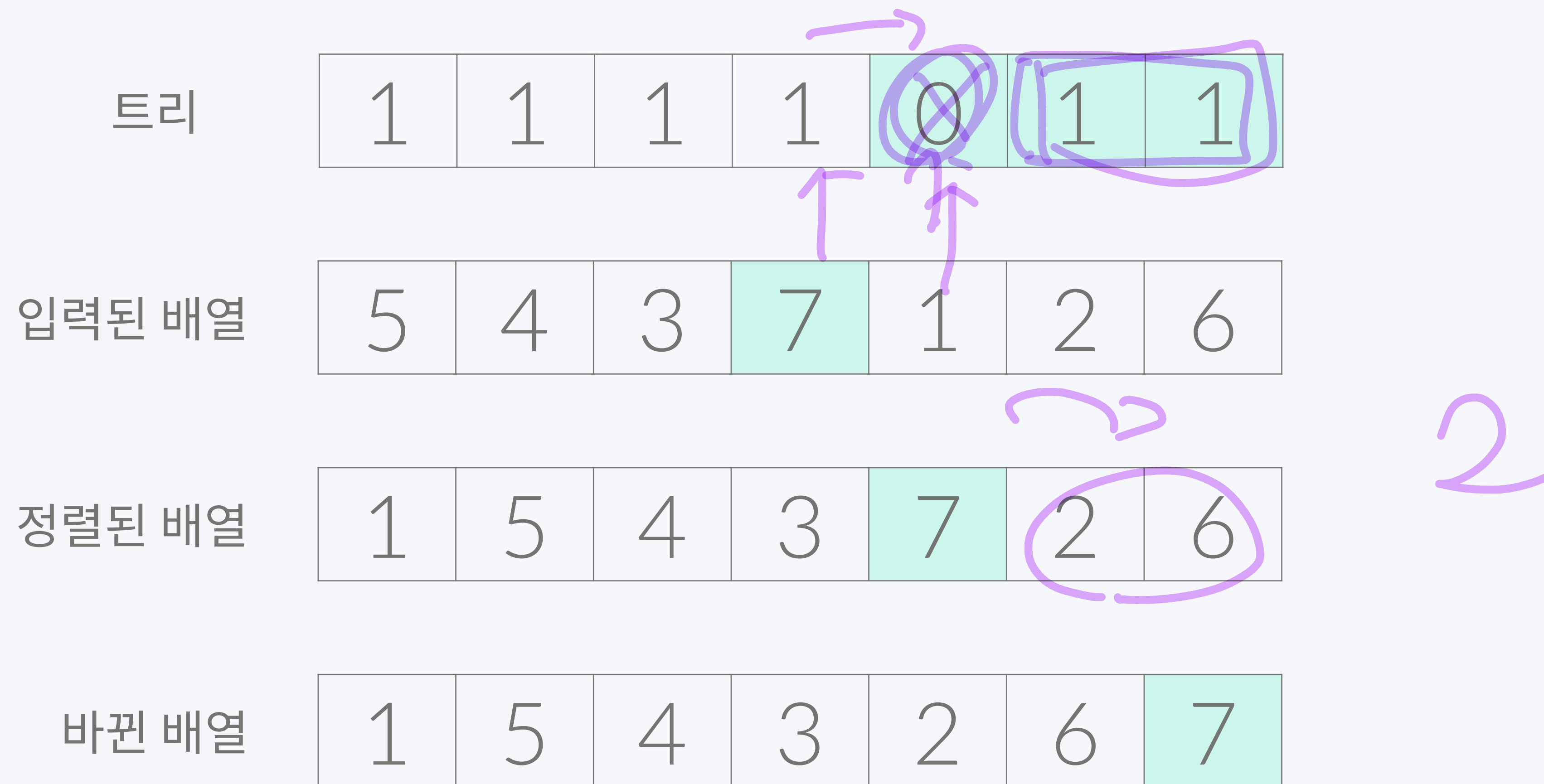


터보소트

41

<https://www.acmicpc.net/problem/3006>

- 2단계

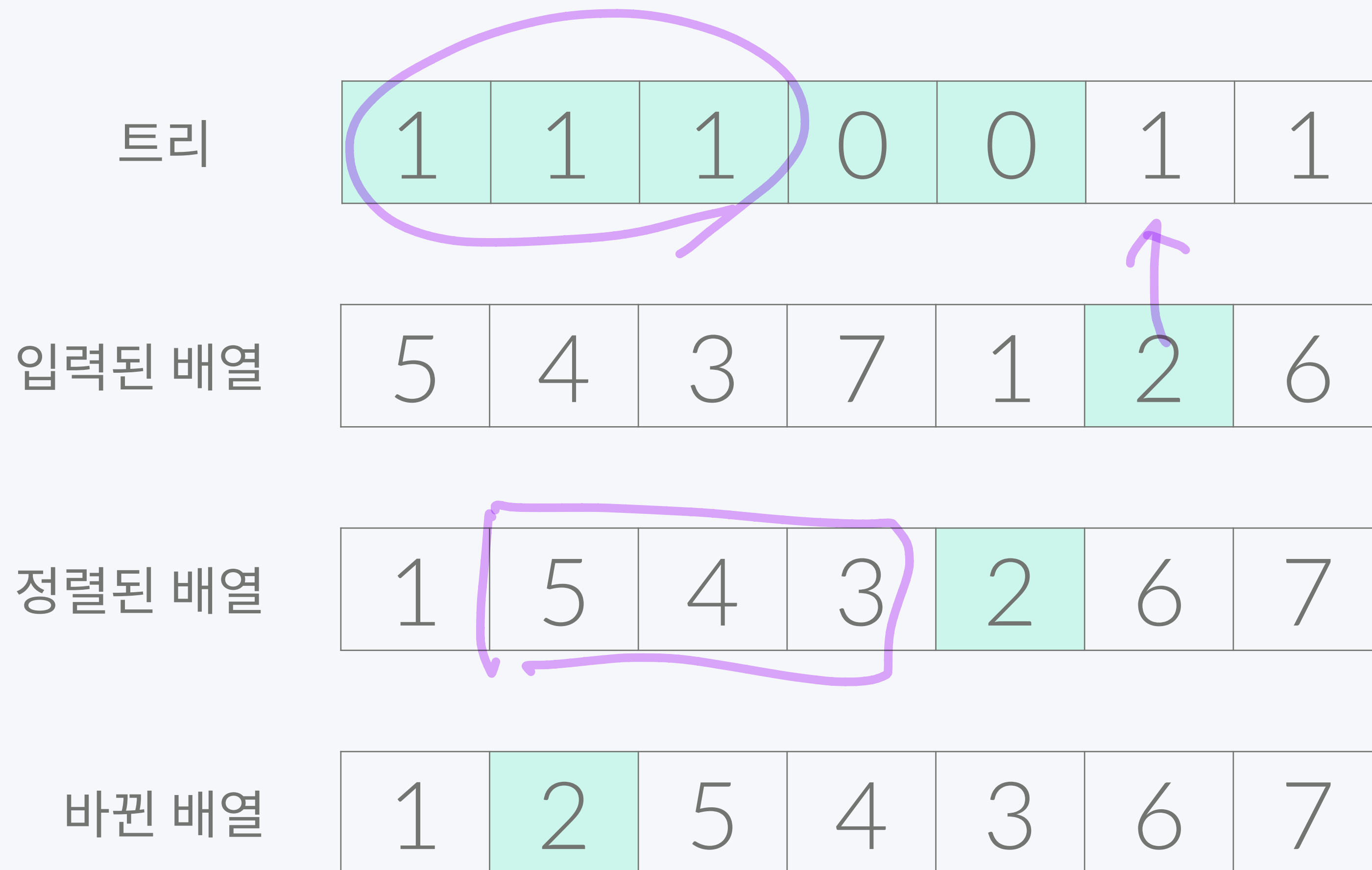


터보소트

42

<https://www.acmicpc.net/problem/3006>

- 3단계



터보소트

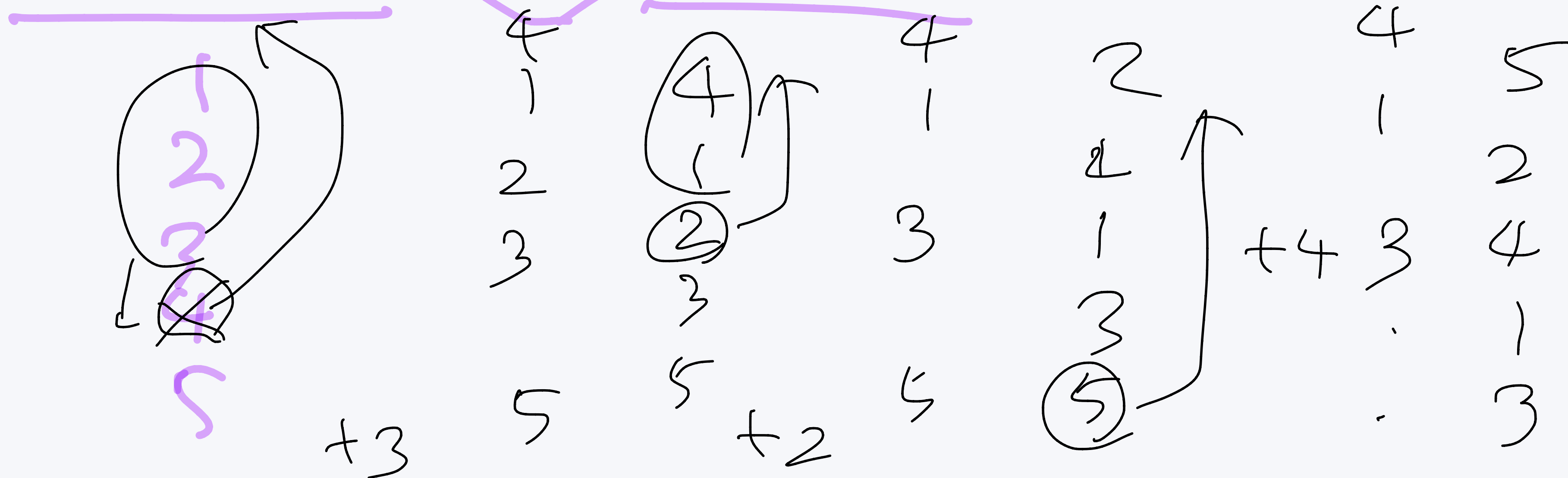
<https://www.acmicpc.net/problem/3006>

- C/C++: <https://gist.github.com/Baekjoon/f9cb420d3ad21ae1bbdb302f58c47bf2>

영화 수집

<https://www.acmicpc.net/problem/3653>

- 보고 싶은 영화가 있을 때는, DVD의 위치를 찾은 다음 쌓아놓은 컬렉션이 무너지지 않게 조심스럽게 DVD를 뺀다
- 가장 처음에 영화가 쌓여진 순서는 1부터 증가하는 순서이다
- 가장 위에 있는 영화의 번호는 1이다
- 상근이가 영화를 한 편 볼 때마다 그 DVD의 위에 몇 개의 DVD가 있었는지를 구해야 한다



영화 수집

45

<https://www.acmicpc.net/problem/3653>

- 영화의 개수: 5개, 보려고 하는 영화의 개수: 3개

1	2	3	4	5	6	7	8
			1	2	3	4	5

영화 수집

<https://www.acmicpc.net/problem/3653>

- 4를 본다.

1	2	3	4	5	6	7	8
			1	2	3	4	5
0	0	0	1	1	1	1	1

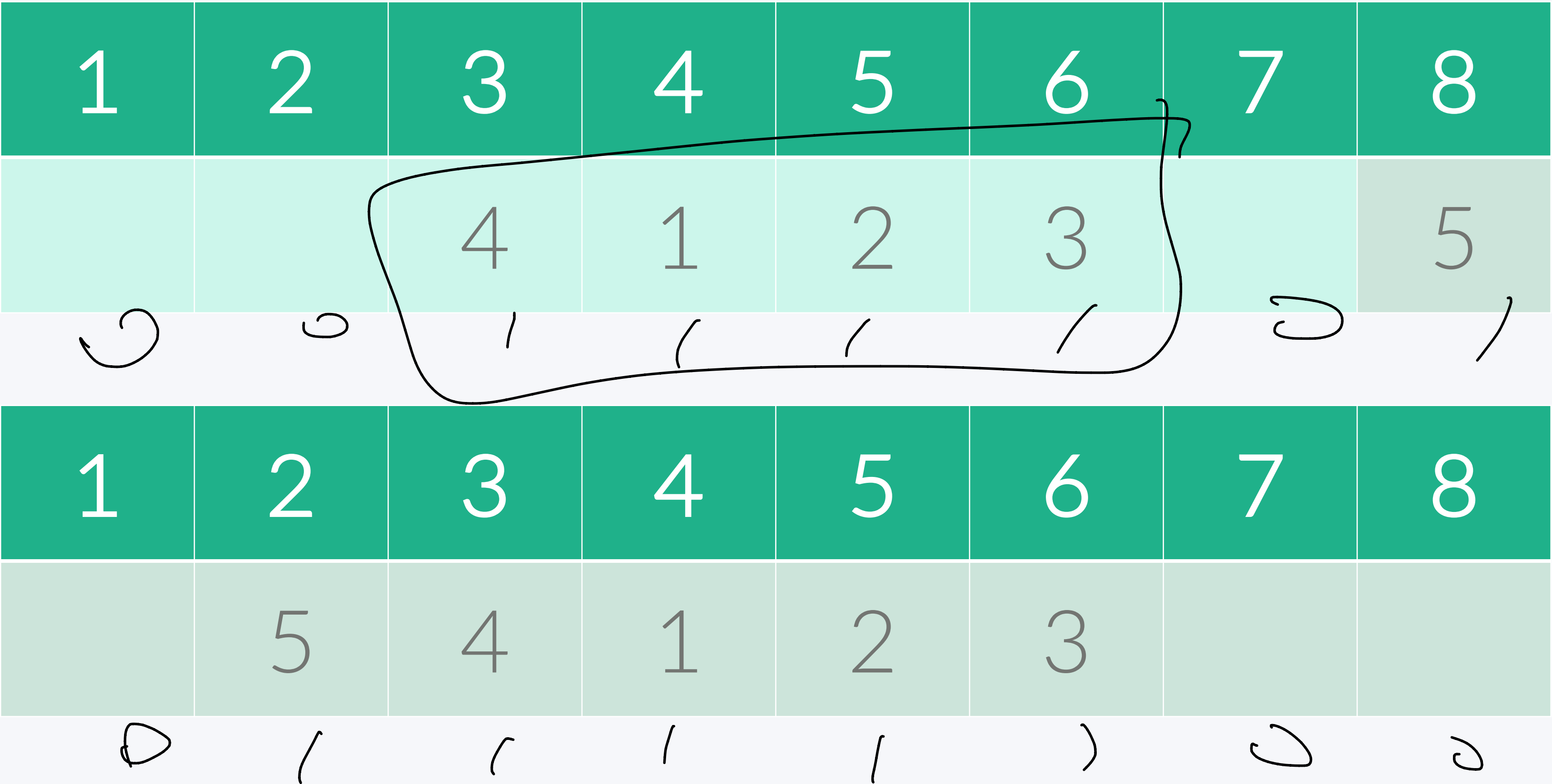
1	2	3	4	5	6	7	8
		4	1	2	3		5
0	0	1	1	1	1	0	1

영화 수집

<https://www.acmicpc.net/problem/3653>

- 5를 본다.

+4



영화 수집

<https://www.acmicpc.net/problem/3653>

- 2를 본다.

1	2	3	4	5	6	7	8
	5	4	1	2	3		
1	2	3	4	5	6	7	8
2	5	4	1		3		

Handwritten annotations below the table:

- Below the first row: 0, 1, 1, 1, 1, 5, 0, 0
- Below the second row: 1, 1, 1, 1, 0, 1, 0, 0

A black line connects the first row's first four columns to the second row's first four columns, and another black line connects the second row's first four columns to the third row's first four columns.

영화 수집

<https://www.acmicpc.net/problem/3653>

- 총 $N+M$ 개라고 생각을 하고
- 가장 처음에는 $M+1 \sim M+N$ 에 영화를 놓는다
- 그 다음, 한 칸씩 앞으로 옮기는 방식

영화 수집

50

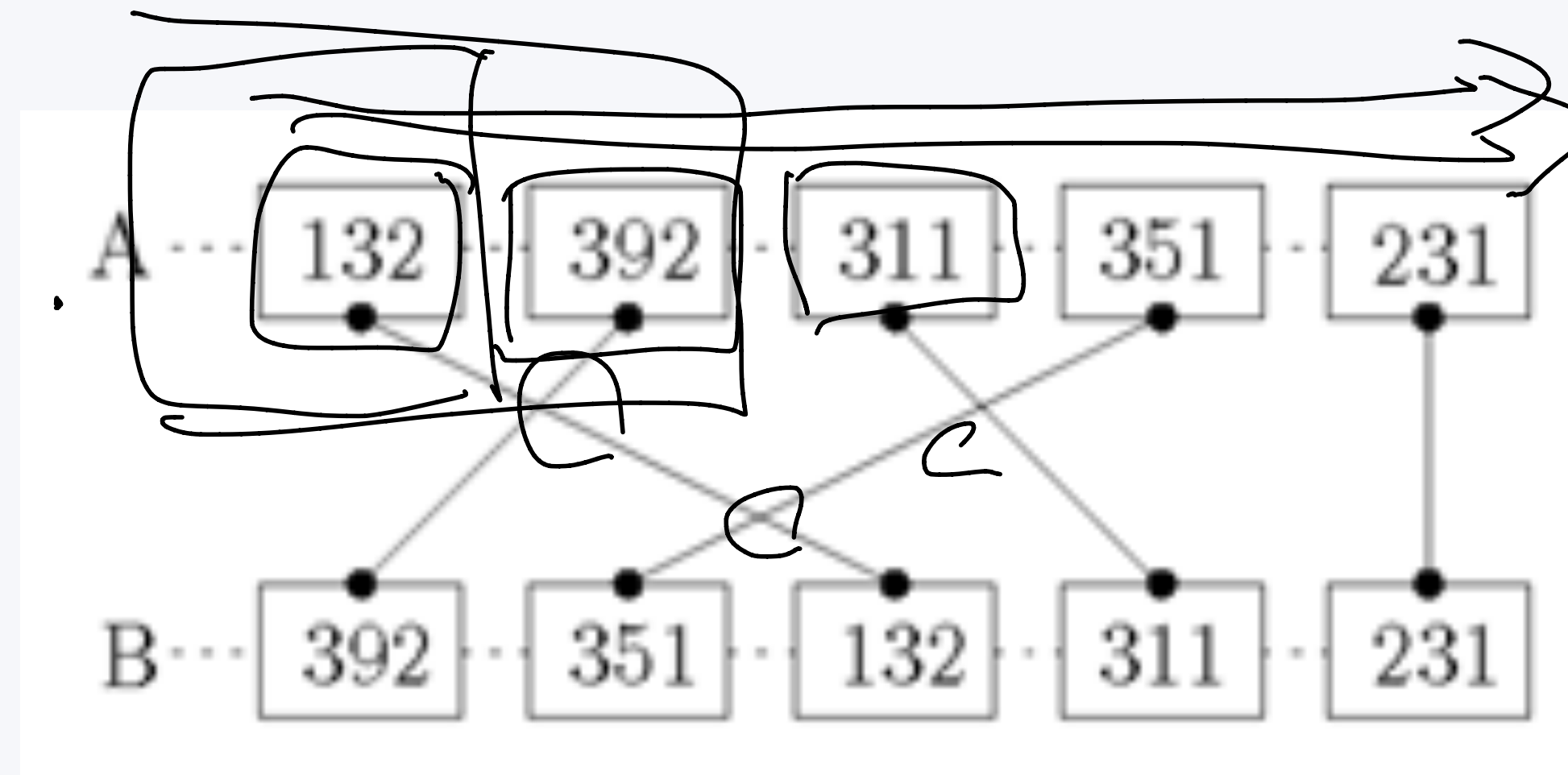
<https://www.acmicpc.net/problem/3653>

- C/C++: <https://gist.github.com/Baekjoon/815fbfada8745f669c6a>

공장

<https://www.acmicpc.net/problem/7578>

- 교차의 개수를 세는 문제
- 교차 하려면
- $A[i] < A[j]$ 이면서, $B[i] > B[j]$
- 의 개수를 세야 한다

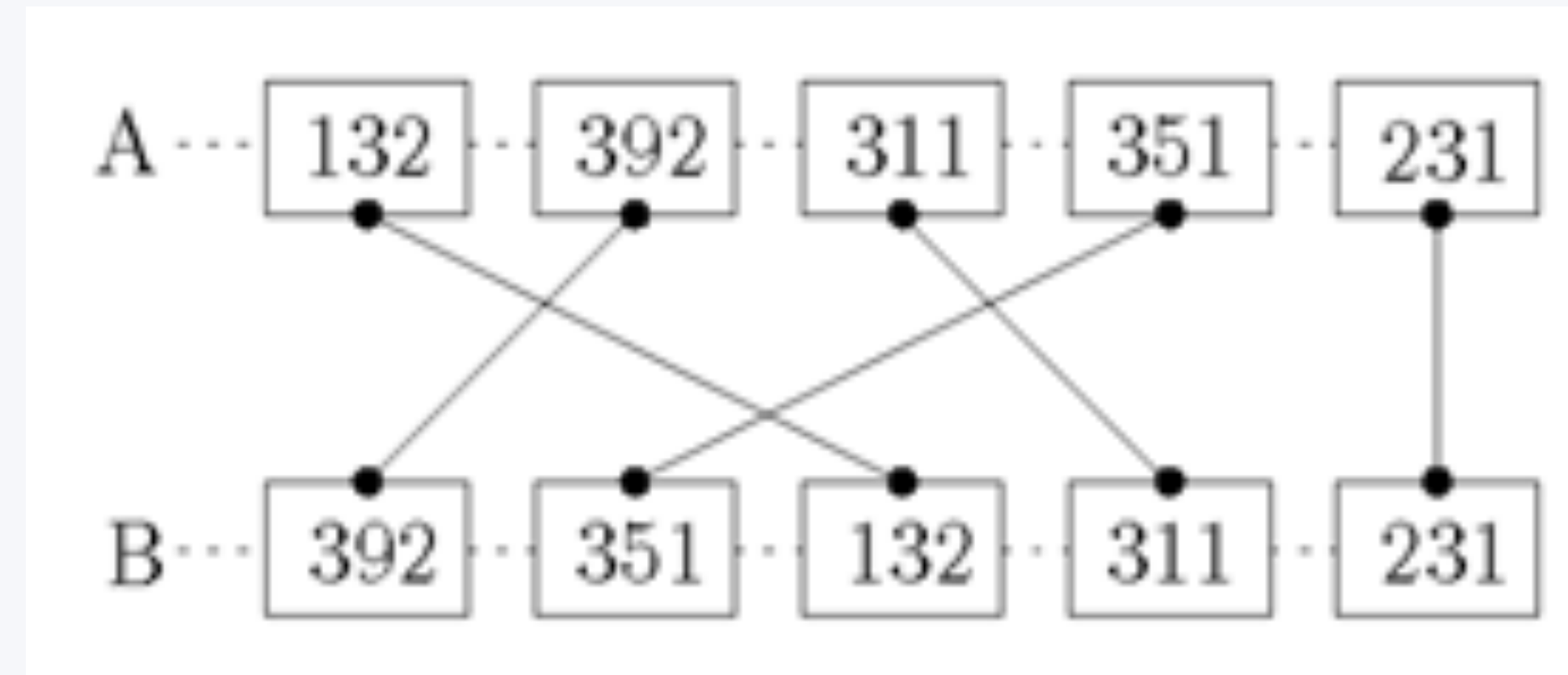


공장

52

<https://www.acmicpc.net/problem/7578>

- 교차의 개수를 세는 문제
- 교차 하려면
- $A[i] < A[j]$ 이면서, $B[i] > B[j]$
- 의 개수를 세야 한다
- $A[i] < A[j]$: 트리에 넣는 순서로 해결 가능
- $B[i] > B[j]$: 트리에서 개수를 세는 방식으로 해결 가능

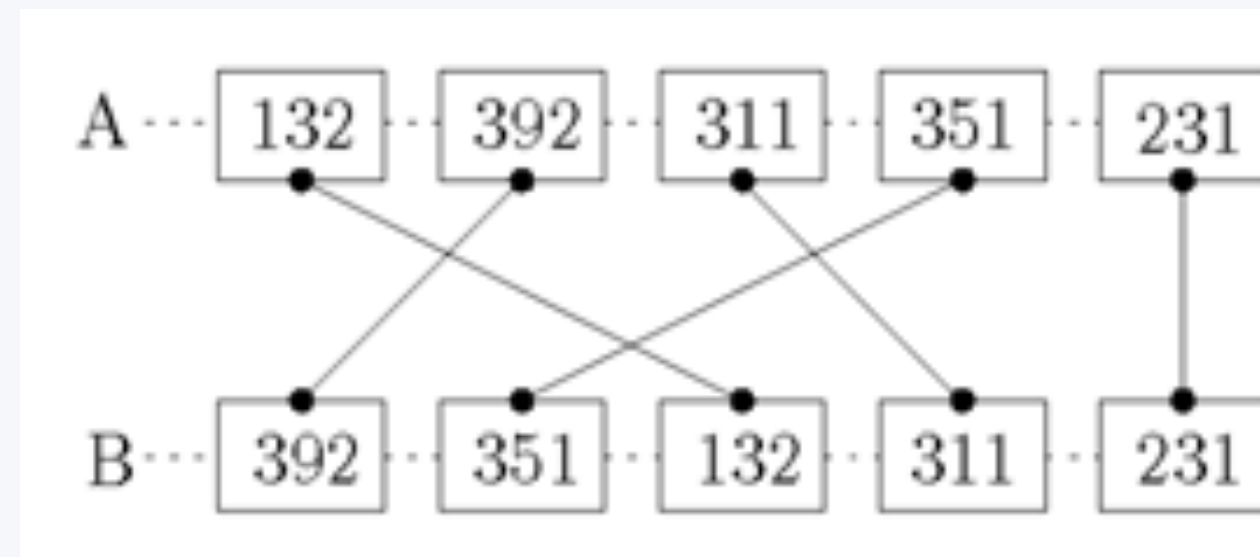


공장

53

<https://www.acmicpc.net/problem/7578>

- 교차의 개수를 세는 문제
- 교차 하려면
- $A[i] < A[j]$ 이면서, $B[i] > B[j]$
- 의 개수를 세야 한다
- $A[i] < A[j]$: 트리에 넣는 순서로 해결 가능
- $B[i] > B[j]$: 트리에서 개수를 세는 방식으로 해결 가능

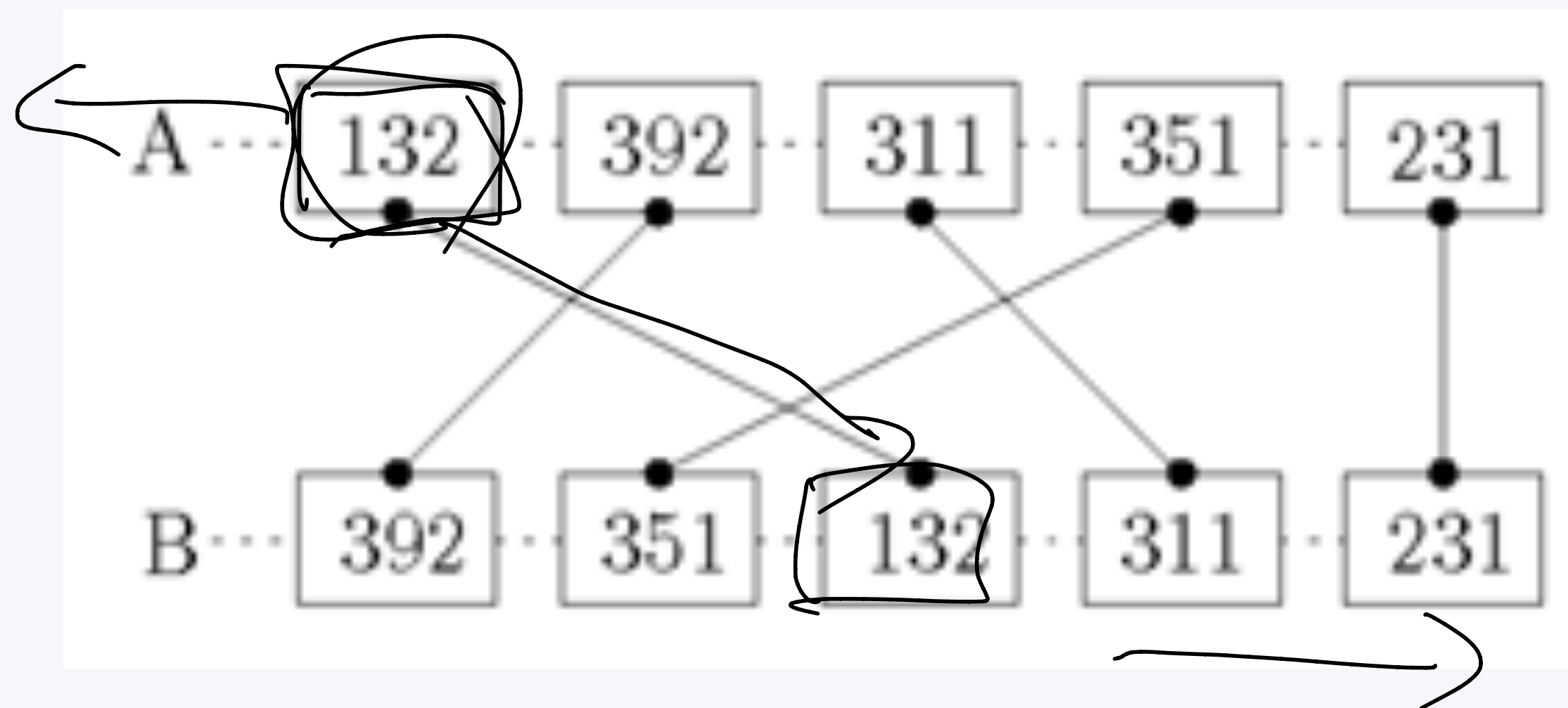


공장

54

<https://www.acmicpc.net/problem/7578>

- 132, 392, 311, 351, 231



i	1	2	3	4	5
B[i]	392	351	132	311	231
Tree[i]	0	0	0	0	0

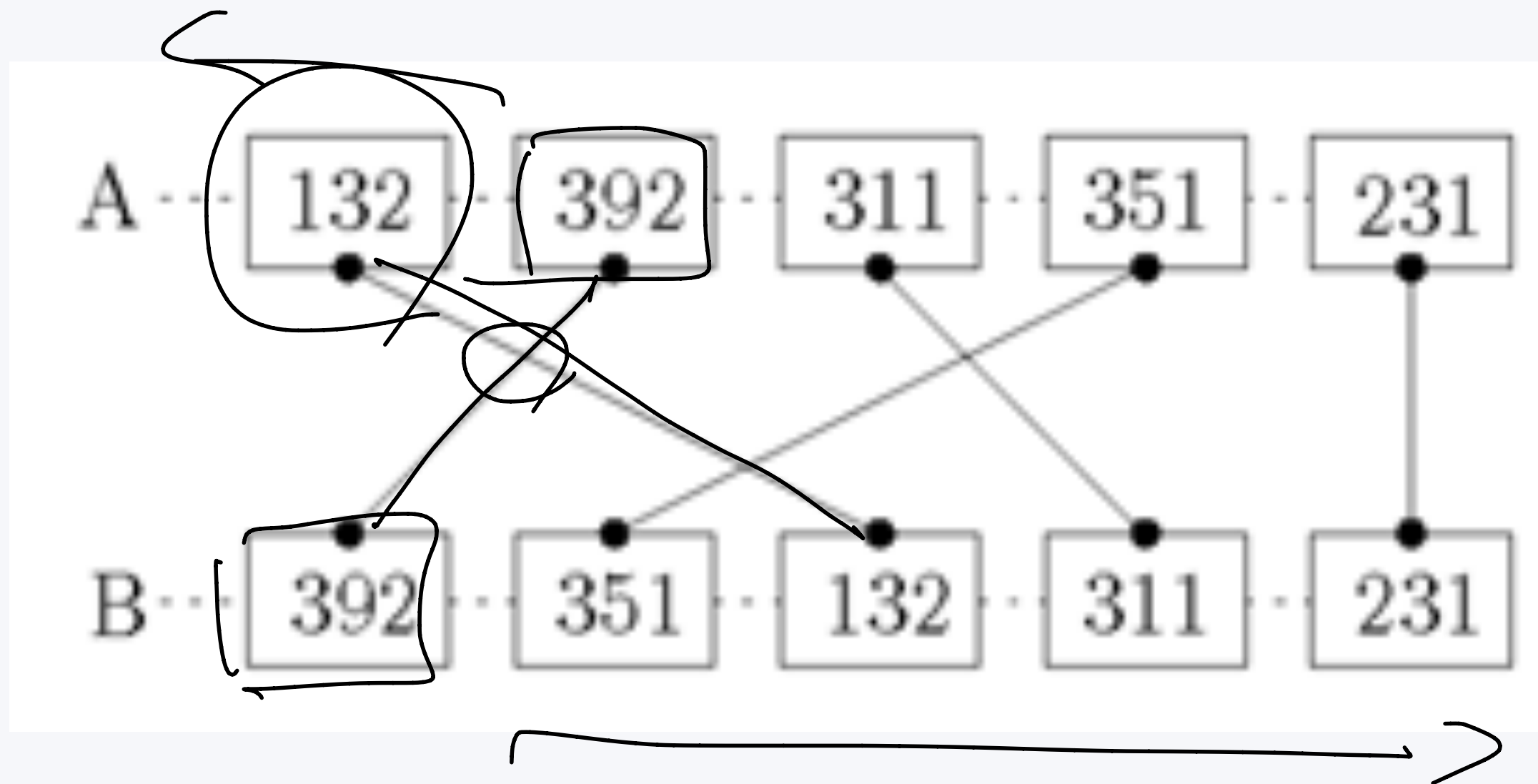
to

공장

55

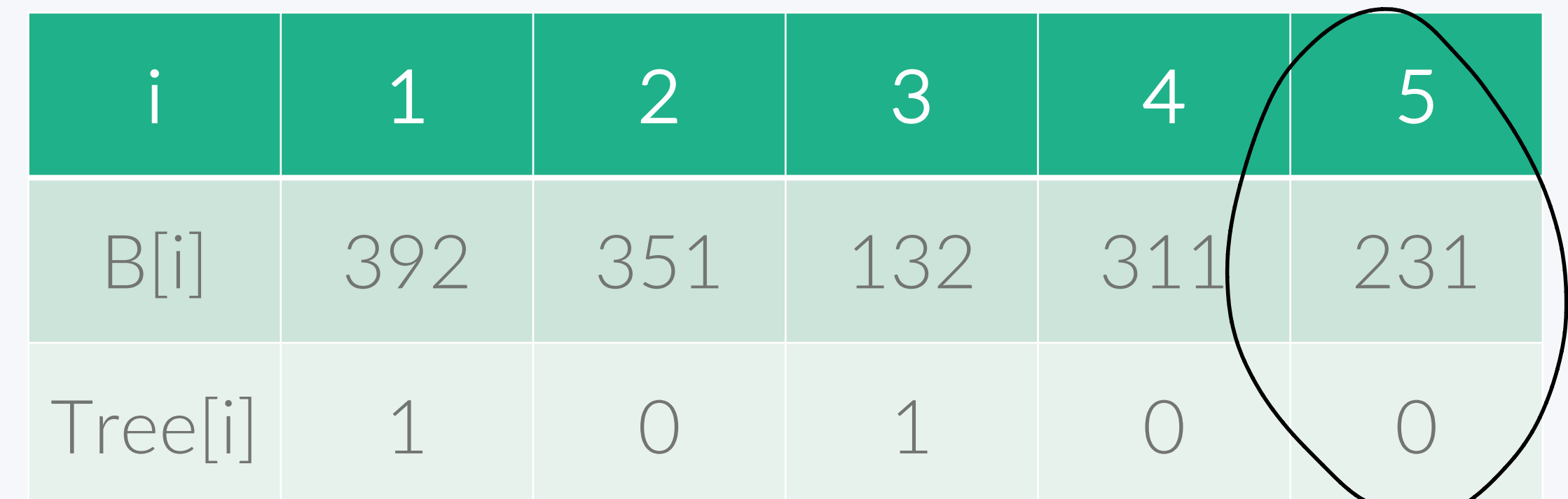
<https://www.acmicpc.net/problem/7578>

- 132, 392, 311, 351, 231
- 311~231까지 합: 0



i	1	2	3	4	5
B[i]	392	351	132	311	231
Tree[i]	0	0	1	0	0

- 132, **392**, 311, 351, 231
- 351~231까지 합: 1
- 교차: 1

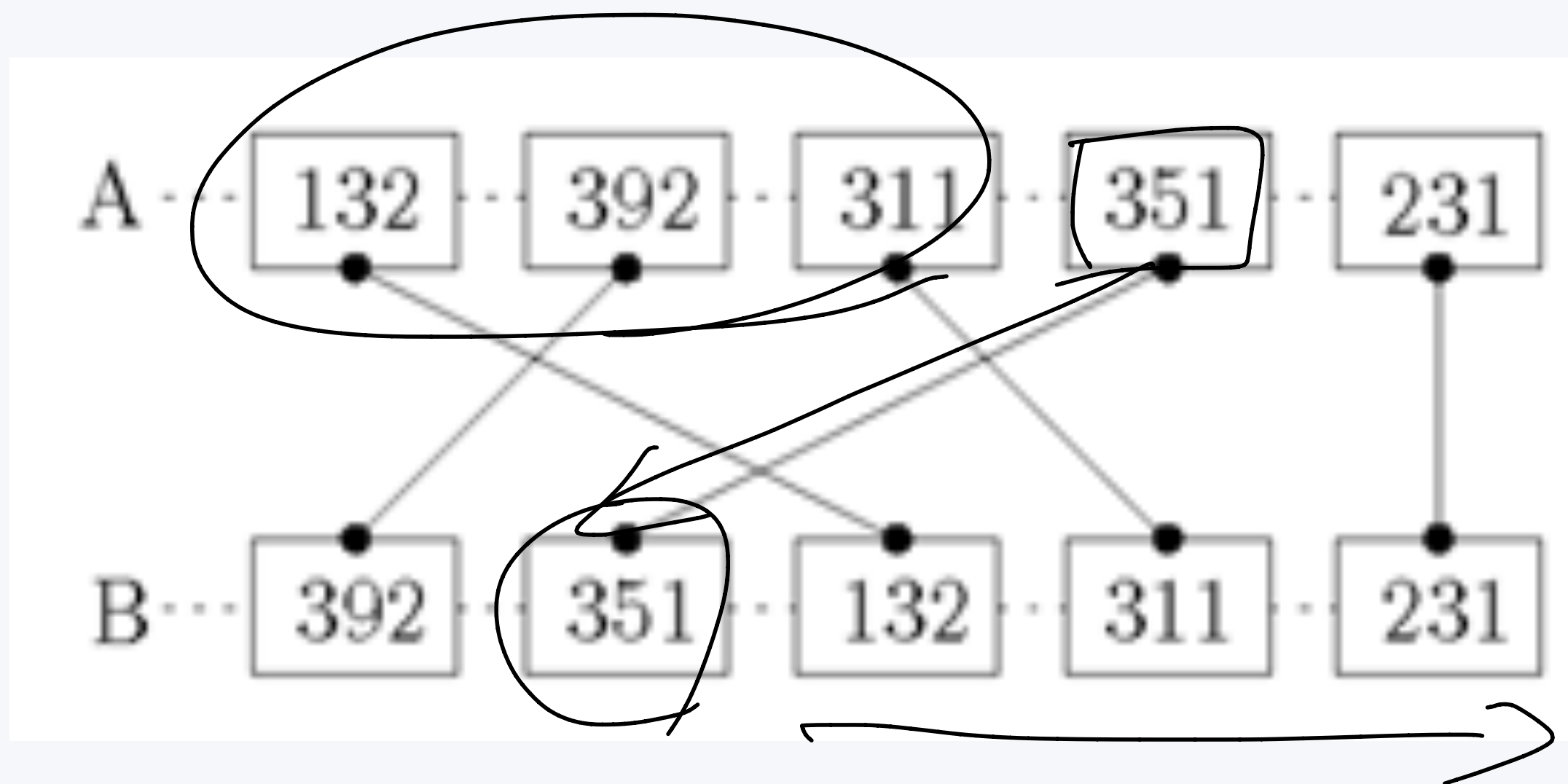


공장

57

<https://www.acmicpc.net/problem/7578>

- 132, 392, **311**, 351, 231
- 231~231까지 합: 0



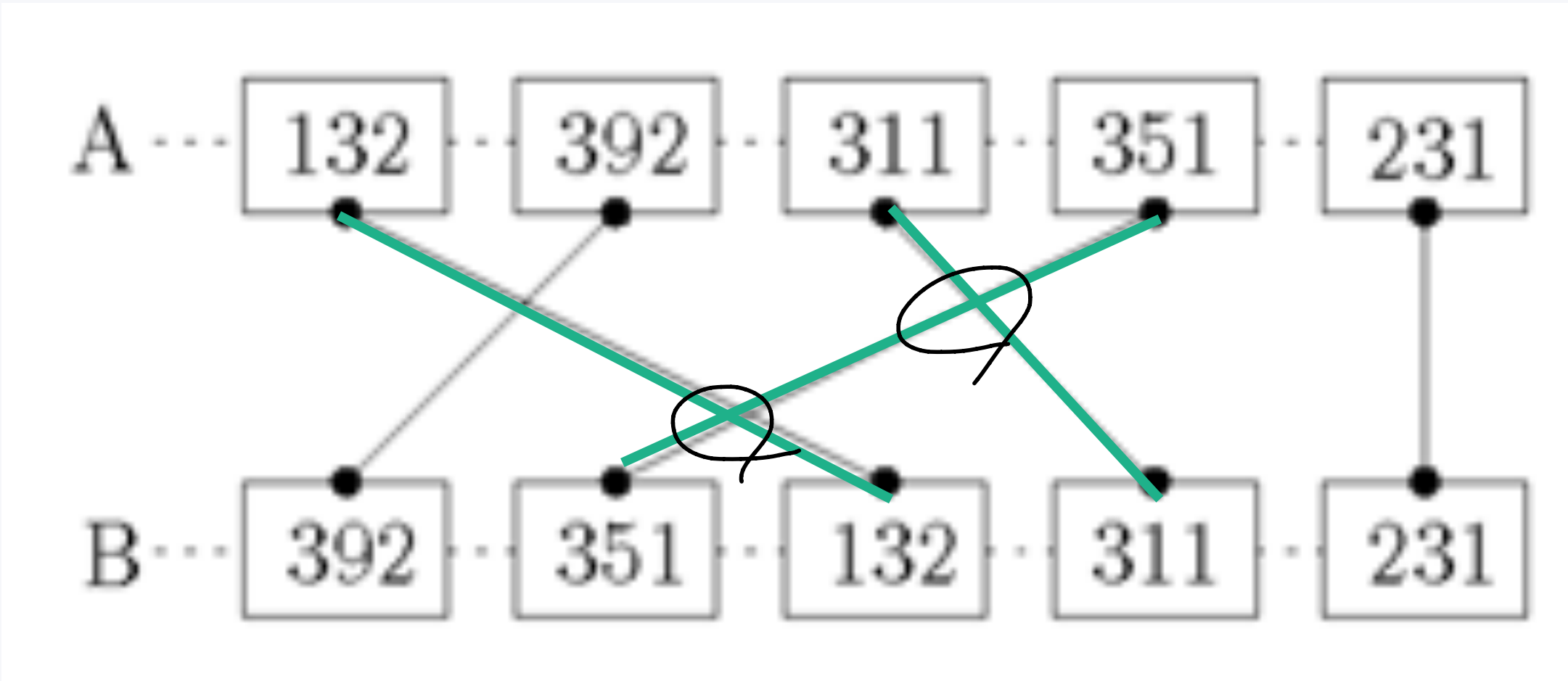
i	1	2	3	4	5
B[i]	392	351	132	311	231
Tree[i]	1	0	1	1	0

2

공장

<https://www.acmicpc.net/problem/7578>

- 132, 392, 311, **351**, 231
- 132~231까지 합: 2
- 교차: 2

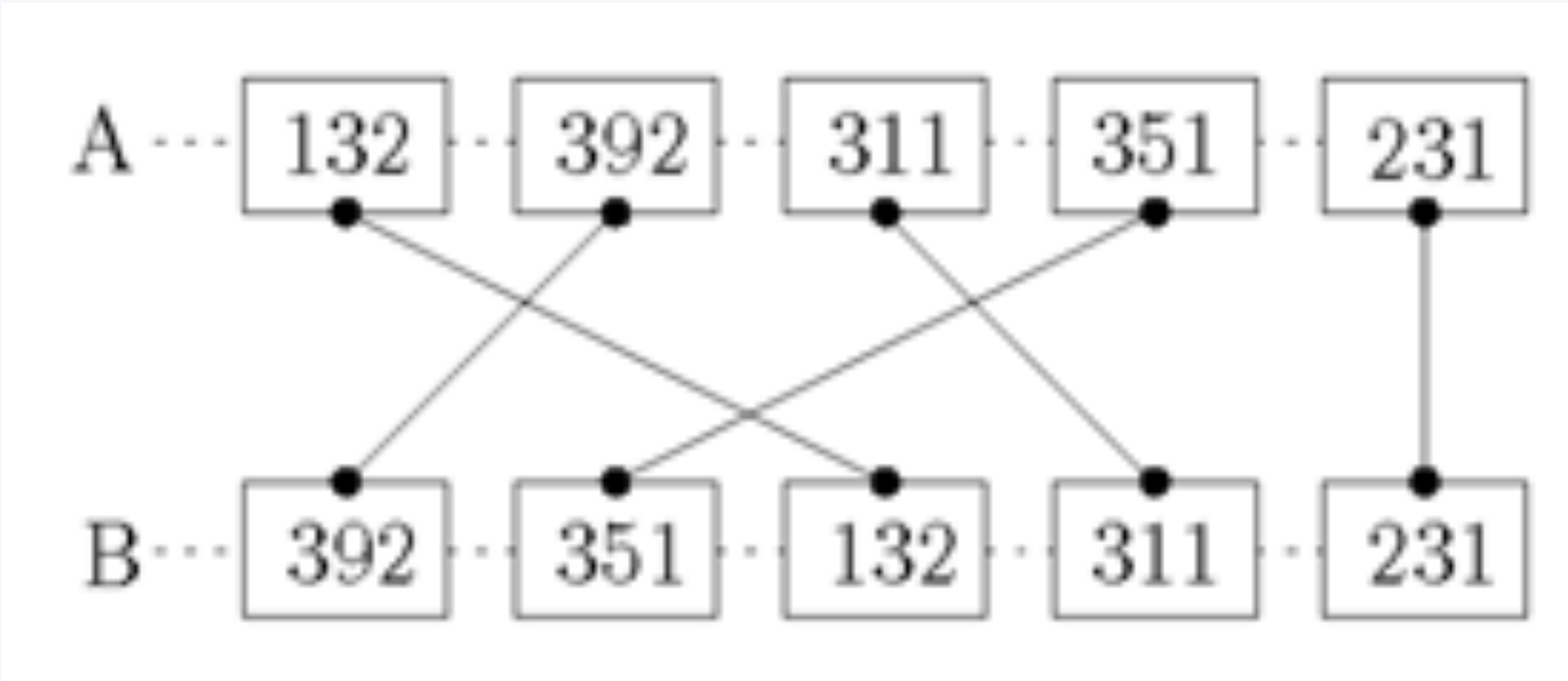


i	1	2	3	4	5
B[i]	392	351	132	311	231
Tree[i]	1	1	1	1	0

공장

<https://www.acmicpc.net/problem/7578>

- 132, 392, 311, 351, **231**
- 231~231까지 합: 0



i	1	2	3	4	5
B[i]	392	351	132	311	231
Tree[i]	1	1	1	1	1

공장

60

<https://www.acmicpc.net/problem/7578>

- C/C++: <https://gist.github.com/Baekjoon/26bdc5e354fc3ea5f1a4>

나무 심기

61

<https://www.acmicpc.net/problem/1280>

- 1번부터 N번까지 번호가 매겨져 있는 N개의 나무가 있다

- i번 나무는 좌표 $X[i]$ 에 심는다 $0 \leq X[i] \leq 10^9$

- 1번 나무를 심는 비용은 없다

- 각각의 나무를 심는데 드는 비용은 현재 심어져있는 모든 나무 까지 거리의 합

- 만약 3번 나무를 심는다면, 1번 나무와의 거리 + 2번 나무와의 거리가 3번 나무를 심는데 드는 비용

- 2번 나무부터 N번 나무까지를 심는 비용의 곱을 구하는 문제

$$2 \leq N \leq 200,000$$

$$N \times O(N)$$

나무 개수

거리를 심는 비용

제한 : 20만

$$\lg N$$

나무 심기

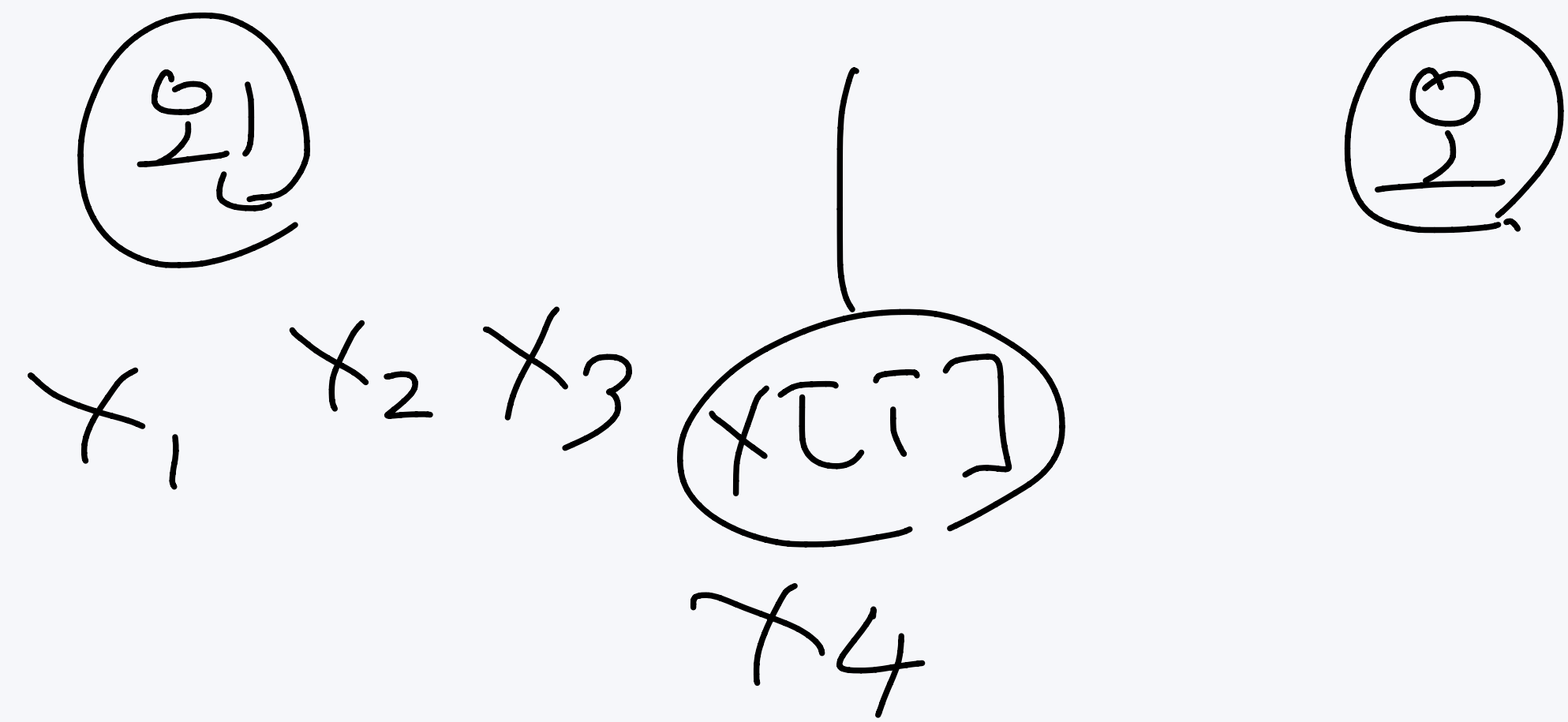
<https://www.acmicpc.net/problem/1280>

- BIT를 2개 써서 푸는 문제
- 좌표 제한 $L = 200,000$
- $\text{cnt}[i]$ = 좌표 i 에 심어져 있는 나무의 개수
- $\text{dist}[i] = \text{cnt}[i] * i$ (i 에 심어져 있는 좌표의 합)

나무 심기

<https://www.acmicpc.net/problem/1280>

- 나무를 tree[i]에 심는다 $x[i]$
- $CL = \text{sum}(\text{cnt}, 1, x[i]-1)$ 개가 왼쪽에
- 좌표가 x_0 에 있는 나무와의 거리는 $x[i] - x_0$
- 거리의 합
- $CL * x[i] - \text{sum}(\text{all } x_0)$
- $= CL * x[i] - \text{sum}(\text{dist}, 1, x[i]-1)$



$$x_4 - x_1 + x_4 - x_2 + x_4 - x_3$$

$$3x_4 - (x_1 + x_2 + x_3)$$

31-201 201

dist

cnt

나무 심기

<https://www.acmicpc.net/problem/1280>

- 나무를 $tree[i]$ 에 심는다
- $CR = \text{sum}(\text{cnt}, x[i]+1, L)$ 개가 오른쪽에
- 거리의 합
- $\text{sum}(\text{dist}, x[i]+1, L) - CR * x[i]$

Handwritten diagram illustrating the calculation of the distance sum for a tree structure.

Diagram showing a vertical line with points x_1, x_2, x_3 on the right and x_4 below it.

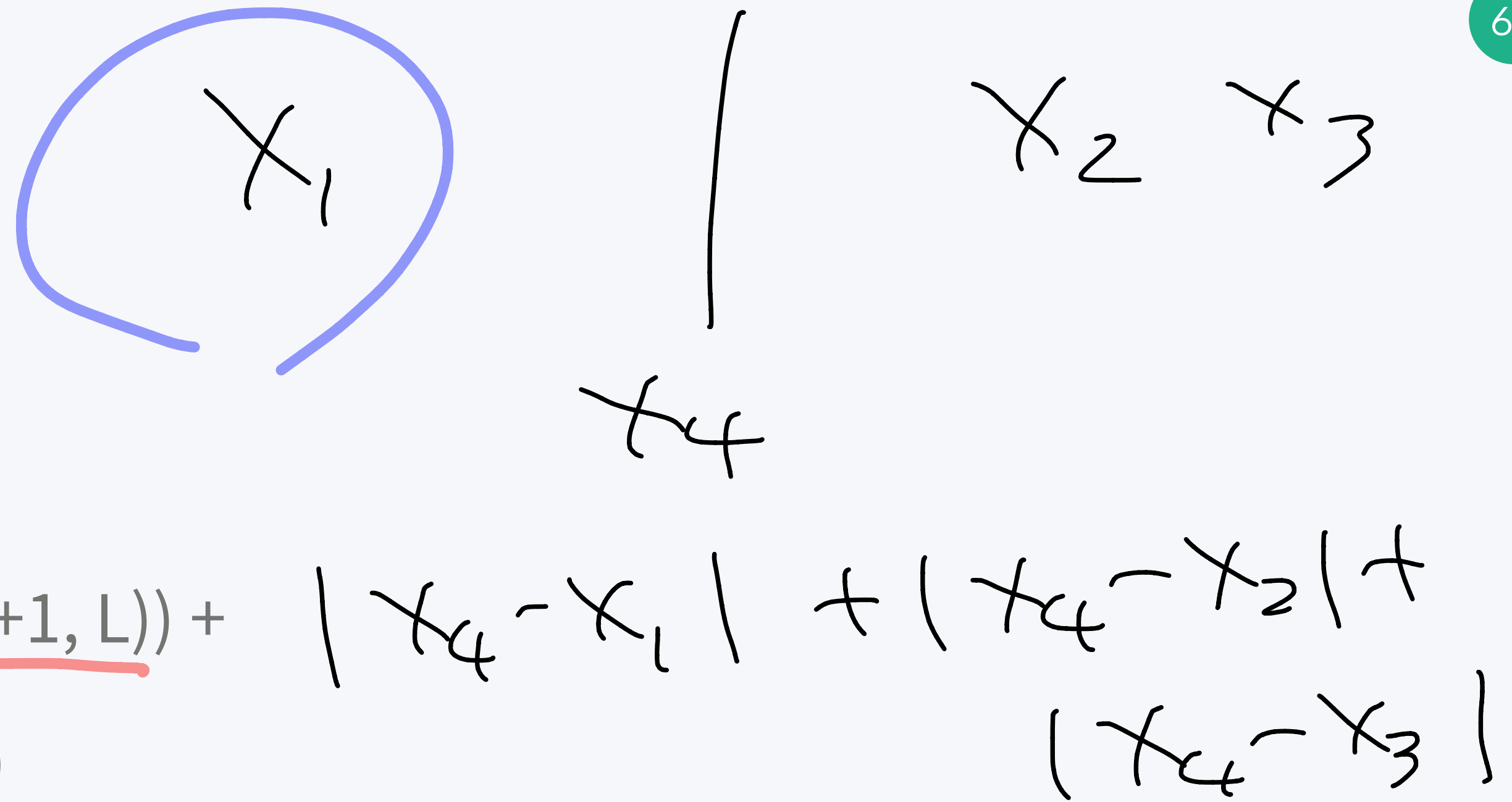
$$x_1 - x_4 + x_2 - x_4 + x_3 - x_4$$
$$\underbrace{(x_1 + x_2 + x_3)}_{\text{dist}} - \underbrace{3x_4}_{\text{cnt}}$$

나무 심기

<https://www.acmicpc.net/problem/1280>

65

- 나무를 $tree[i]$ 에 심는다
- 비용
- $x[i] * (\text{sum}(\text{cnt}, 1, x[i]-1) - \text{sum}(\text{cnt}, x[i]+1, L)) +$
- $\text{sum}(\text{dist}, x[i]+1, L) - \text{sum}(\text{dist}, 1, x[i]-1)$



$$\underbrace{(x_4 - x_1)}_{\text{blue underline}} + x_2 \underbrace{(-x_4)}_{\text{blue underline}} + x_3 - x_4$$

나무 심기

<https://www.acmicpc.net/problem/1280>

- C/C++: <https://gist.github.com/Baekjoon/37013b75cdbcb2b56d6d>

라운드 로빈 스케줄러

<https://www.acmicpc.net/problem/12016>

- 이번 문제에서 살펴볼 스케줄러는 라운드 로빈 스케줄러이다
- 총 작업은 N 개가 있으며, 0번부터 $N-1$ 번까지 번호가 매겨져 있다
- 스케줄러는 각 작업을 0번 작업부터 순서대로 한 번에 1초씩 실행시킨다
- 모든 작업을 순서대로 실행시킨 후에는 다시 0번 작업부터 실행을 시작한다
- 이 때, 완료된 작업이 있으면, 그 작업은 앞으로 실행시키지 않는다.
- 각 작업을 수행해야하는 시간이 주어졌을 때, 각 작업이 언제 완료되는지 구하는 문제

라운드 로빈 스케줄러

<https://www.acmicpc.net/problem/12016>

- 작업이 8, 1, 3, 6, 3, 8 인 경우
- 먼저 완료 되는 순서부터 살펴본다.

$0 \rightarrow 1$ 우선: 6
 $1 \xrightarrow{2} 3$ 우선: 5
 $6 + 10 = 16$
 $16 - 3 = 13$
 $16 - 1 = 15$ 우선: 3
 $3 \rightarrow 6$
 $16 + 3 + 3 = 22$ 22 - 1 = 21
 $21 - 1 = 20$

(NlgN)

$28 \quad 2 \quad 13 \quad 24 \quad 15 \quad 29$
 $0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$

소요 시간	8	1	3	6	3	8
완료 순서	5	1	2	4	3	6

1	1	2	3	4	5	6
2	2		8	9	10	11
3	12		13	14	15	16
4	17			18		19
5	20			21		22
6	23			24		25
7	26					27
8	28					29

라운드 로빈 스케줄러

<https://www.acmicpc.net/problem/12016>

- 작업이 8, 1, 3, 6, 3, 8 인 경우
- 먼저 완료 되는 순서부터 살펴본다.
- 총 지난 시간 = 0
- 남은 작업의 개수 = 6
- 모든 작업을 1초씩 실행시키는데 걸리는 시간 = 총 지난 시간 + 남은 작업의 개수 * 1 = 6
- 2번 작업이 완료되는 시간 = 6 - (3~6번 작업 중에 완료되지 않은 작업의 개수) = 2

소요 시간	8	1	3	6	3	8
완료 순서	5	1	2	4	3	6

라운드 로빈 스케줄러

<https://www.acmicpc.net/problem/12016>

- 작업이 8, 1, 3, 6, 3, 8 인 경우
- 먼저 완료 되는 순서부터 살펴본다.
- 총 지난 시간 = 6
- 남은 작업의 개수 = 5
- 모든 작업을 3초씩 실행시키는데 걸리는 시간 = 총 지난 시간 + 남은 작업의 개수 * (3-1) = 16
- 3번 작업이 완료되는 시간 = 16 - (4~6번 작업 중에 완료되지 않은 작업의 개수) = 13
- 5번 작업이 완료되는 시간 = 16 - (6~6번 작업 중에 완료되지 않은 작업의 개수) = 15

소요 시간	8	1	3	6	3	8
완료 순서	5	1	2	4	3	6

라운드 로빈 스케줄러

<https://www.acmicpc.net/problem/12016>

- 작업이 8, 1, 3, 6, 3, 8 인 경우
- 먼저 완료 되는 순서부터 살펴본다.
- 총 지난 시간 = 16
- 남은 작업의 개수 = 3
- 모든 작업을 6초씩 실행시키는데 걸리는 시간 = 총 지난 시간 + 남은 작업의 개수 * (6-3) = 25
- 4번 작업이 완료되는 시간 = 25 - (5~6번 작업 중에 완료되지 않은 작업의 개수) = 24

소요 시간	8	1	3	6	3	8
완료 순서	5	1	2	4	3	6

라운드 로빈 스케줄러

<https://www.acmicpc.net/problem/12016>

- 작업이 8, 1, 3, 6, 3, 8 인 경우
- 먼저 완료 되는 순서부터 살펴본다.
- 총 지난 시간 = 25
- 남은 작업의 개수 = 2
- 모든 작업을 8초씩 실행시키는데 걸리는 시간 = 총 지난 시간 + 남은 작업의 개수 * (8-6) = 29
- 1번 작업이 완료되는 시간 = 29 - (2~6번 작업 중에 완료되지 않은 작업의 개수) = 28
- 6번 작업이 완료되는 시간 = 29 - (7~6번 작업 중에 완료되지 않은 작업의 개수) = 29

소요 시간	8	1	3	6	3	8
완료 순서	5	1	2	4	3	6

라운드 로빈 스케줄러

73

<https://www.acmicpc.net/problem/12016>

- C/C++: <https://gist.github.com/Baekjoon/fa236de594ad526c597d93aca72d2015>

불만 정렬

<https://www.acmicpc.net/problem/5012>

- $i < j < k$ 이면서 $A[i] > A[j] > A[k]$ 인 쌍의 개수를 구하는 문제

$i < j$ $A[i] > A[j]$
 $A[i] < A[j]$, $B[i] > B[j]$

불만 정렬

75

<https://www.acmicpc.net/problem/5012>

- 7578번 공장 문제와 비슷하다
- 공장: $i < j$ 이면서 $A[i] > A[j]$ 쌍의 개수를 찾는 문제
- BIT를 이용해서 각각의 j 마다 $A[i] > A[j]$ 인 쌍의 개수를 구했다.

불만 정렬

<https://www.acmicpc.net/problem/5012>

- $i < j < k$ 이면서 $A[i] > A[j] > A[k]$ 쌍의 개수를 찾는 문제
- BIT를 이용해서 각각의 j 마다 $A[i] > A[j]$ 인 쌍의 개수를 구해서 tree2에 저장한다.
- tree2에서 각각의 k 마다 $A[i] > A[j] > A[k]$ 인 쌍의 개수를 구한다.

불만 정렬

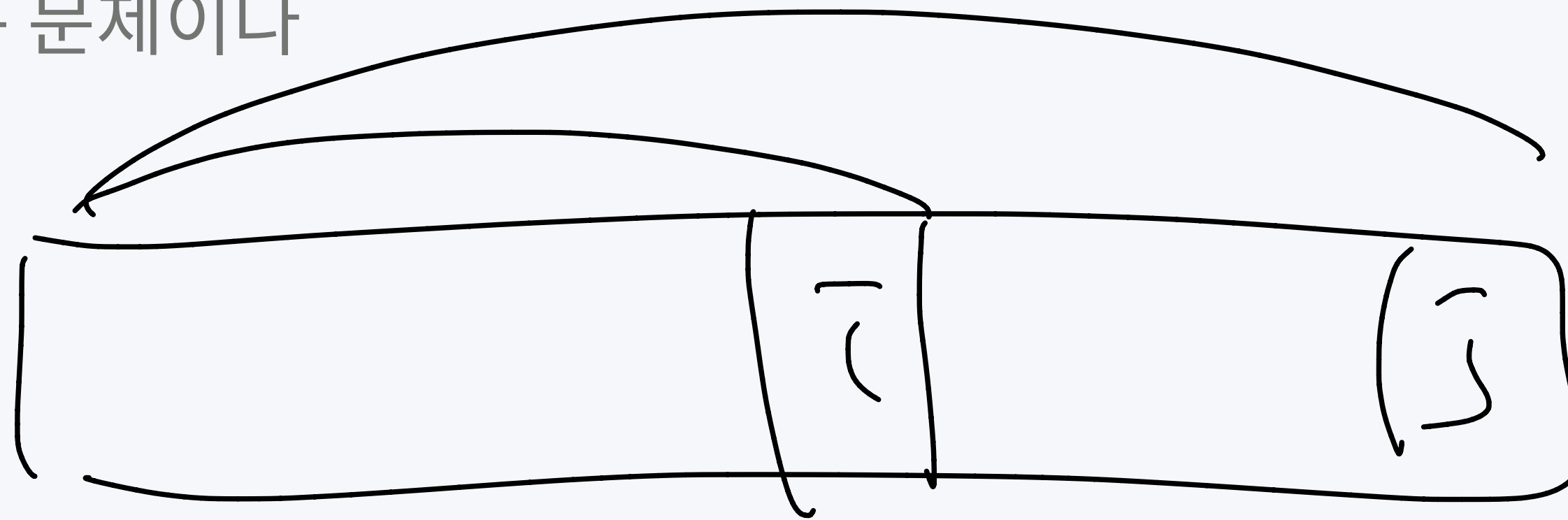
<https://www.acmicpc.net/problem/5012>

- C/C++: <https://gist.github.com/Baekjoon/0c0133afea9f45713fbb252d4bd1b72e>

음주 코딩

<https://www.acmicpc.net/problem/5676>

- 구간의 곱을 구하는 문제이다



o/o

음주 코딩

<https://www.acmicpc.net/problem/5676>

- 실제로 어떤 수가 들어있는지가 중요한 것이 아니고, 부호만 중요하기 때문에
- 1, 0, -1만 저장해서 문제를 푼다.
- 세그먼트 트리를 사용해야 한다.

$$\frac{0}{5} \frac{1}{7} : = \frac{0}{2} \frac{1}{7}$$

$$\frac{0}{6} \frac{1}{7} : = \frac{2}{7} \frac{1}{7}$$

$$0: \quad 0 \mid \frac{1}{\sigma^2} 2 \frac{1}{\sigma}$$

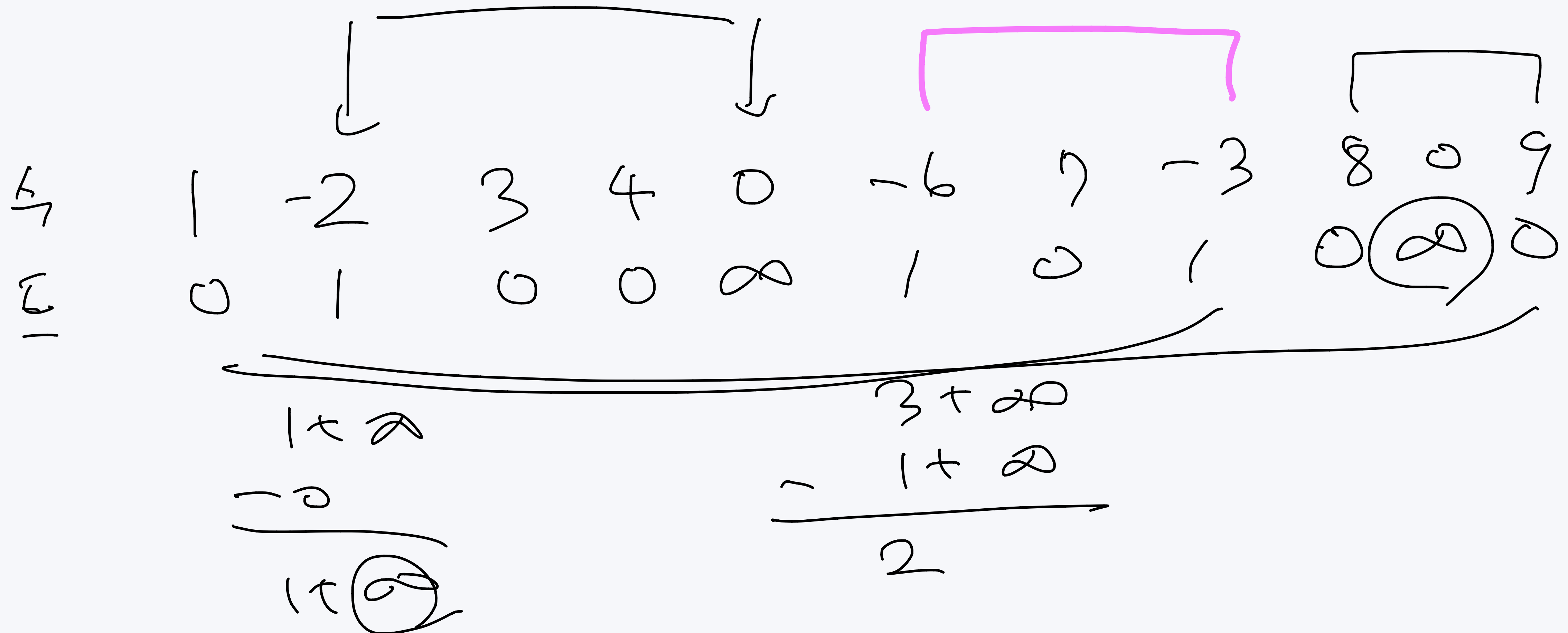
$\frac{\sigma}{\tau} \rightarrow 1$
 $\frac{\sigma}{\tau} \rightarrow \infty$
 $\sigma \rightarrow \infty$

음주 코딩

80

<https://www.acmicpc.net/problem/5676>

- C/C++: <https://gist.github.com/Baekjoon/d8b36c41ee06019118f7147bc8210084>



음주 코딩

<https://www.acmicpc.net/problem/5676>

- 바이너리 인덱스 트리를 이용해서도 문제를 풀 수 있다

음주 코딩

82

<https://www.acmicpc.net/problem/5676>

- C/C++: <https://gist.github.com/Baekjoon/d408371fd054876bf0c6bde01349488f>

K번째 찾기

17 | 20 번째 \in 21

$(\lg N)$

사탕상자

<https://www.acmicpc.net/problem/2243>

- 각각의 사탕은 그 맛의 좋고 나쁨이 1부터 1,000,000까지의 정수로 구분된다.
- 1이 가장 맛있는 사탕을 의미하며, 1,000,000은 가장 맛있는 사탕을 의미한다.
- 수정이는 동생이 말을 잘 들은 정도에 따라서, 사탕상자 안에 있는 사탕들 중 몇 번째로 맛있는 사탕을 꺼내주곤 한다.
- 예를 들어 말을 매우 잘 들었을 때에는 사탕상자에서 가장 맛있는 사탕을 꺼내주고, 말을 조금 잘 들었을 때에는 사탕상자에서 여섯 번째로 맛있는 사탕을 꺼내주는 식이다.
-

사탕상자

<https://www.acmicpc.net/problem/2243>

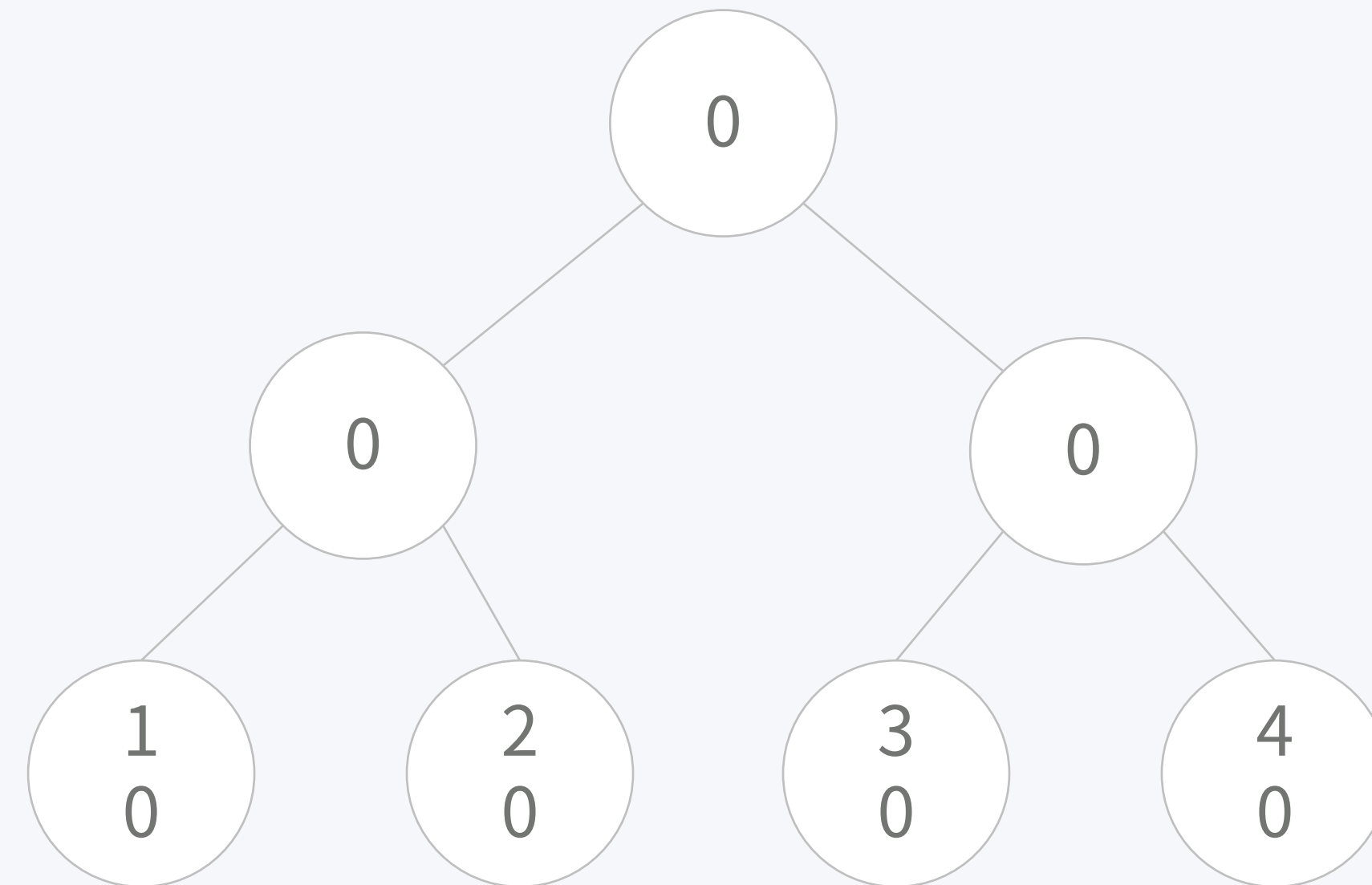
- 사탕을 꺼내는 경우
- 꺼내는 순위
- 사탕을 넣는 경우
- 맛과 개수

사탕상자

86

<https://www.acmicpc.net/problem/2243>

- 처음 상태

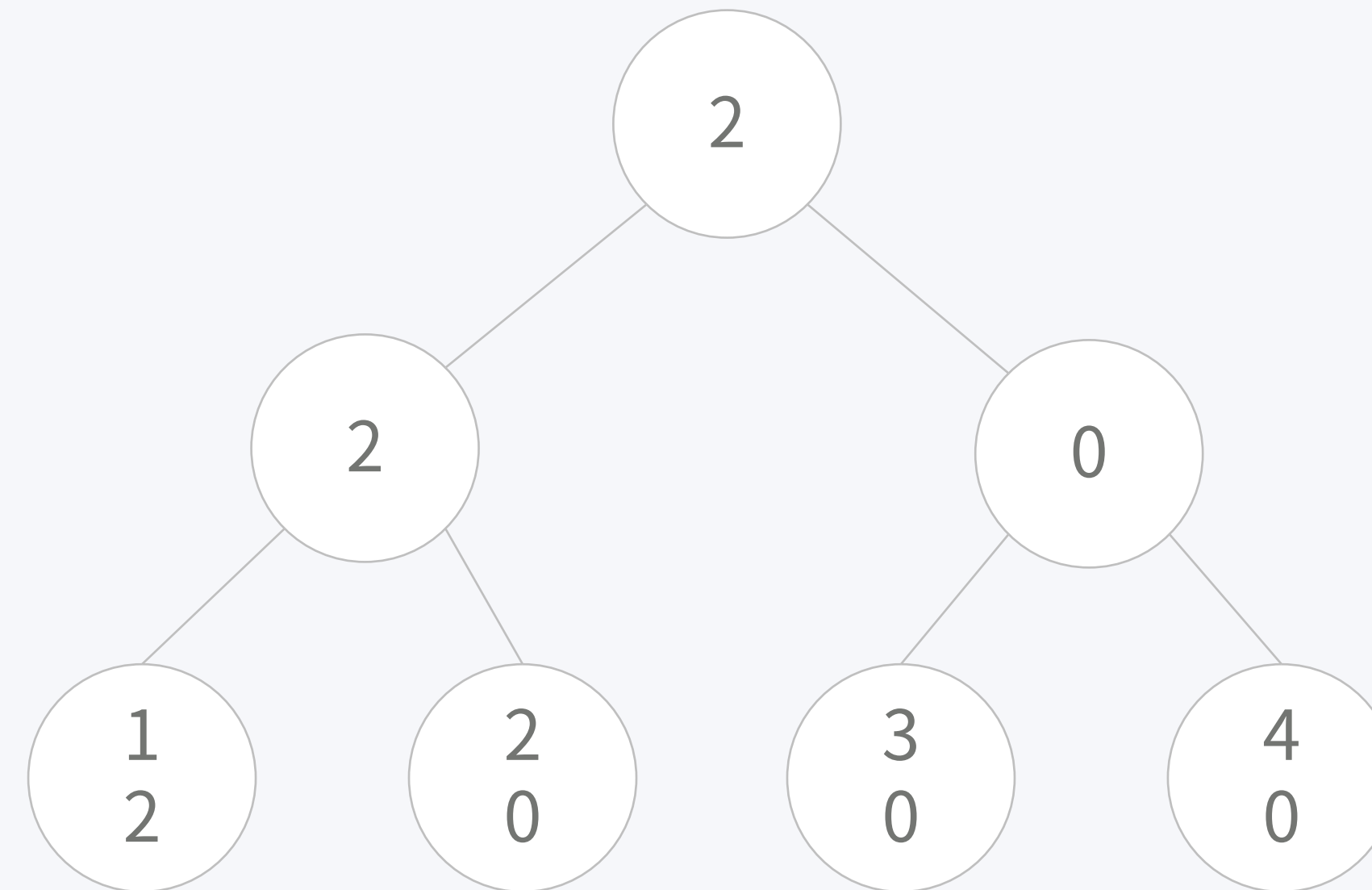


사탕상자

87

<https://www.acmicpc.net/problem/2243>

- 맛이 1인 사탕을 2개 넣는다



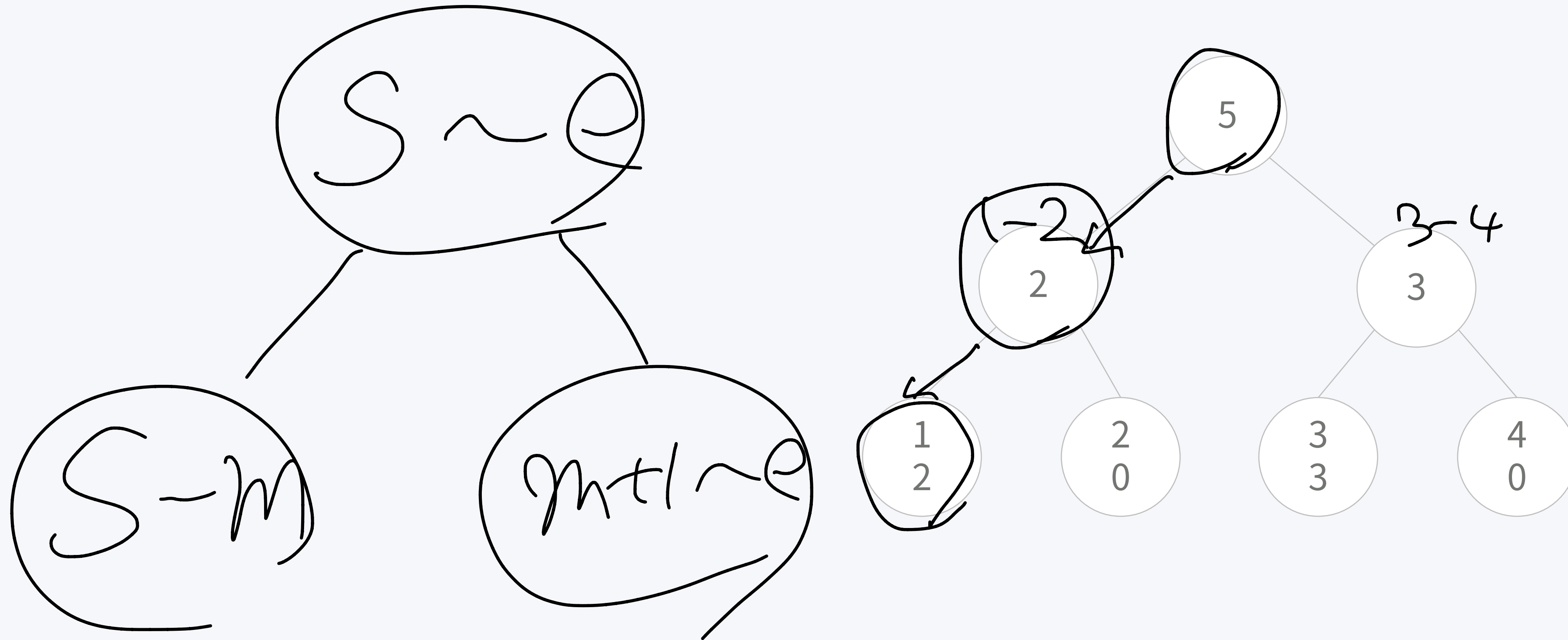
사탕상자

<https://www.acmicpc.net/problem/2243>

38

2

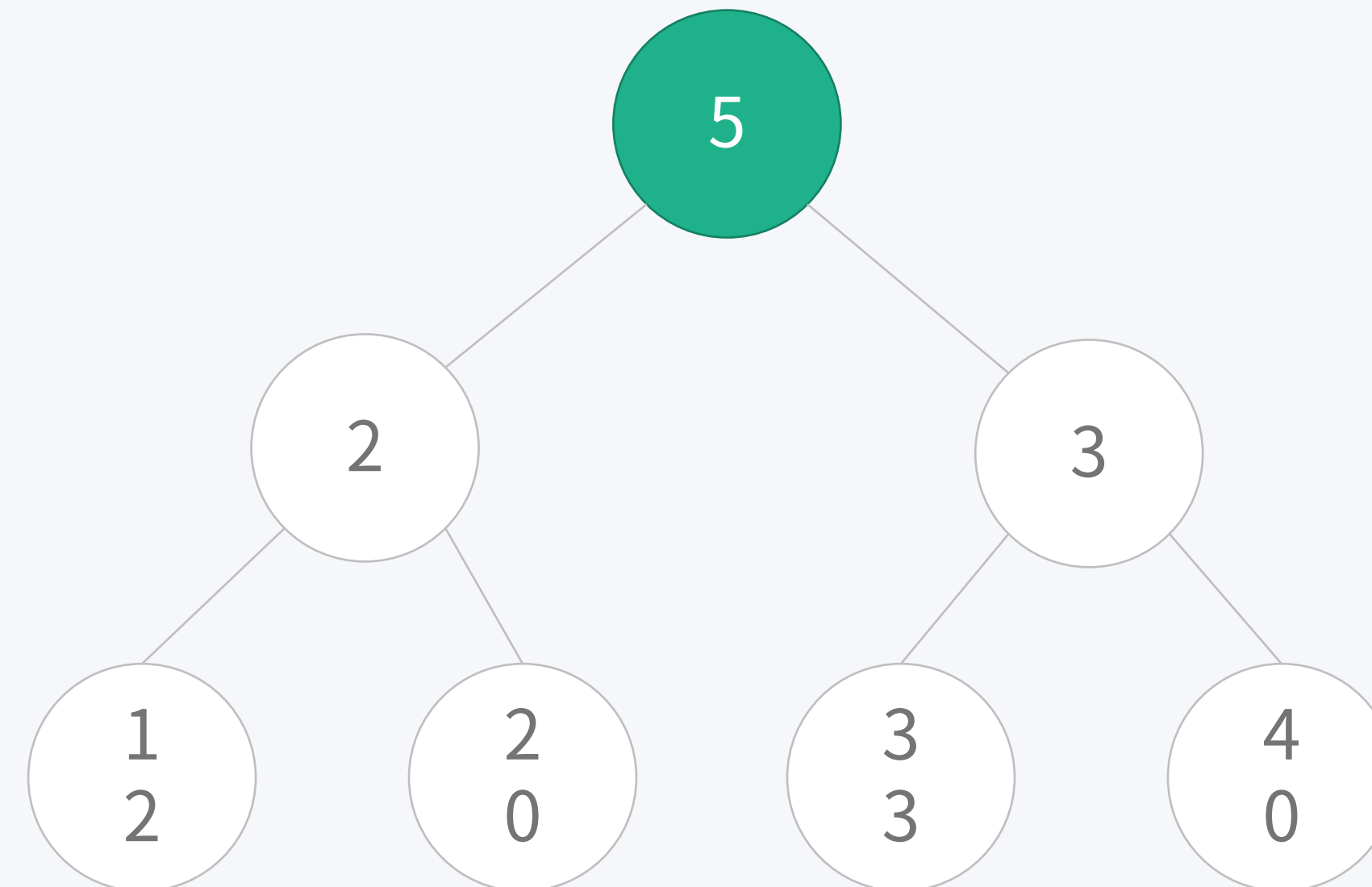
- 맛이 3인 사탕을 3개 넣는다



사탕상자

<https://www.acmicpc.net/problem/2243>

- 2번째 맛있는 사탕을 뺀다
- 왼쪽에 있다

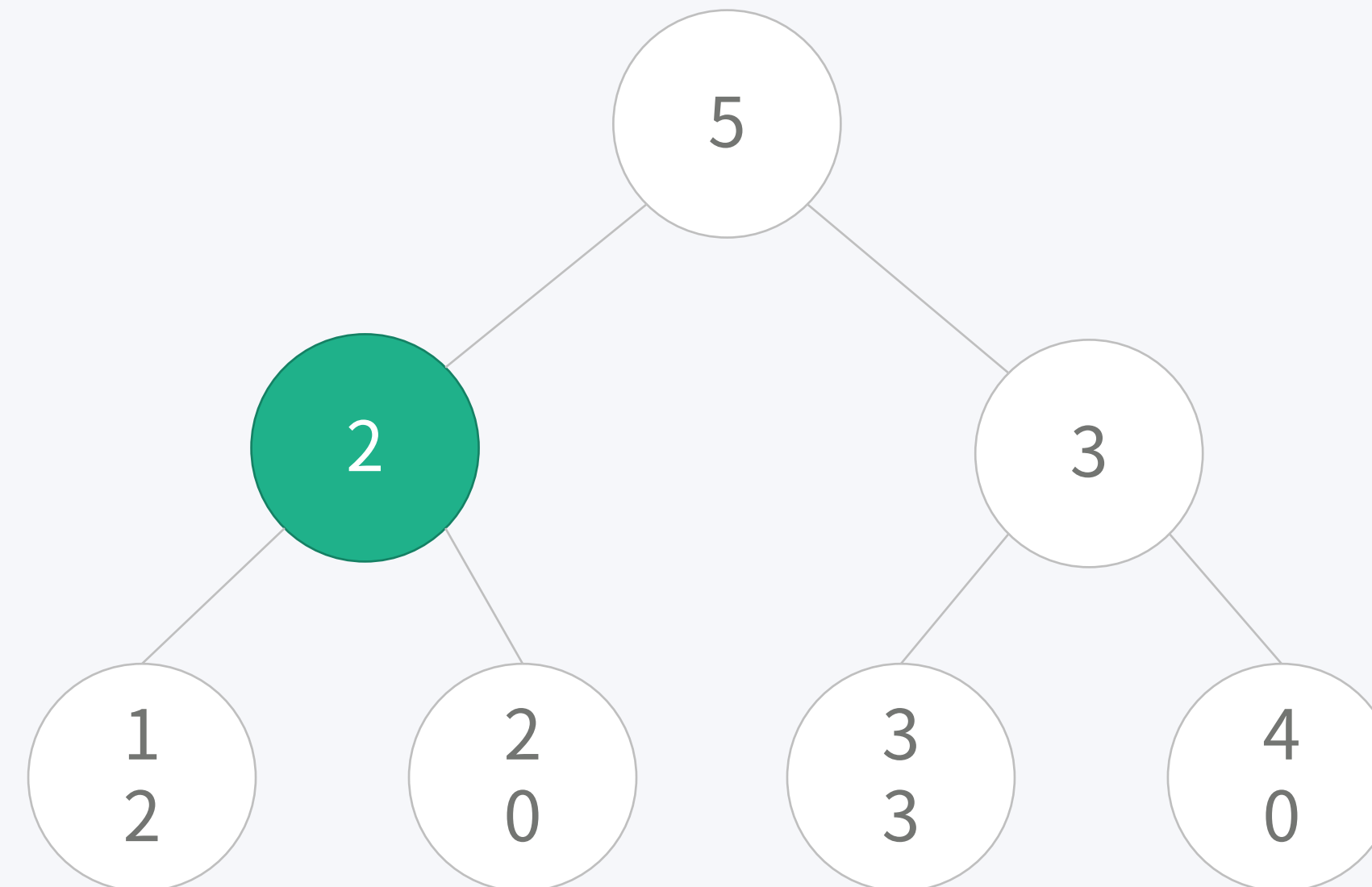


사탕상자

90

<https://www.acmicpc.net/problem/2243>

- 2번째 맛있는 사탕을 뺀다
- 왼쪽에 있다

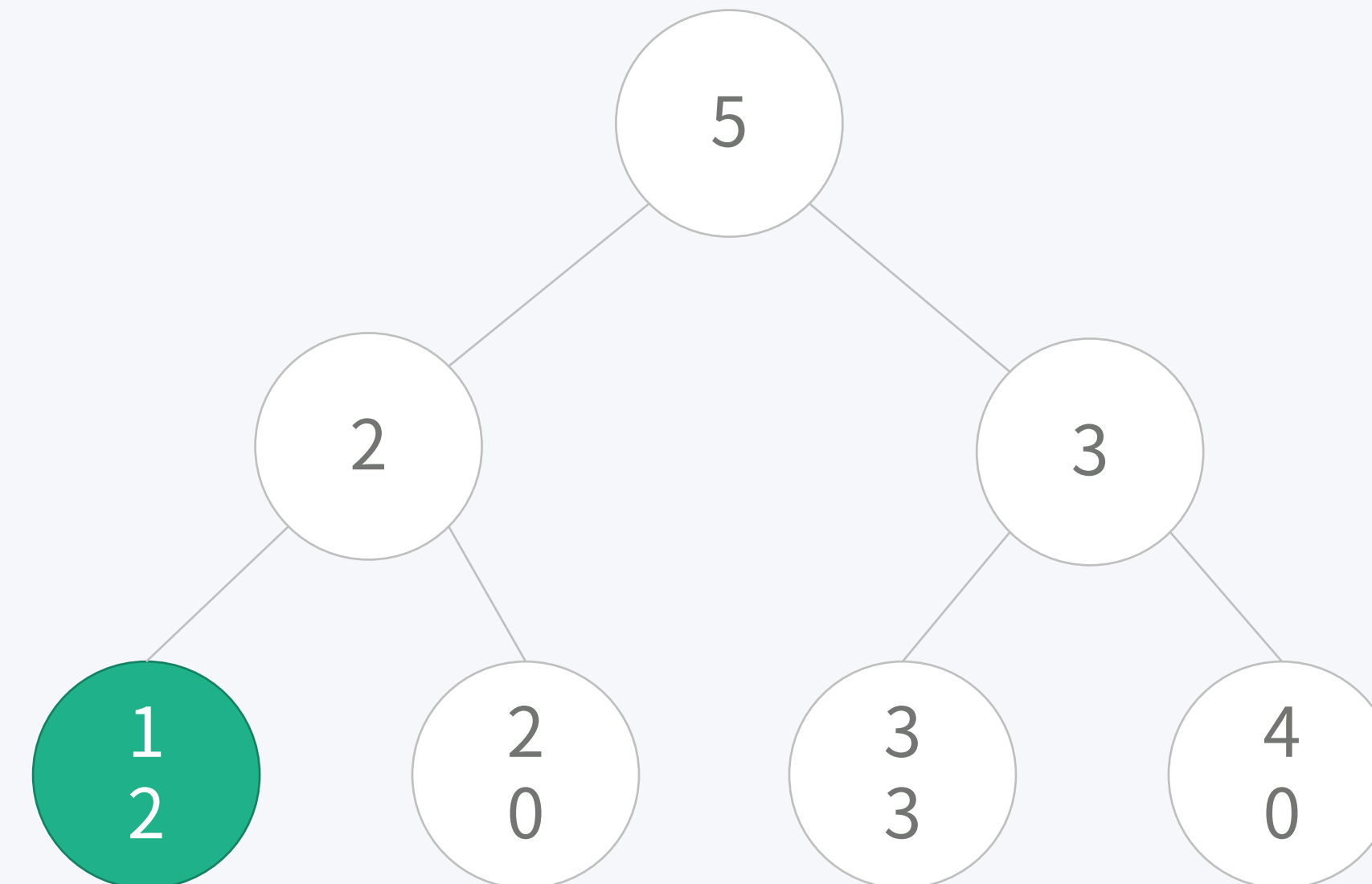


사탕상자

91

<https://www.acmicpc.net/problem/2243>

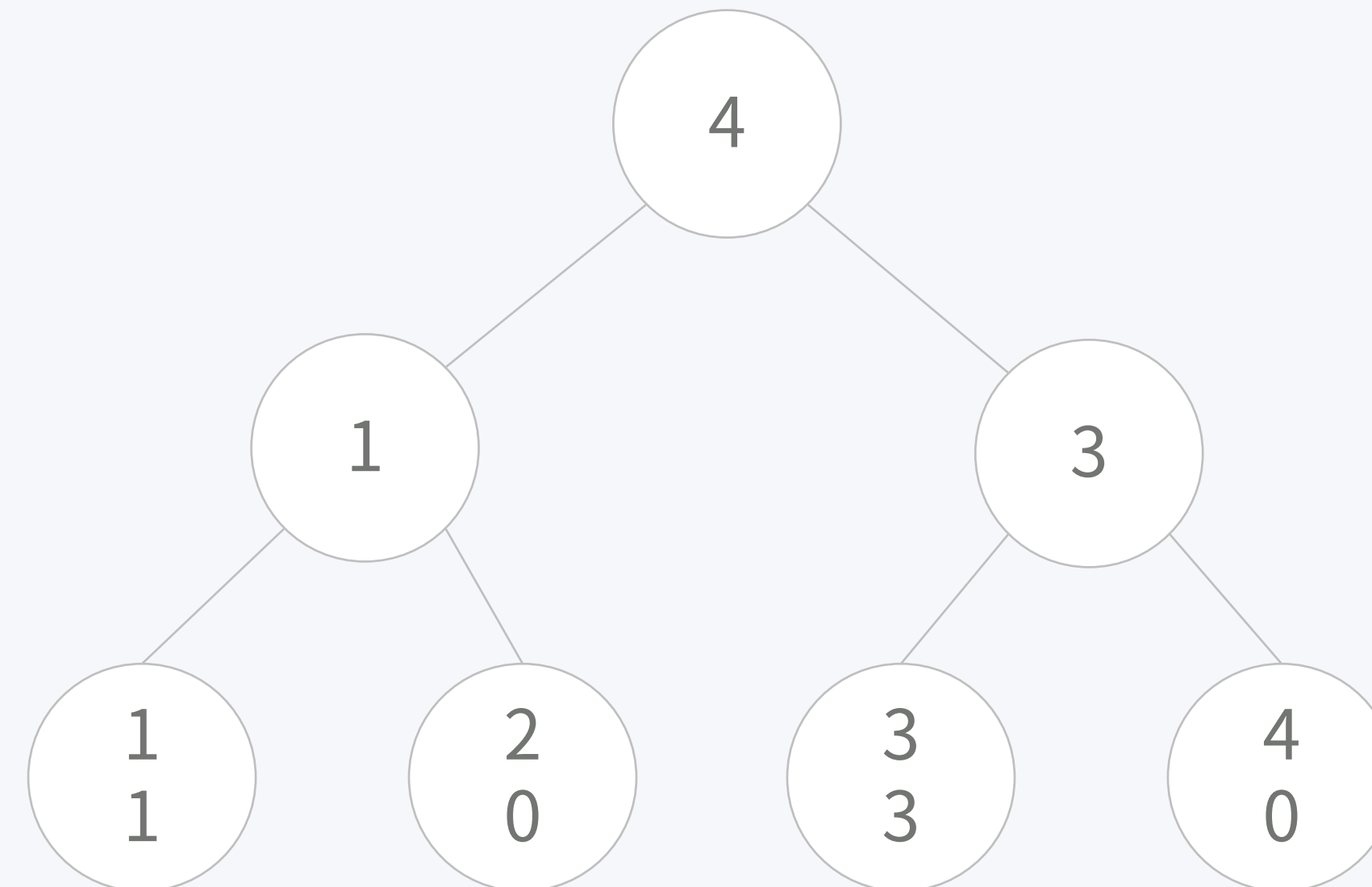
- 2번째 맛있는 사탕을 뺀다
- 왼쪽에 있다



사탕상자

<https://www.acmicpc.net/problem/2243>

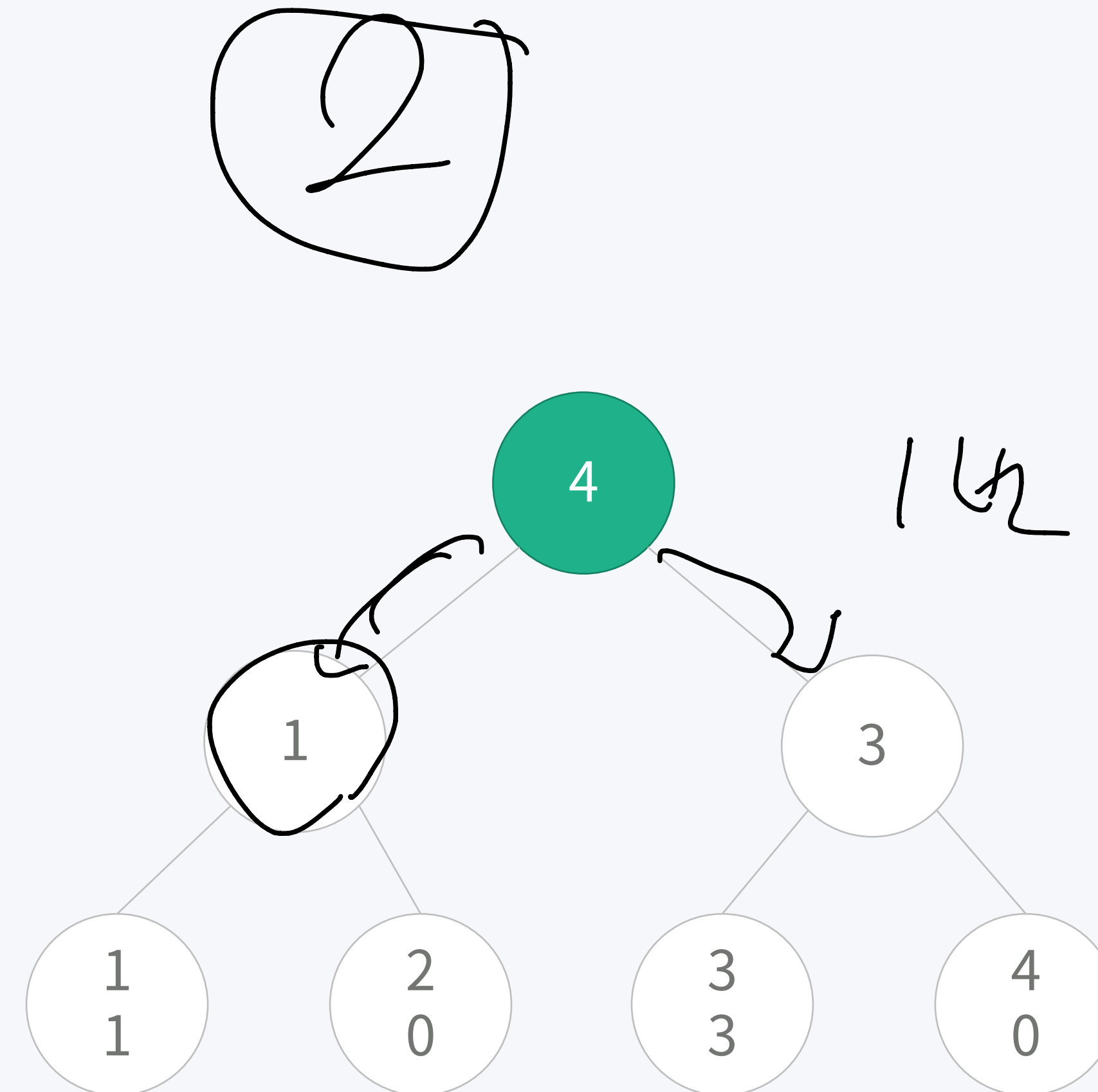
- 2번째 맛있는 사탕을 뺀다
- 왼쪽에 있다
- 사탕을 빼고 트리를 업데이트 한다



사탕상자

<https://www.acmicpc.net/problem/2243>

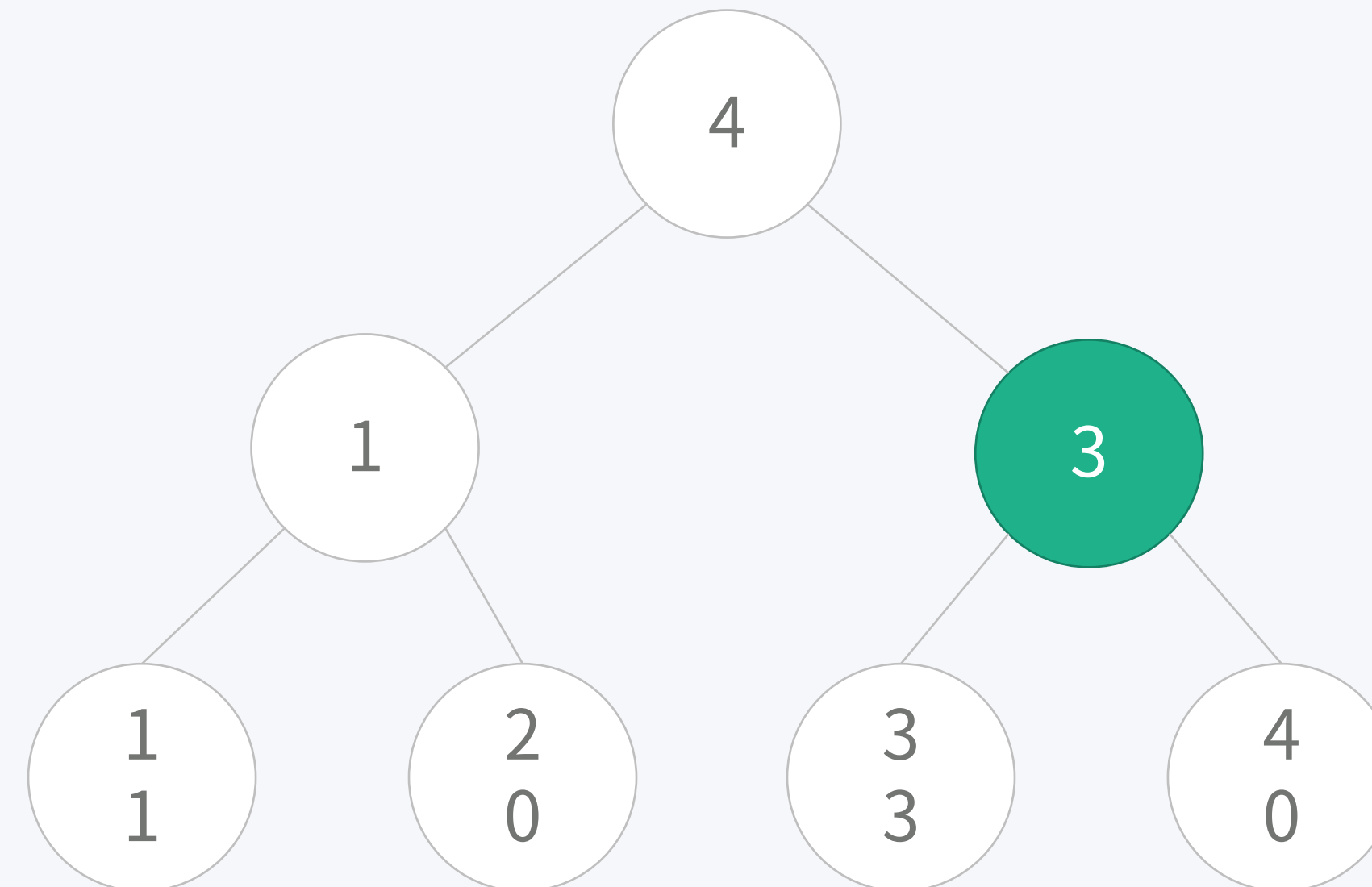
- 2번째 맛있는 사탕을 뺀다
- 오른쪽에 있다
- 오른쪽에서 $2-1 = 1$ 번째 사탕이다.



사탕상자

<https://www.acmicpc.net/problem/2243>

- 1번째 맛있는 사탕을 뺀다
- 왼쪽에 있다
- 왼쪽에서 1번째 사탕이다.

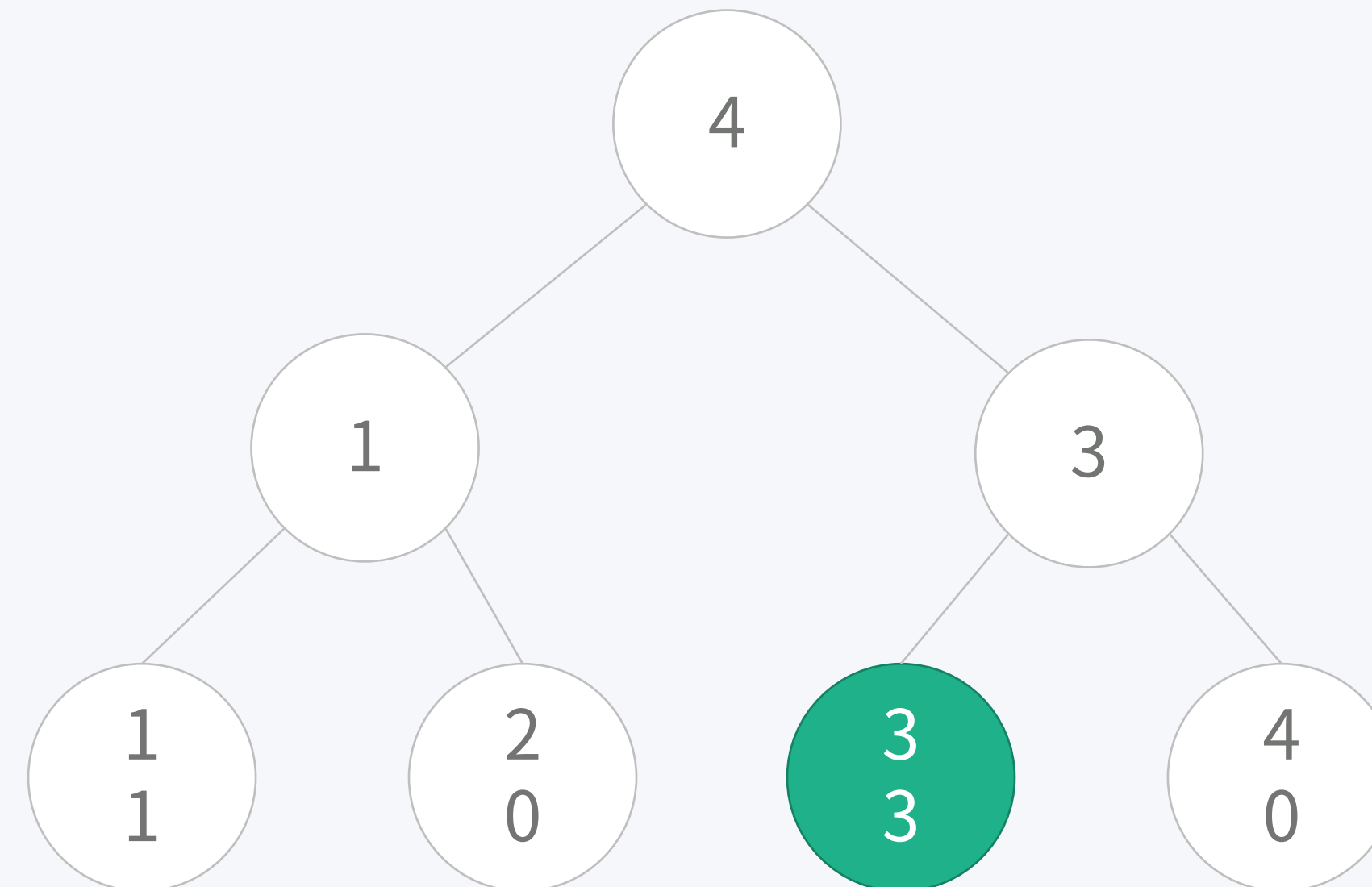


사탕상자

95

<https://www.acmicpc.net/problem/2243>

- 1번째 맛있는 사탕을 뺀다
- 사탕을 빼고 트리를 업데이트 한다

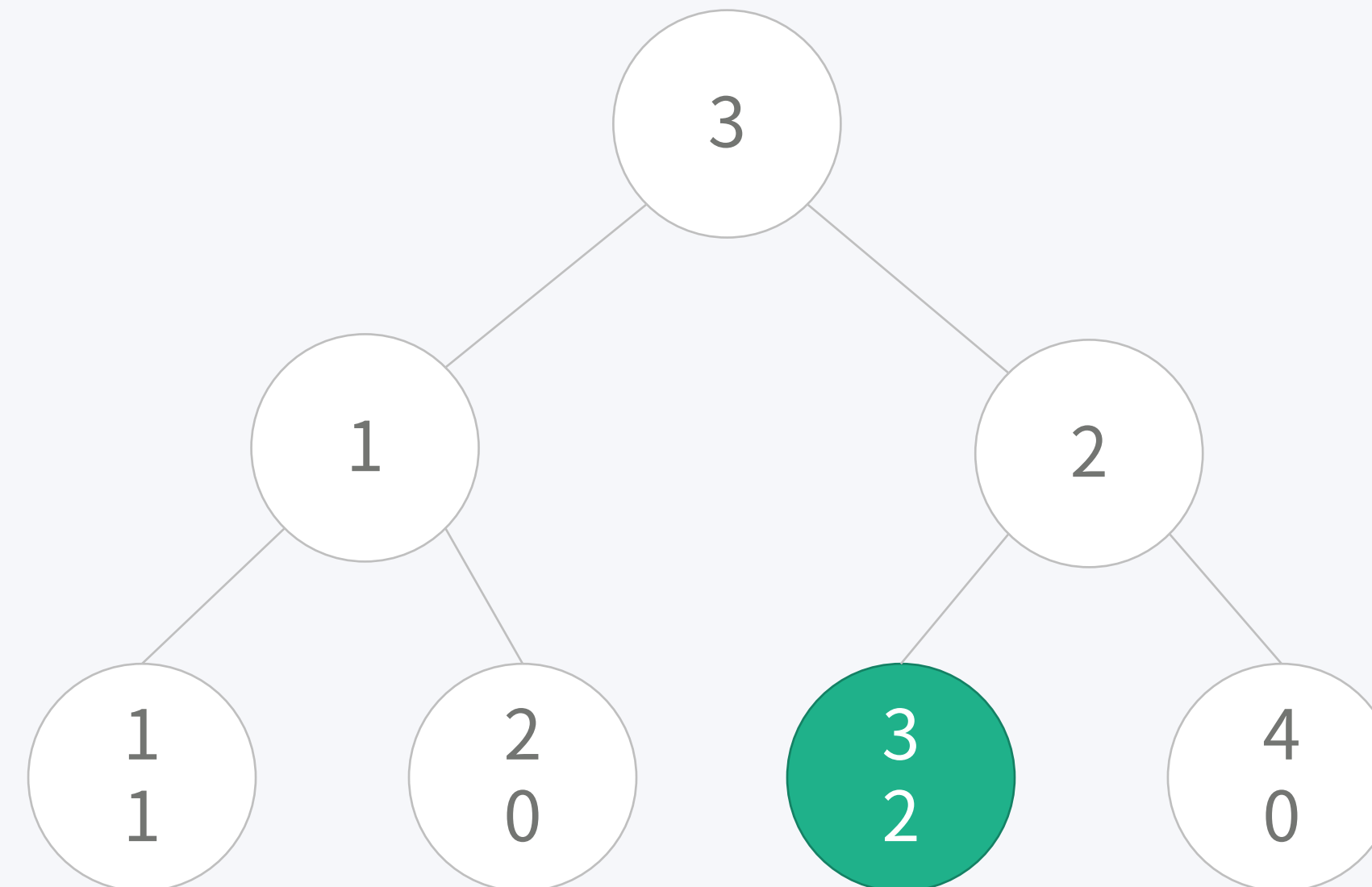


사탕상자

96

<https://www.acmicpc.net/problem/2243>

- 1번째 맛있는 사탕을 뺀다
- 사탕을 빼고 트리를 업데이트 한다



사탕상자

<https://www.acmicpc.net/problem/2243>

- C/C++: <https://gist.github.com/Baekjoon/90f48983c26a60acffe2>

중앙값 측정

98

<https://www.acmicpc.net/problem/9426>

- C/C++: <https://gist.github.com/Baekjoon/6d34b492c1aef6062fa3e612d29d5e17>

순열

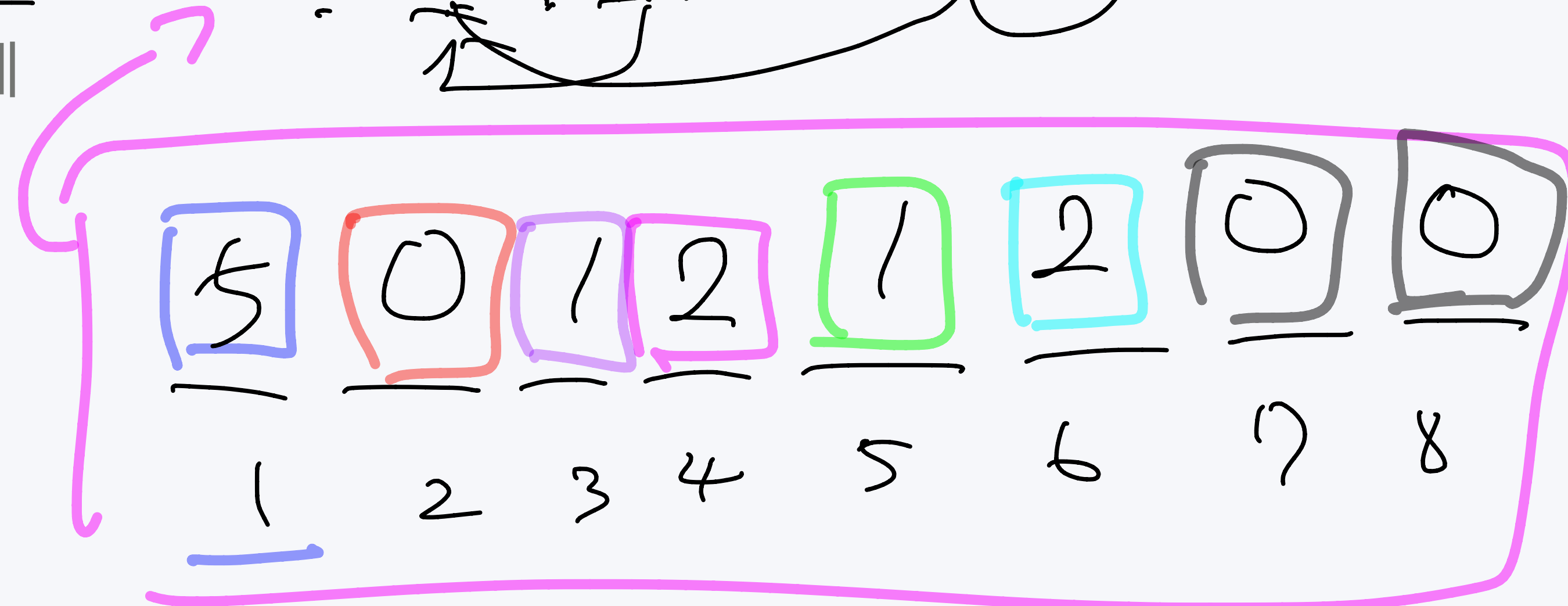
<https://www.acmicpc.net/problem/1849>

- 1~N까지 이루어진 수열
- i앞에 있는 수 중에서 i보다 큰 수의 개수가 a[i]
- a[i]가 주어졌을 때, 원래 수열을 구하는 문제

- a: 5 0 1 2 1 2 0 0 $N \leq 10만$
- 원래 수열: 2 7 3 5 4 1 8 6

N^2

2 7 3 5 4 1 8 6



2 7 3 5 4 1 8 6

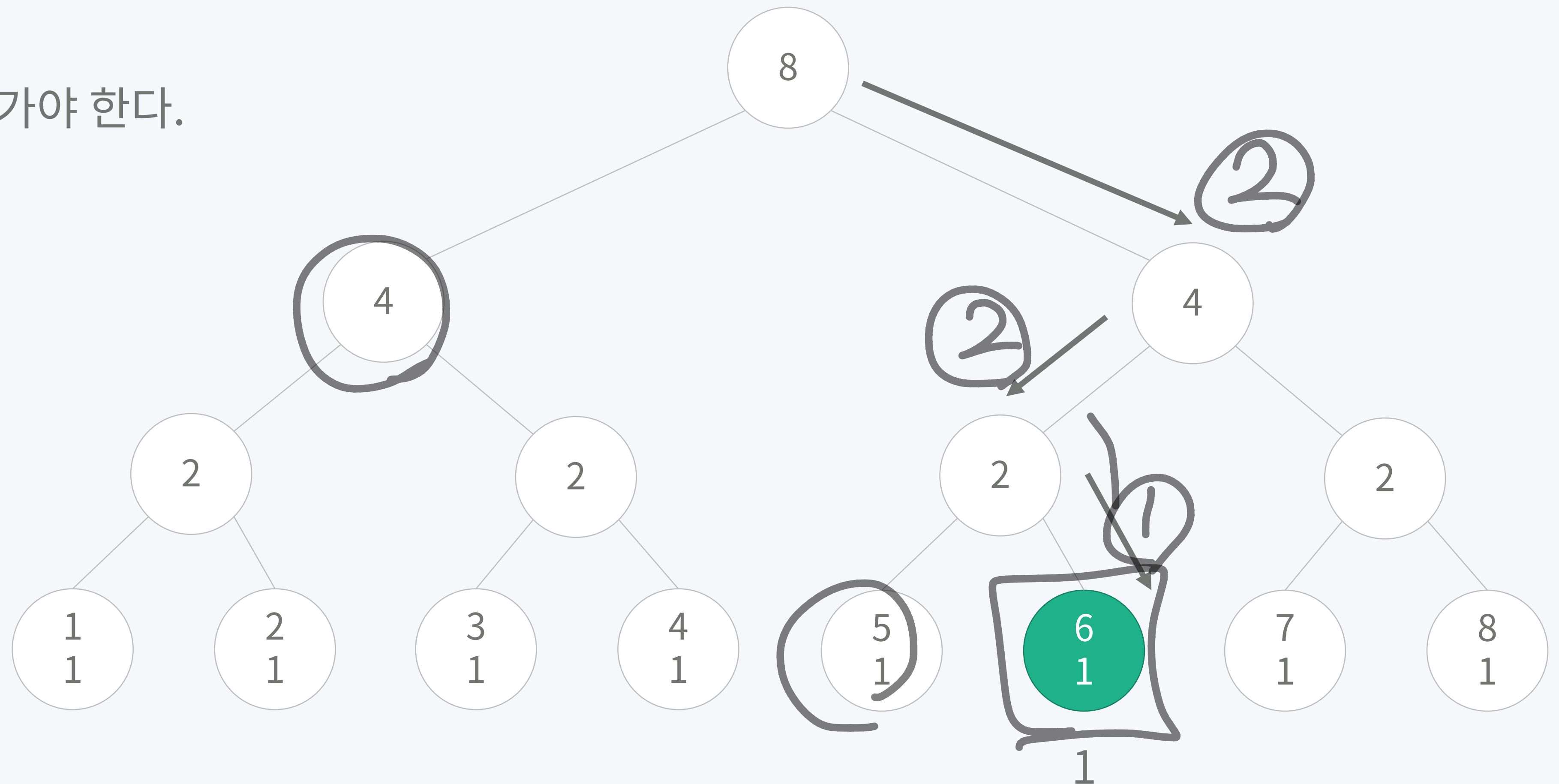
1 2 3 4 5 6 7 8

순열

100

<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[1] = 5$
- 6번째 위치에 들어가야 한다.

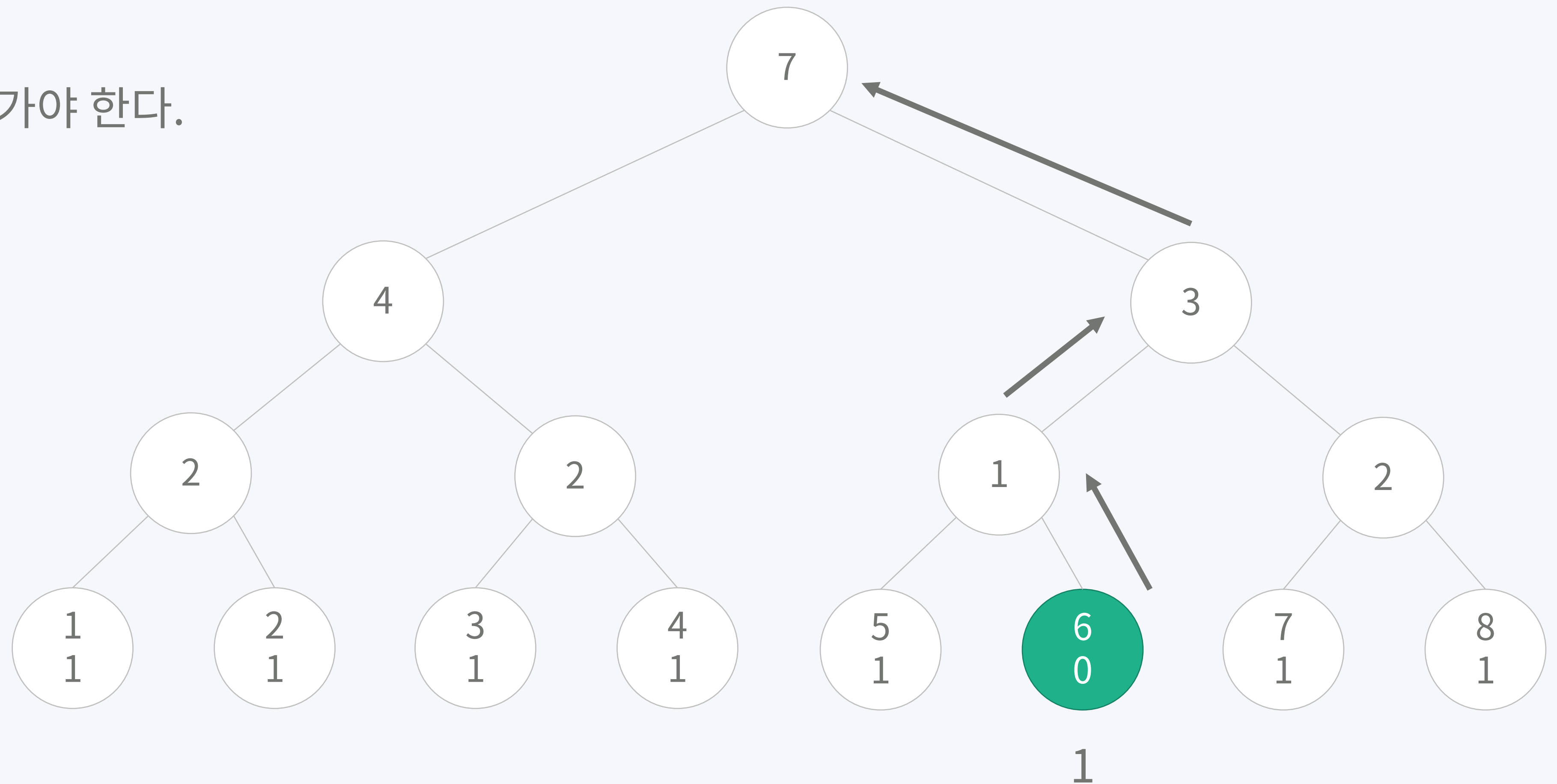


순열

101

<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[1] = 5$
- 6번째 위치에 들어가야 한다.

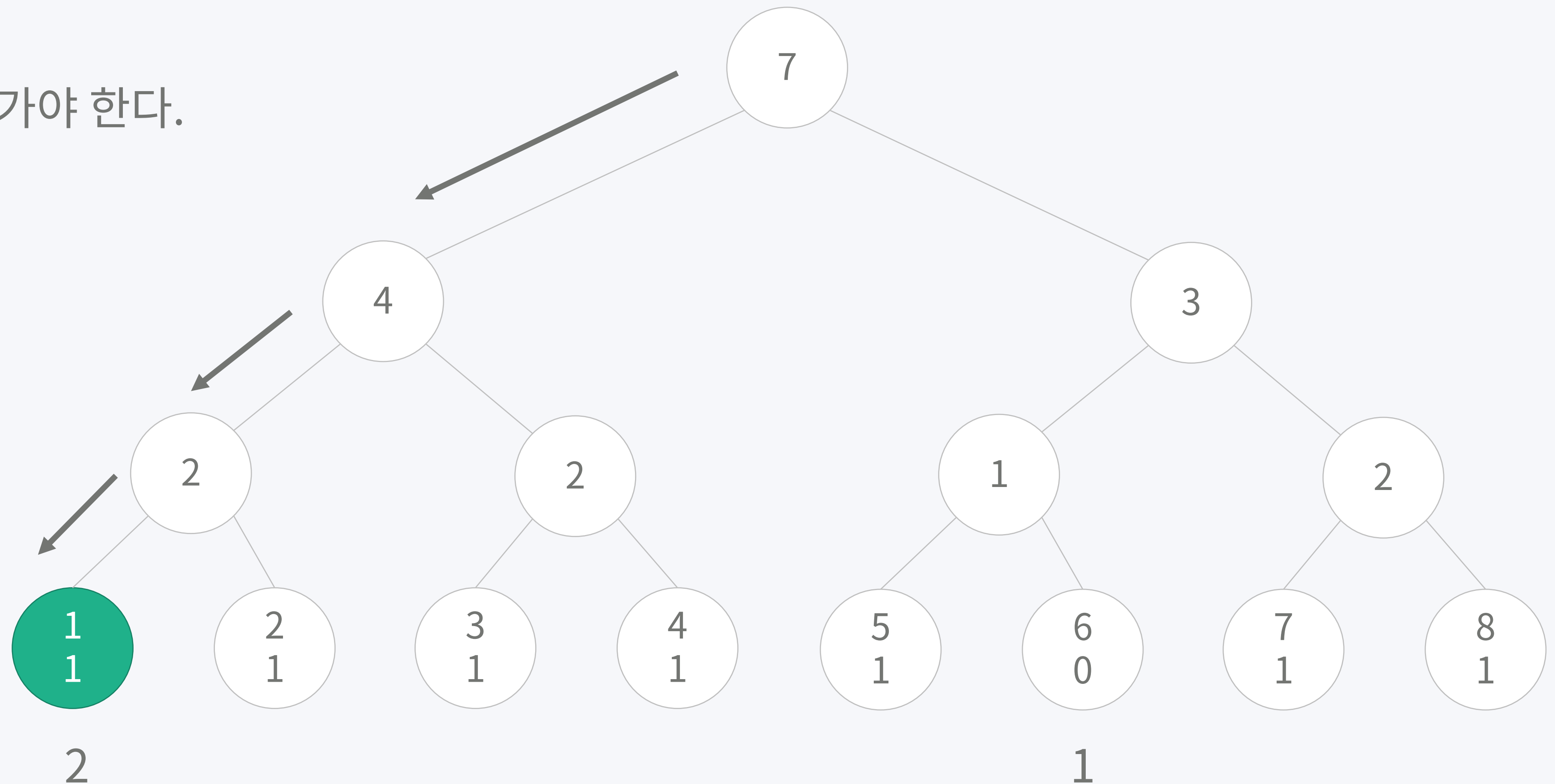


순열

102

<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[2] = 0$
- 1번째 위치에 들어가야 한다.

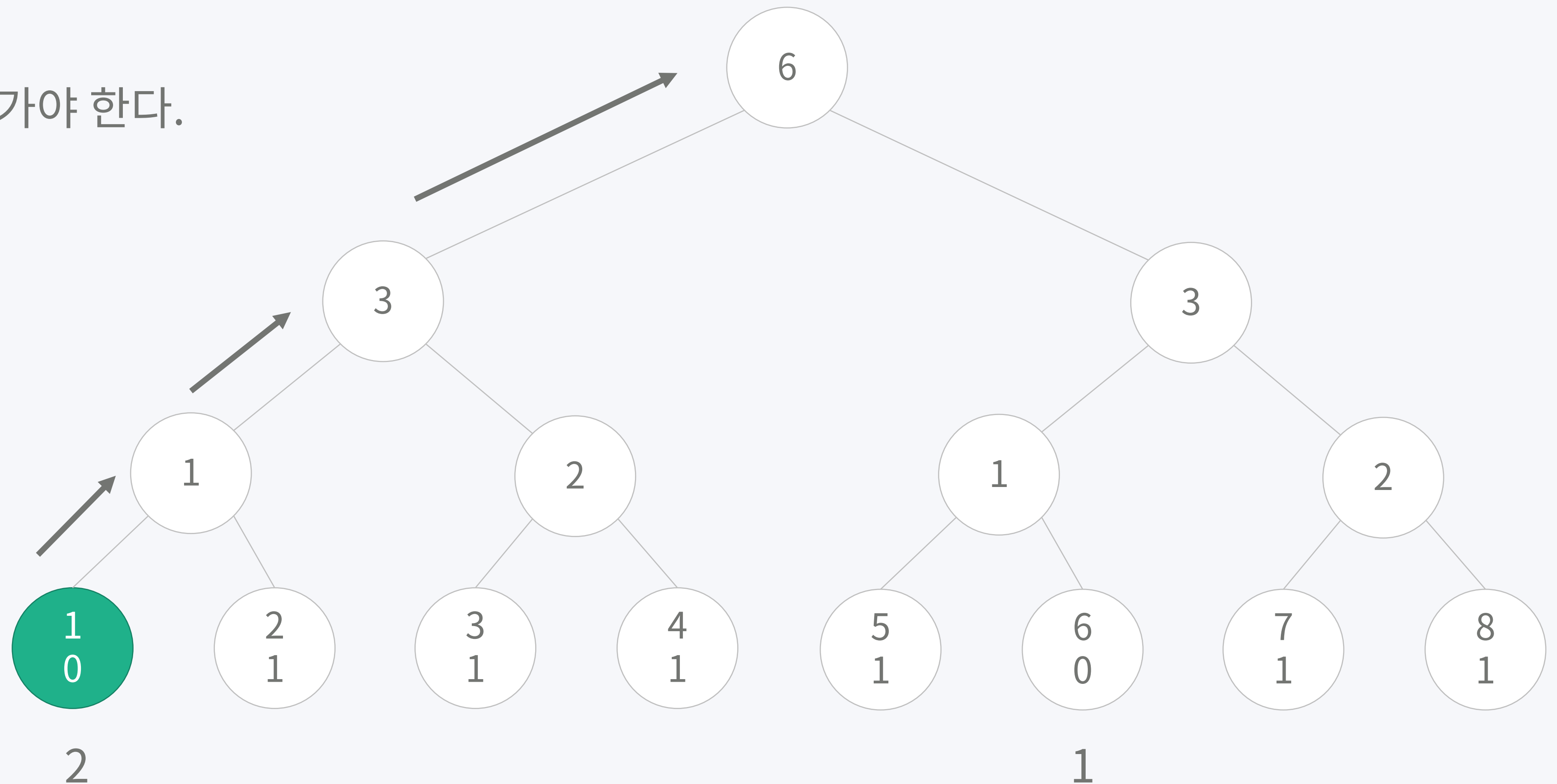


순열

103

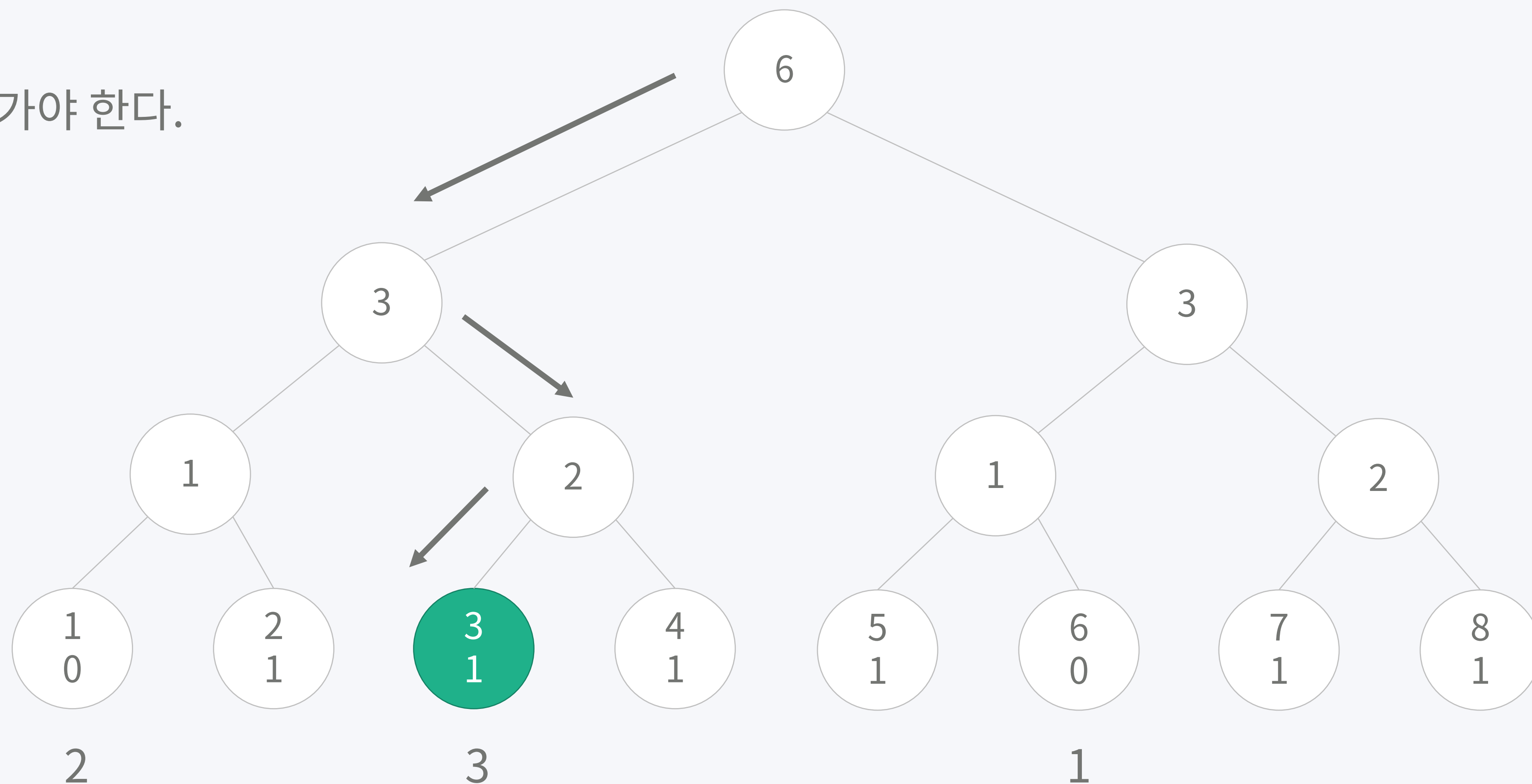
<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[2] = 0$
- 1번째 위치에 들어가야 한다.



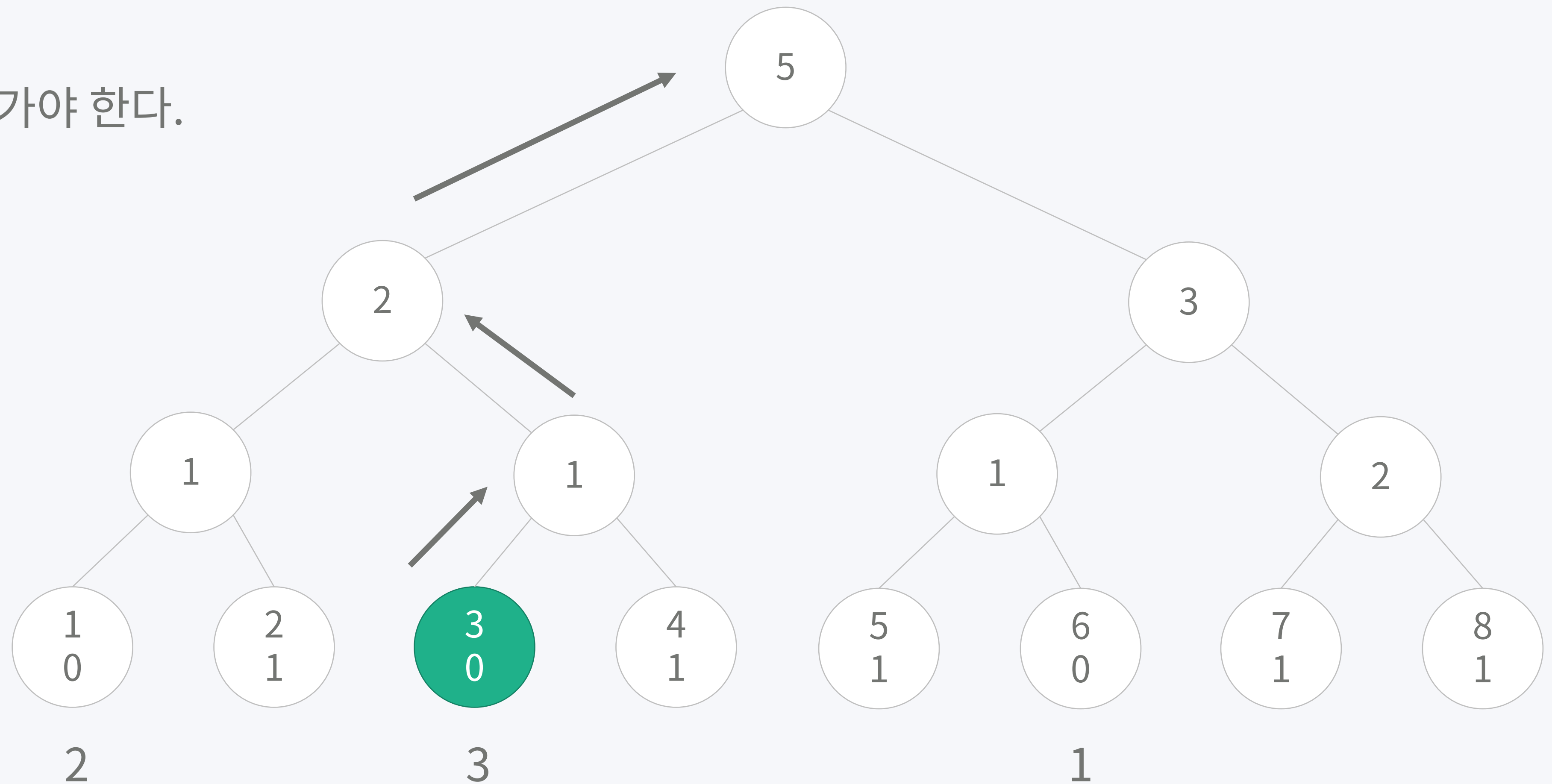
<https://www.acmicpc.net/problem/1849>

- 5 0 **1** 2 1 2 0 0
- $a[3] = 1$
- 2번째 위치에 들어가야 한다.



<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[3] = 1$
- 2번째 위치에 들어가야 한다.

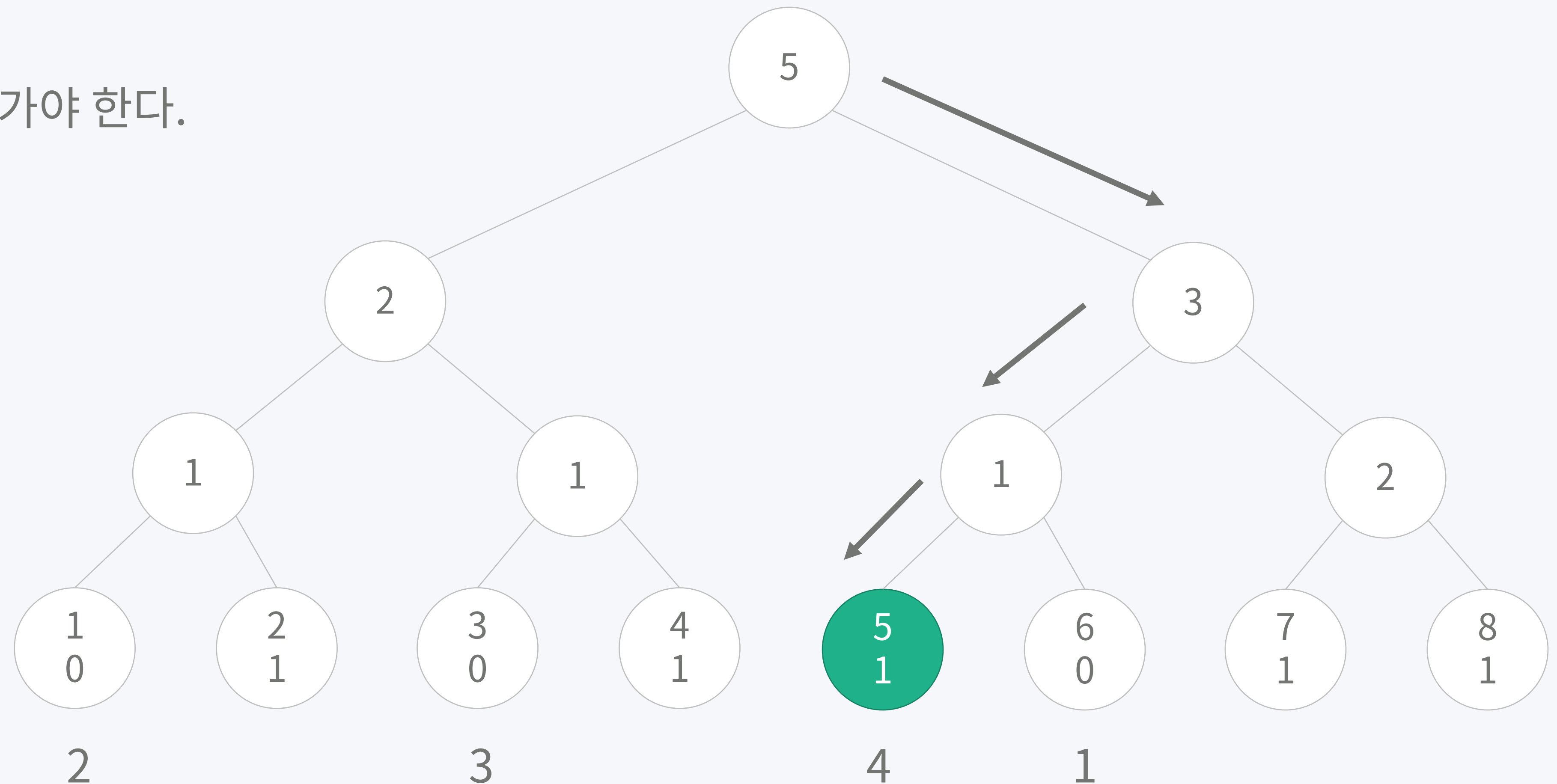


순열

106

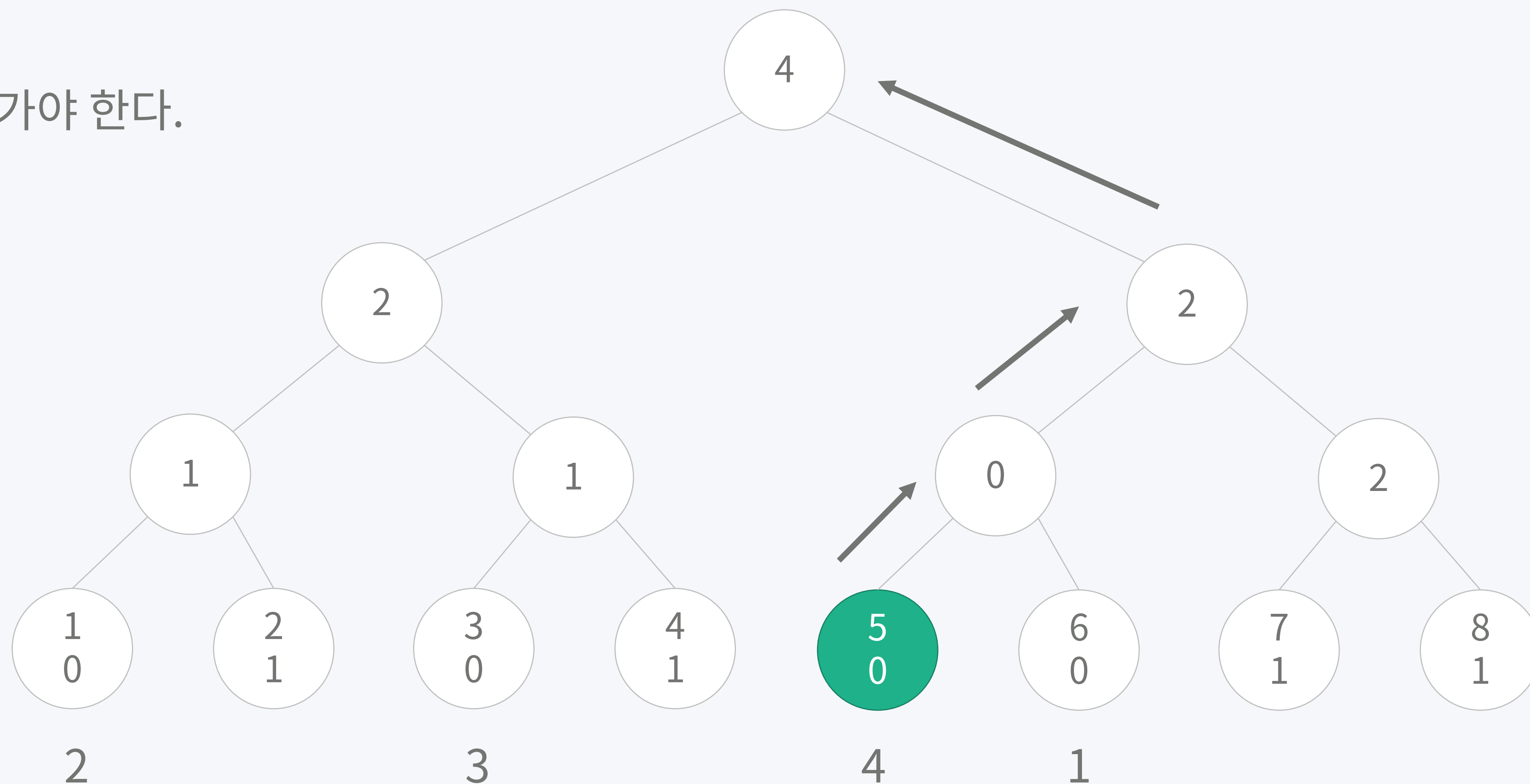
<https://www.acmicpc.net/problem/1849>

- 5 0 1 **2** 1 2 0 0
- $a[4] = 2$
- 3번째 위치에 들어가야 한다.



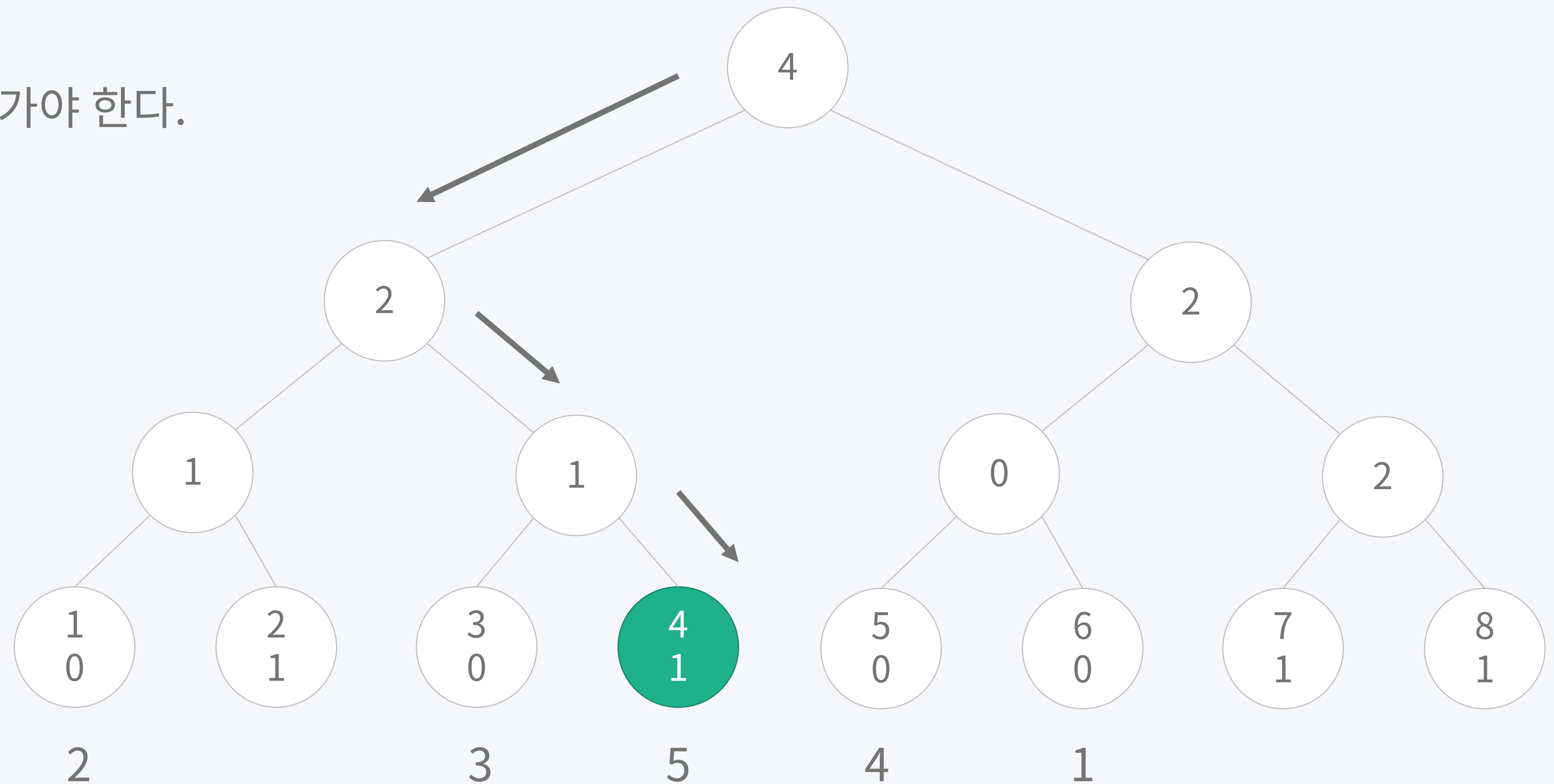
<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[4] = 2$
- 3번째 위치에 들어가야 한다.



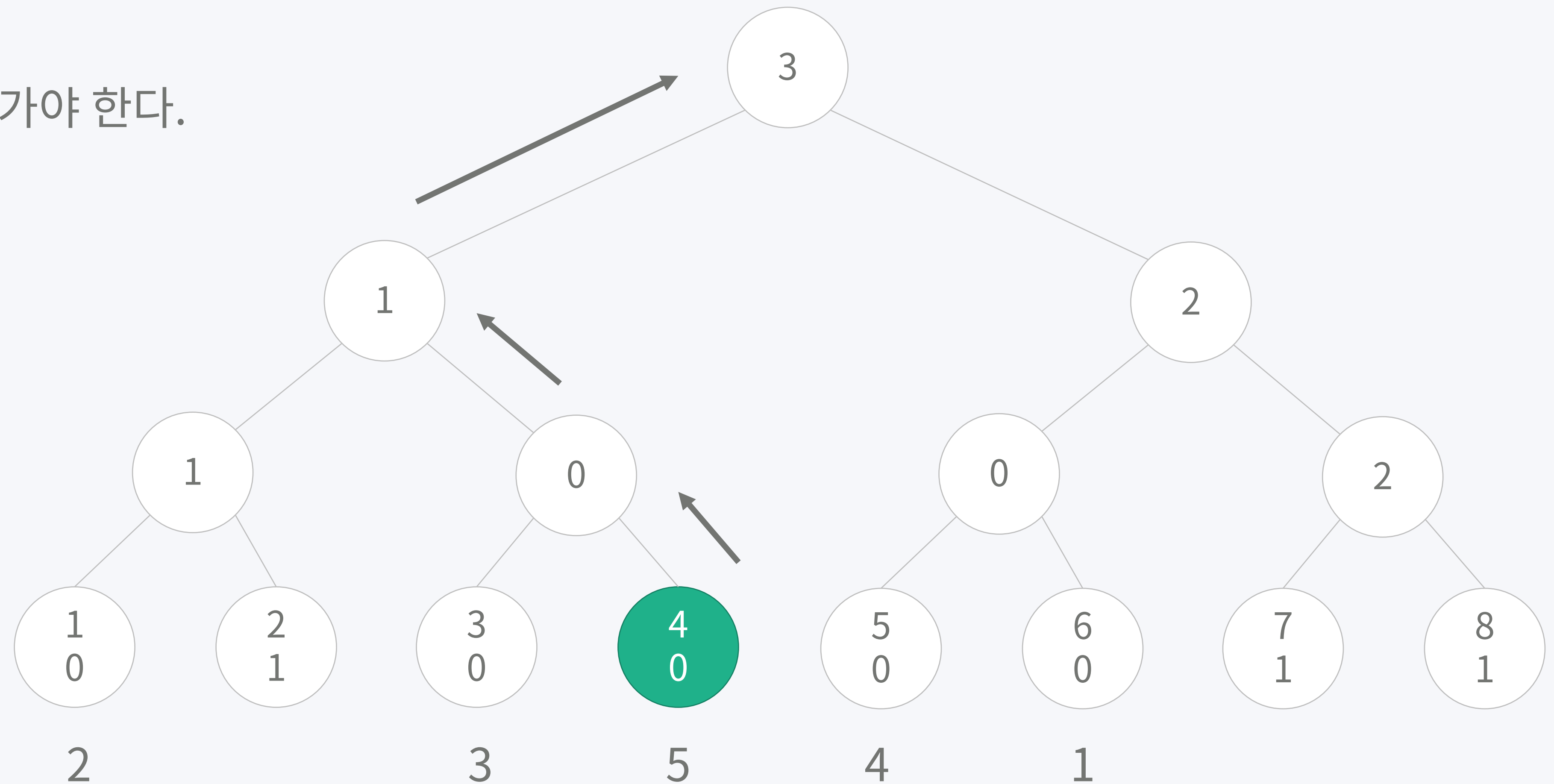
<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 **1** 2 0 0
- $a[5] = 1$
- 2번째 위치에 들어가야 한다.



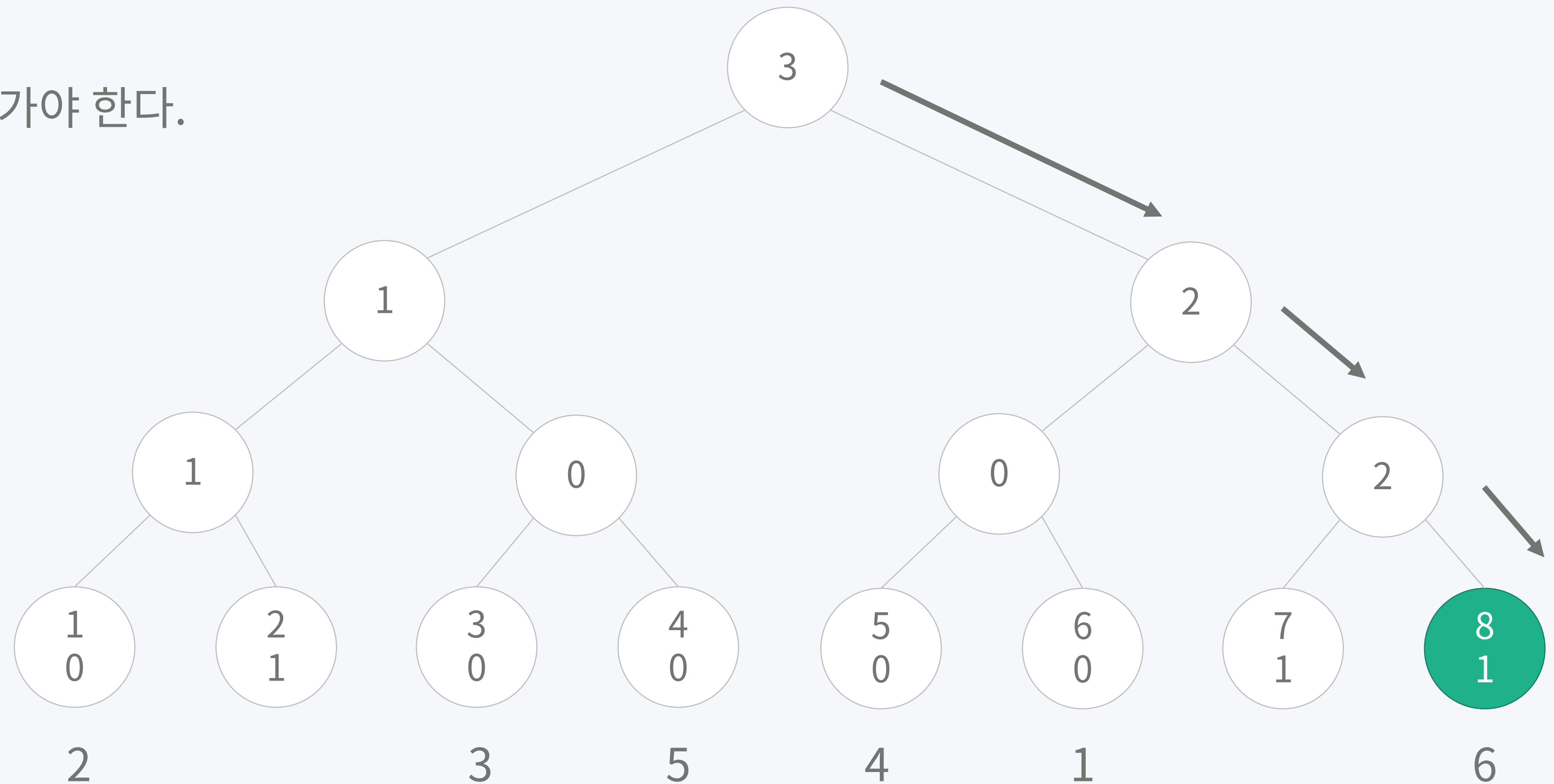
<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 **1** 2 0 0
- $a[5] = 1$
- 2번째 위치에 들어가야 한다.



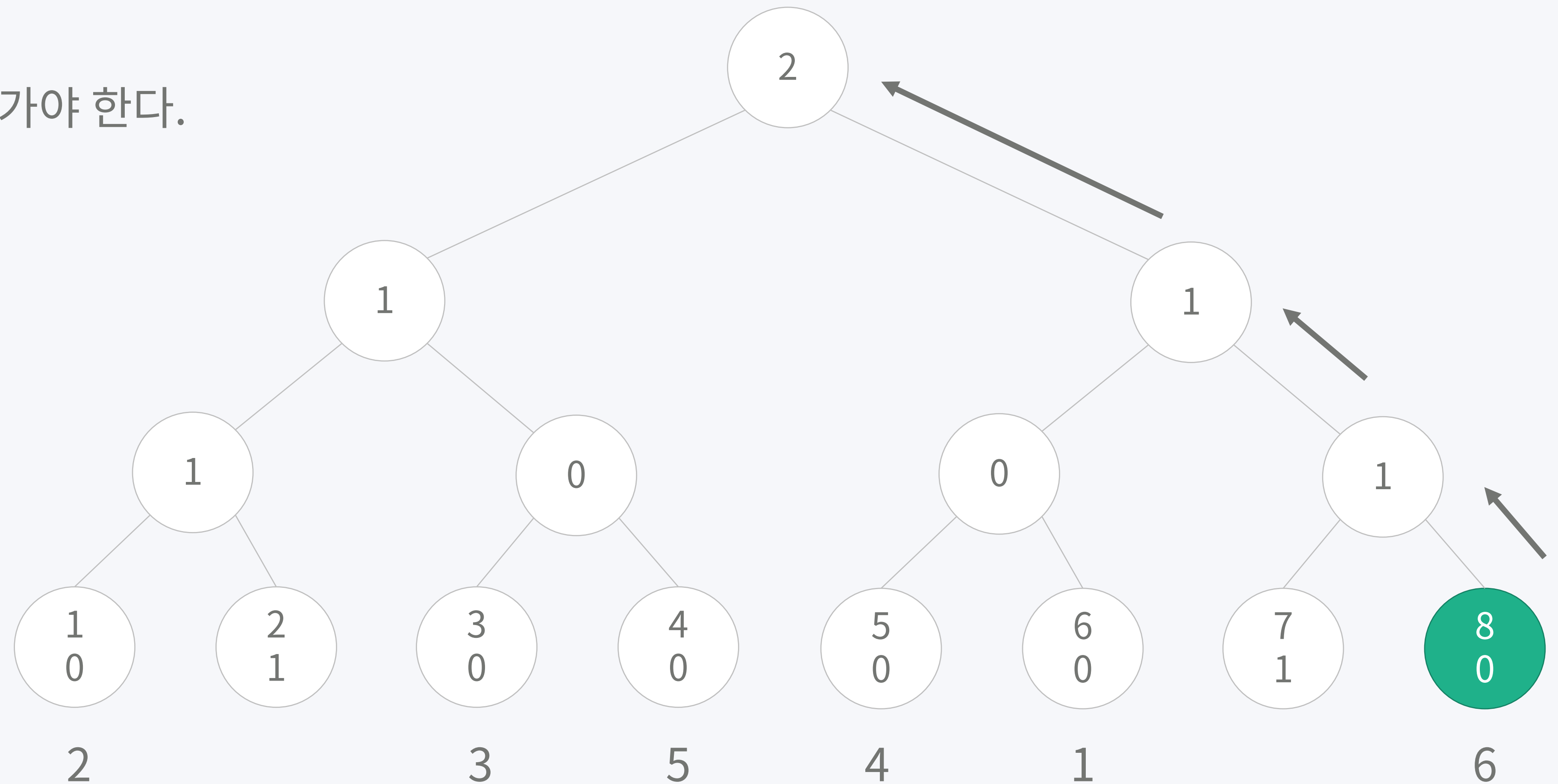
<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[6] = 2$
- 3번째 위치에 들어가야 한다.



<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[6] = 2$
- 3번째 위치에 들어가야 한다.

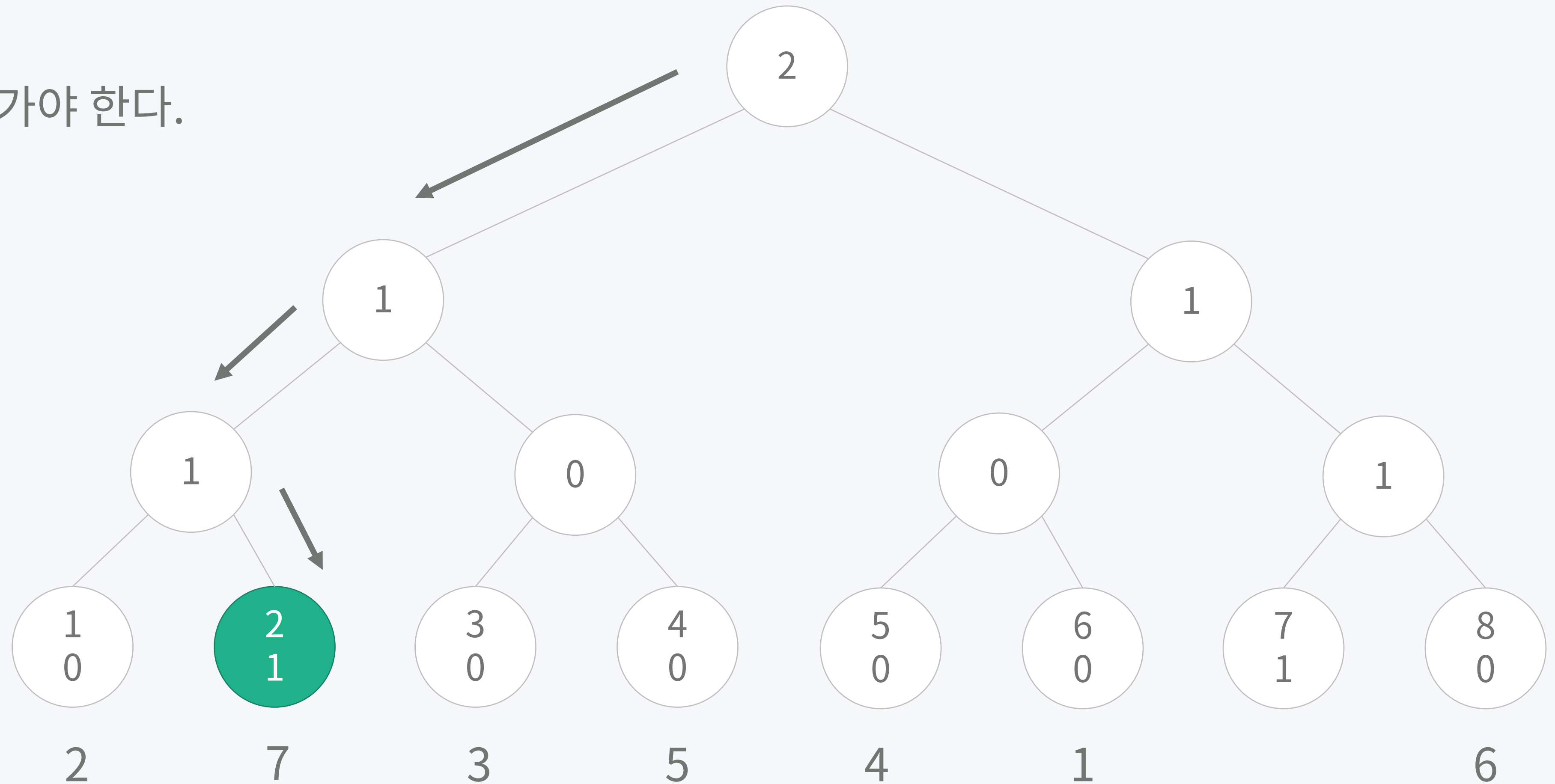


순열

112

<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[7] = 0$
- 1번째 위치에 들어가야 한다.

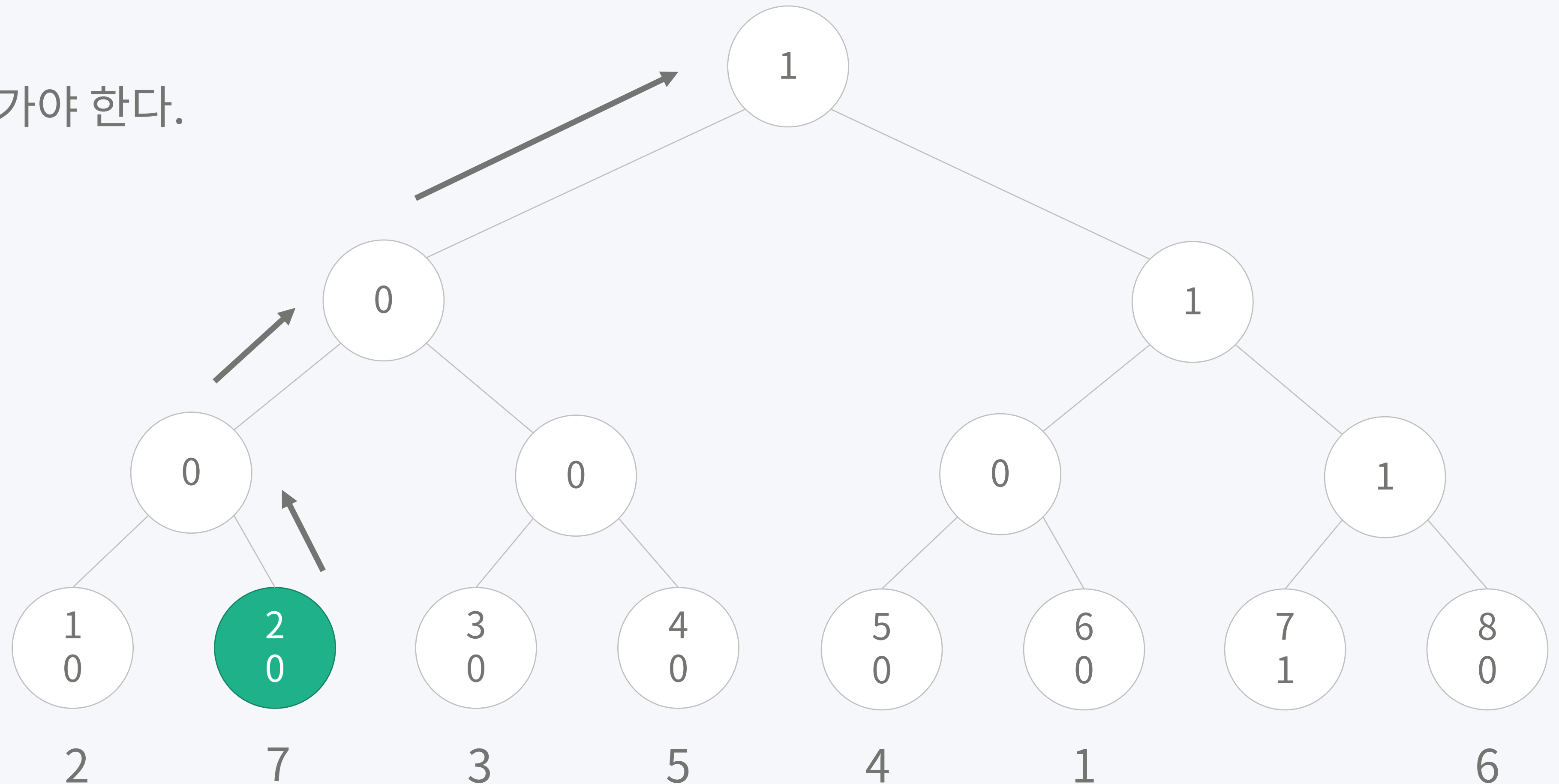


순열

113

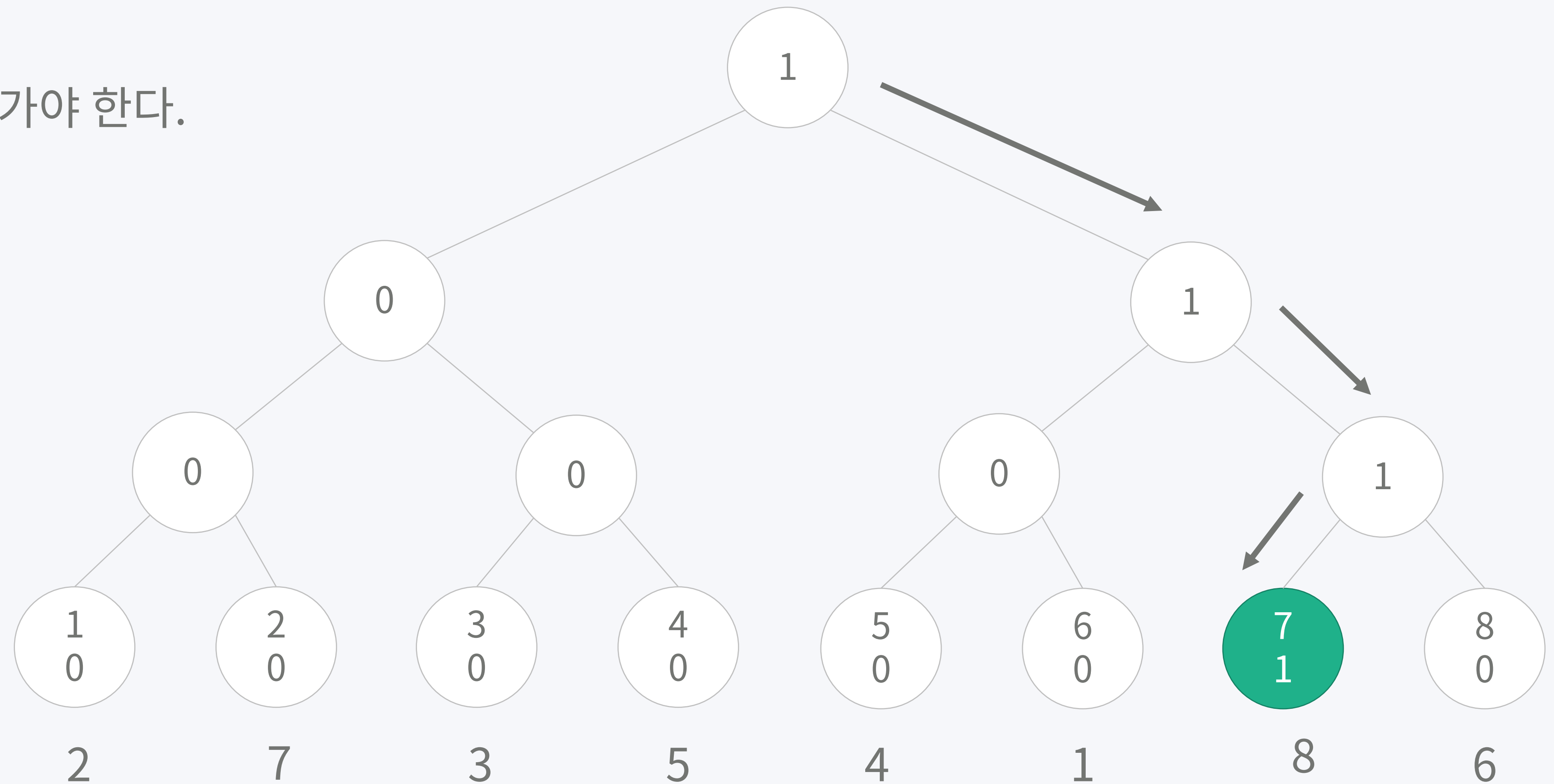
<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[7] = 0$
- 1번째 위치에 들어가야 한다.



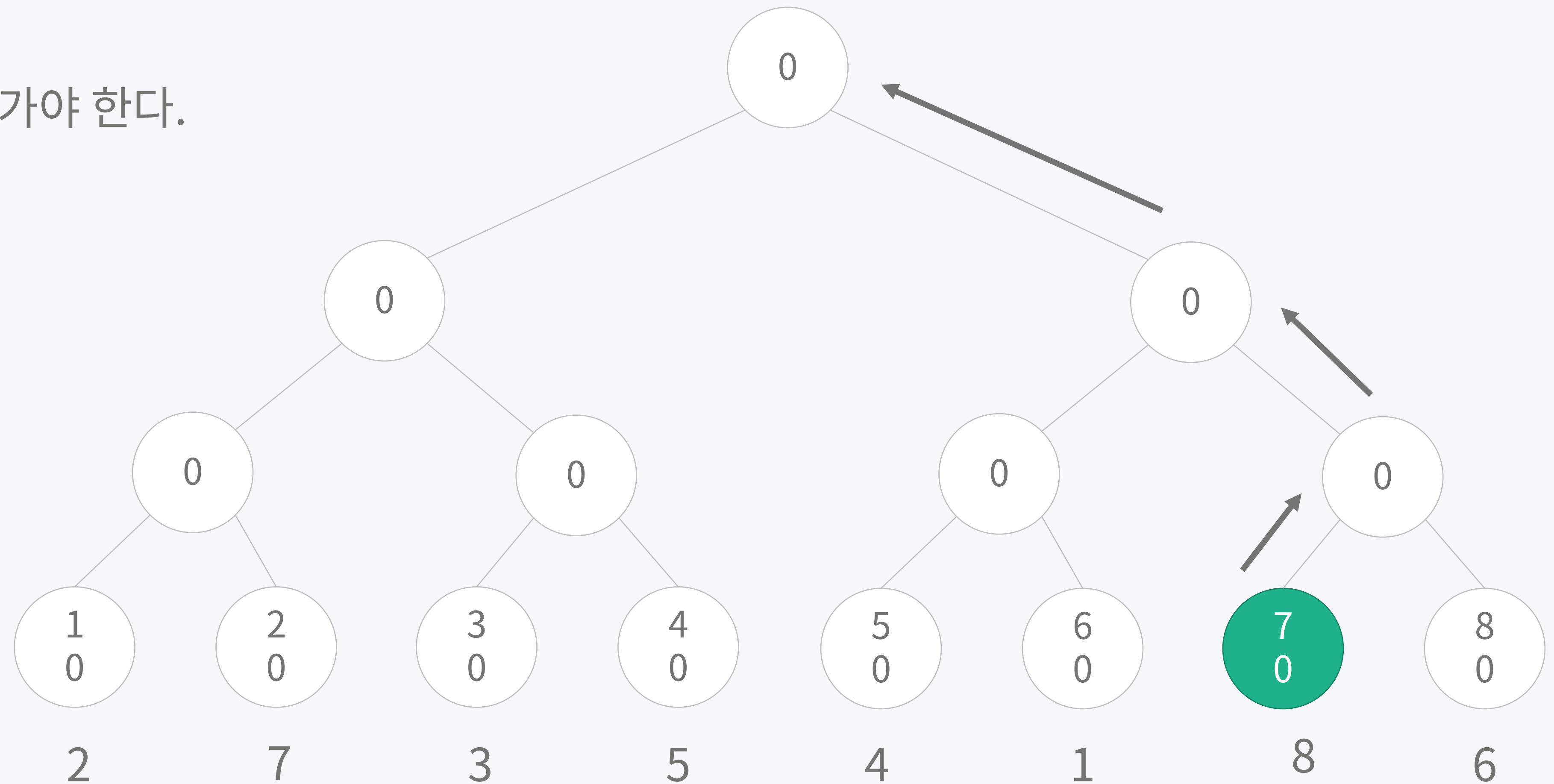
<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[8] = 0$
- 1번째 위치에 들어가야 한다.



<https://www.acmicpc.net/problem/1849>

- 5 0 1 2 1 2 0 0
- $a[8] = 0$
- 1번째 위치에 들어가야 한다.



<https://www.acmicpc.net/problem/1849>

- C/C++: <https://gist.github.com/Baekjoon/3ec9a5bf774c5b1f00e9>
- Java: <https://gist.github.com/Baekjoon/3065c0c01f6f7847541829318811be58>

조세퍼스 문제

<https://www.acmicpc.net/problem/1168>

- N과 K가 주어졌을 때, 조세퍼스 순열을 구하는 문제
- $N = 7, K = 3$
- $\langle 3, 6, 2, 7, 5, 1, 4 \rangle$

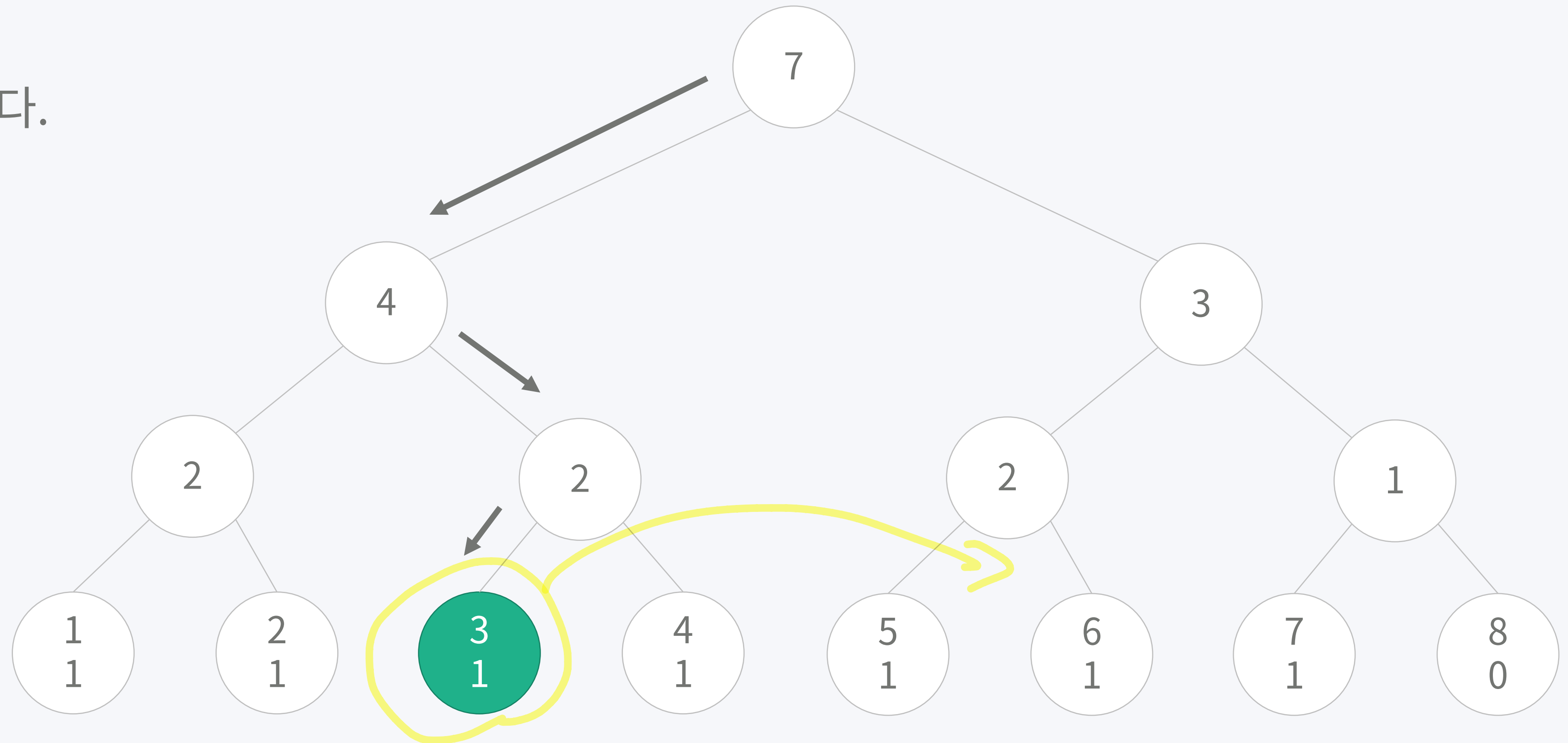
조세퍼스 문제

118

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$

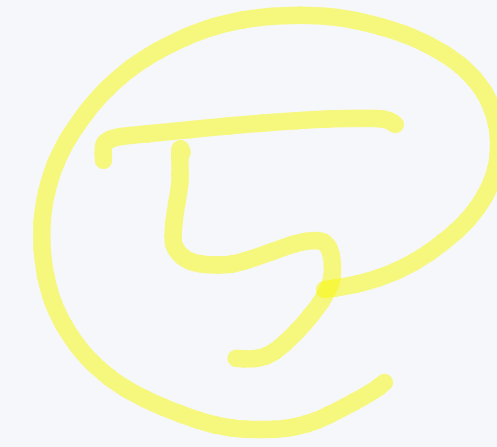
- 3번째 위치를 찾는다.



조세퍼스 문제

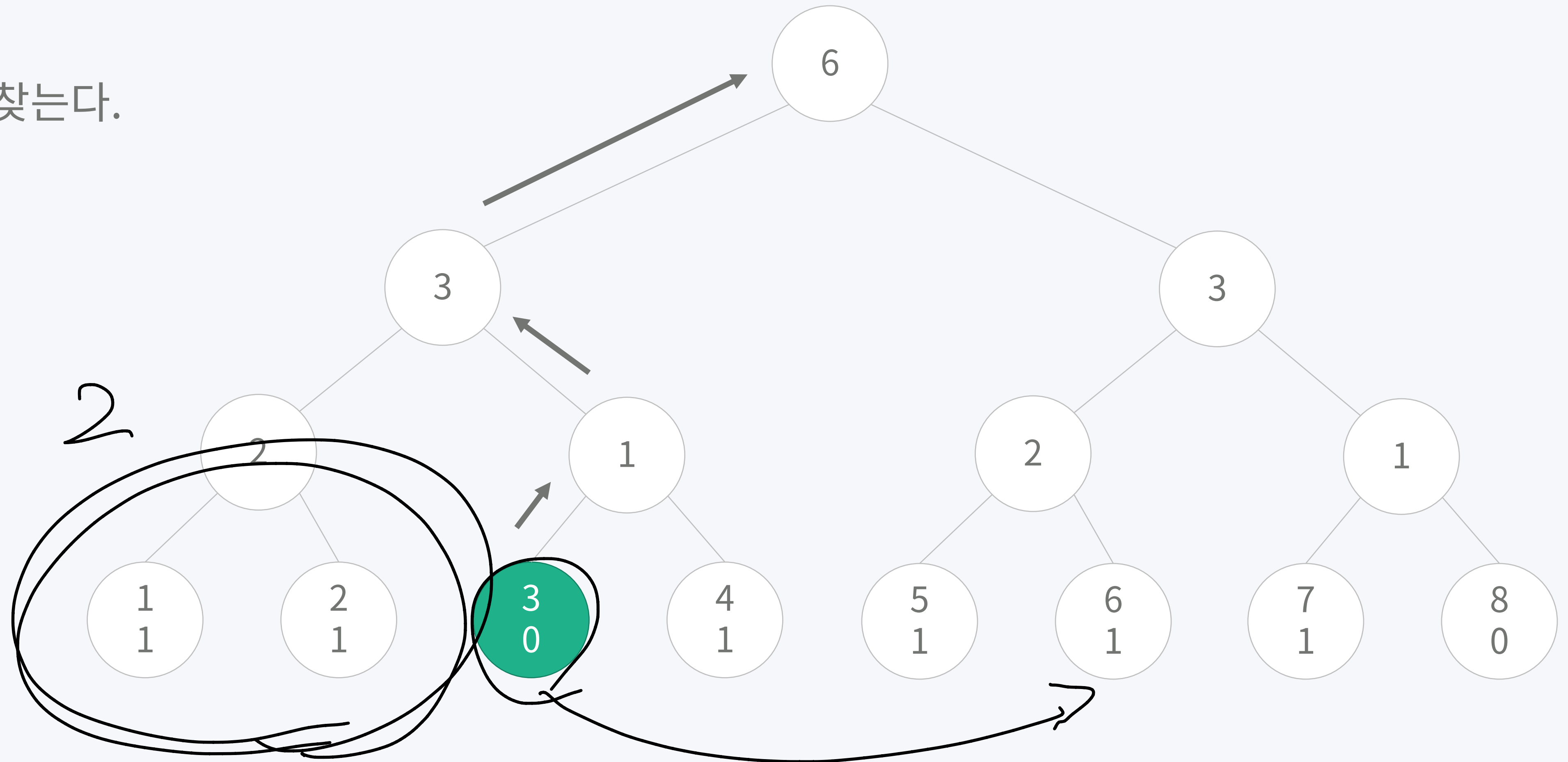
<https://www.acmicpc.net/problem/1168>

119



- $N = 7, K = 3$

- 3번째 위치를 찾는다.

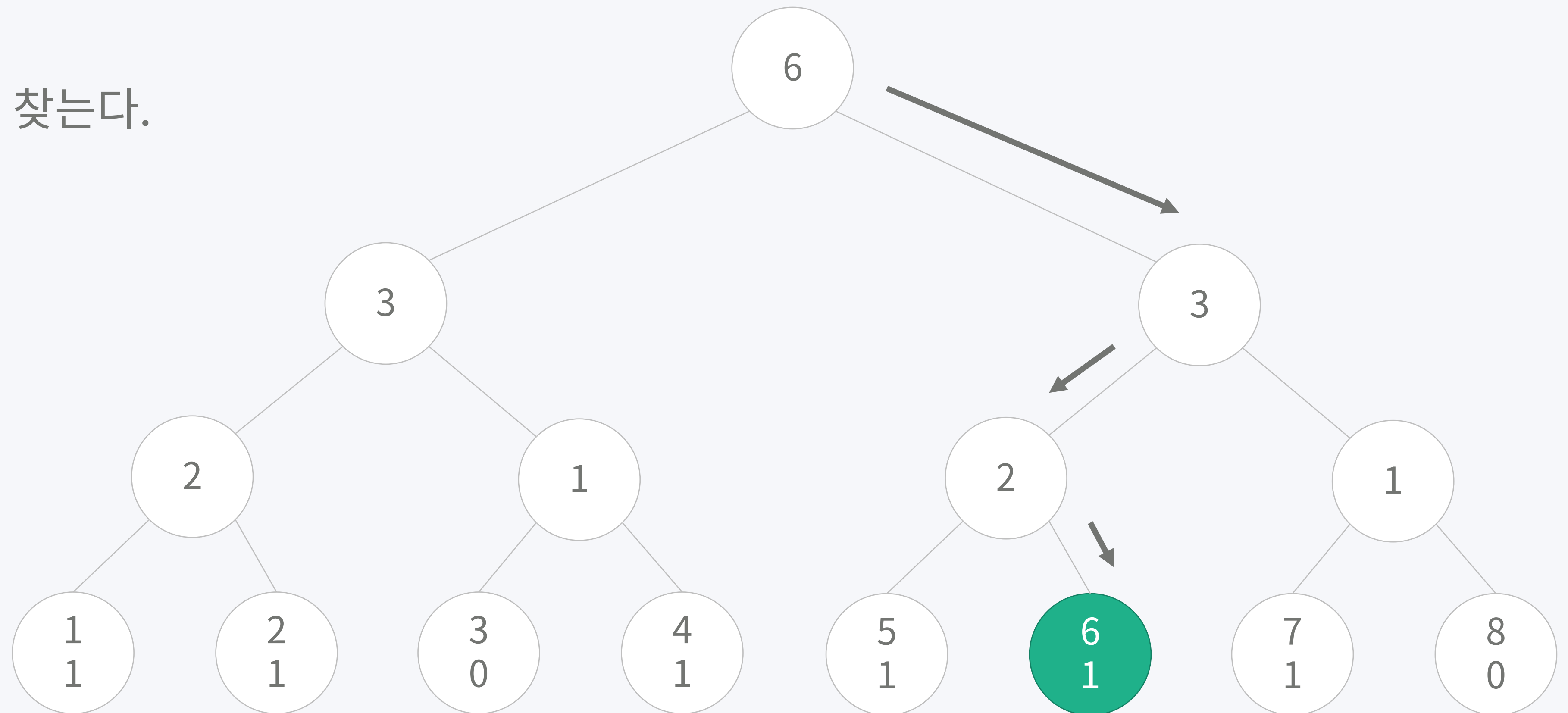


조세퍼스 문제

120

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~3까지 합 = 2
- $2+3=5$ 번째 위치를 찾는다.

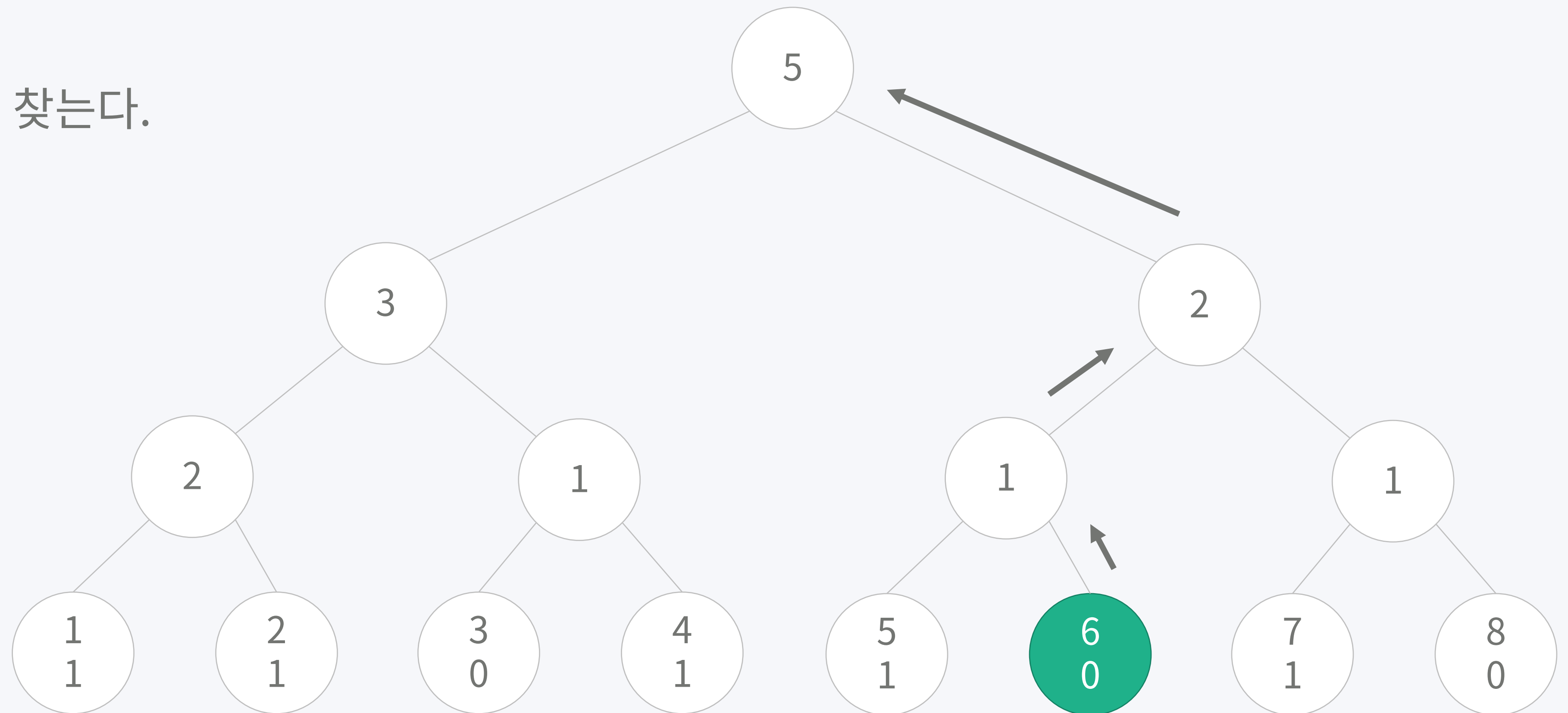


조세퍼스 문제

121

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~3까지 합 = 2
- $2+3=5$ 번째 위치를 찾는다.

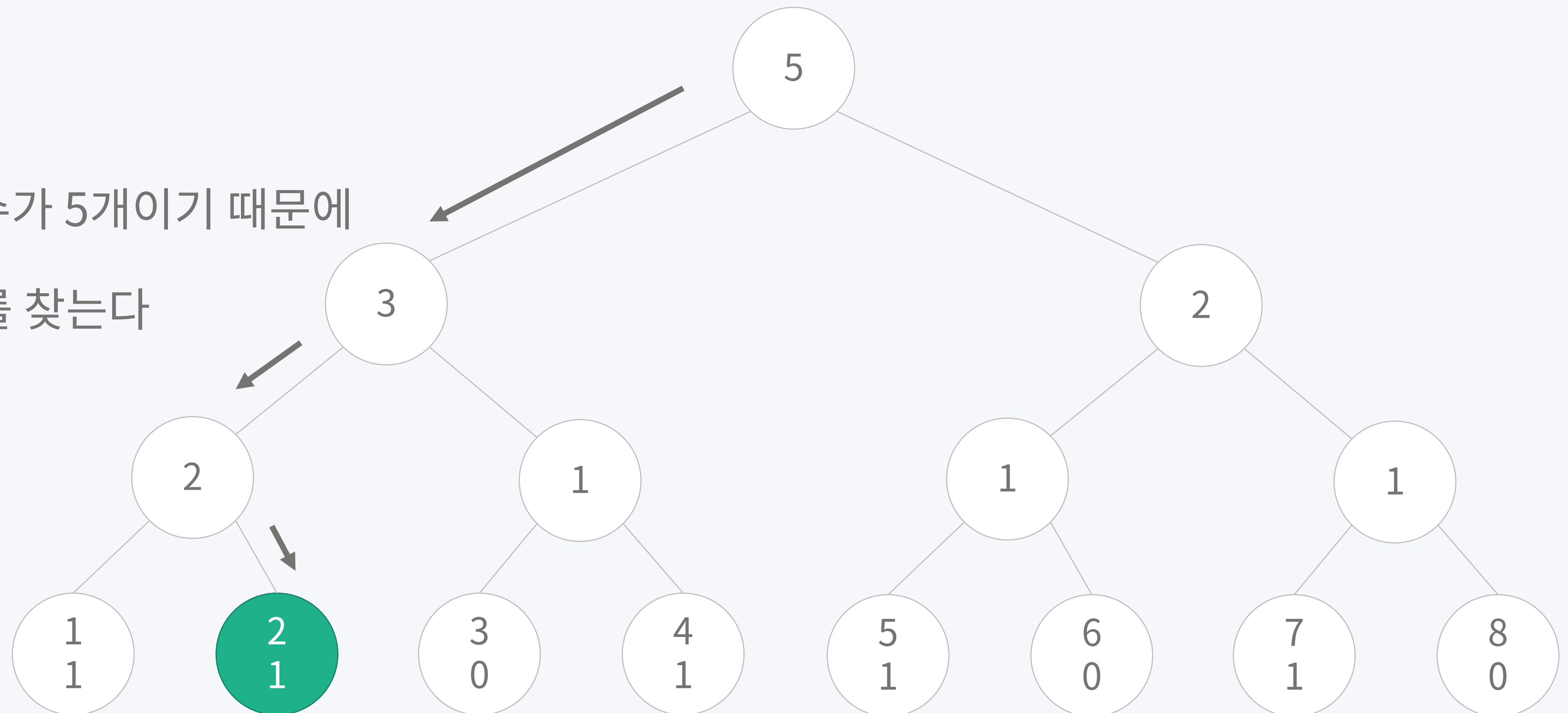


조세퍼스 문제

122

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~6까지 합 = 4
- $4 + 3 = 7$ 번째 위치
- 남아있는 수의 개수가 5개이기 때문에
- $7 - 5 = 2$ 번째 위치를 찾는다

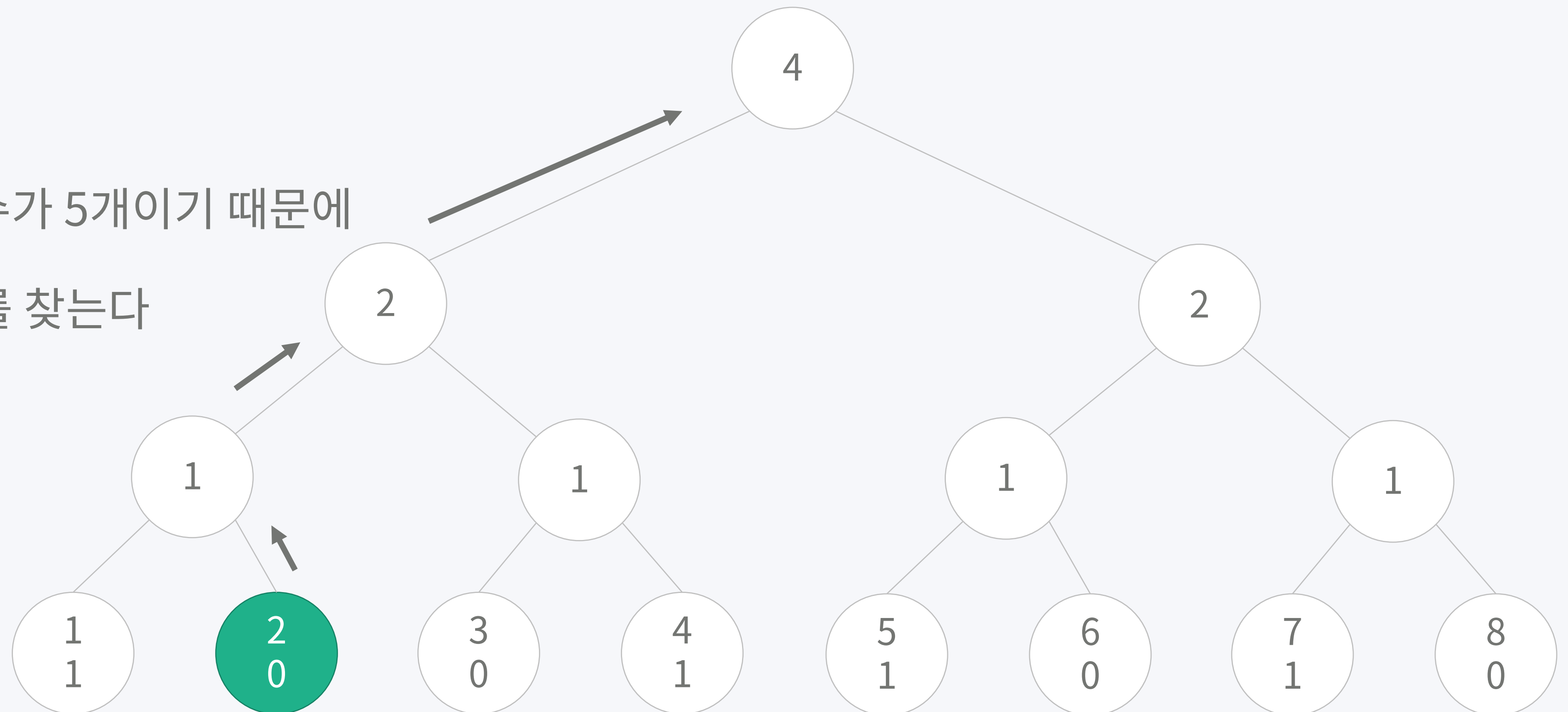


조세퍼스 문제

123

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~6까지 합 = 4
- $4 + 3 = 7$ 번째 위치
- 남아있는 수의 개수가 5개이기 때문에
- $7 - 5 = 2$ 번째 위치를 찾는다

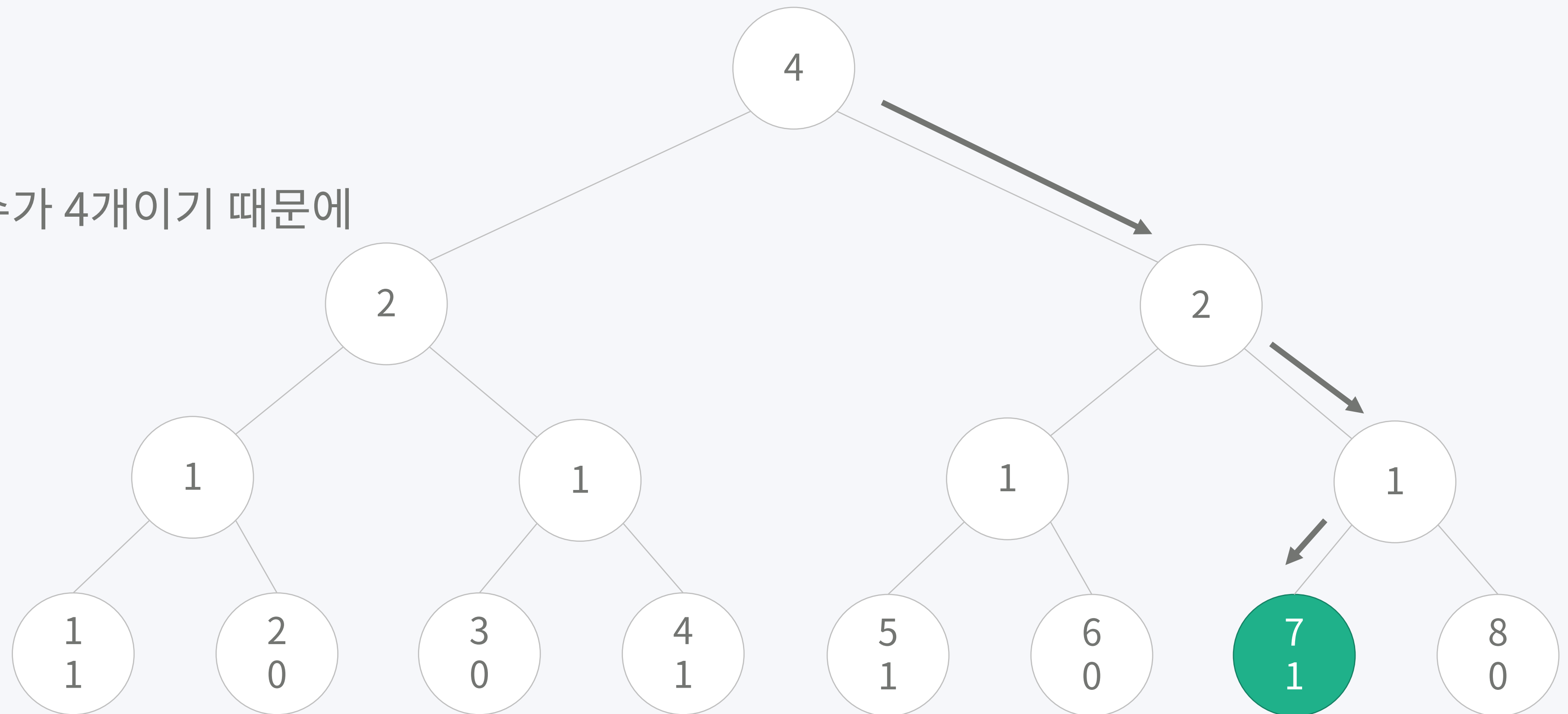


조세퍼스 문제

124

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~2까지 합 = 1
- $1+3 = 4$ 번째 위치
- 남아있는 수의 개수가 4개이기 때문에
- 찾을 수 있다.

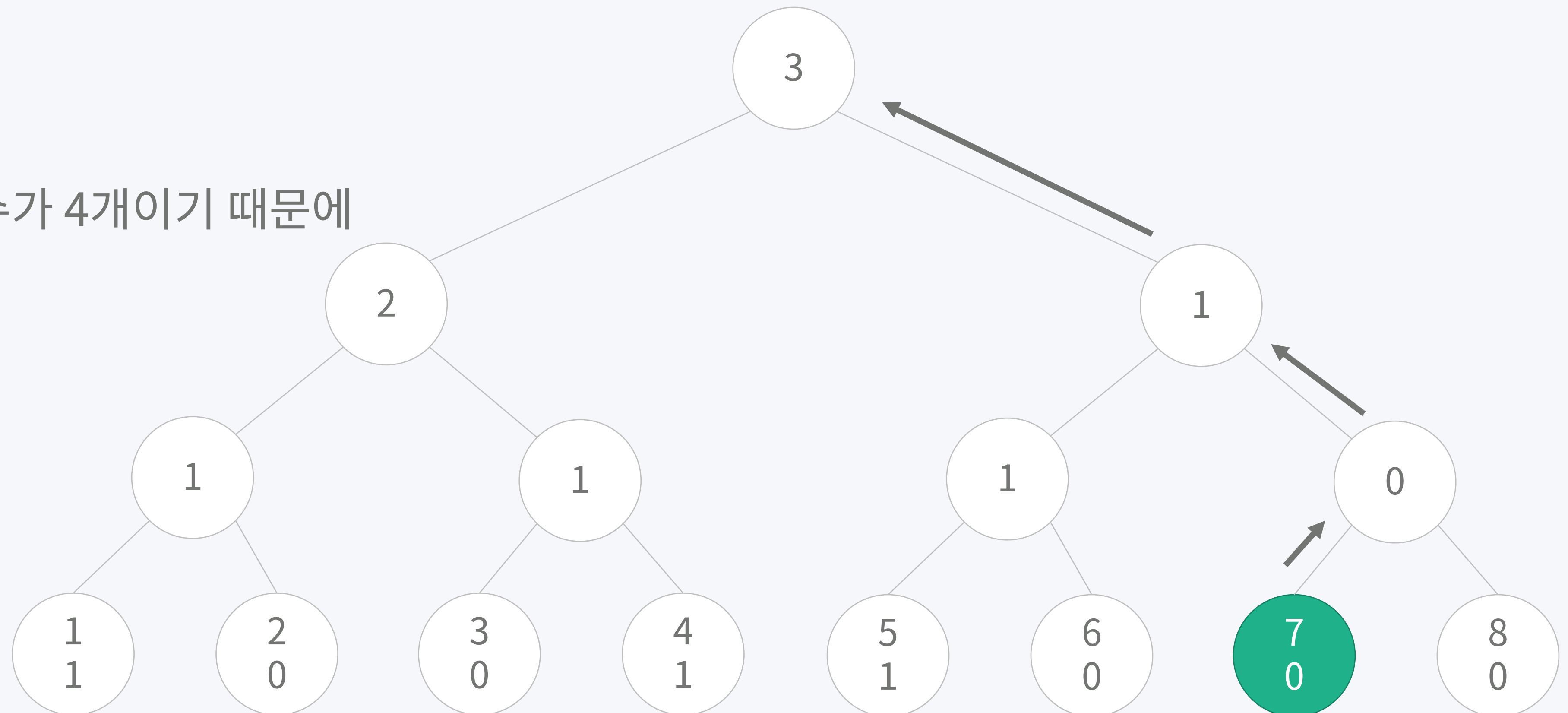


조세퍼스 문제

125

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~2까지 합 = 1
- $1+3 = 4$ 번째 위치
- 남아있는 수의 개수가 4개이기 때문에
- 찾을 수 있다.

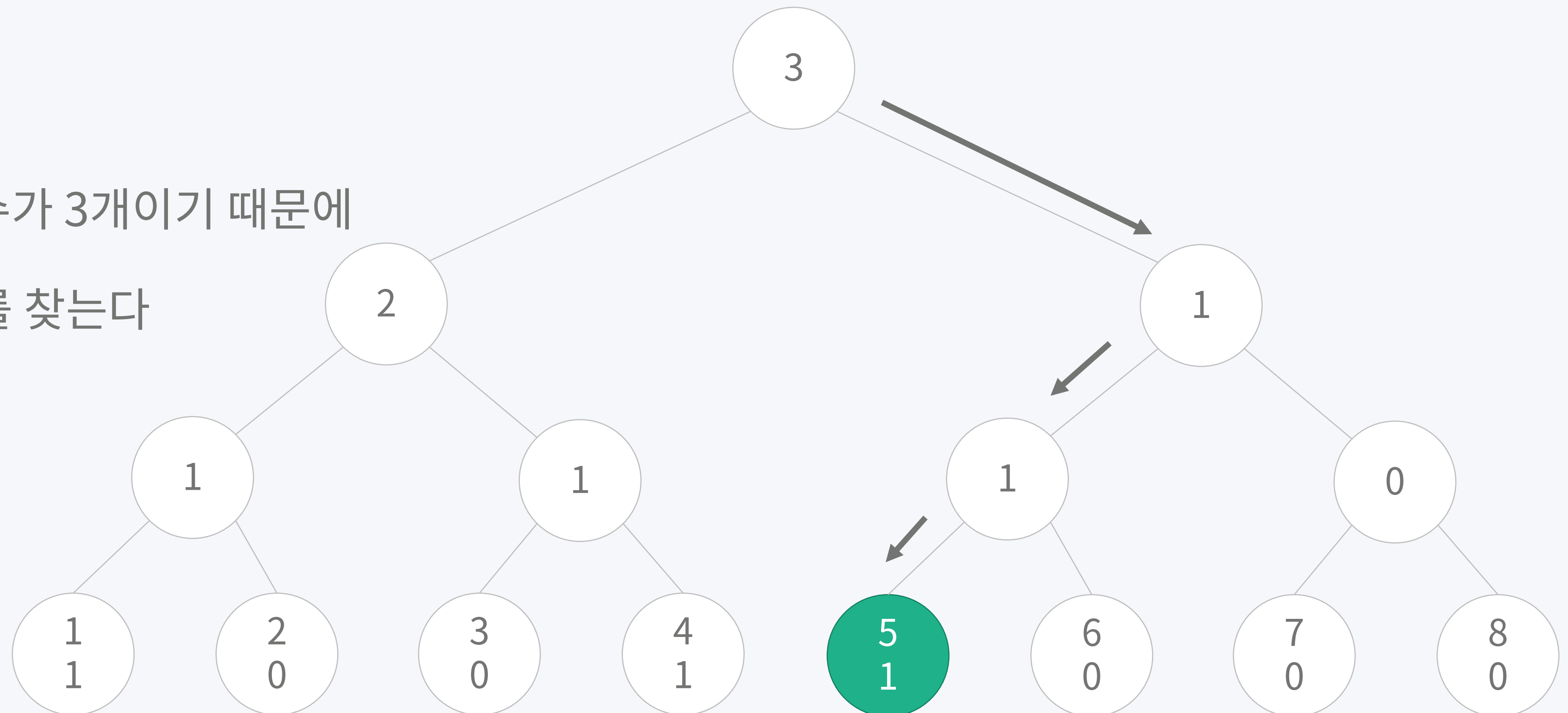


조세퍼스 문제

126

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~7까지 합 = 3
- $3+3 = 6$ 번째 위치
- 남아있는 수의 개수가 3개이기 때문에
- $6-3 = 3$ 번째 위치를 찾는다

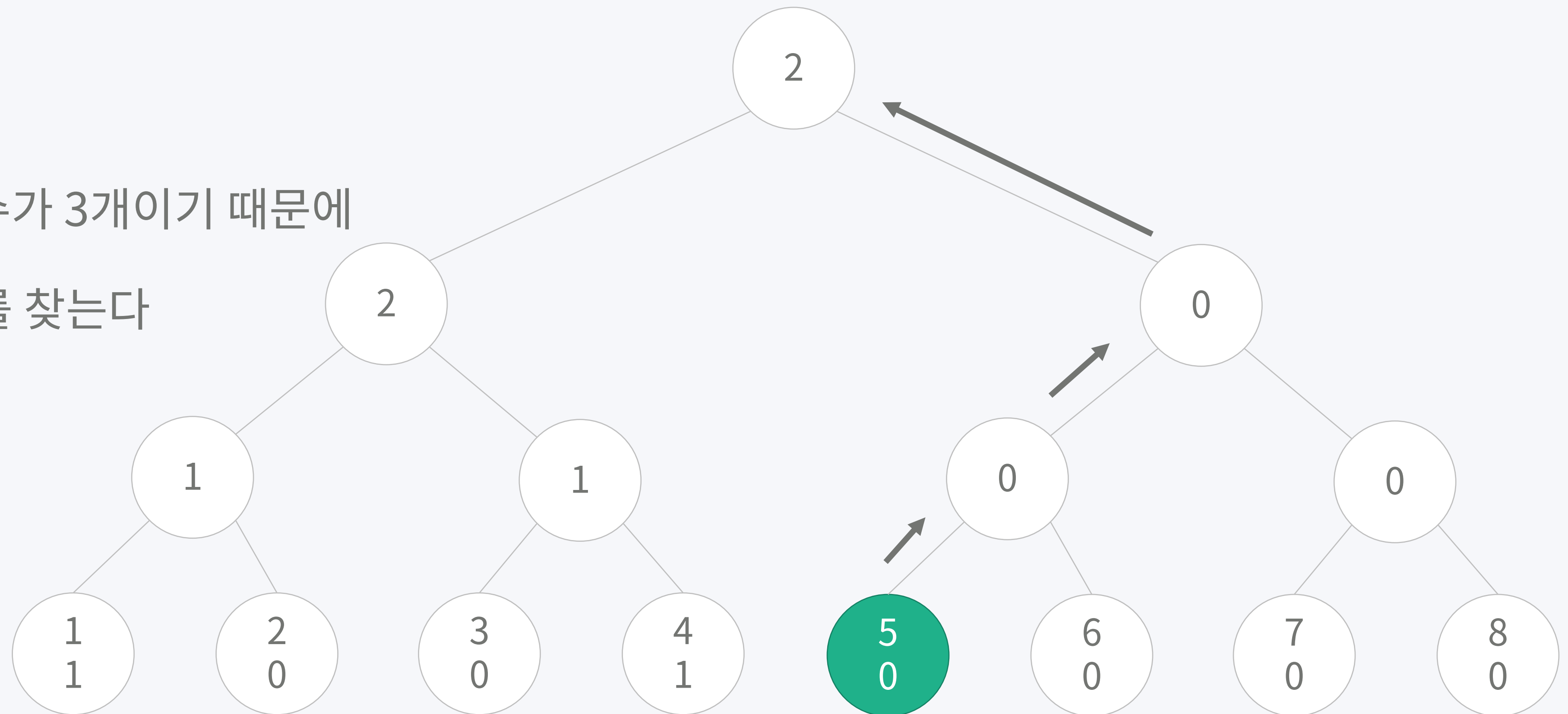


조세퍼스 문제

127

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~7까지 합 = 3
- $3+3 = 6$ 번째 위치
- 남아있는 수의 개수가 3개이기 때문에
- $6-3 = 3$ 번째 위치를 찾는다

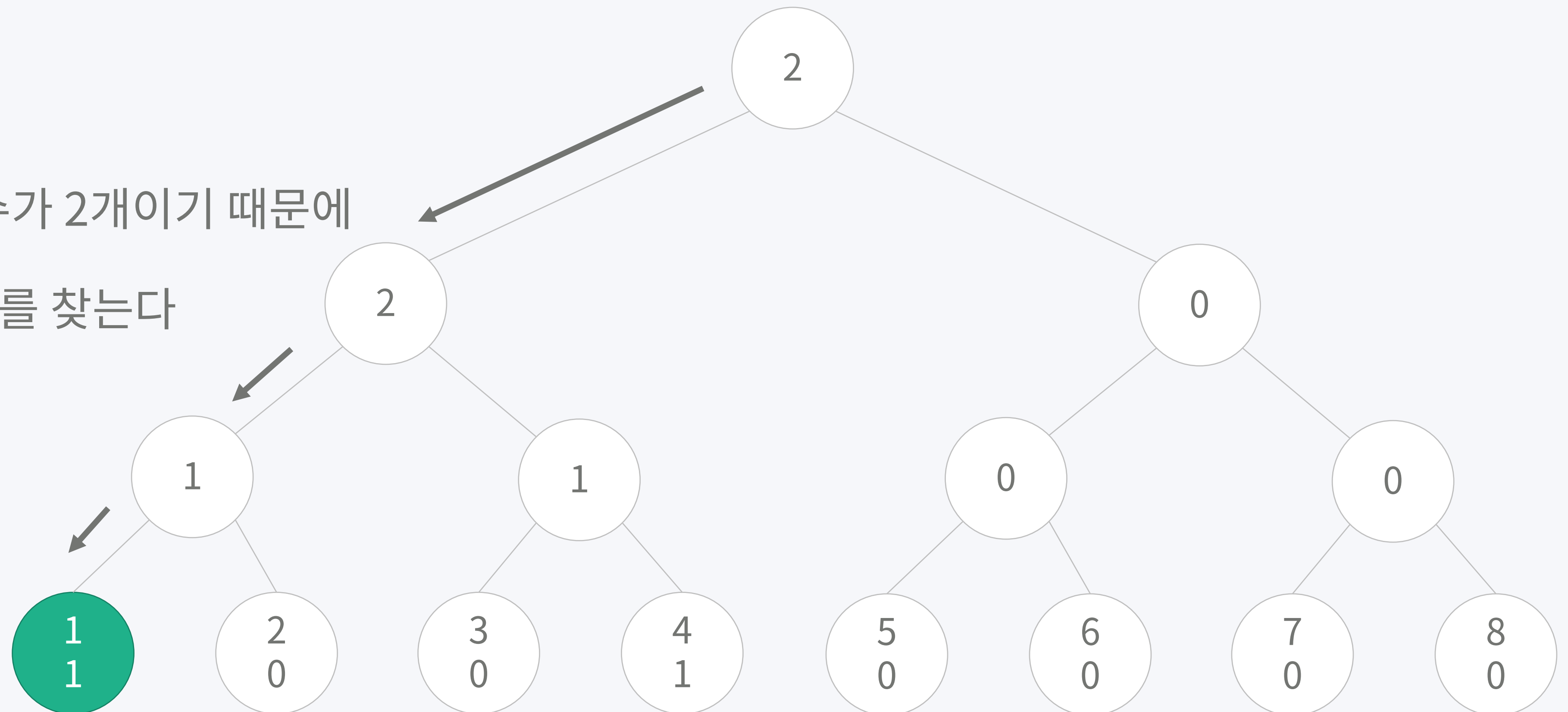


조세퍼스 문제

128

<https://www.acmicpc.net/problem/1168>

- $N = 7, K = 3$
- 1~5까지 합 = 2
- $2+3 = 5$ 번째 위치
- 남아있는 수의 개수가 2개이기 때문에
- $5\%2 = 1$ 번째 위치를 찾는다



조세퍼스 문제

129

<https://www.acmicpc.net/problem/1168>

- C/C++: <https://gist.github.com/Baekjoon/498c99f68e003ce3a236f5428d52f95f>