

SEGMENT TREE

24. 구간 트리

구간 트리 : 구간에 대한 질문 대답하기

- ▶ RMQ
- ▶ RSQ
- ▶ Fenwick tree

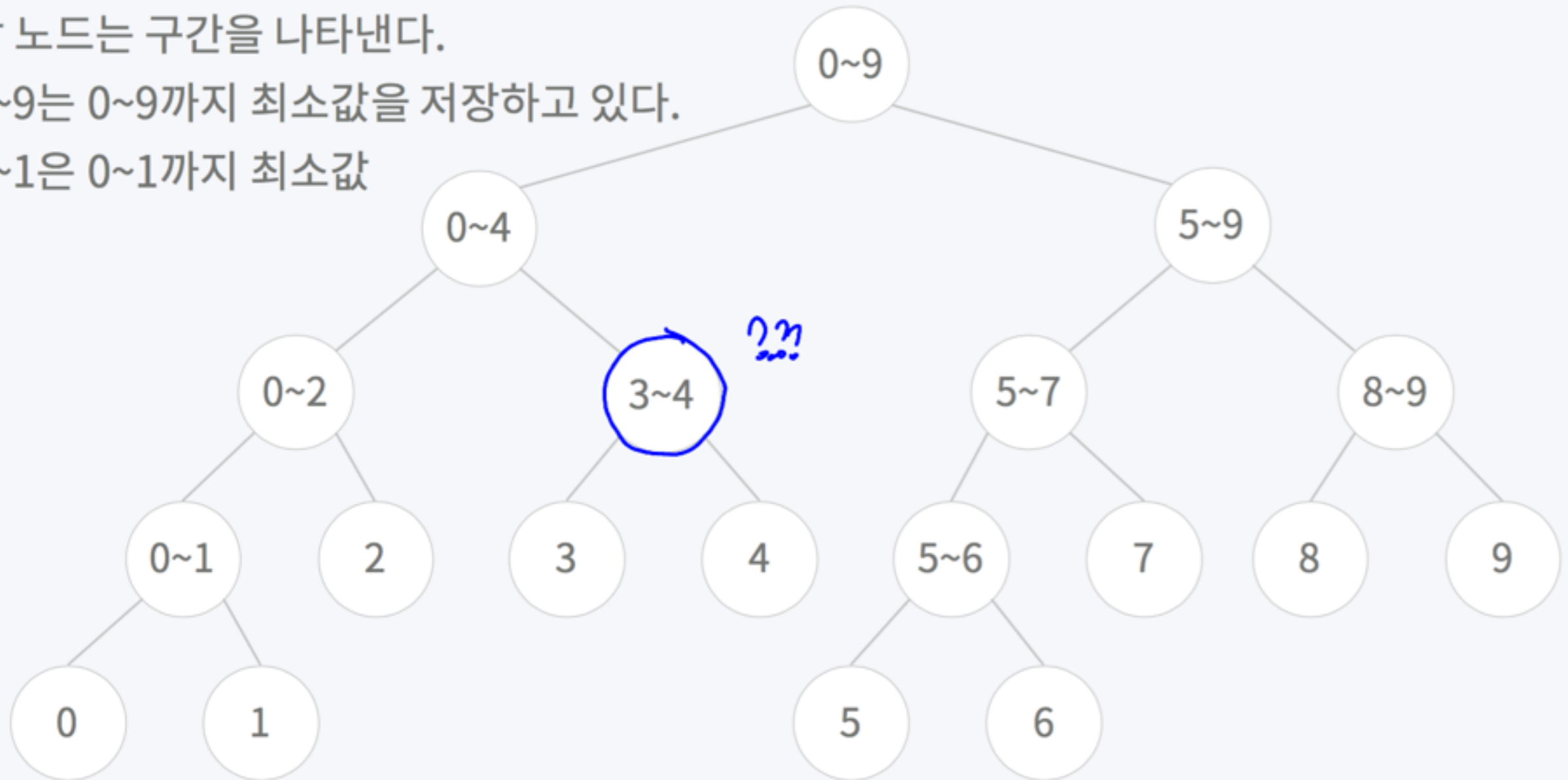
구간 트리

- ▶ 특정 구간에 대한 질문을 빠르게 대답 $O(\lg n)$
- ▶ 배열을 통해 구현 가능

세그먼트 트리

구간의 최소값 구하기 (RMQ)

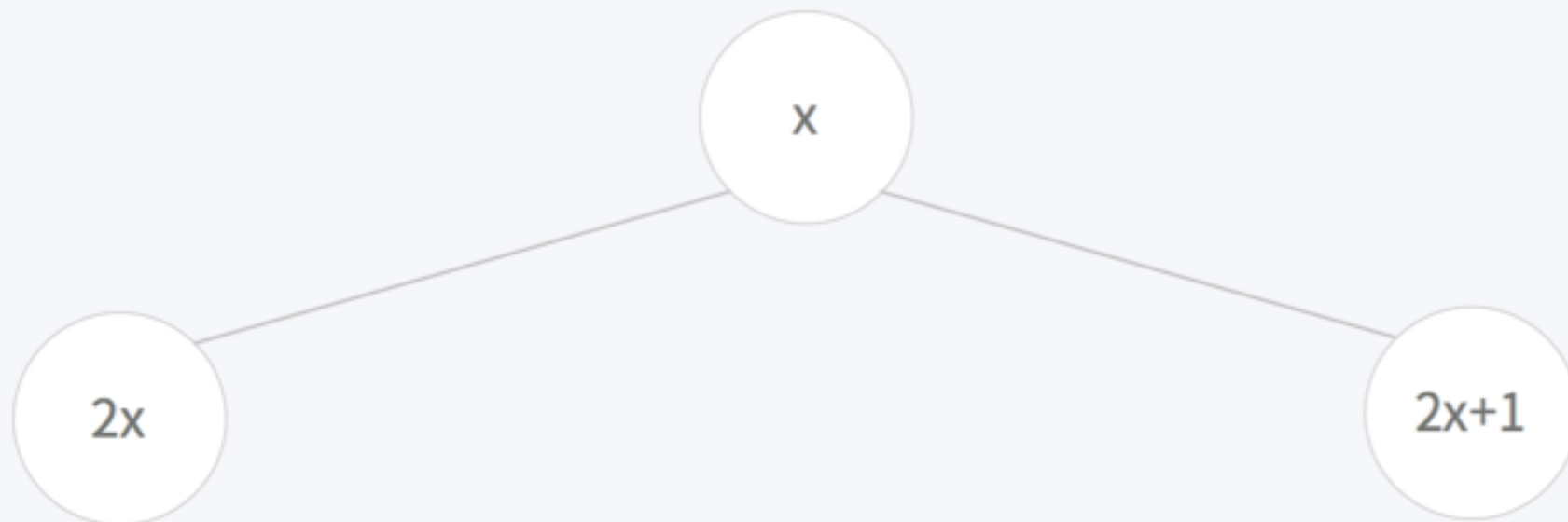
- 각 노드는 구간을 나타낸다.
- 0~9는 0~9까지 최소값을 저장하고 있다.
- 0~1은 0~1까지 최소값



세그먼트 트리

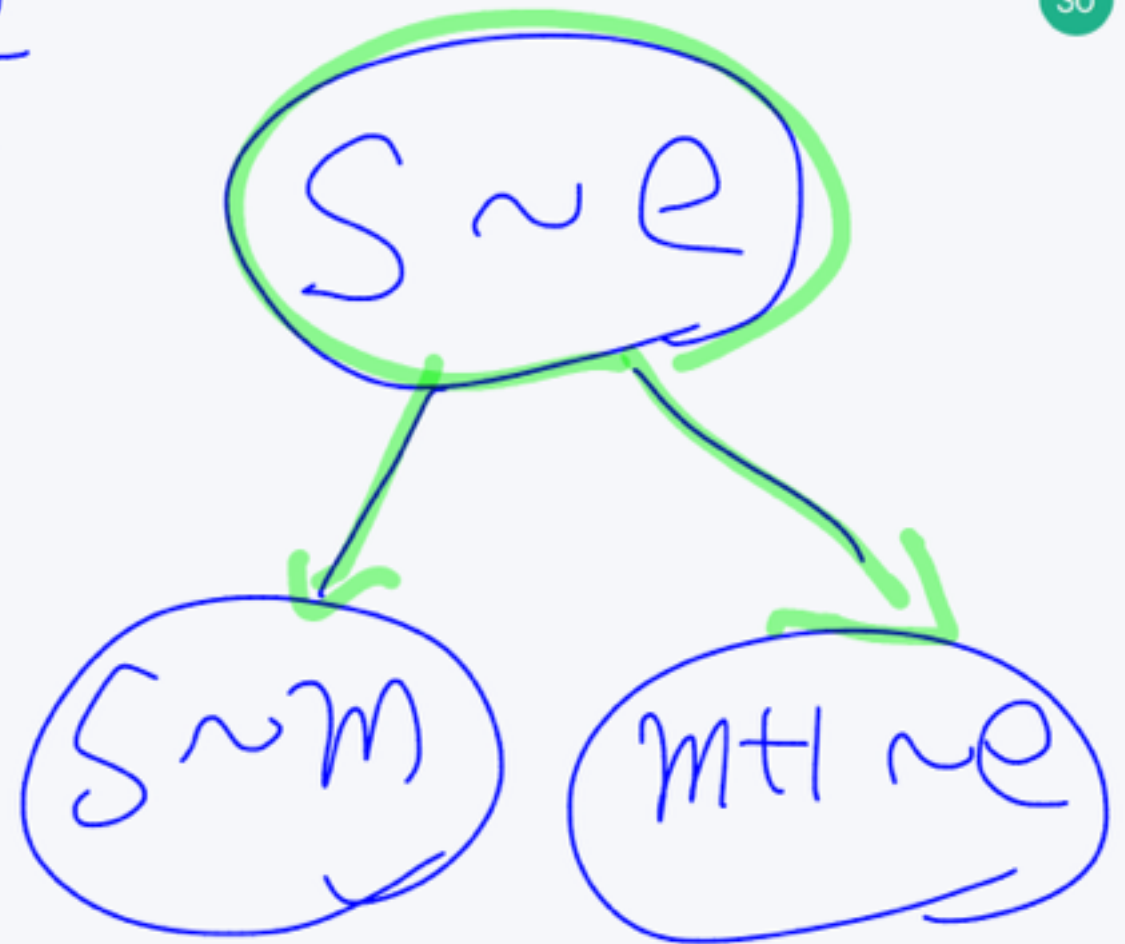
구간의 최소값 구하기 (RMQ)

- $start == end$ 인 경우에는 리프 노드
- 노드 $node$ 의 왼쪽 자식: $2*node$, 오른쪽 자식: $2*node+1$
- 어떤 노드가 $[start, end]$ 를 담당한다면
- 왼쪽 자식: $[start, (start+end)/2]$, 오른쪽 자식: $[(start+end)/2+1, end]$ 를 담당해야 한다



$$m = \frac{s+e}{2}$$

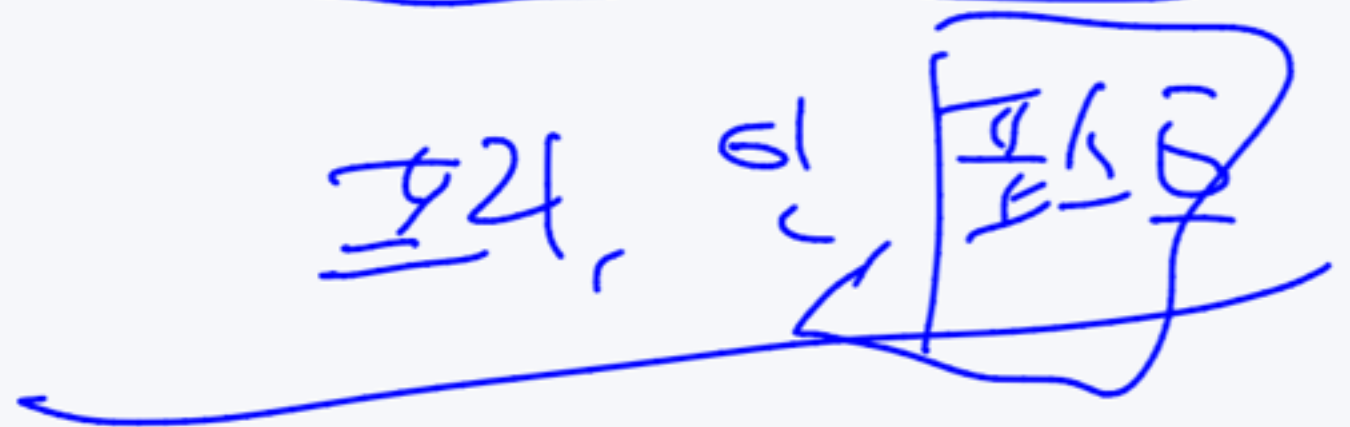
세그먼트 트리



```

void init(int node, int start, int end) {
    if (start == end) {
        tree[node] = a[start];
    } else {
        init(node*2, start, (start+end)/2);
        init(node*2+1, (start+end)/2+1, end);
        tree[node] = min(tree[node*2], tree[node*2+1]);
    }
}

```



세그먼트 트리

구간의 최소값 구하기 (RMQ)

```
int query(int node, int start, int end, int i, int j) {  
    if (i > end || j < start)   
        return -1;  
    if (i <= start && end <= j)   
        return d[node];  
    int m1 = query(2*node, start, (start+end)/2, i, j);  
    int m2 = query(2*node+1, (start+end)/2+1, end, i, j);  
    if (m1 == -1) return m2;  
    else if (m2 == -1) return m1;  
    else return min(m1, m2);  
}
```

포함 X

포함 O

$m1 == -1 \text{ \&\& } m2 ==$

[HTTPS://WWW.ACMICPC.NET/PROBLEM/10868](https://www.acmicpc.net/problem/10868)

[HTTPS://WWW.ACMICPC.NET/PROBLEM/2357](https://www.acmicpc.net/problem/2357)

RSQ

- ▶ 구간 트리로 구할 수 있으나
- ▶ 펜윅 트리를 쓰는 것이 더 간단

Fenwick Tree (BIT)

- i 의 마지막 자리: i 를 2진수로 나타냈을 때, 가장 마지막 1이 나타내는 값

- $3 = 11_2$

- $5 = 101_2$

1101100

- $6 = 110_2$

- $8 = 1000_2$

- $9 = 1001_2$

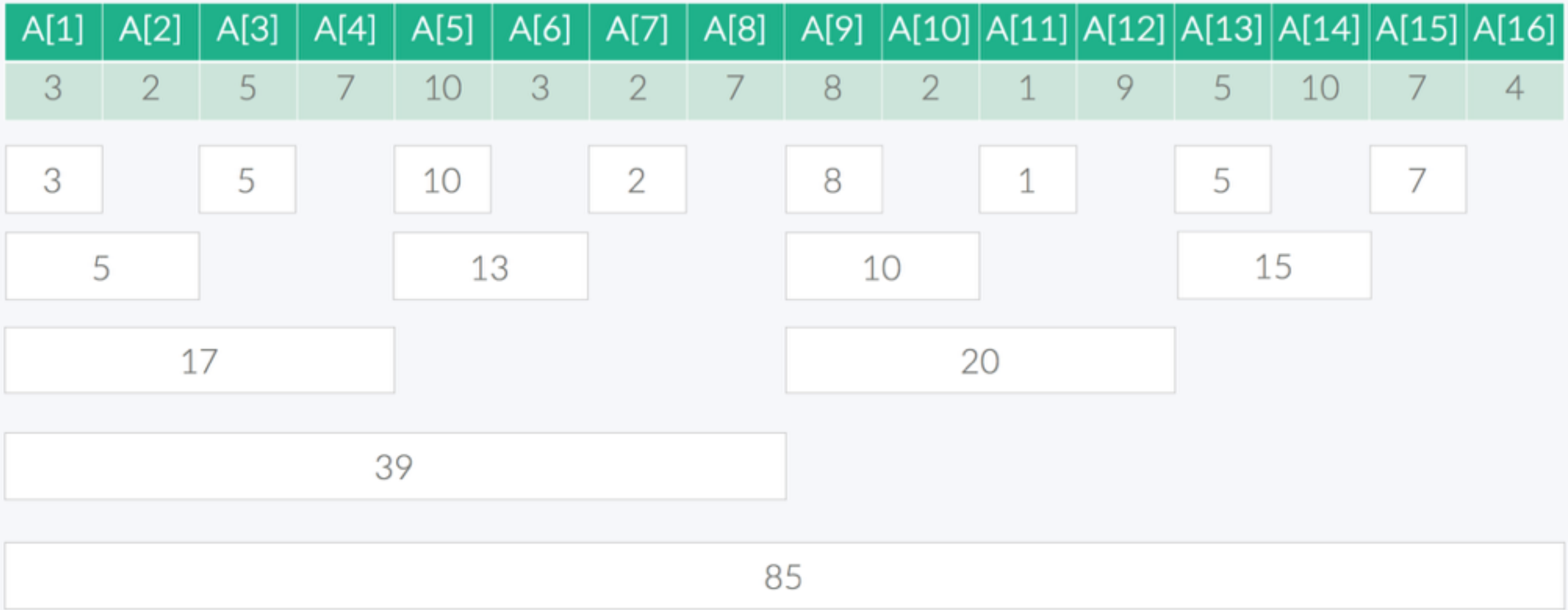
- $10 = 1010_2$

- $11 = 1011_2$

- $12 = 1100_2$

- $16 = 10000_2$

12 = 1100
4



TREE[IDX] : IDX부터 앞으로 IDX의 마지막 비트 만큼의 합