

세그먼트 트리와 펜윅 트리 (연습)

최백준 choi@startlink.io

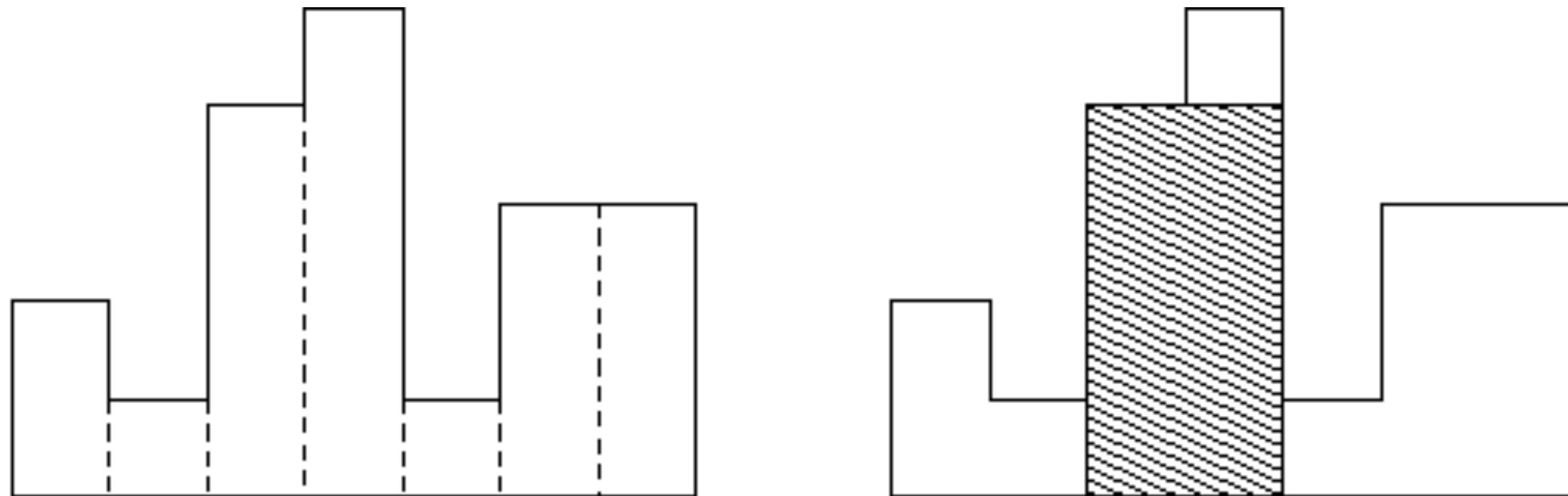
최소값 찾기

히스토그램에서 가장 큰 직사각형

3

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

- N개의 히스토그램이 주어졌을 때, 가장 큰 직사각형을 찾는 문제
- $1 \leq N \leq 100,000$

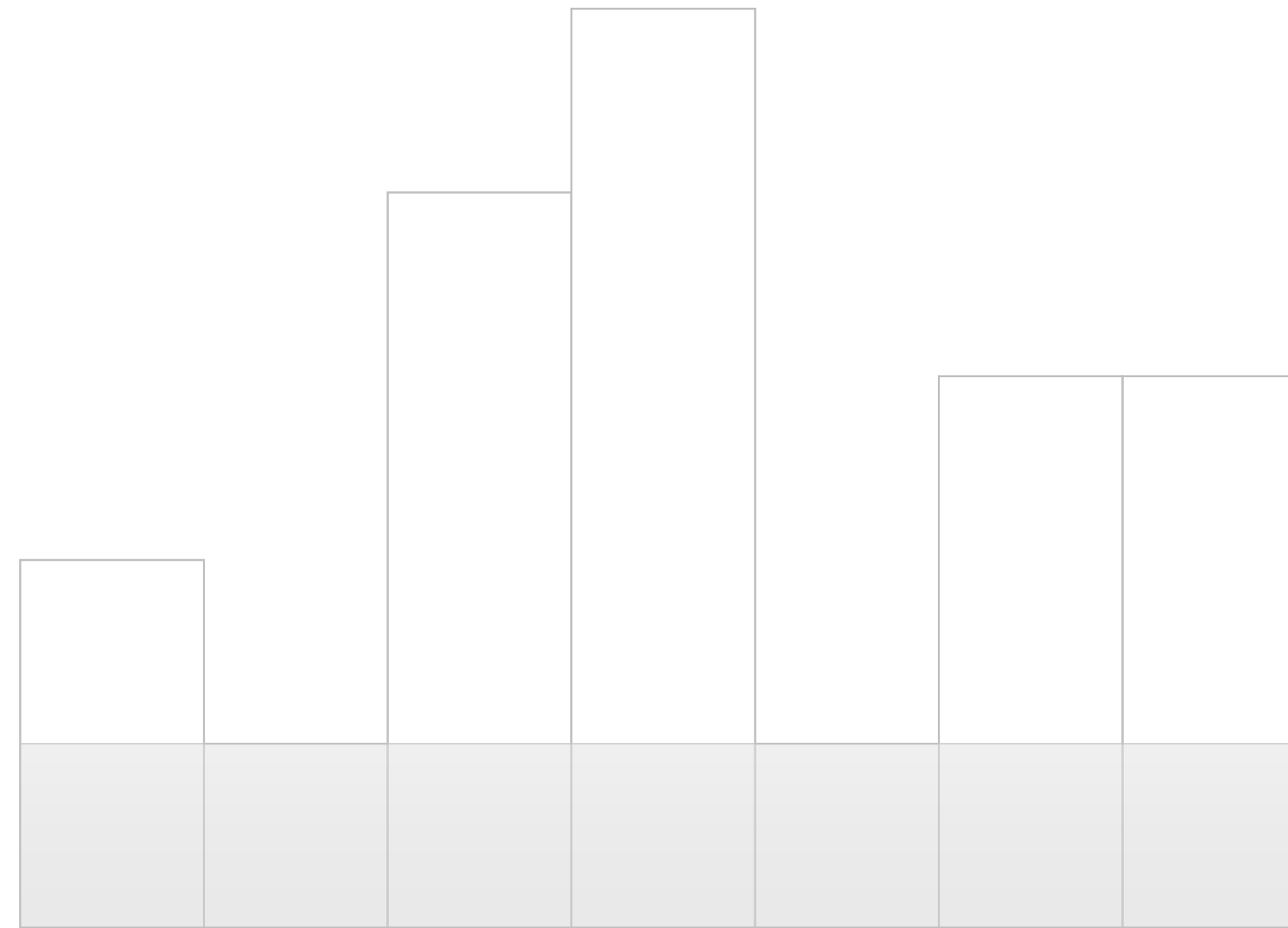


히스토그램에서 가장 큰 직사각형

4

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

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

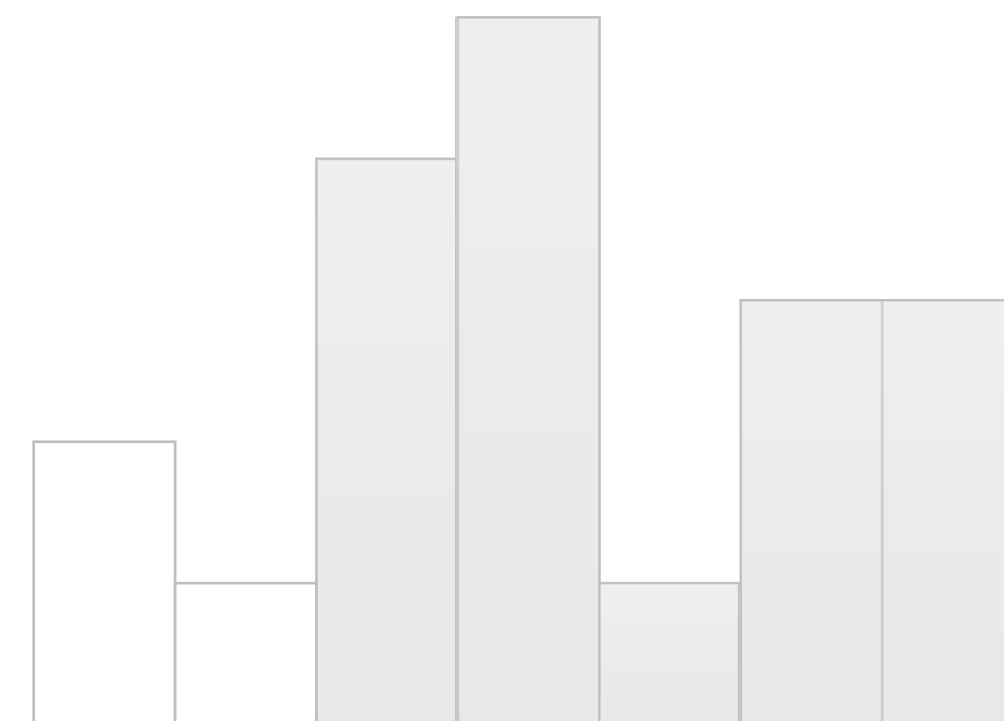
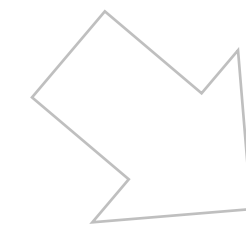
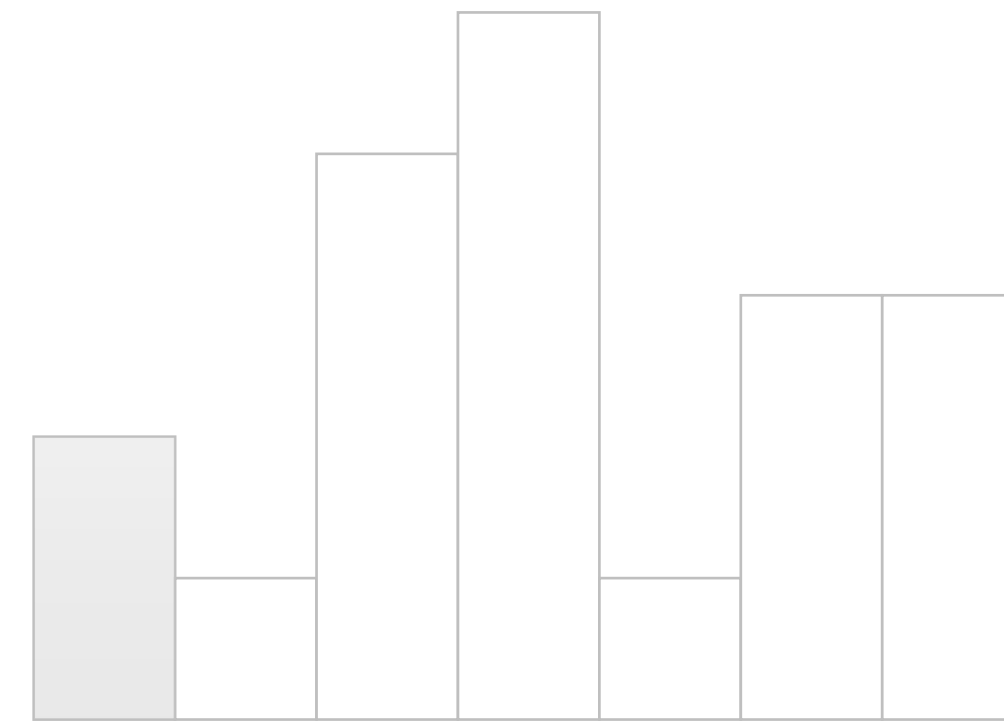
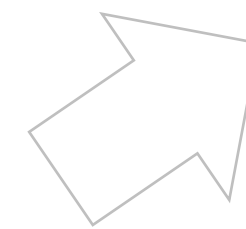
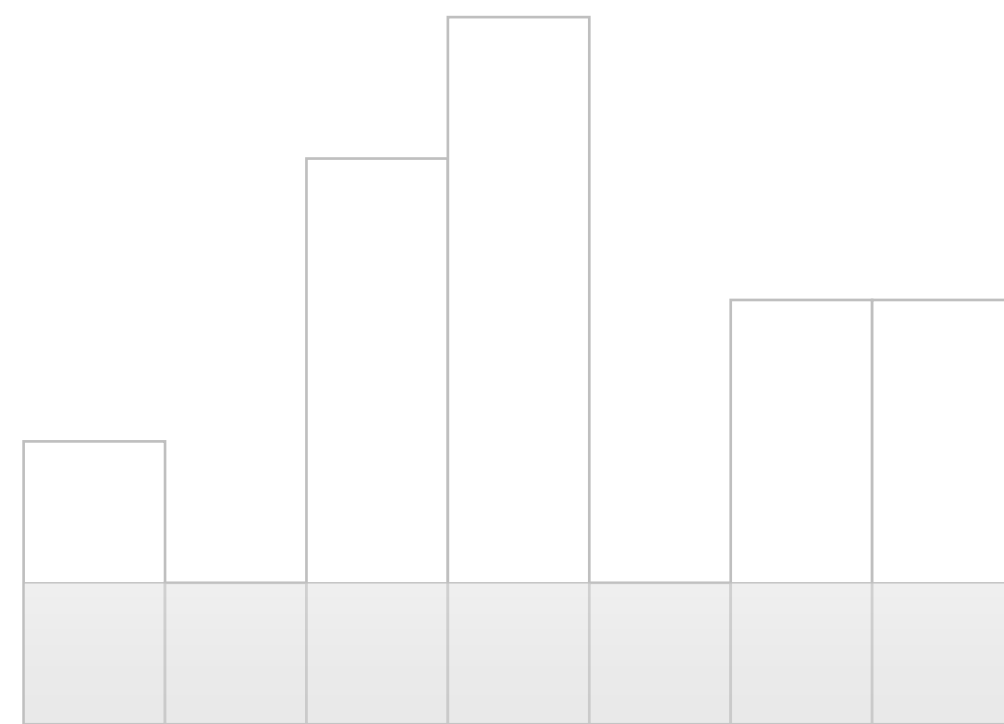


히스토그램에서 가장 큰 직사각형

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.

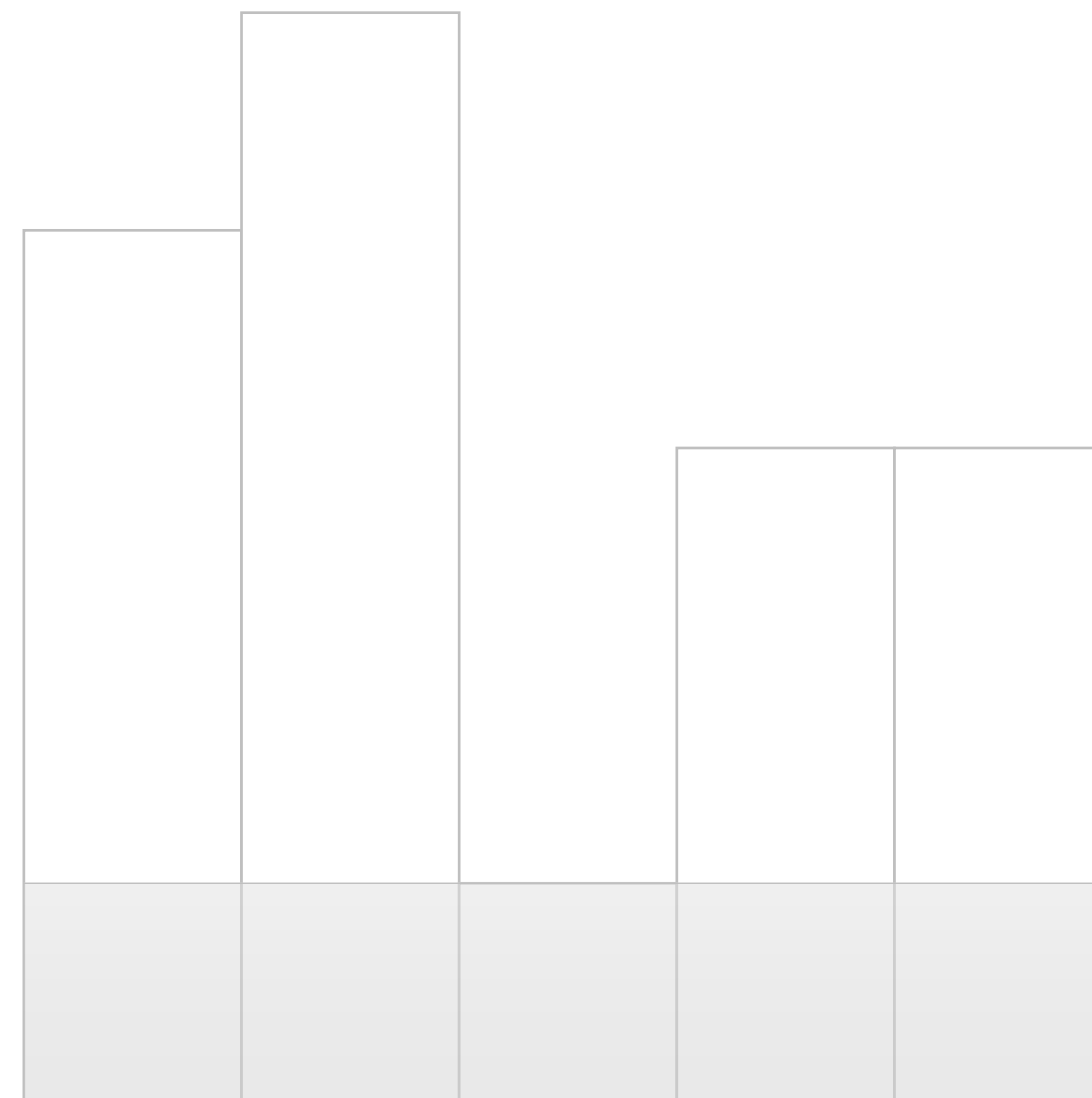


히스토그램에서 가장 큰 직사각형

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.

히스토그램에서 가장 큰 직사각형

9

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

- 소스: <http://codeplus.codes/a789f9ee9b1b48a48df70fbc15b4b163>

수열과 쿼리 24

10

<https://www.acmicpc.net/problem/17408>

- 길이가 N 인 수열 A 가 있고, 다음 두 가지 쿼리가 있다. $2 \leq N \leq 100,000$
 - 1 $i \ v: A_i$ 를 v 로 바꾼다
 - 2 $l \ r: l \leq i < j \leq r$ 을 만족하는 모든 $A_i + A_j$ 중에서 최댓값을 출력
- 쿼리 M 개를 수행해야 한다. $1 \leq M \leq 100,000$

수열과 쿼리 24

<https://www.acmicpc.net/problem/17408>

- 만약 쿼리가 다음과 같다면 세그먼트 트리를 이용할 수 있다.
- 길이가 N 인 수열 A 가 있고, 다음 두 가지 쿼리가 있다. $2 \leq N \leq 100,000$
 - 1 $i \ v: A_i$ 를 v 로 바꾼다
 - 2 $l \ r: l \leq i \leq r$ 을 만족하는 모든 A_i 중에서 최댓값을 출력
- 쿼리 M 개를 수행해야 한다. $1 \leq M \leq 100,000$
- 이를 이용해 원래 문제를 해결할 수 있다.

수열과 쿼리 24

<https://www.acmicpc.net/problem/17408>

- 세그먼트 트리에 최댓값이 아니고, 최댓값의 위치를 저장했다고 하자
- $l \leq i \leq r$ 에서 최댓값의 위치가 p 라고 한다면
- $l \leq j < p$ 에서 최댓값의 위치 p_l
- $p < j \leq r$ 에서 최댓값의 위치 p_r 을 찾고
- $\max(a[p] + a[p_l], a[p] + a[p_r])$ 을 비교해보면 된다.

수열과 쿼리 24

<https://www.acmicpc.net/problem/17408>

- 소스: <http://codeplus.codes/45e2a30db1ee444387d9e31a67e60e04>

합 구하기

버블 소트

<https://www.acmicpc.net/problem/1517>

- N개로 이루어진 수열 $A[1], A[2], \dots, A[N]$ 이 있을 때
- Swap이 총 몇 번 발생하는지 알아내는 문제
- $3\ 2\ 1 \rightarrow 2\ 3\ 1 \rightarrow 2\ 1\ 3 \rightarrow 1\ 2\ 3$ (3번)

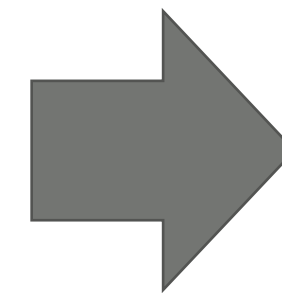
버블 소트

16

<https://www.acmicpc.net/problem/1517>

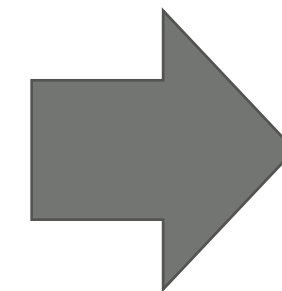
- $A = [10, 1, 5, 2, 3]$

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



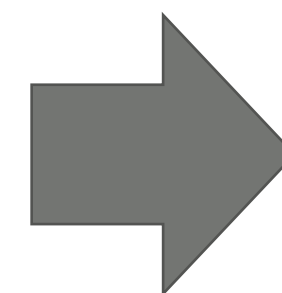
1	10	5	2	3
---	----	---	---	---

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



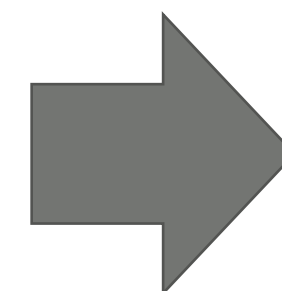
1	5	10	2	3
---	---	----	---	---

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



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

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



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

$i=1$

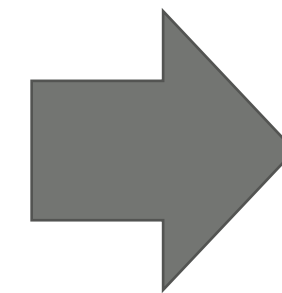
버블 소트

17

<https://www.acmicpc.net/problem/1517>

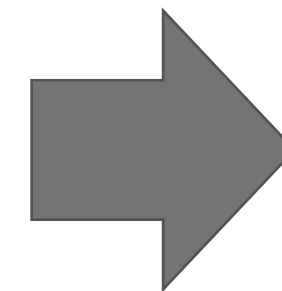
- $A = [10, 1, 5, 2, 3]$

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



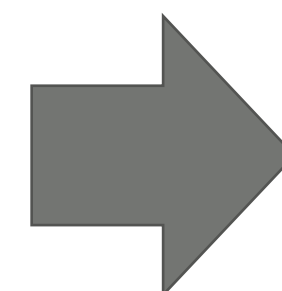
1	5	2	3	10
---	---	---	---	----

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



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

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



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

$i=2$

버블 소트

<https://www.acmicpc.net/problem/1517>

- 이 문제는 수열에서 inversion의 개수를 세는 문제이다.
- inversion: $A[i] > A[j]$ ($i < j$)
- 이 문제는 공장 문제와 같은 방식으로 해결할 수 있다.
- 공장: $A[i] < A[j]$ 이면서 $B[i] > B[j]$ 인 (i, j) 쌍의 개수를 찾는 문제
- 버블 소트: $i < j$ 이면서 $A[i] > A[j]$ 인 (i, j) 쌍의 개수를 찾는 문제

버블 소트

<https://www.acmicpc.net/problem/1517>

- 소스: <http://codeplus.codes/801d5dc3078d4d939bf4e3979be0b879>

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>

- 1단계

트리

1	1	1	1	1	1	1
---	---	---	---	---	---	---

입력된 배열

5	4	3	7	1	2	6
---	---	---	---	---	---	---

정렬된 배열

5	4	3	7	1	2	6
---	---	---	---	---	---	---

바뀐 배열

1	5	4	3	7	2	6
---	---	---	---	---	---	---

터보소트

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

- 2단계

트리

1	1	1	1	0	1	1
---	---	---	---	---	---	---

입력된 배열

5	4	3	7	1	2	6
---	---	---	---	---	---	---

정렬된 배열

1	5	4	3	7	2	6
---	---	---	---	---	---	---

바뀐 배열

1	5	4	3	2	6	7
---	---	---	---	---	---	---

터보소트

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

- 3단계

트리

1	1	1	0	0	1	1
---	---	---	---	---	---	---

입력된 배열

5	4	3	7	1	2	6
---	---	---	---	---	---	---

정렬된 배열

1	5	4	3	2	6	7
---	---	---	---	---	---	---

바뀐 배열

1	2	5	4	3	6	7
---	---	---	---	---	---	---

터보소트

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

- 소스: <http://codeplus.codes/5436ffd159aa4417a56543c40f92b358>

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

26

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

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

소요 시간	8	1	3	6	3	8
----------	---	---	---	---	---	---

라운드 로빈 스케줄러

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

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

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

소요 시간	8	1	3	6	3	8
완료 순서	5	1	2	4	3	6

- 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

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

32

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

- 소스: <http://codeplus.codes/994d65465f7648819bcc4a872c4faa58>

나무 심기

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

- 1번부터 N번까지 번호가 매겨져 있는 N개의 나무가 있다
- i번 나무는 좌표 $X[i]$ 에 심는다
- 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

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
- $\text{dist}[i] = \text{cnt}[i] \times i$ (i 에 심어져 있는 좌표의 합)

나무 심기

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

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

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
- $\text{sum}(\text{dist}, x[i]+1, L) - CR \times x[i]$

나무 심기

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

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

나무 심기

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

- 소스: <http://codeplus.codes/6e19da105c434400807a014c9f00ed2e>

불만 정렬

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

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

불만 정렬

<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

42

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

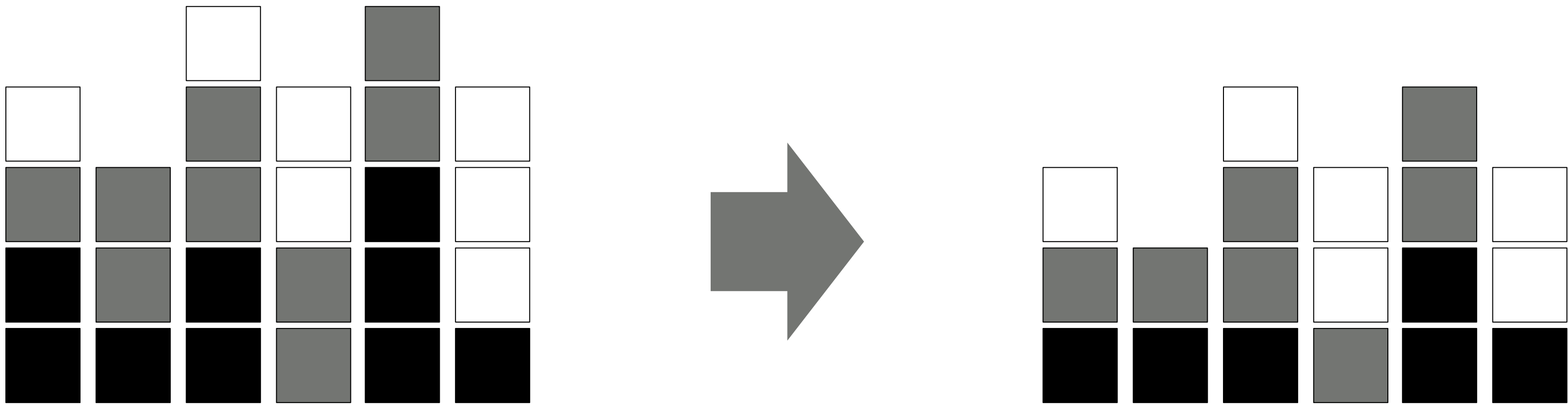
- 소스: <http://codeplus.codes/1201872e4195453aba622046bd36546c>

<https://www.acmicpc.net/problem/1273>

- 맥주캔은 검은색, 회색, 흰색으로 총 3가지 종류가 있고, N개의 열에 쌓았다. $1 \leq N \leq 30$ 만
- 쌓은 순서는 검은색, 회색, 흰색의 순서이다.
- 높이를 정해서 공기총을 발사하면, 해당 높이의 캔이 모두 바깥으로 떨어지고, 위에 있는 캔이 그대로 한 칸 내려온다.
- 검은색은 1점, 회색은 2점, 흰색은 5점
- 각 높이를 쏠 때마다 얻을 수 있는 점수를 구하는 문제
- 공기총의 발사 횟수 $\leq 300,000$
- 모든 수 $\leq 10^6$

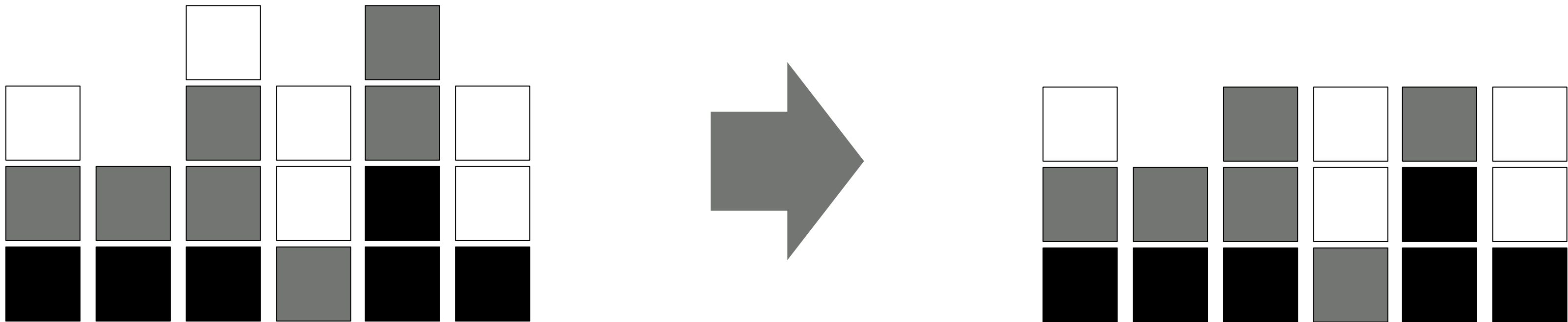
<https://www.acmicpc.net/problem/1273>

- 2번 높이를 총으로 쓴 경우 (점수: 12)



<https://www.acmicpc.net/problem/1273>

- 4번 높이를 총으로 쓴 경우 (점수: 7)



<https://www.acmicpc.net/problem/1273>

- $B[i]$, $G[i]$, $W[i]$ 를 i 번 열에 있는 검정색, 회색, 흰색 캔의 개수
- $X_i[h]$ = i 번 열의 높이 h 에 있는 캔의 점수
- $X[h]$ = 높이 h 에 총을 발사했을 때 얻는 점수
- $X[h] = \sum X_i[h]$ ($1 \leq i \leq N$) 으로 구할 수 있다.
- 매번 $X[h]$ 를 구하는 것은 너무 비효율적이다.

<https://www.acmicpc.net/problem/1273>

- $B[i]$, $G[i]$, $W[i]$ 를 i 번 열에 있는 검정색, 회색, 흰색 캔의 개수
- $X_i[h]$ = i 번 열의 높이 h 에 있는 캔의 점수
- $X[h]$ = 높이 h 에 총을 발사했을 때 얻는 점수
- $X[h] = \sum X_i[h]$ ($1 \leq i \leq N$) 으로 구할 수 있다.
- $Y_i[h] = X_i[h] - X_i[h-1]$
- $Y_i[h]$ 는 0을 제외하면 4가지 서로 다른 값을 가질 수 있다.
 - 검회흰: 1, 3
 - 검흰: 1, 4
 - 회흰: 2, 3
 - 흰: 5

<https://www.acmicpc.net/problem/1273>

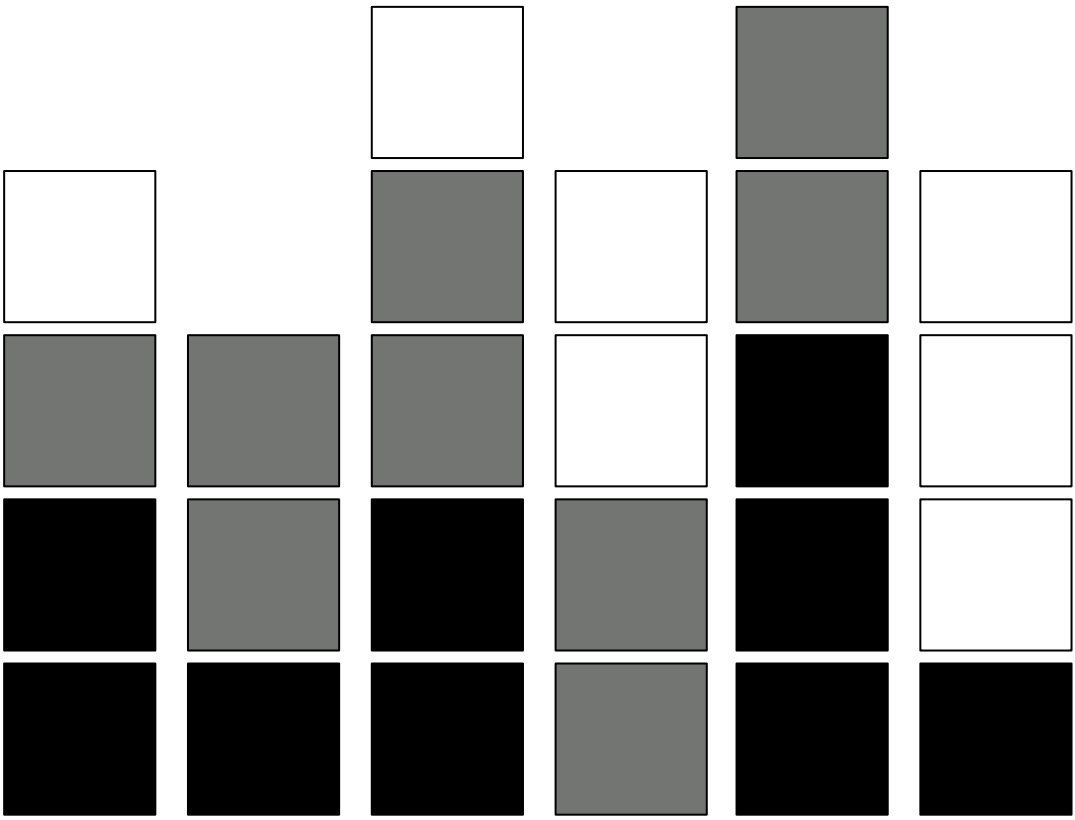
- $B[i]$, $G[i]$, $W[i]$ 를 i 번 열에 있는 검정색, 회색, 흰색 캔의 개수
- $X_i[h]$ = i 번 열의 높이 h 에 있는 캔의 점수
- $X[h]$ = 높이 h 에 총을 발사했을 때 얻는 점수
- $X[h] = \sum X_i[h]$ ($1 \leq i \leq N$) 으로 구할 수 있다.
- $Y_i[h] = X_i[h] - X_i[h-1]$
- $X_i[h] = \sum Y_i[j]$ ($1 \leq j \leq h$)
- $Y[h] = \sum Y_i[h]$ ($1 \leq i \leq N$)라면 $Y[h]$ 를 효율적으로 구할 수 있다.
- $Y_i[h]$ 의 값이 변하는 i 는 1 , $B[i]+1$, $B[i]+G[i]+1$, $B[i]+G[i]+W[i]+1$ 이기 때문

<https://www.acmicpc.net/problem/1273>

- $B[i], G[i], W[i]$ 를 i 번 열에 있는 검정색, 회색, 흰색 캔의 개수
- $X_i[h]$ = i 번 열의 높이 h 에 있는 캔의 점수
- $X[h]$ = 높이 h 에 총을 발사했을 때 얻는 점수
- $X[h] = \sum X_i[h] \ (1 \leq i \leq N)$ 으로 구할 수 있다.
- $Y_i[h] = X_i[h] - X_i[h-1]$
- $X_i[h] = \sum Y_i[j] \ (1 \leq j \leq h)$
- $Y[h] = \sum Y_i[h] \ (1 \leq i \leq N)$
- $X[h] = \sum X_i[h] \ (1 \leq i \leq N) = \sum \sum Y_i[j] \ (1 \leq i \leq N) \ (1 \leq j \leq h) = \sum Y[j] \ (1 \leq j \leq h) = X[h-1] + Y[h]$



<https://www.acmicpc.net/problem/1273>



i	1	2	3	4	5	6
B[i]	2	1	2	0	3	1
G[i]	1	2	2	2	2	0
W[i]	1	0	1	2	0	3

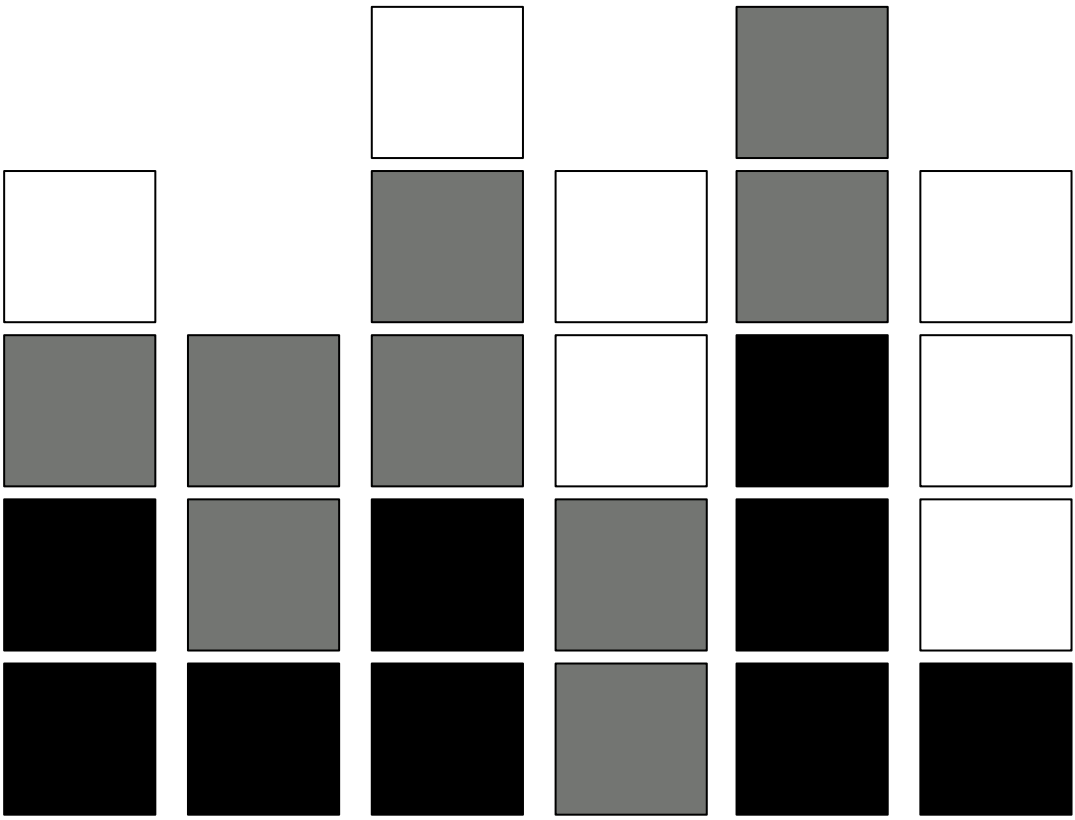
h	1	2	3	4	5
$X_1[h]$	1	1	2	5	
$X_2[h]$	1	2	2		
$X_3[h]$	1	1	2	2	5
$X_4[h]$	2	2	5	5	
$X_5[h]$	1	1	1	2	2
$X_6[h]$	1	5	5	5	
$X[h]$	7	12	17	19	7

h	1	2	3	4	5	6
$Y_1[h]$	1	0	1	3	-5	0
$Y_2[h]$	1	1	0	-2	0	0
$Y_3[h]$	1	0	1	0	3	-5
$Y_4[h]$	2	0	3	0	-5	0
$Y_5[h]$	1	0	0	1	0	-2
$Y_6[h]$	1	4	0	0	-5	0
$Y[h]$	7	5	5	2	-12	-7



<https://www.acmicpc.net/problem/1273>

• $i = 1$



i	1	2	3	4	5	6
B[i]	2	1	2	0	3	1
G[i]	1	2	2	2	2	0
W[i]	1	0	1	2	0	3

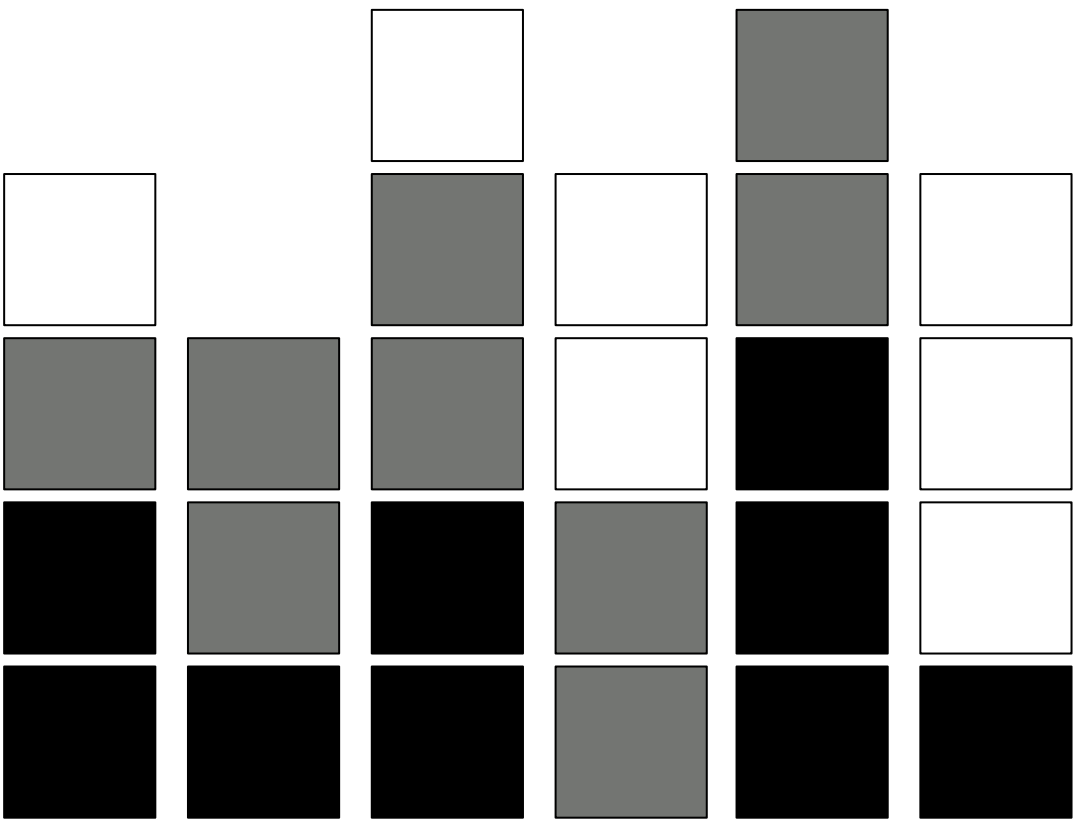
h	1	2	3	4	5
$X_1[h]$	1	1	2	5	
$X_2[h]$	1	2	2		
$X_3[h]$	1	1	2	2	5
$X_4[h]$	2	2	5	5	
$X_5[h]$	1	1	1	2	2
$X_6[h]$	1	5	5	5	
$X[h]$	7	12	17	19	7

h	1	2	3	4	5	6
$Y_1[h]$	1	0	1	3	-5	0
$Y_2[h]$	1	1	0	-2	0	0
$Y_3[h]$	1	0	1	0	3	-5
$Y_4[h]$	2	0	3	0	-5	0
$Y_5[h]$	1	0	0	1	0	-2
$Y_6[h]$	1	4	0	0	-5	0
$Y[h]$	7	5	5	2	-12	-7



<https://www.acmicpc.net/problem/1273>

• $i = 4$



i	1	2	3	4	5	6
B[i]	2	1	2	0	3	1
G[i]	1	2	2	2	2	0
W[i]	1	0	1	2	0	3

h	1	2	3	4	5
$X_1[h]$	1	1	2	5	
$X_2[h]$	1	2	2		
$X_3[h]$	1	1	2	2	5
$X_4[h]$	2	2	5	5	
$X_5[h]$	1	1	1	2	2
$X_6[h]$	1	5	5	5	
$X[h]$	7	12	17	19	7

h	1	2	3	4	5	6
$Y_1[h]$	1	0	1	3	-5	0
$Y_2[h]$	1	1	0	-2	0	0
$Y_3[h]$	1	0	1	0	3	-5
$Y_4[h]$	2	0	3	0	-5	0
$Y_5[h]$	1	0	0	1	0	-2
$Y_6[h]$	1	4	0	0	-5	0
$Y[h]$	7	5	5	2	-12	-7

삿

<https://www.acmicpc.net/problem/1273>

- $Y[i]$ 를 만들고, 이를 이용해 $X[i]$ 를 만들 수 있다.
- 트리를 이용해 원하는 높이를 찾을 수 있다.
- $tree[i] = i$ 번째 높이가 있으면 1, 없으면 0



<https://www.acmicpc.net/problem/1273>

- 소스: <http://codeplus.codes/dc96ed5846e54169a19bc03ddc91b4b2>

K번째 찾기

중앙값 측정

56

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

- N개의 수가 주어졌을 때, 길이가 K인 연속 부분 수열 N-K+1개의 중앙값을 구하는 문제
- $1 \leq N \leq 250,000$
- $0 \leq \text{수} \leq 65535$

중앙값 측정

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

- 세그먼트 트리를 만들고

중앙값 측정

58

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

- N개의 수가 주어졌을 때, 길이가 K인 연속 부분 수열 N-K+1개의 중앙값을 구하는 문제
- N = 7, K = 3인 경우

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
------	------	------	------	------	------	------

트리에 넣음

중앙값 측정

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

- N개의 수가 주어졌을 때, 길이가 K인 연속 부분 수열 N-K+1개의 중앙값을 구하는 문제
- N = 7, K = 3인 경우

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
빼고	그대로		넣고			

중앙값 측정

60

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

- N개의 수가 주어졌을 때, 길이가 K인 연속 부분 수열 N-K+1개의 중앙값을 구하는 문제
- N = 7, K = 3인 경우

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
	빼고	그대로	넣고			

중앙값 측정

61

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

- N개의 수가 주어졌을 때, 길이가 K인 연속 부분 수열 N-K+1개의 중앙값을 구하는 문제
- N = 7, K = 3인 경우

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
		빼고	그대로	넣고		

중앙값 측정

62

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

- N개의 수가 주어졌을 때, 길이가 K인 연속 부분 수열 N-K+1개의 중앙값을 구하는 문제
- N = 7, K = 3인 경우

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
			빼고	그대로	넣고	

중앙값 측정

63

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

- 소스: <http://codeplus.codes/263aaea9722b4948a15260a619f3a6bc>

요세푸스 문제 2

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

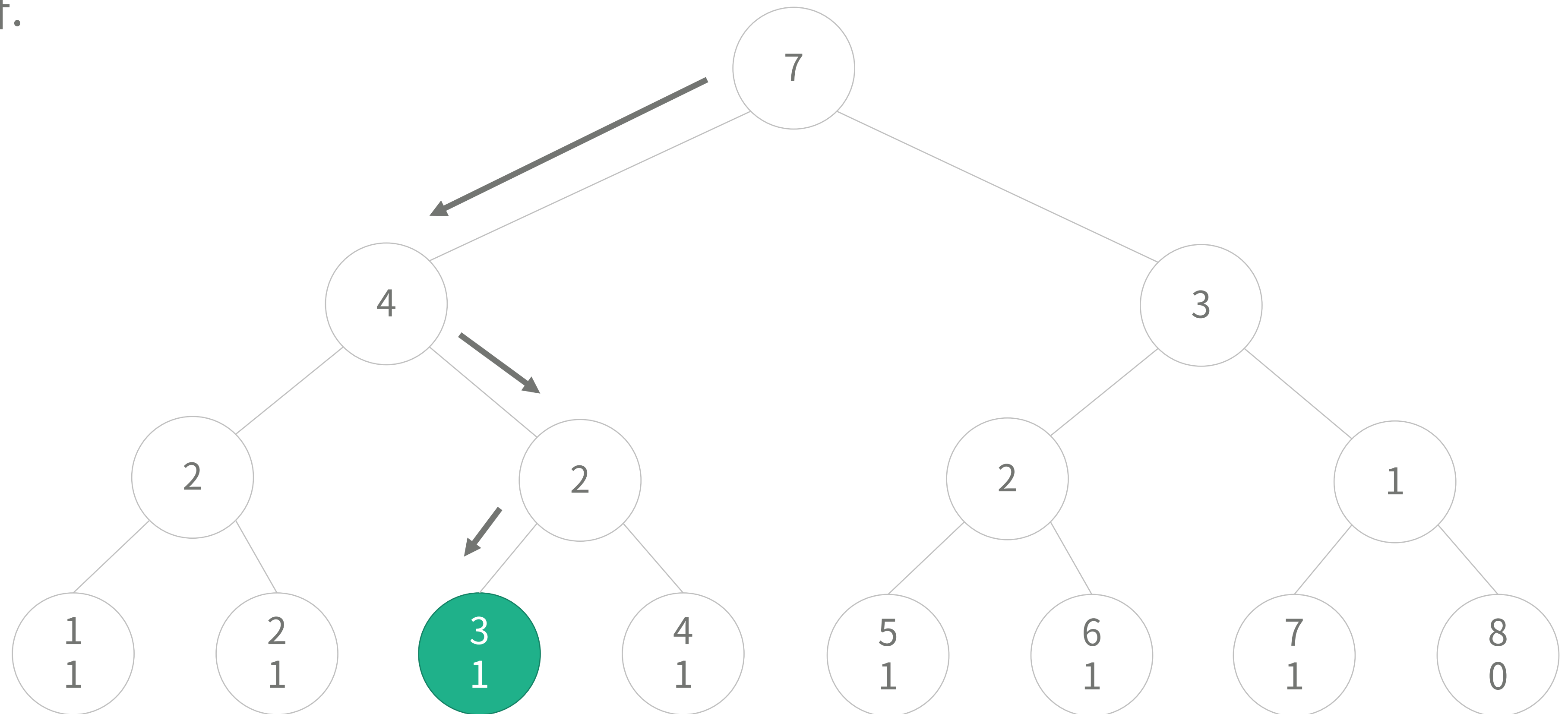
- N과 K가 주어졌을 때, 요세푸스 순열을 구하는 문제
- $N = 7, K = 3$
- $\langle 3, 6, 2, 7, 5, 1, 4 \rangle$

요세푸스 문제 2

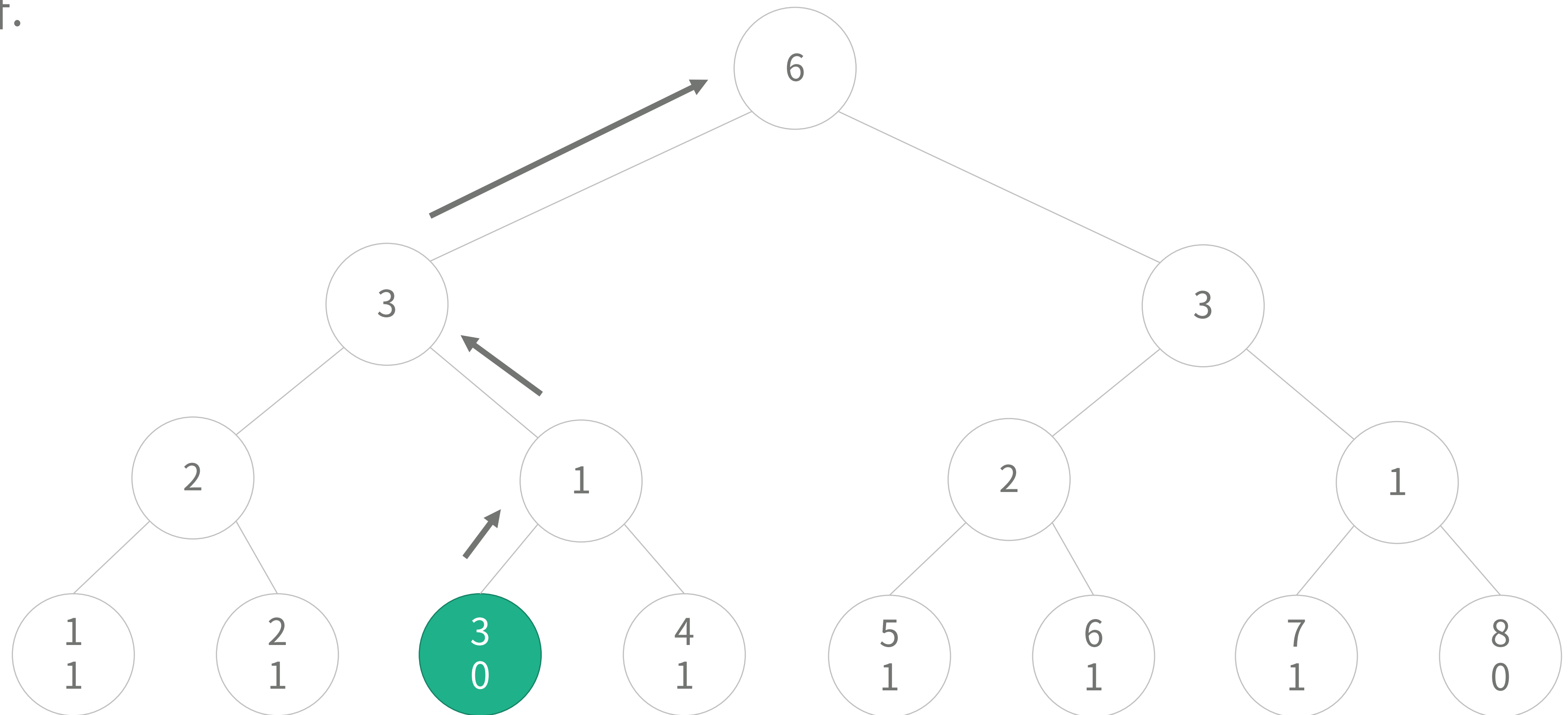
65

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

- $N = 7, K = 3$
- 3번째 위치를 찾는다.



- $N = 7, K = 3$
- 3번째 위치를 찾는다.

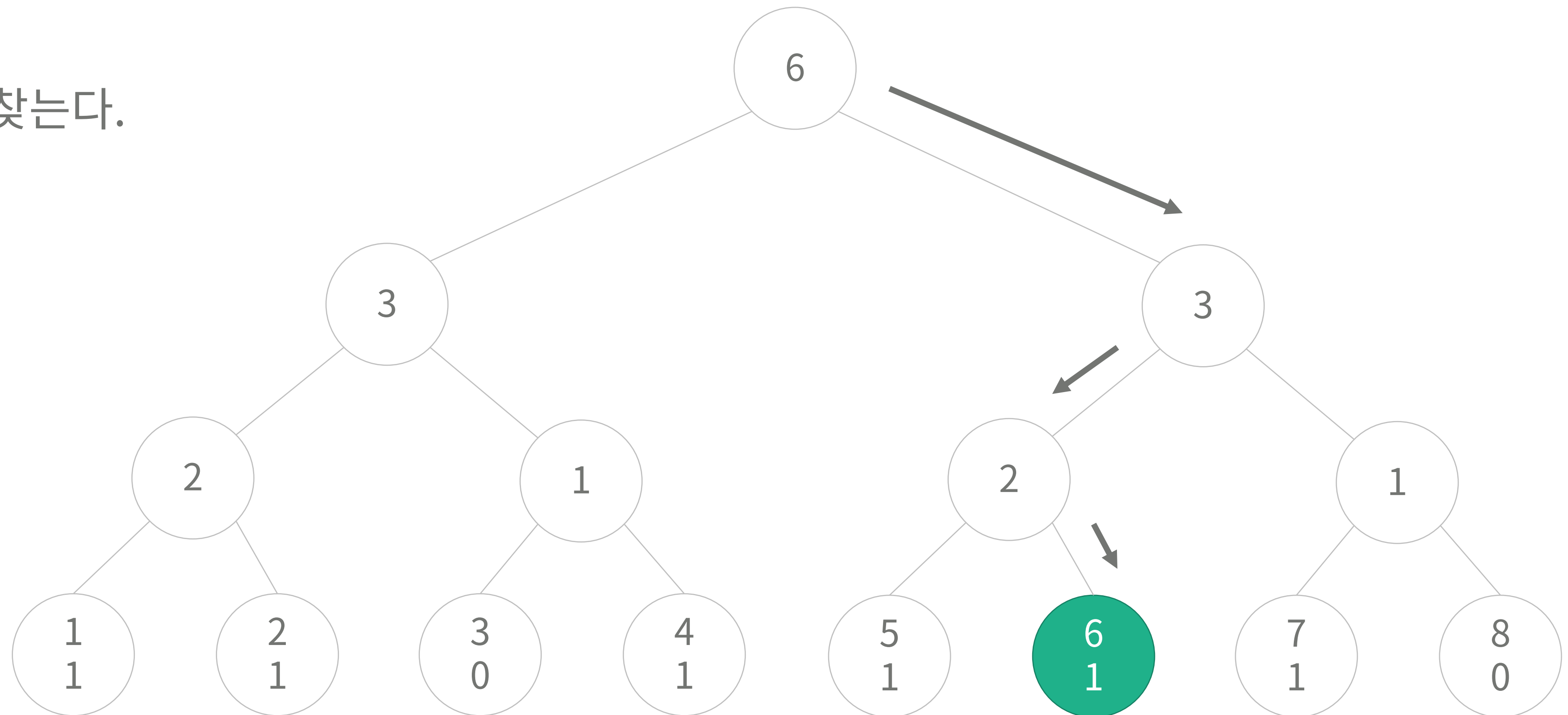


요세푸스 문제 2

67

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

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

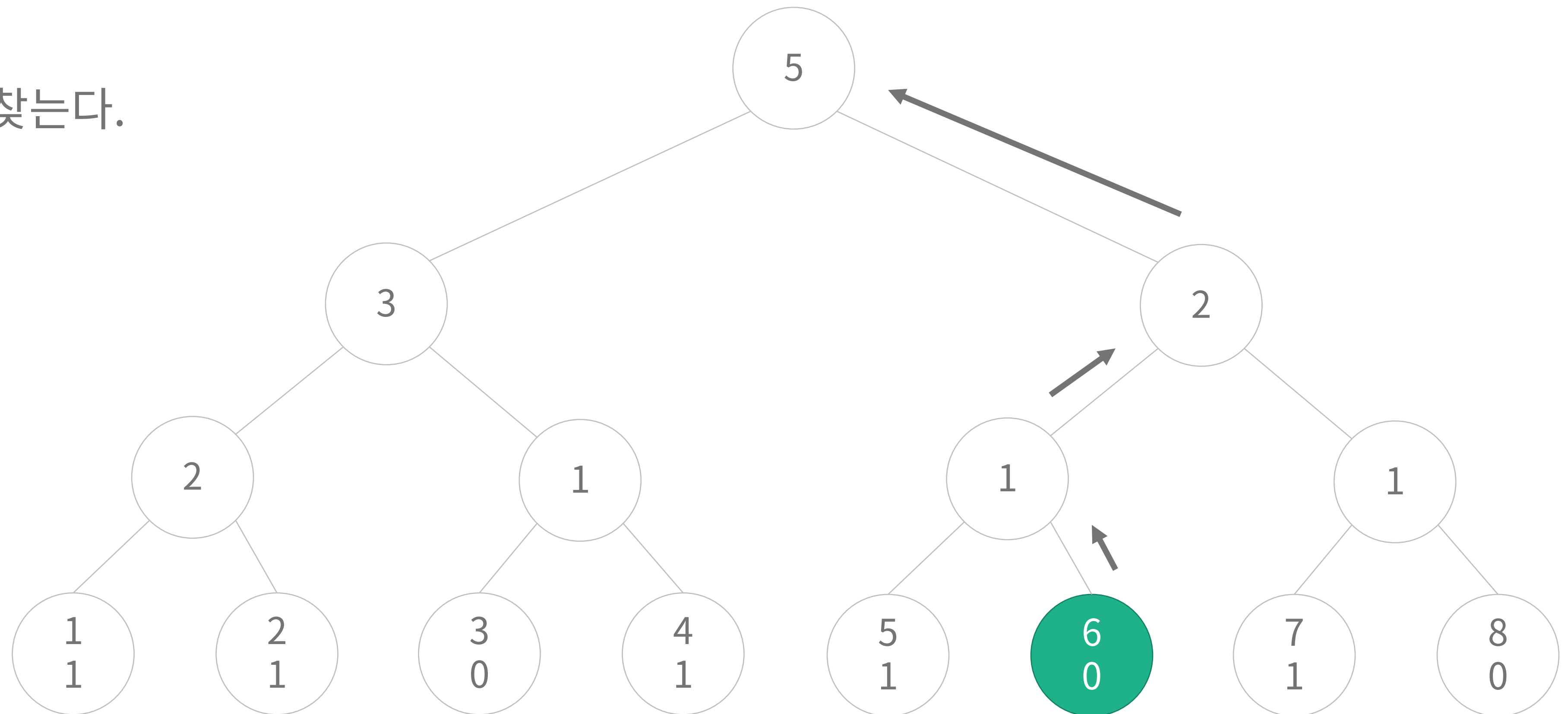


요세푸스 문제 2

68

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

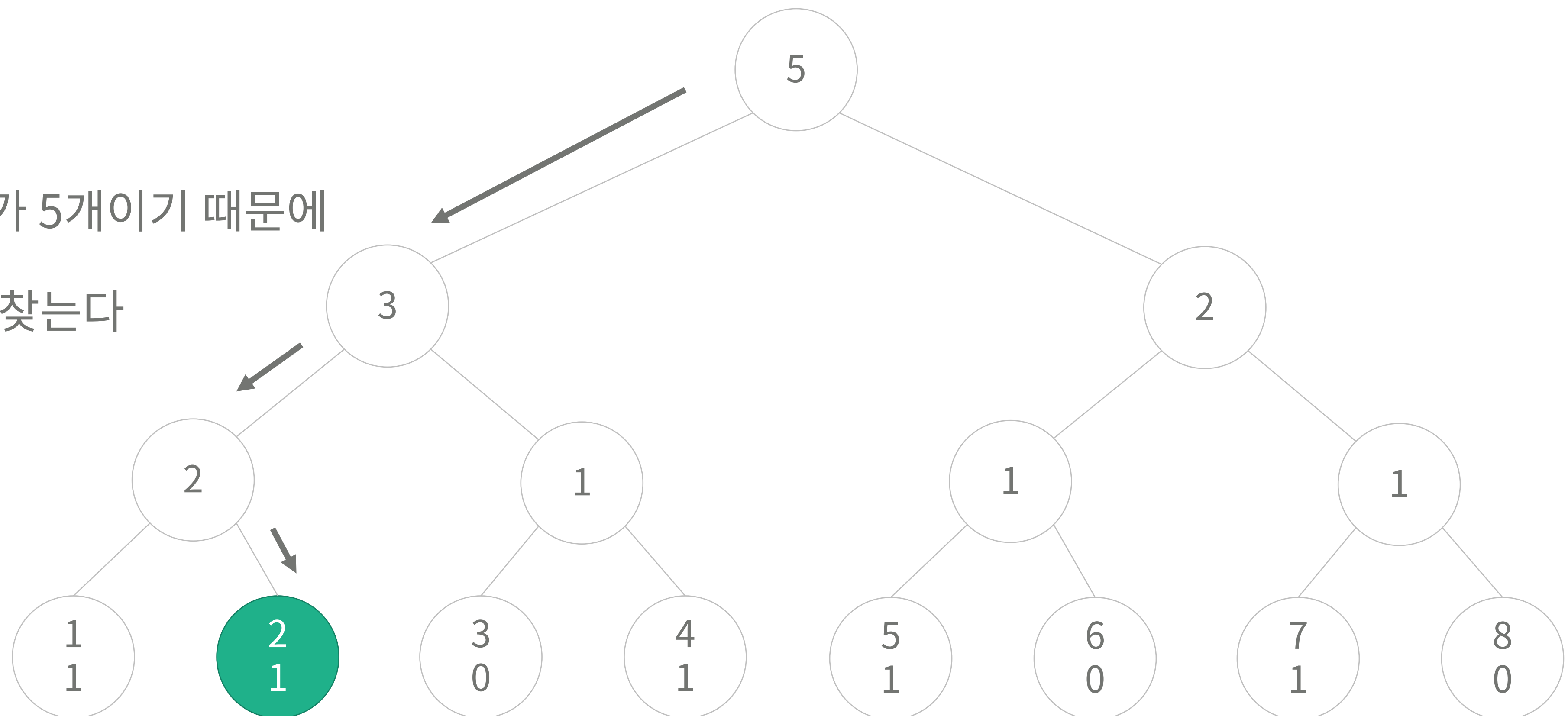
- $N = 7, K = 3$
- 1~3까지 합 = 2
- $2+3=5$ 번째 위치를 찾는다.



요세푸스 문제 2

<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

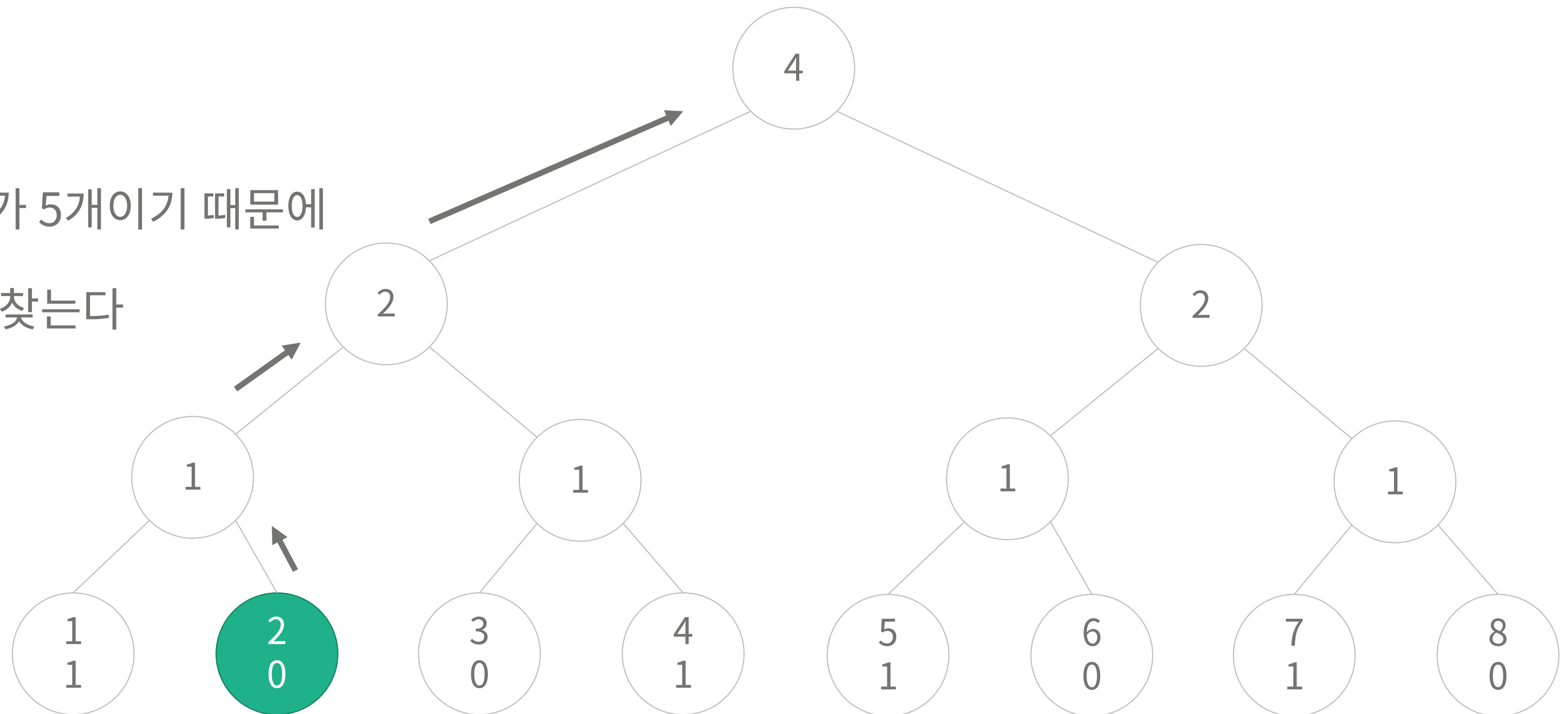


요세푸스 문제 2

70

<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

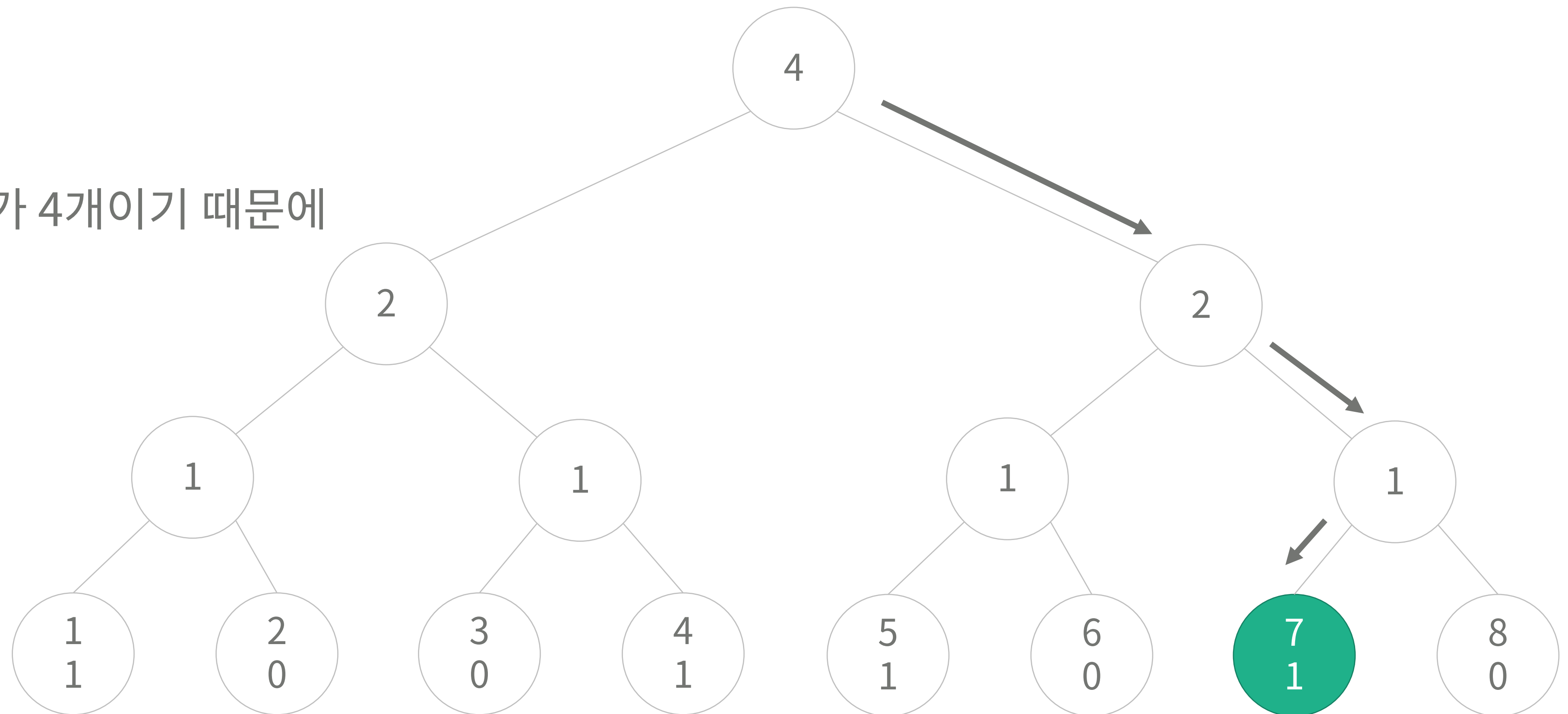


요세푸스 문제 2

71

<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.

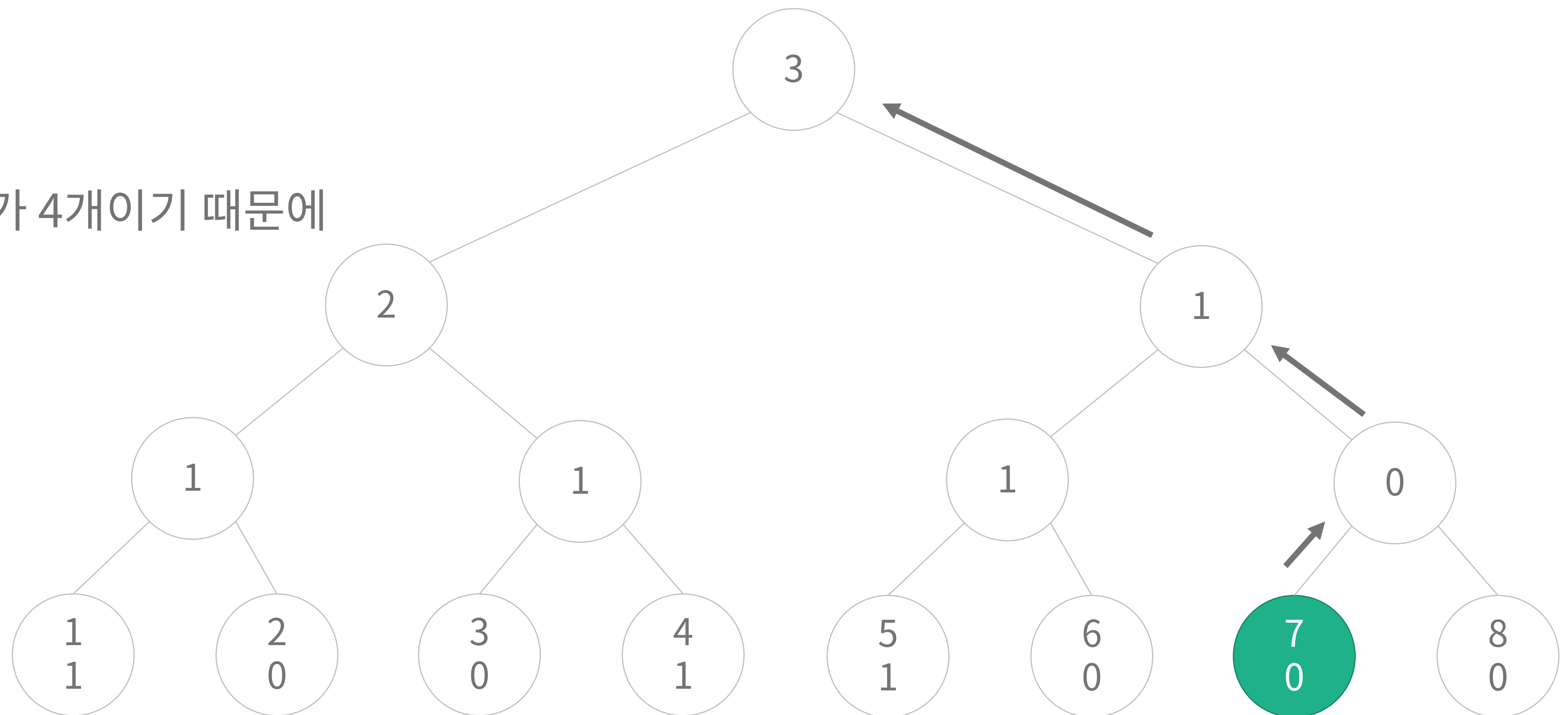


요세푸스 문제 2

72

<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.

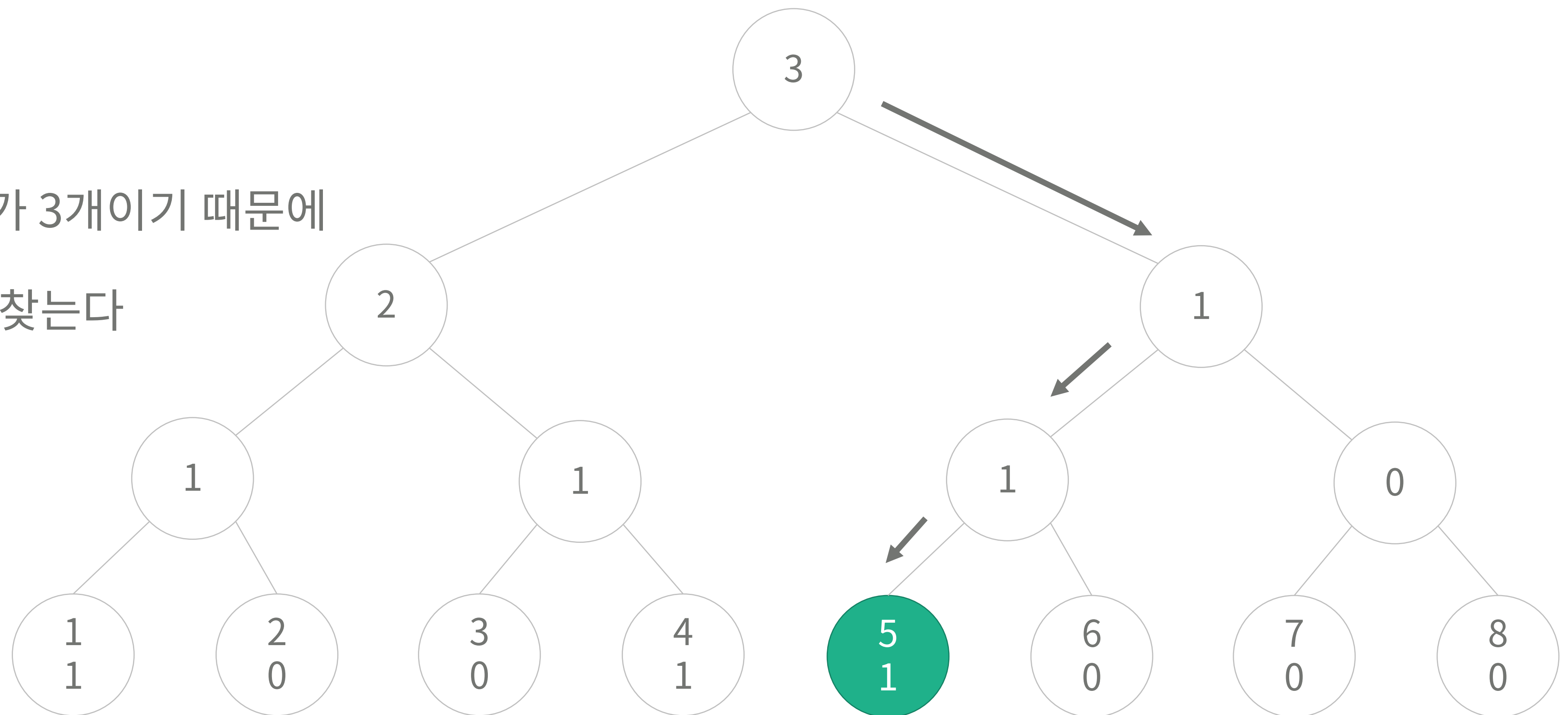


요세푸스 문제 2

73

<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

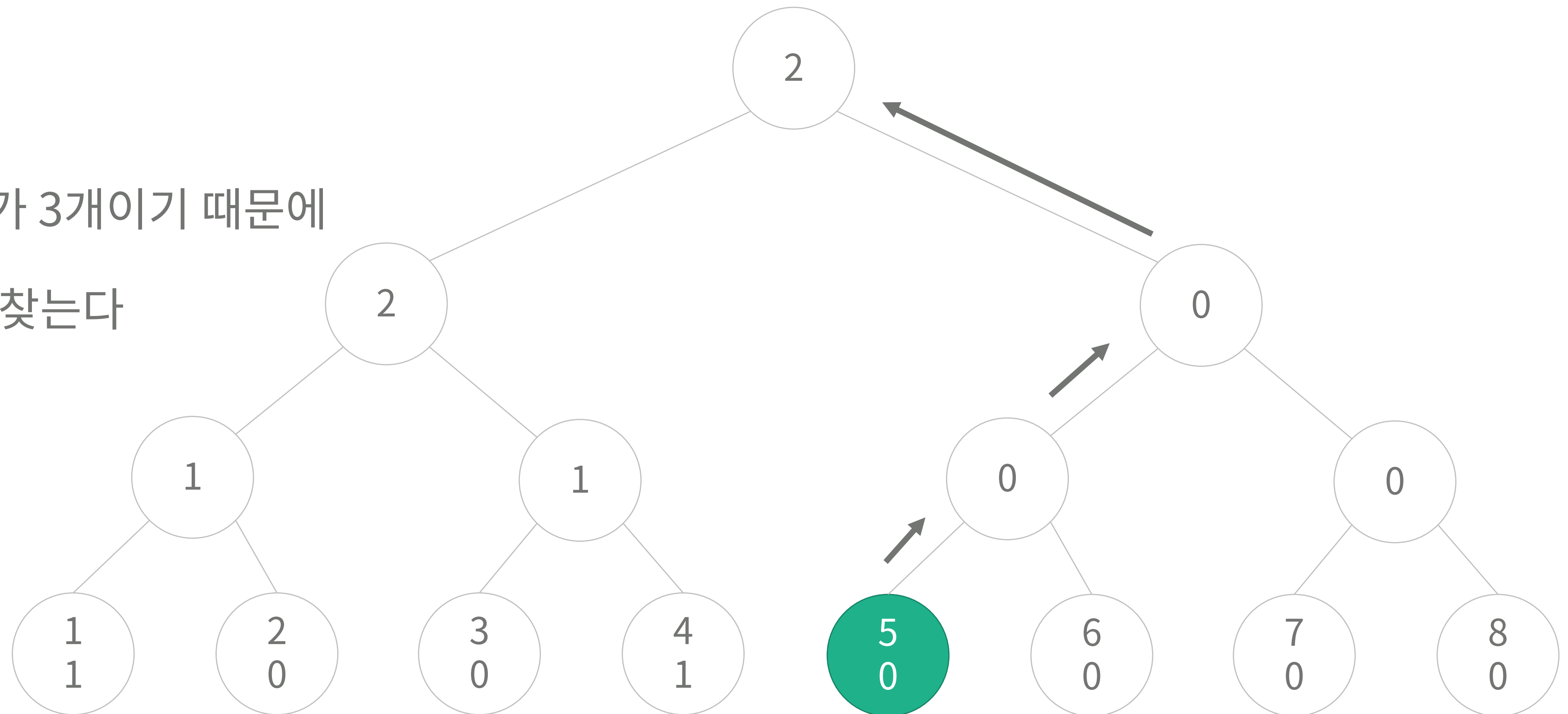


요세푸스 문제 2

74

<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

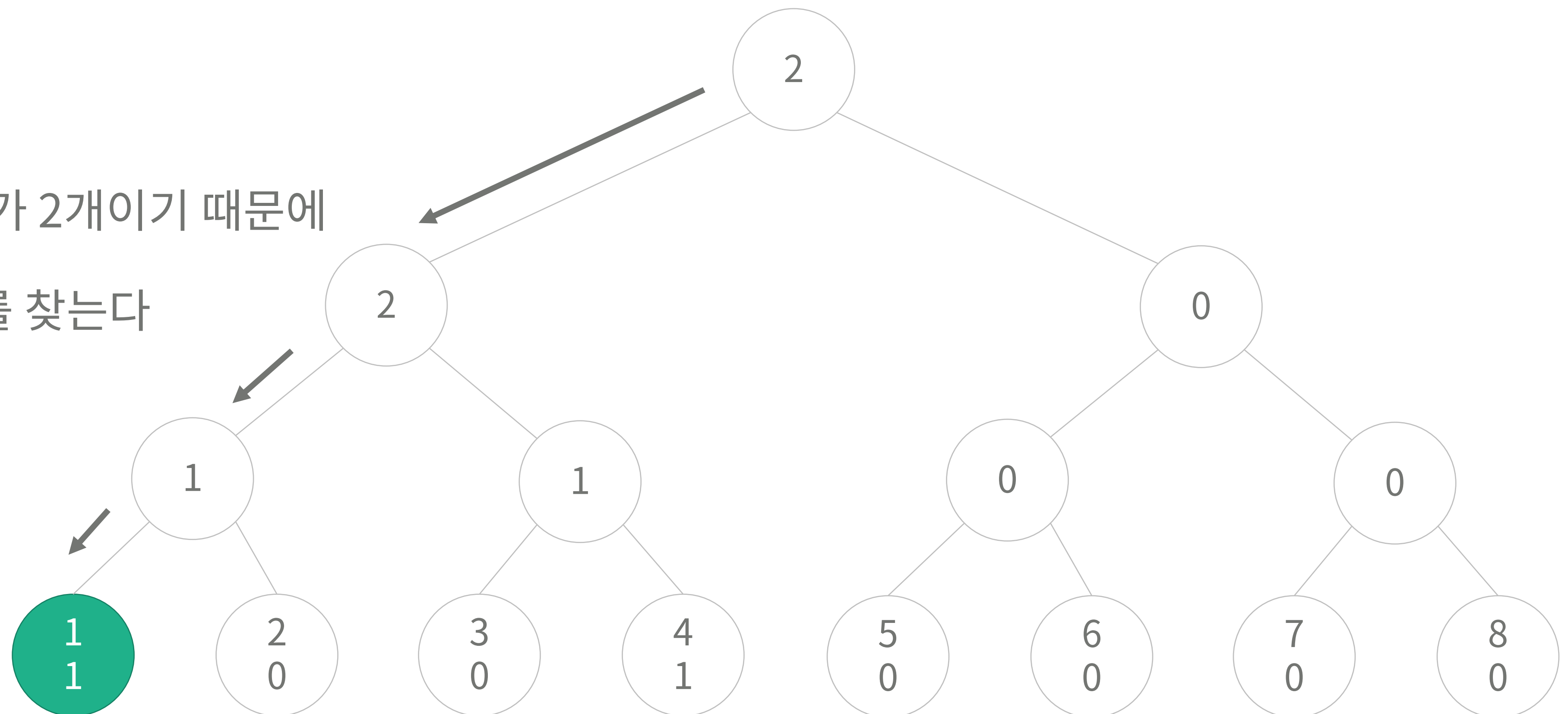


요세푸스 문제 2

75

<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



요세푸스 문제 2

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

- 소스: <http://codeplus.codes/9658c61ea2234b18833f1692a6d891d1>