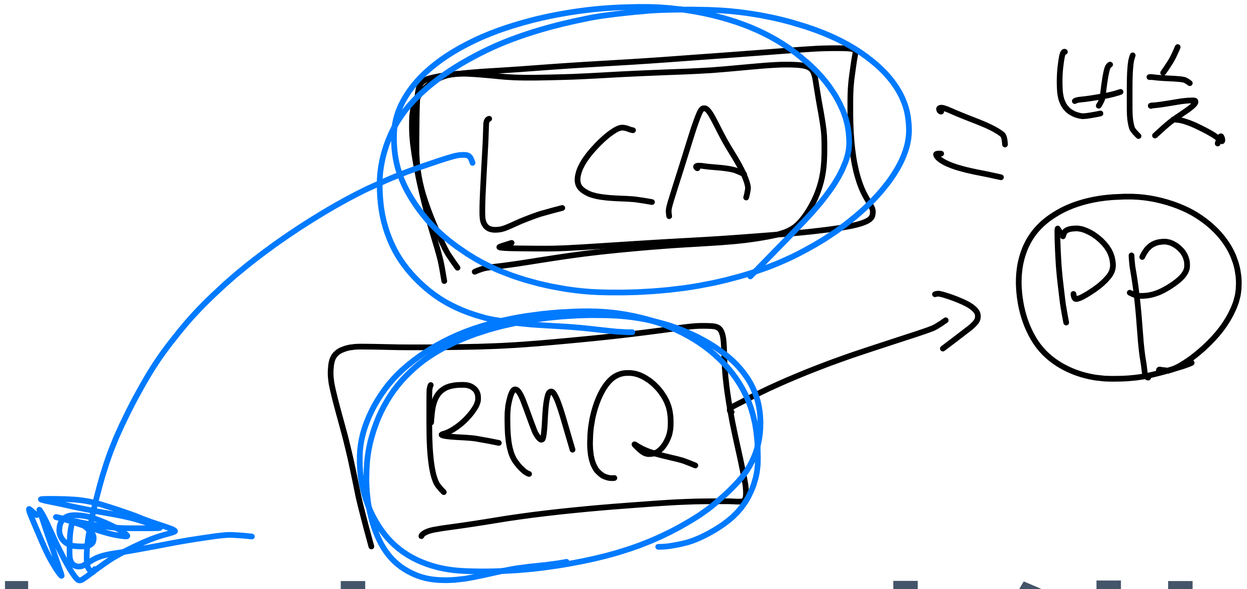


세그먼트 트리 (참고)

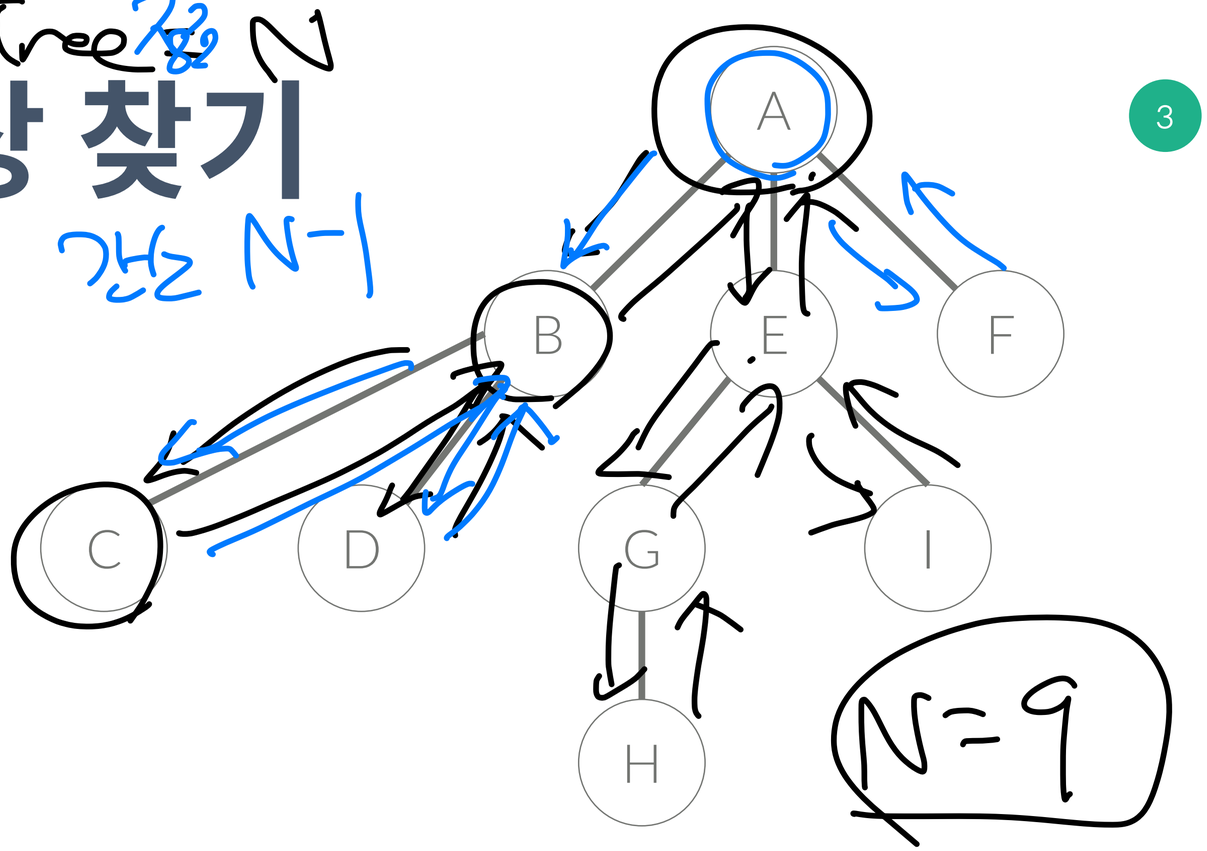
최백준 choi@startlink.io

LCA

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- 세그먼트 트리를 이용해서 LCA를 찾을 수 있다.



Tree 정점 N

간선 N-1

① 원리 DFS → 트리오더

N=9

A B C B D B A E G H G E I E A F A

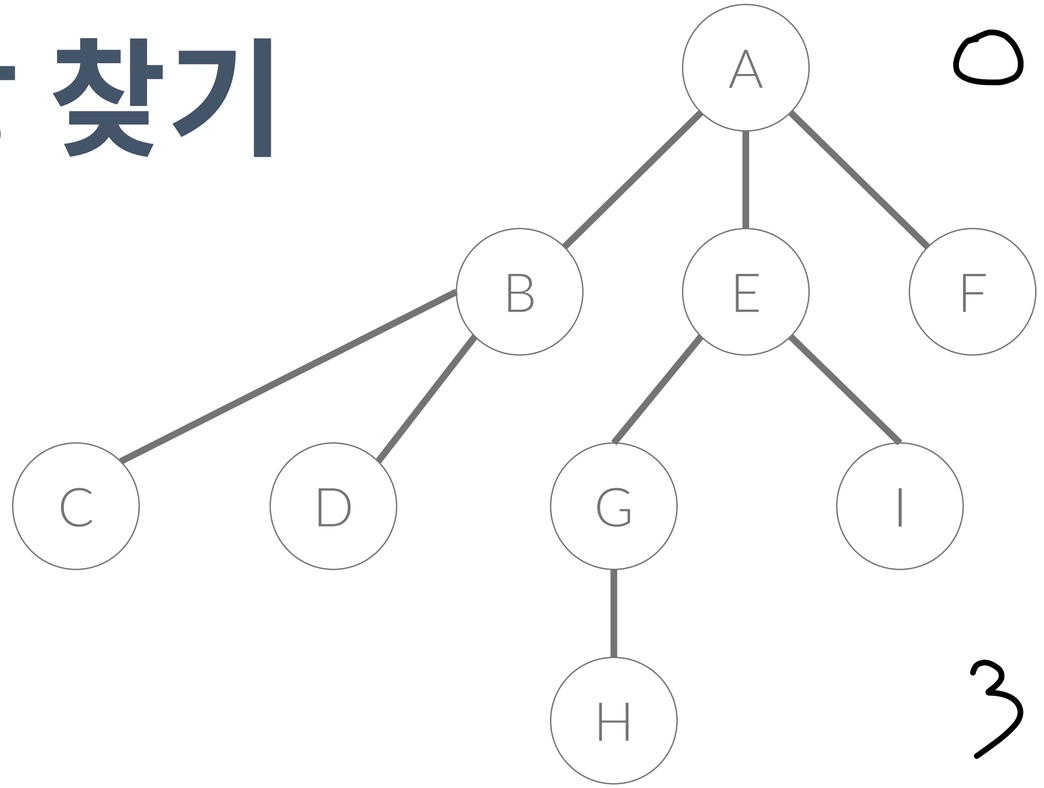
17

1 + (N-1)2 = 1 + 2N - 2 = 2N - 1

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- 트리를 프리오더 순회하고
- 순회하면서 방문한 정점을 모두 적는다.
- 이때, 중간 정점도 모두 기록한다.
- 그리고, 깊이도 모두 기록한다.



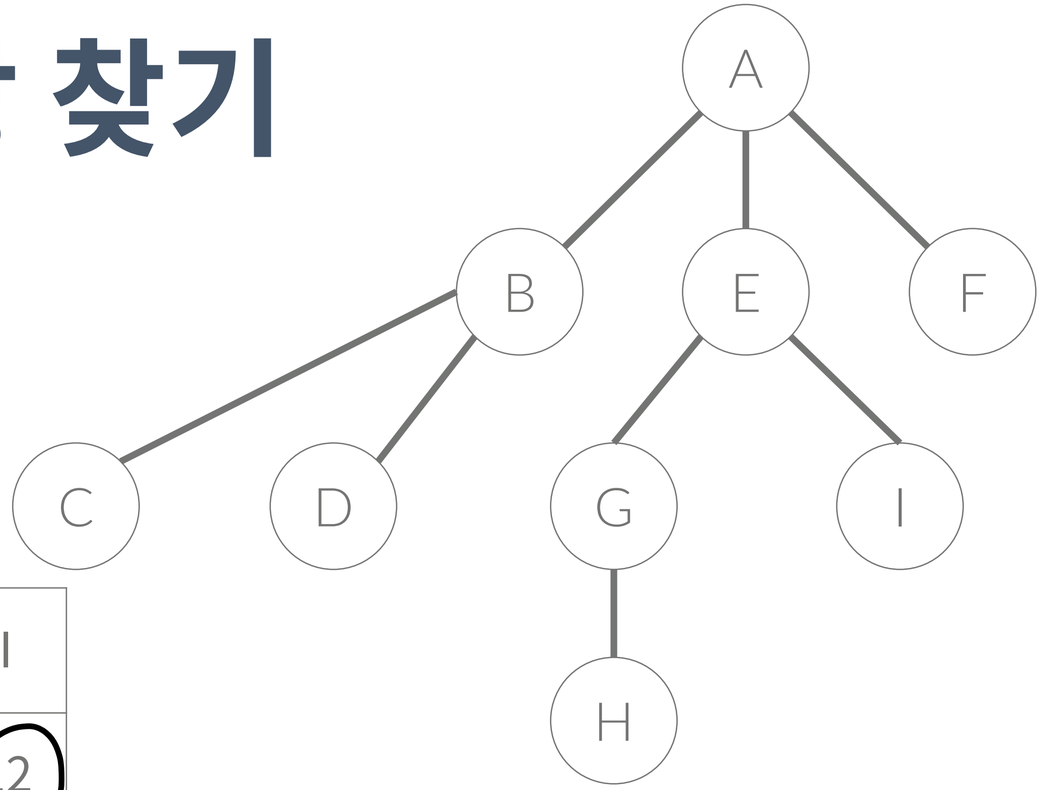
순서	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
정점	A	B	C	B	D	B	A	E	G	H	G	E	I	E	A	F	A
깊이	0	1	2	1	2	1	0	1	2	3	2	1	2	1	0	1	0

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- 각 정점을 처음 방문한 순서도 기록한다.

RMQ



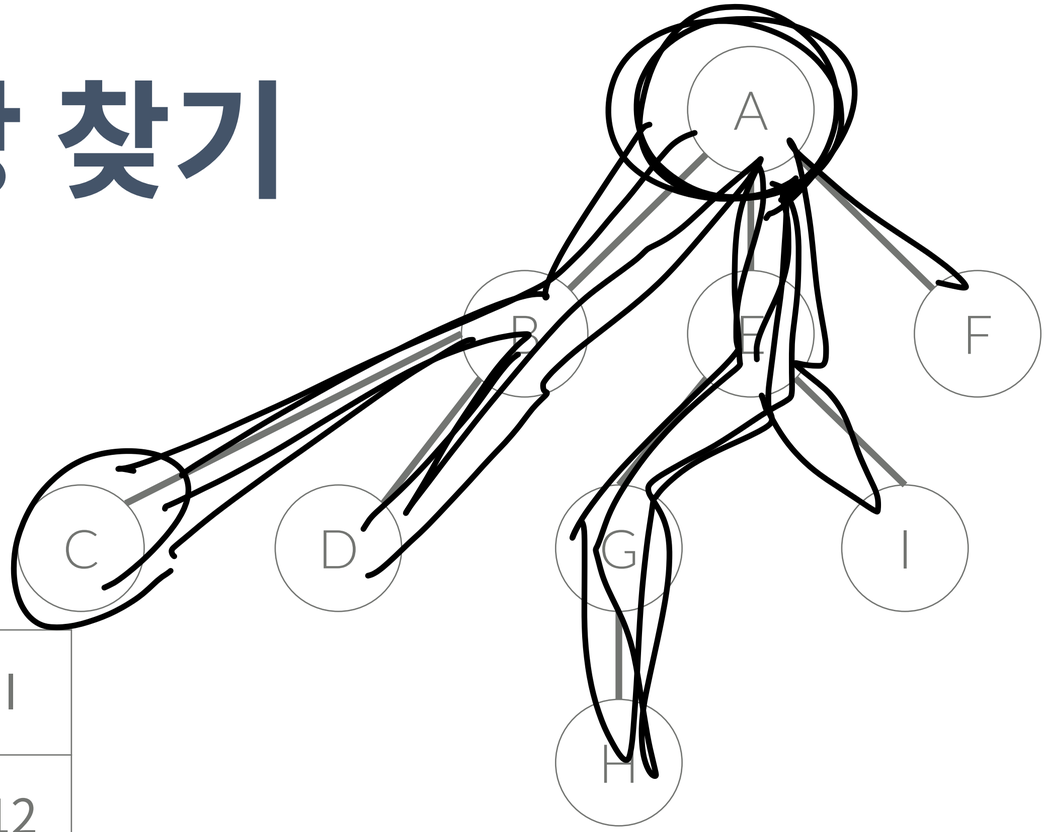
정점	A	B	C	D	E	F	G	H	I
처음	0	1	2	4	7	15	8	9	12

순서	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
정점	A	B	C	B	D	B	A	E	G	H	G	E	I	E	A	F	A
깊이	0	1	2	1	2	1	0	1	2	3	2	1	2	1	0	1	0

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- LCA(u, v)는 u와 v가 첫 등장한 인덱스 사이에서
- 깊이가 최소인 정점이다.



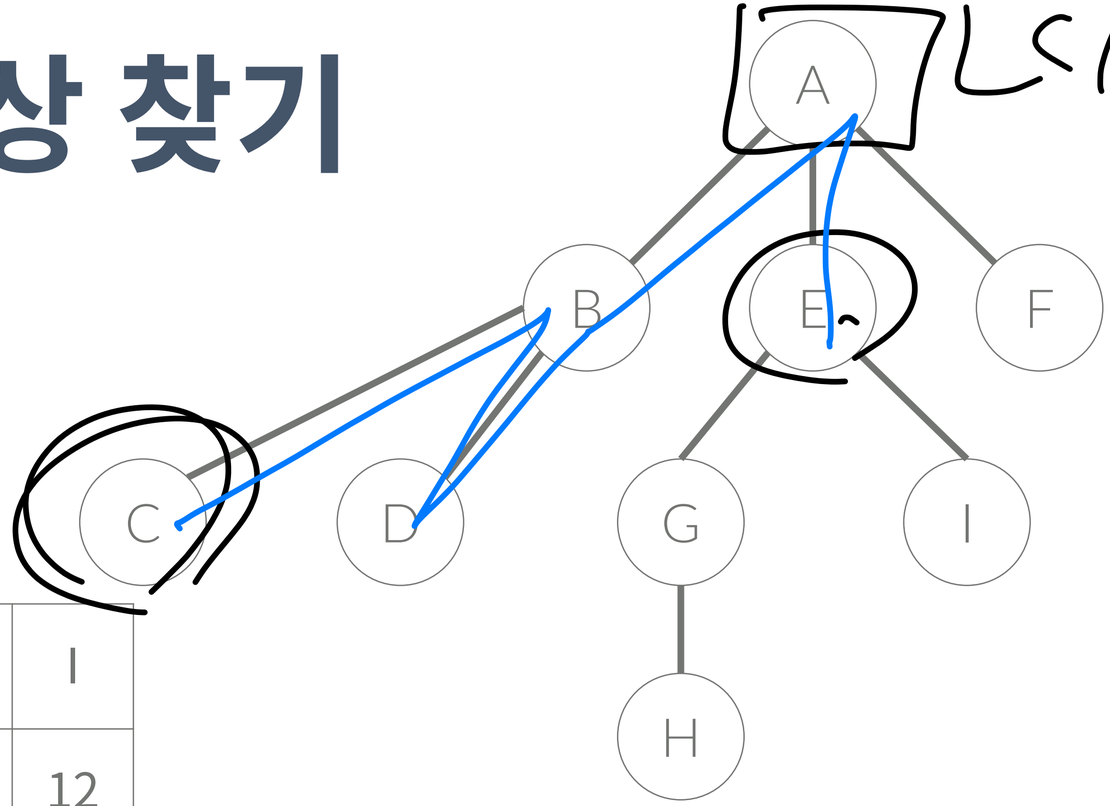
정점	A	B	C	D	E	F	G	H	I
처음	0	1	2	4	7	15	8	9	12

순서	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
정점	A	B	C	B	D	B	A	E	G	H	G	E	I	E	A	F	A
깊이	0	1	2	1	2	1	0	1	2	3	2	1	2	1	0	1	0

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- LCA(C, E) = A



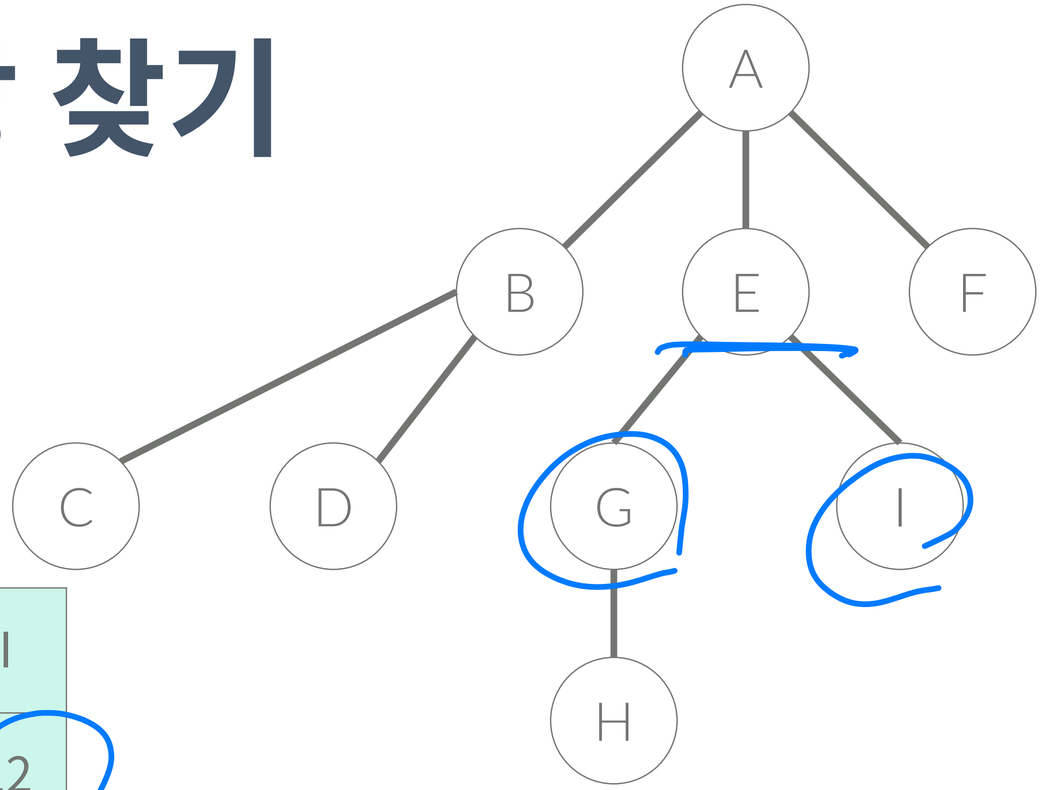
정점	A	B	C	D	E	F	G	H	I
처음	0	1	2	4	7	15	8	9	12

순서	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
정점	A	B	C	B	D	B	A	E	G	H	G	E	I	E	A	F	A
깊이	0	1	2	1	2	1	0	1	2	3	2	1	2	1	0	1	0

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- LCA(G, I) = E



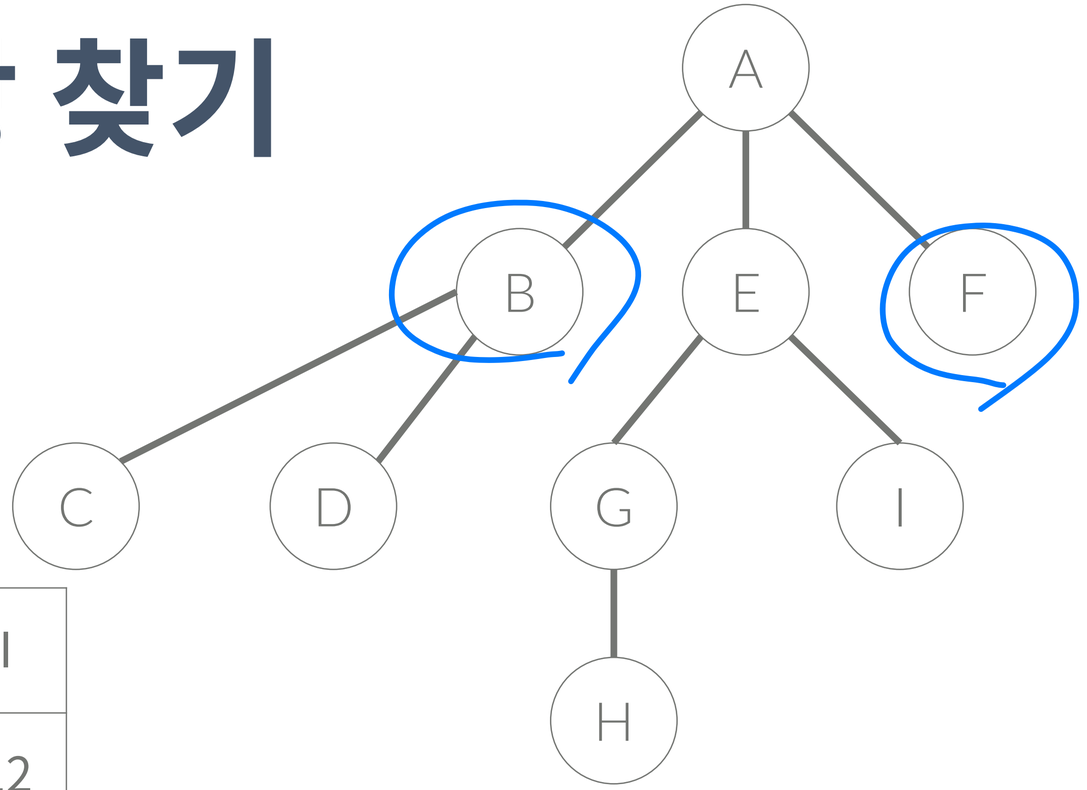
정점	A	B	C	D	E	F	G	H	I
처음	0	1	2	4	7	15	8	9	12

순서	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
정점	A	B	C	B	D	B	A	E	G	H	G	E	I	E	A	F	A
깊이	0	1	2	1	2	1	0	1	2	3	2	1	2	1	0	1	0

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- LCA(B, F) = A



정점	A	B	C	D	E	F	G	H	I
처음	0	1	2	4	7	15	8	9	12

순서	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
정점	A	B	C	B	D	B	A	E	G	H	G	E	I	E	A	F	A
깊이	0	1	2	1	2	1	0	1	2	3	2	1	2	1	0	1	0

LCA 2

https://www.acmicpc.net/problem/11438

- 소스: http://codeplus.codes/ec22e64529c94875bc1daea8e19cb6ba

시간복잡도 트레이

비재귀 구현

세그먼트 트리 비재귀 구현

구간의 합 구하기

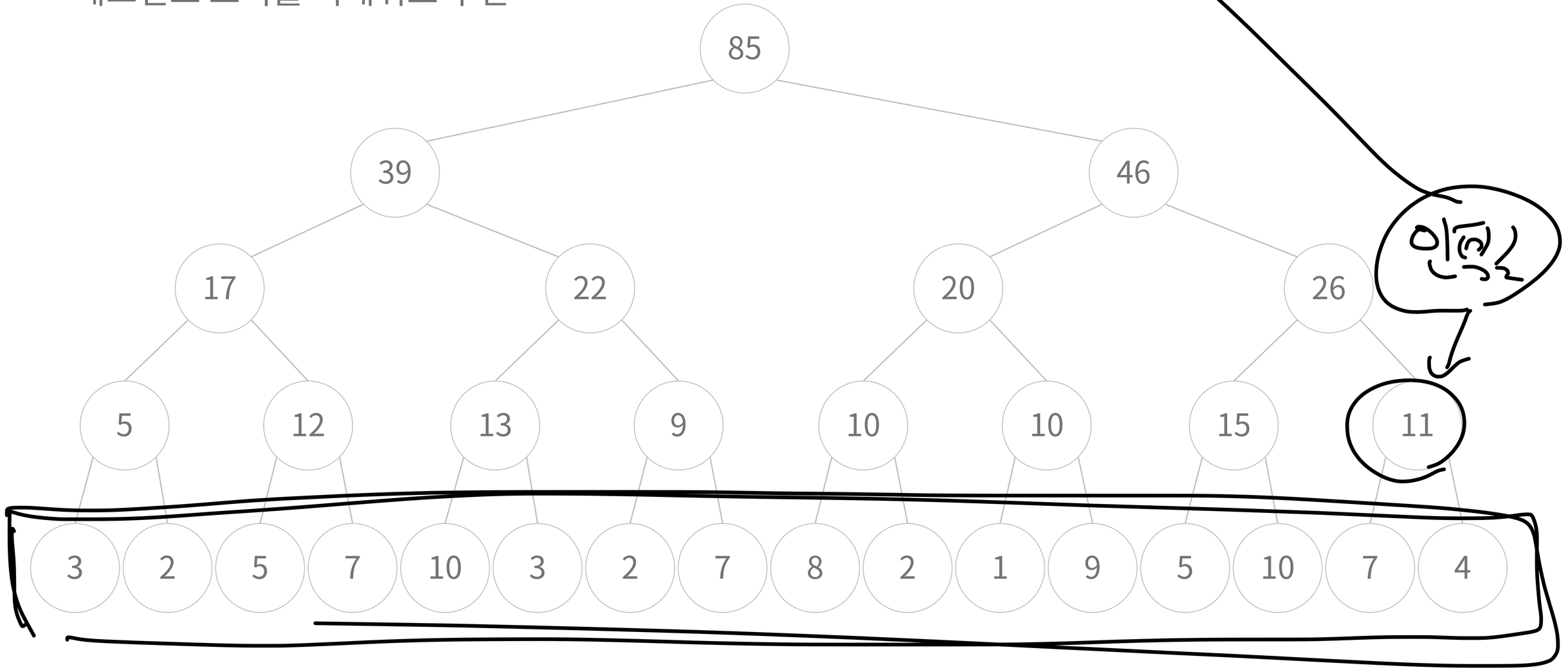
- N을 가장 가까운 2^k의 꼴로 나타내면 세그먼트 트리는 Full Binary Tree가 된다.
- N을 크게 하고, 뒤쪽 부분은 제외하게 코드를 구현하면 비재귀로 구현할 수 있다.

모든 리프노드가 같은 레벨) 같은 길이

세그먼트 트리 비재귀 구현

구간의 합 구하기

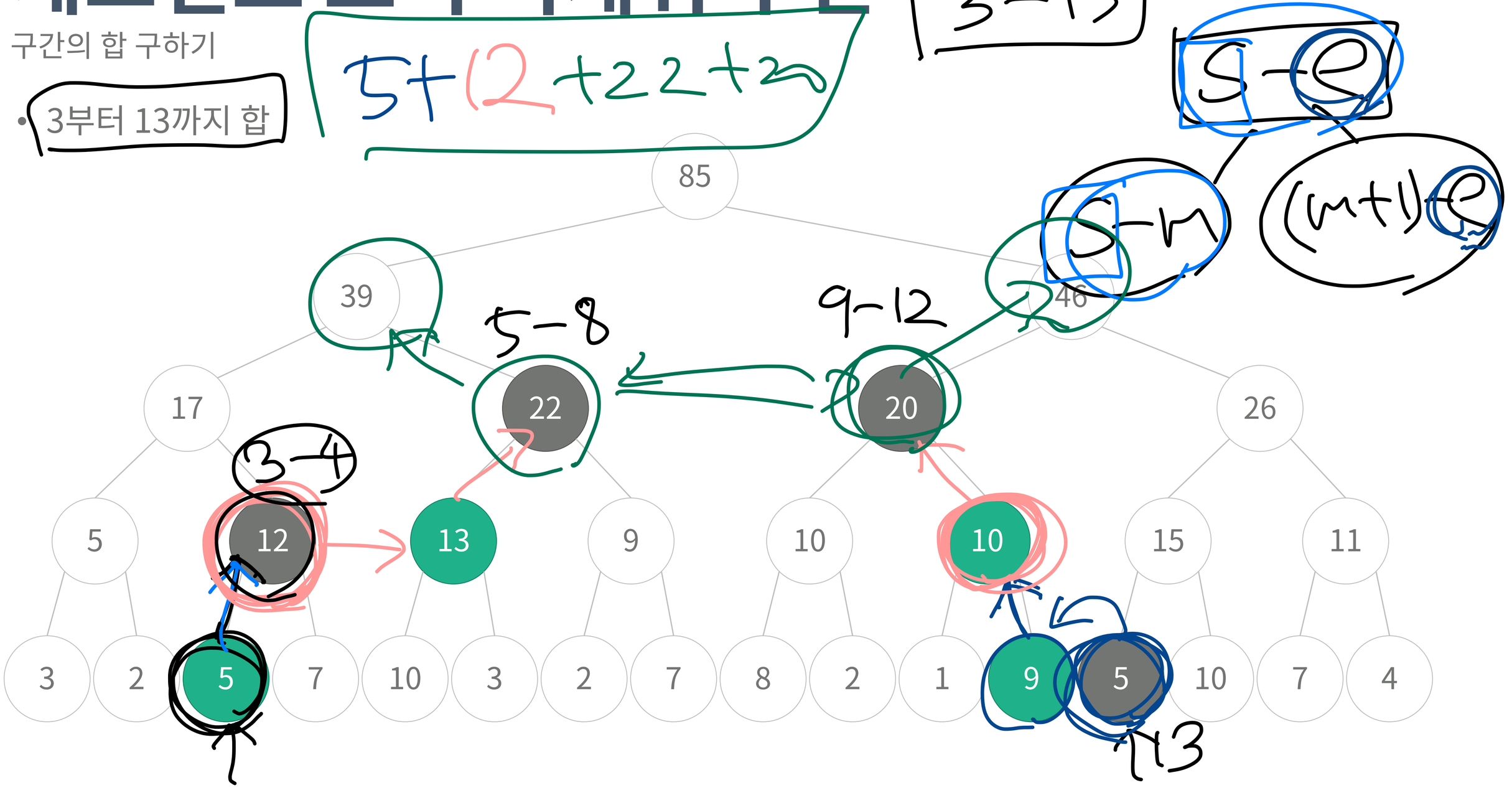
- 세그먼트 트리를 비재귀로 구현



세그먼트 트리 비재귀 구현

구간의 합 구하기

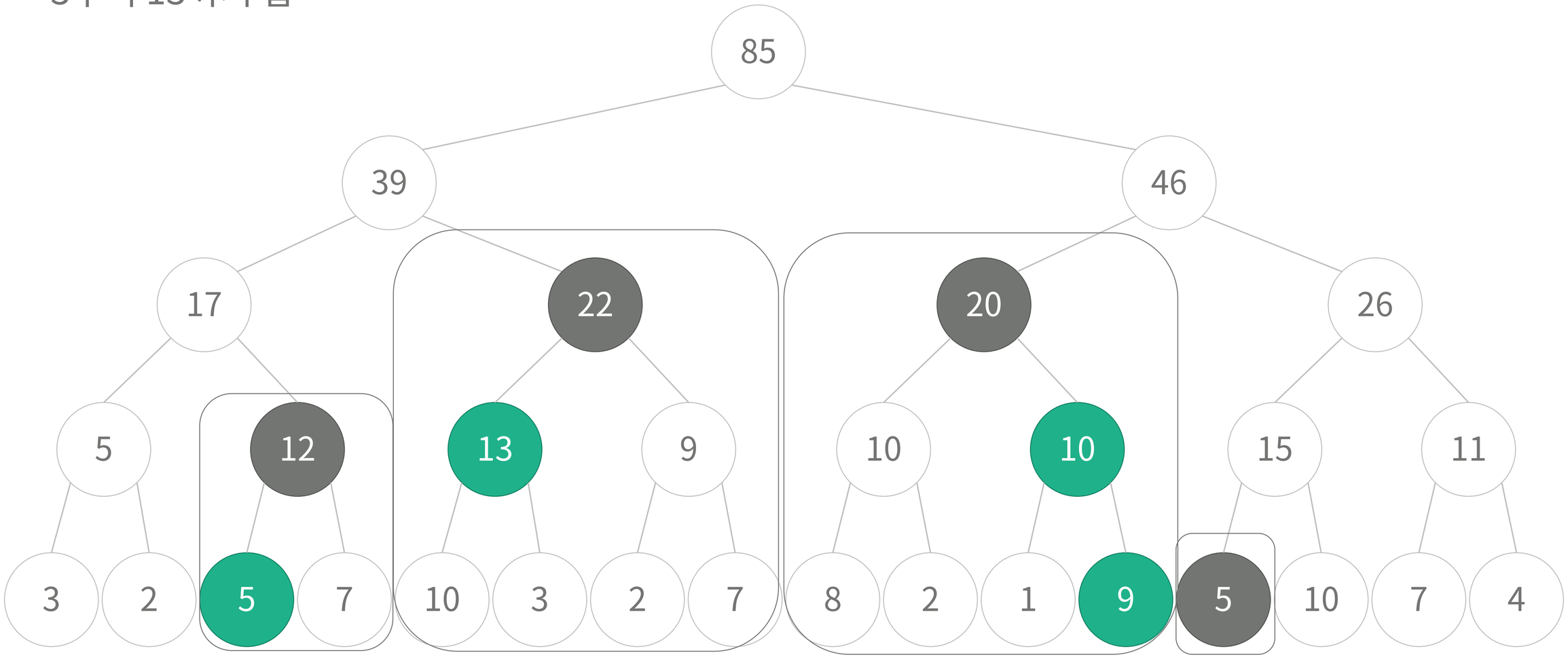
- 3부터 13까지 합



세그먼트 트리 비재귀 구현

구간의 합 구하기

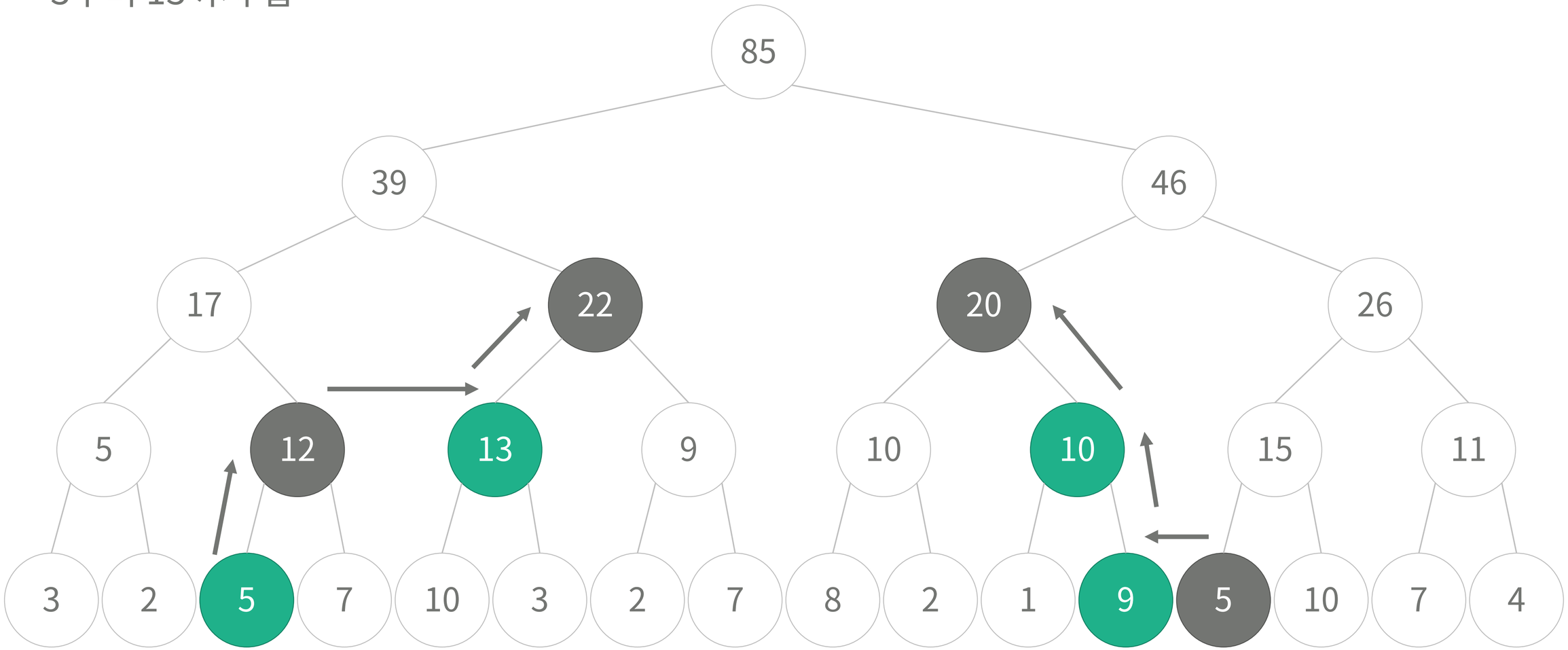
- 3부터 13까지 합



세그먼트 트리 비재귀 구현

구간의 합 구하기

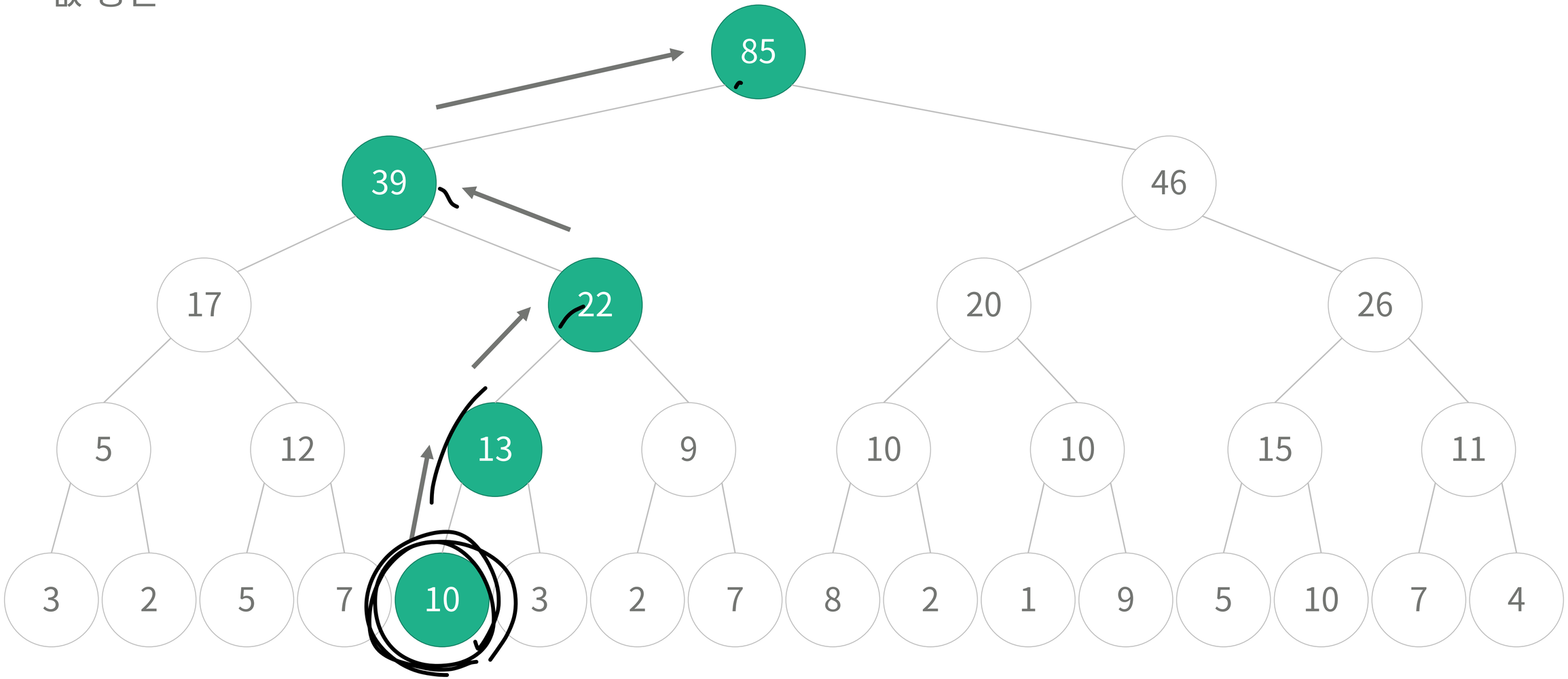
- 3부터 13까지 합



세그먼트 트리 비재귀 구현

구간의 합 구하기

- 값 갱신



세그먼트 트리 비재귀 구현

구간의 합 구하기

- 합 구하는 방법
- 왼쪽
  - 왼쪽 자식이면 올라간다
  - 오른쪽 자식이면 답을 더하고, 오른쪽 칸으로 이동
- 오른쪽
  - 오른쪽 자식이면 올라간다
  - 왼쪽 자식이면 답을 더하고, 왼쪽 칸으로 이동

세그먼트 트리 비재귀 구현

구간의 합 구하기

- 소스: http://codeplus.codes/40242b5512f54024adc41df3d27d3141