트리 2

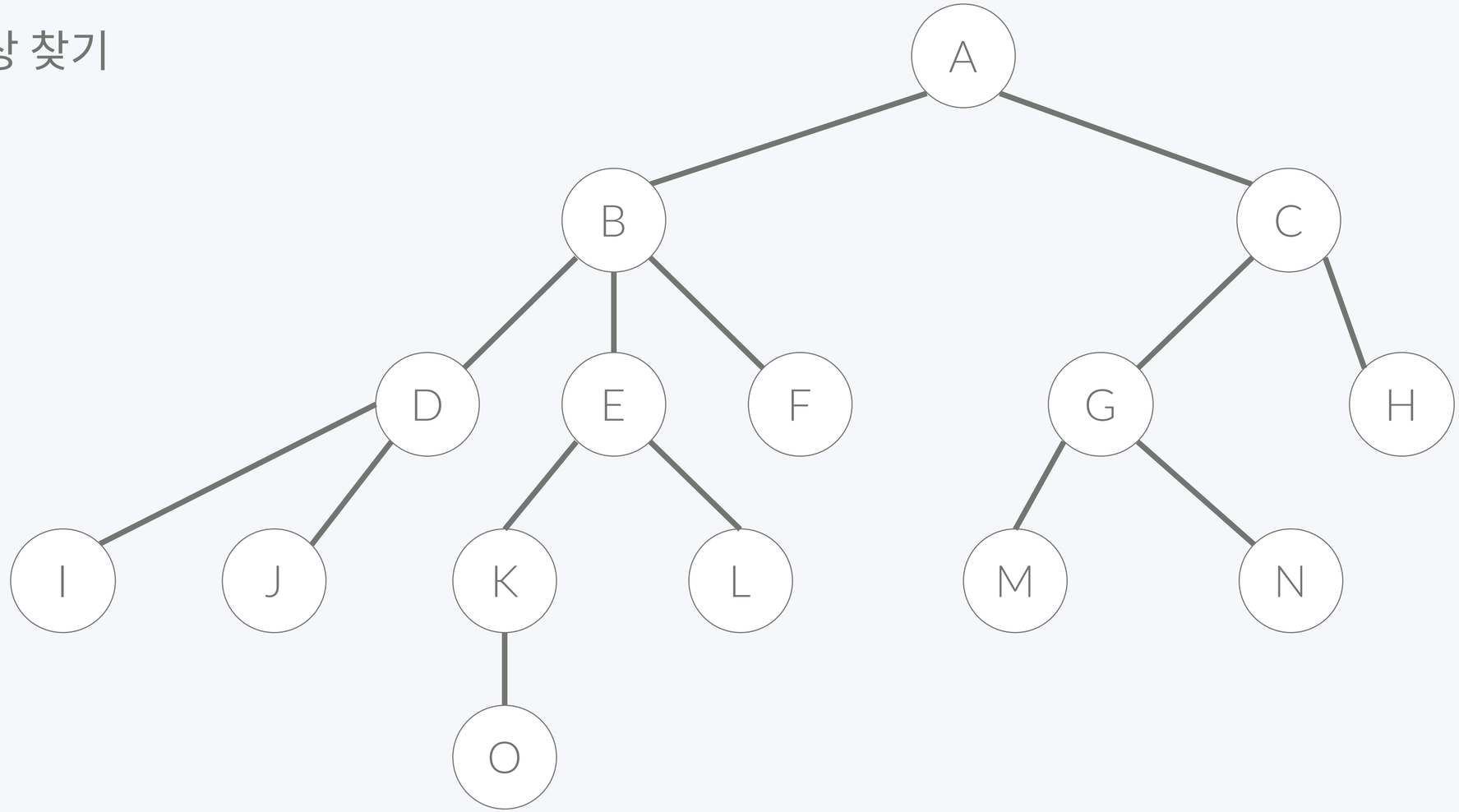
최백준 choi@startlink.io

LCA

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

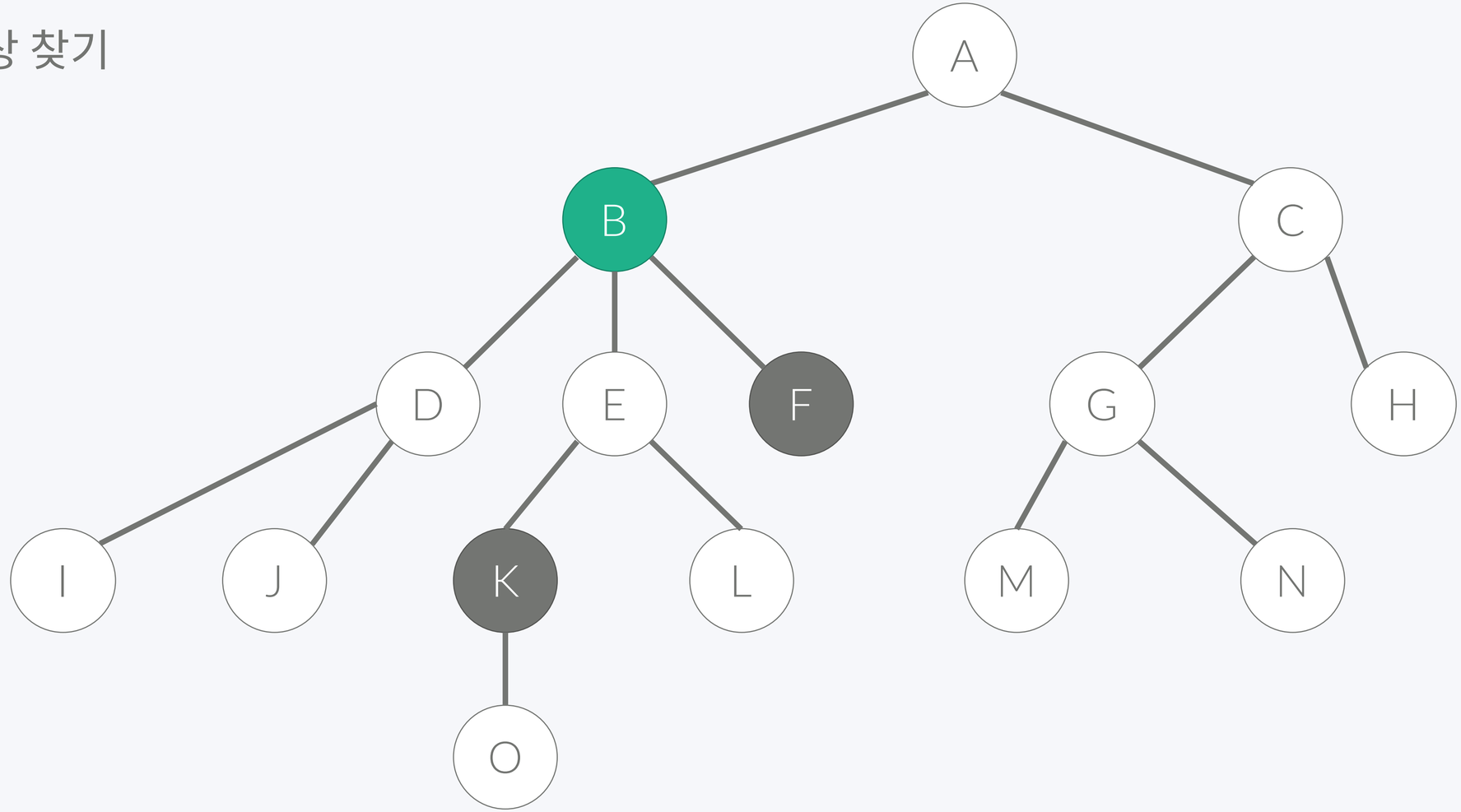
- 가장 가까운 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

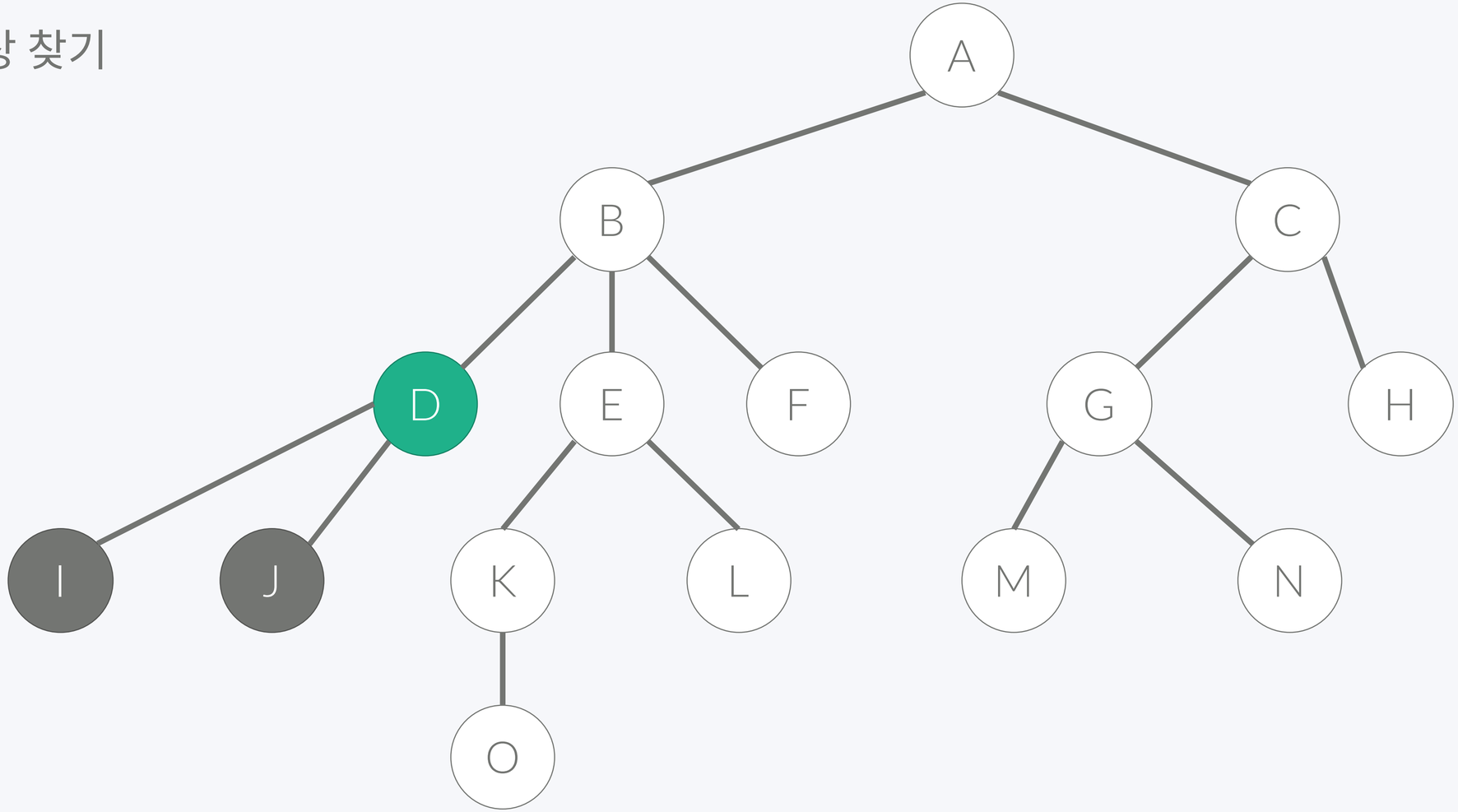
- 가장 가까운 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

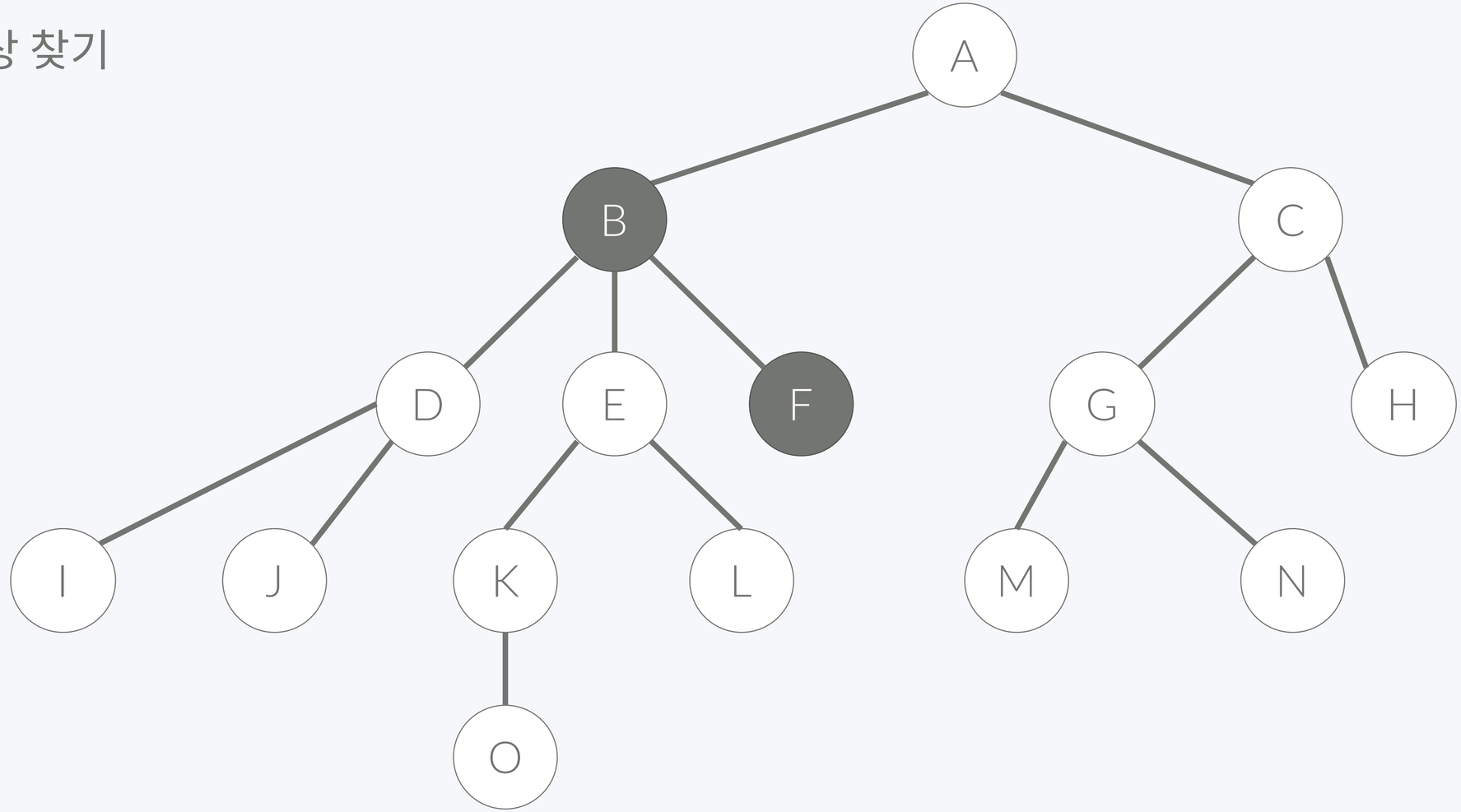
- 가장 가까운 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

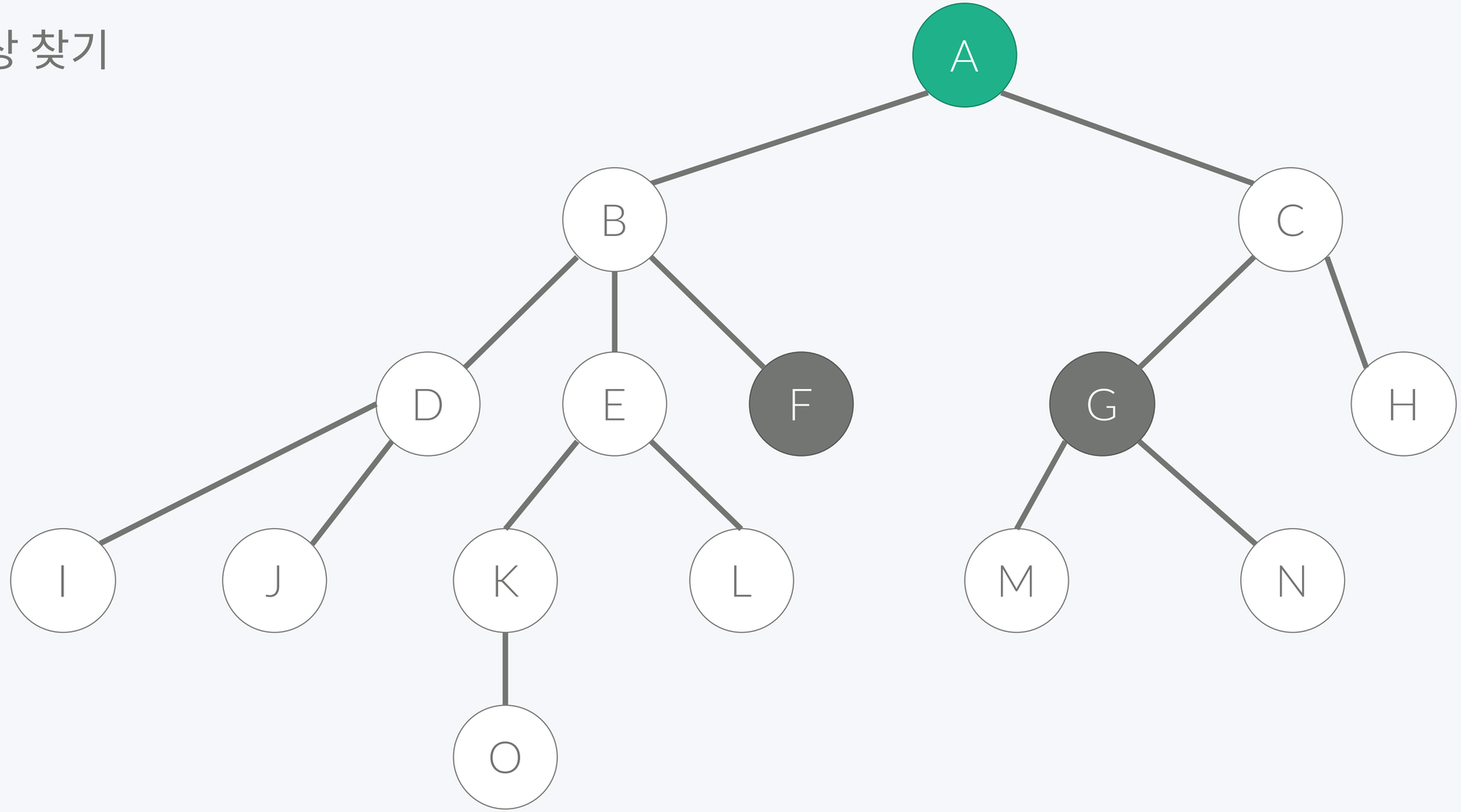
- 가장 가까운 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

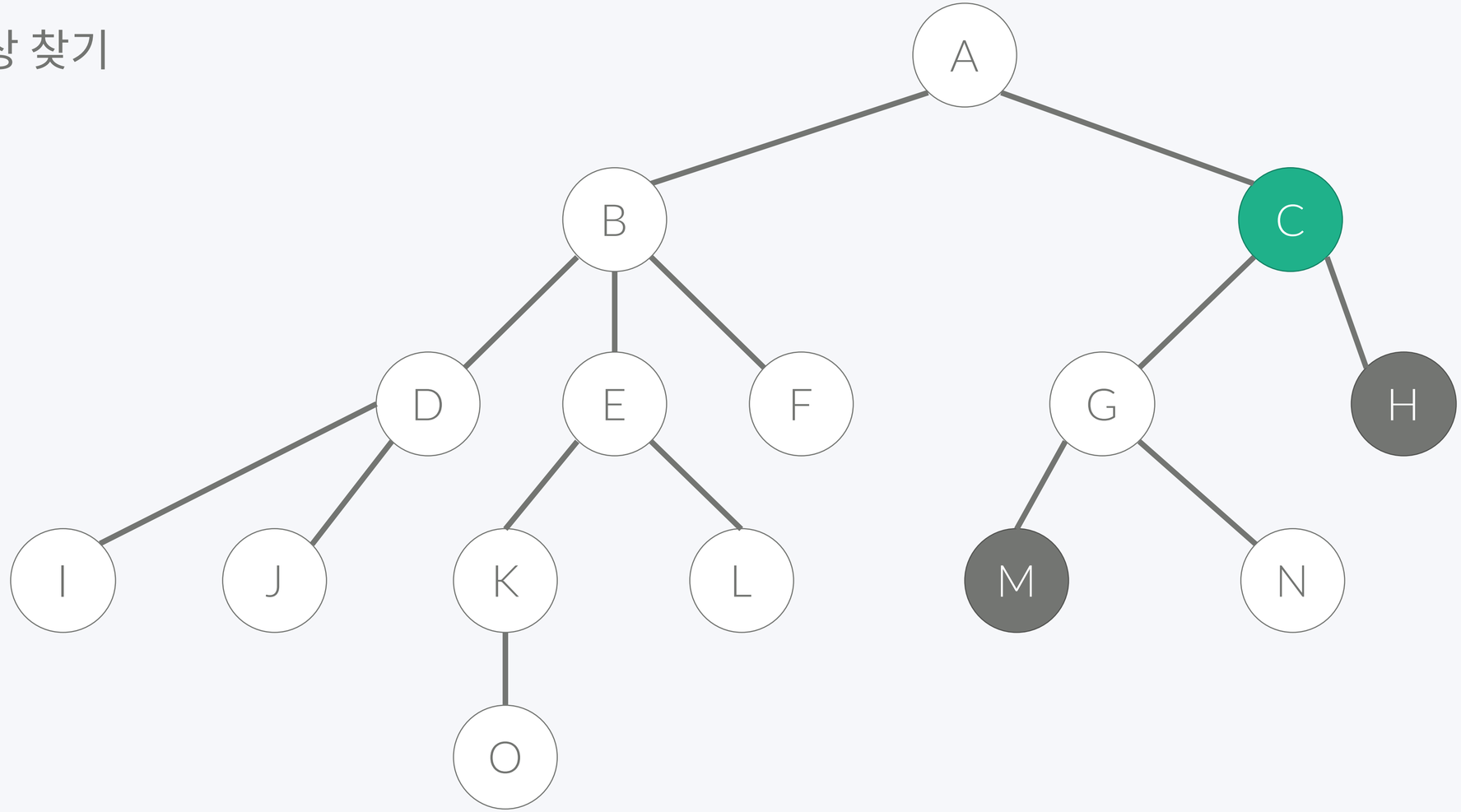
- 가장 가까운 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

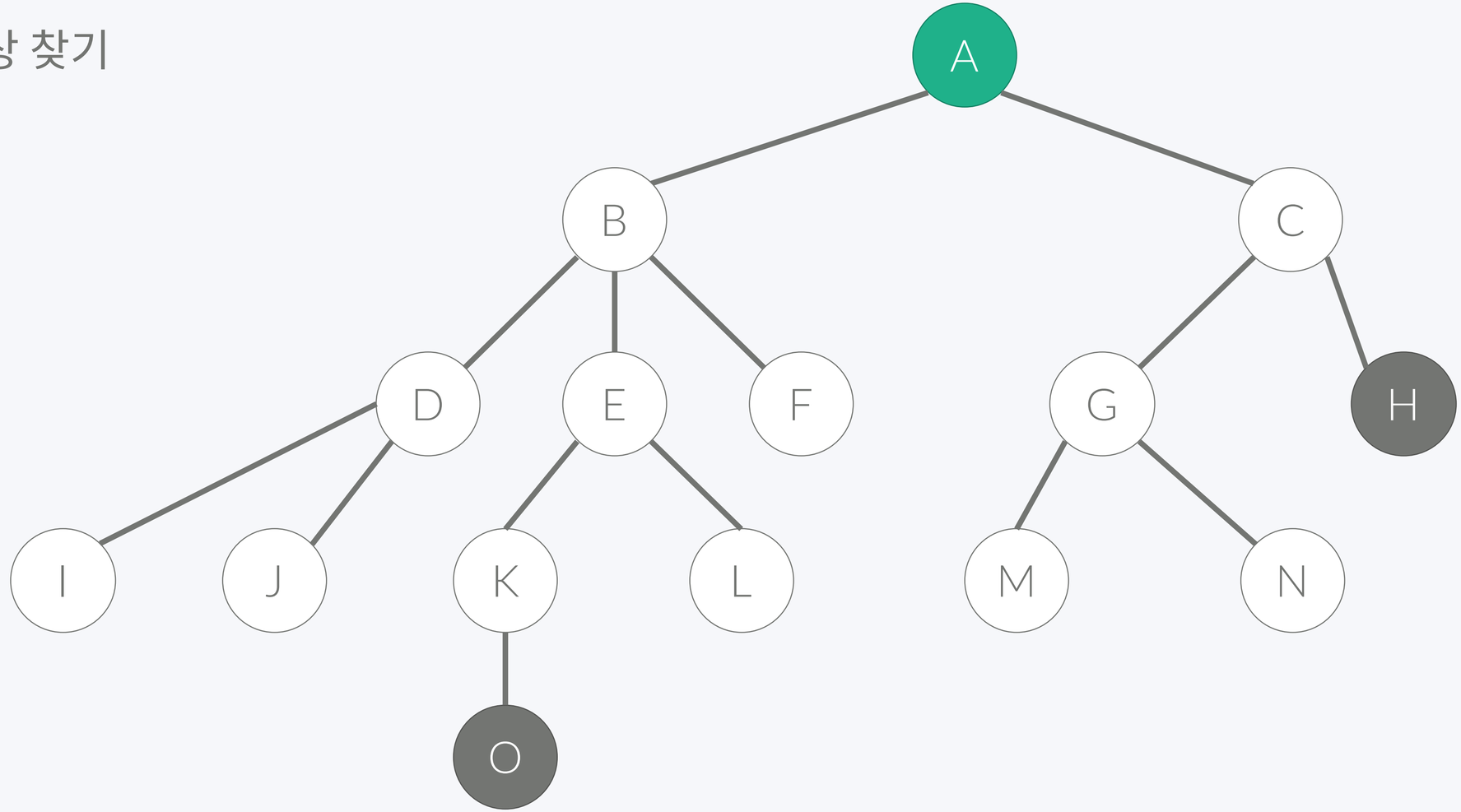
- 가장 가까운 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- 가장 가까운 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- x와 y의 LCA 구하기

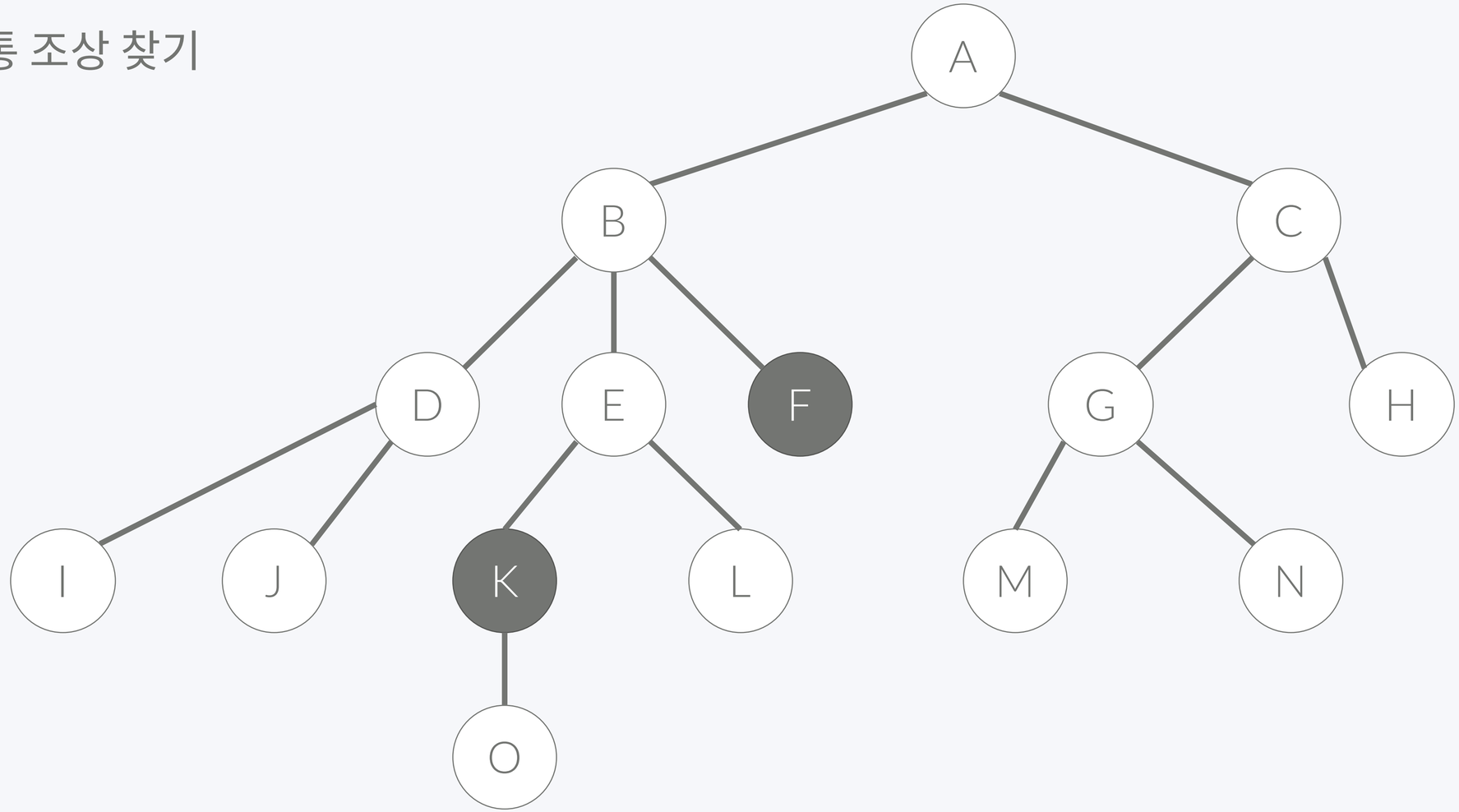
- 두 노드의 레벨이 다르면
- 레벨이 같을 때까지 레벨이 큰 것을 한 칸 씩 위로 올린다

- 두 노드의 레벨이 같아졌으면
- 같은 노드가 될 때까지 한 칸씩 위로 올린다.

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

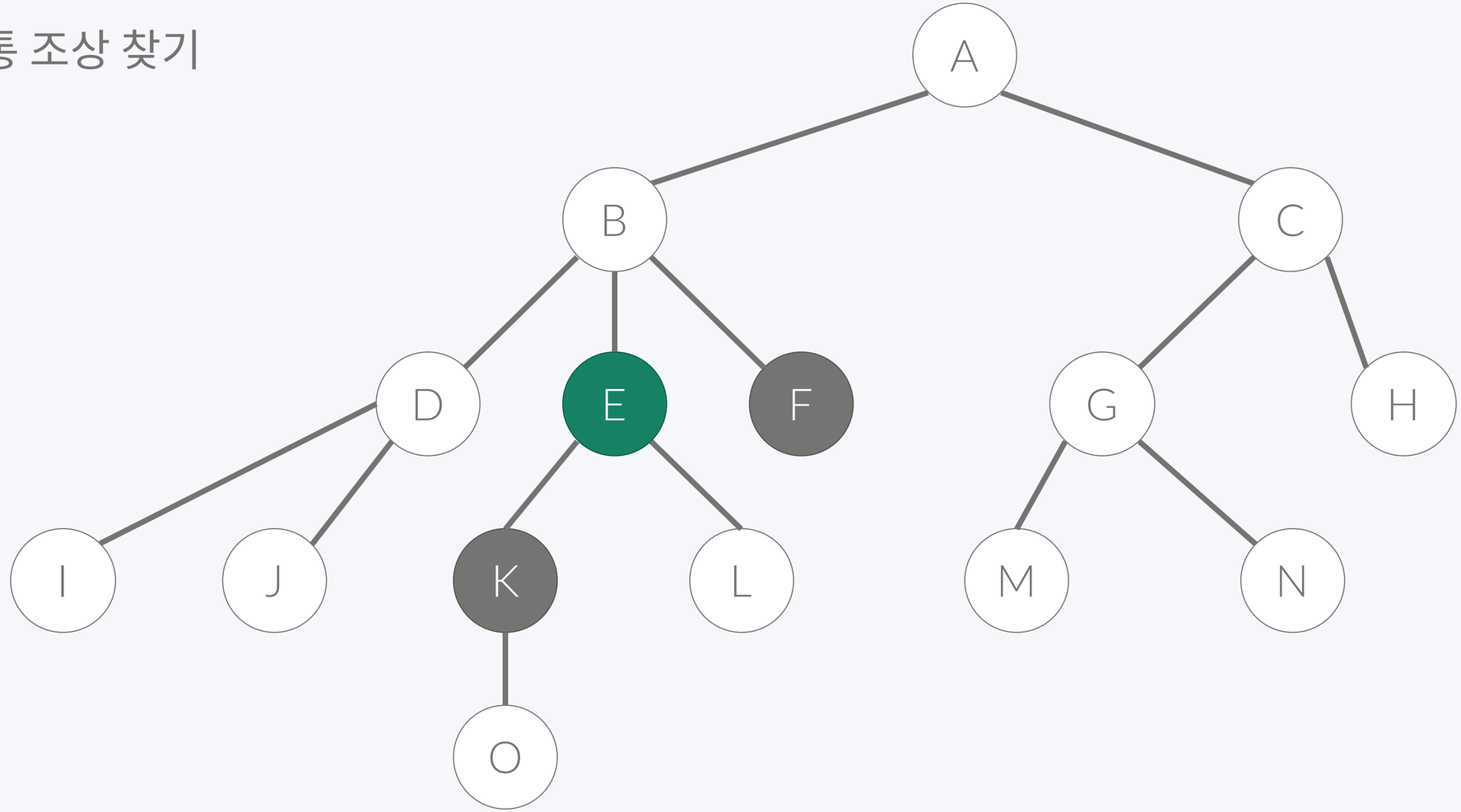
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

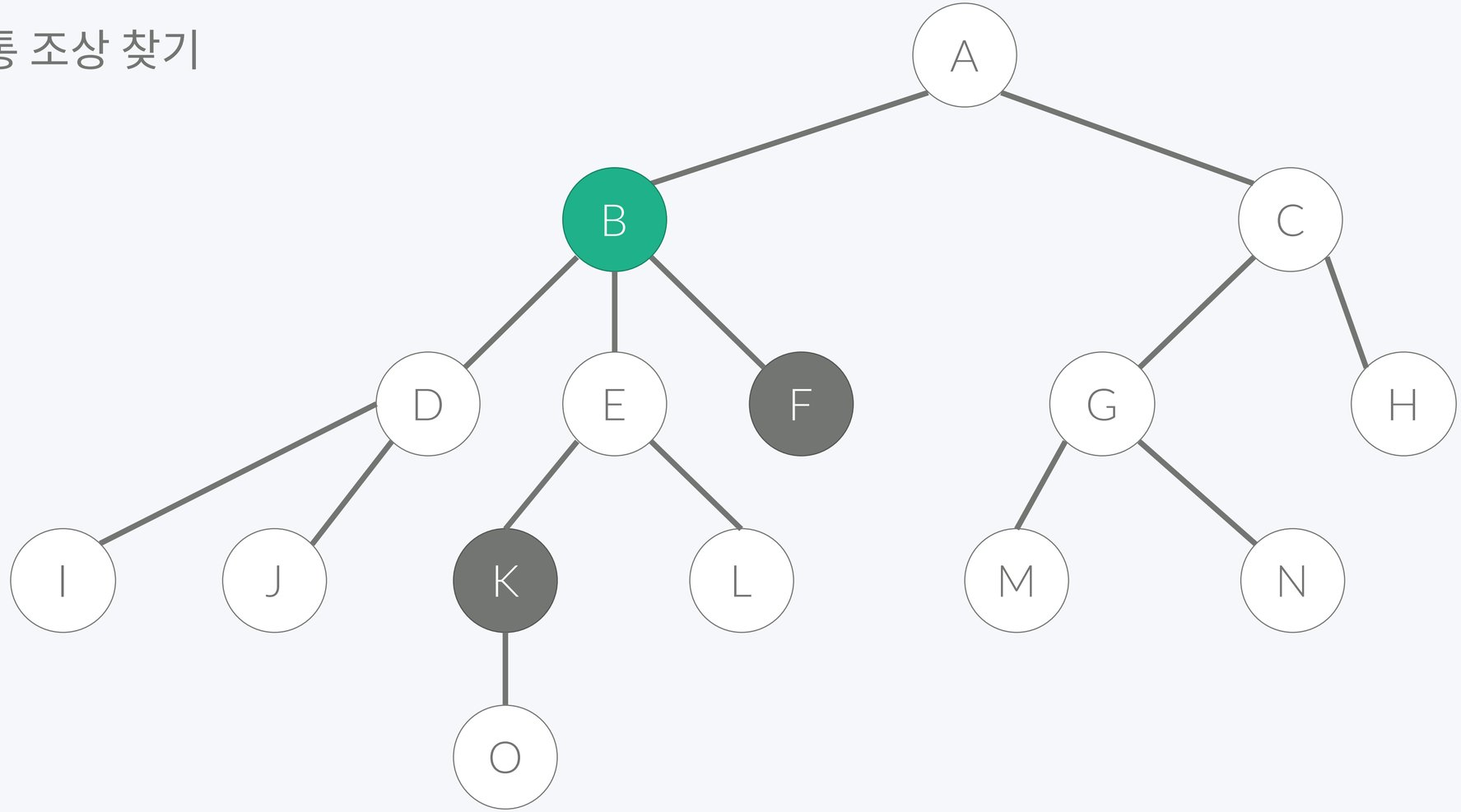
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

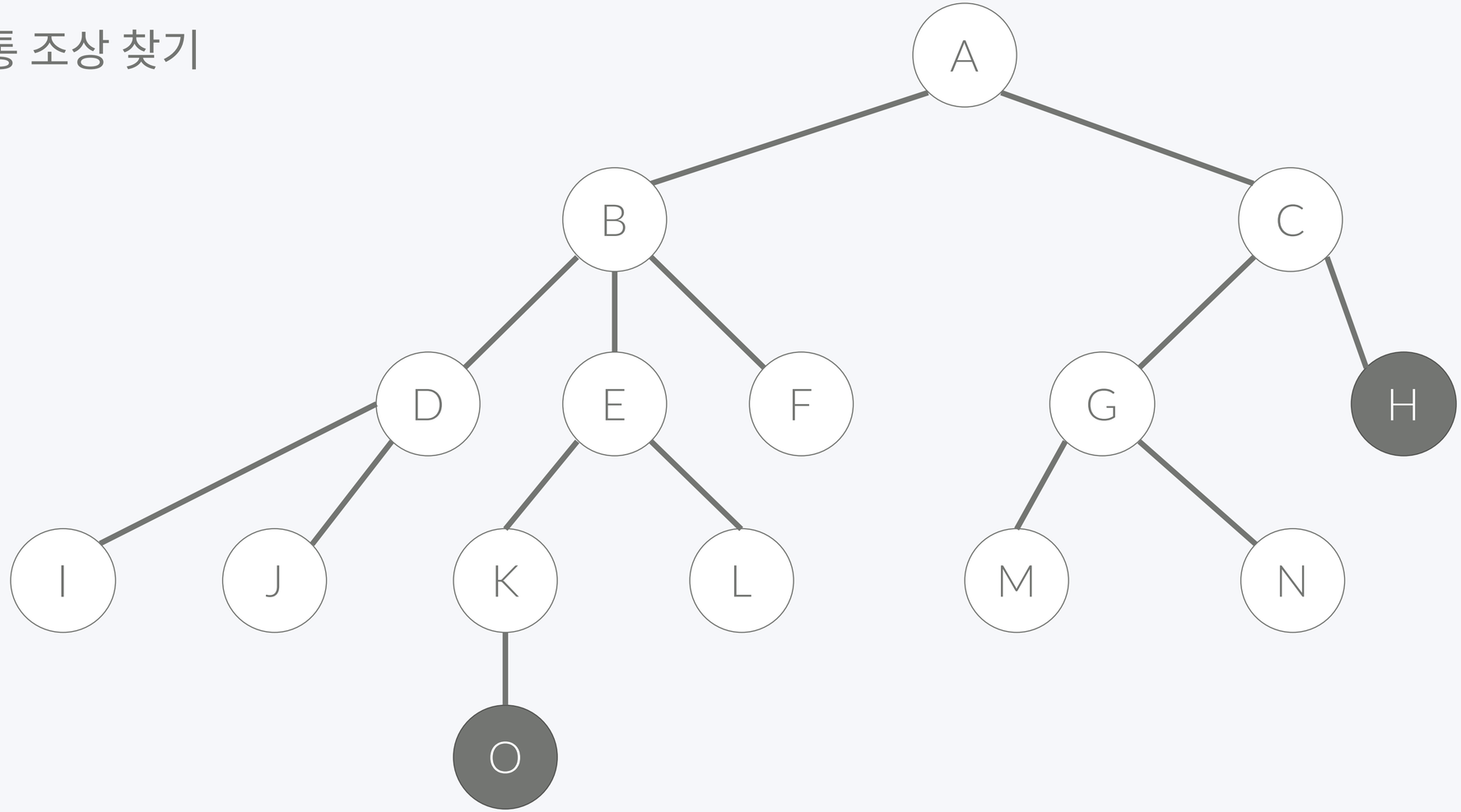
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

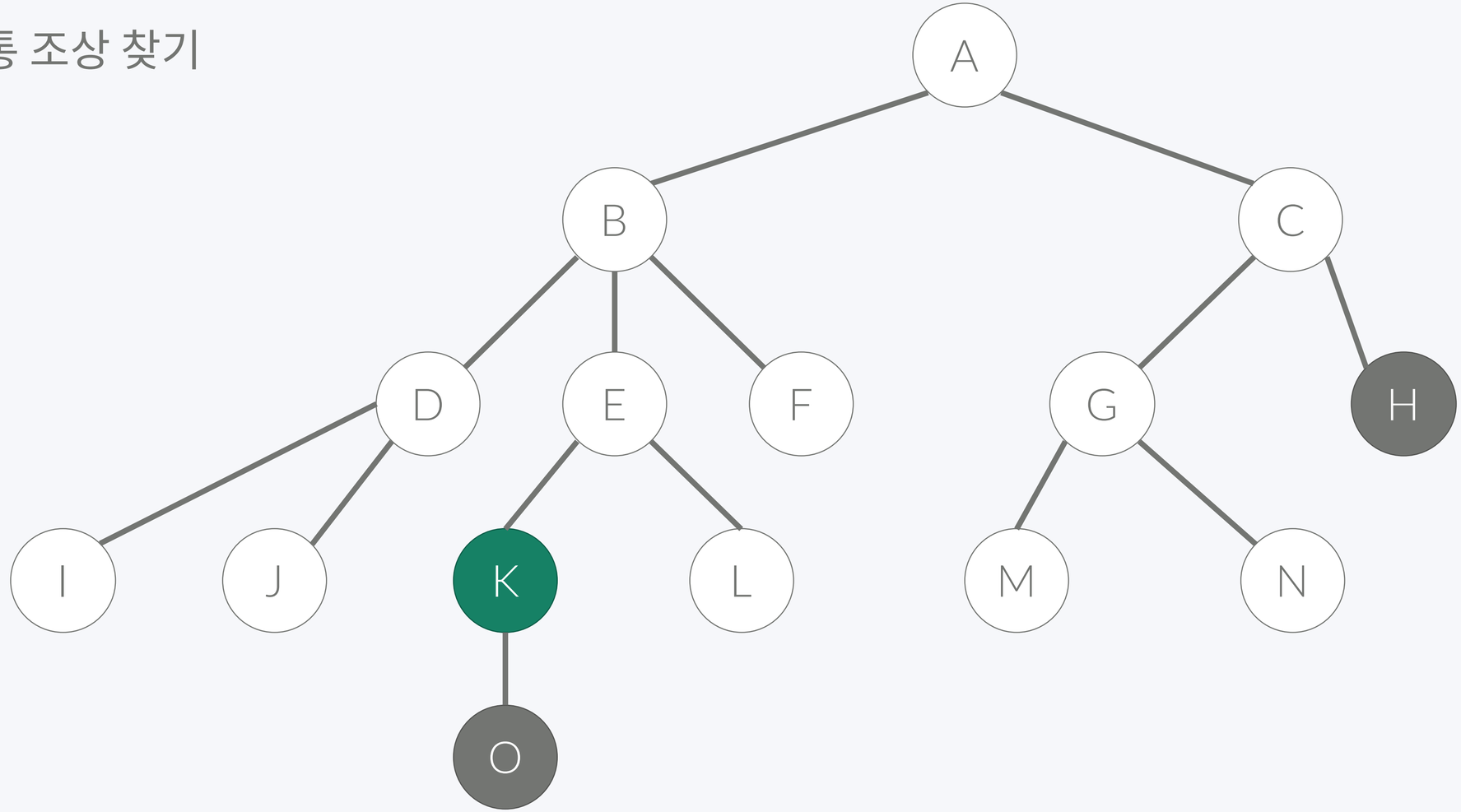
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

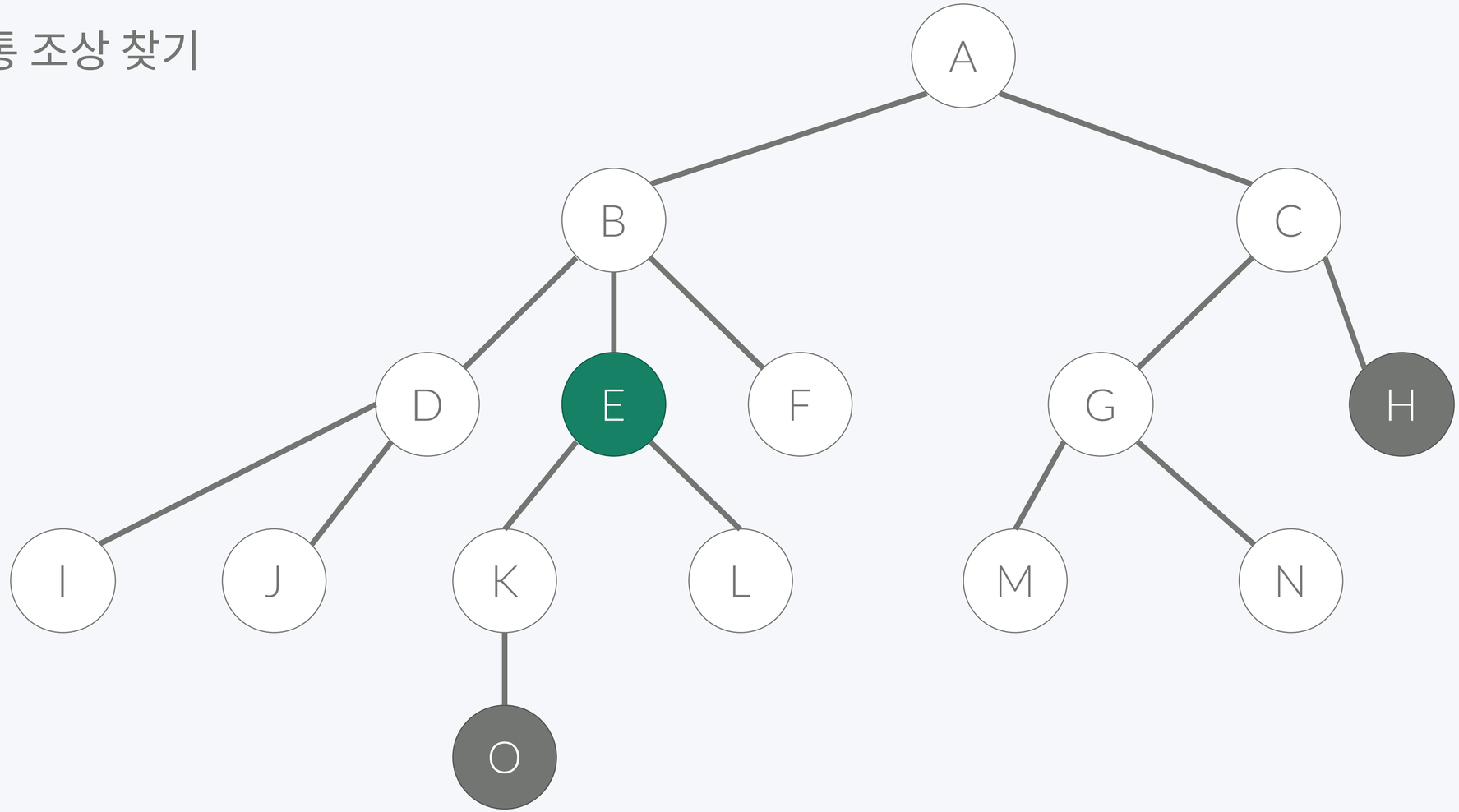
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

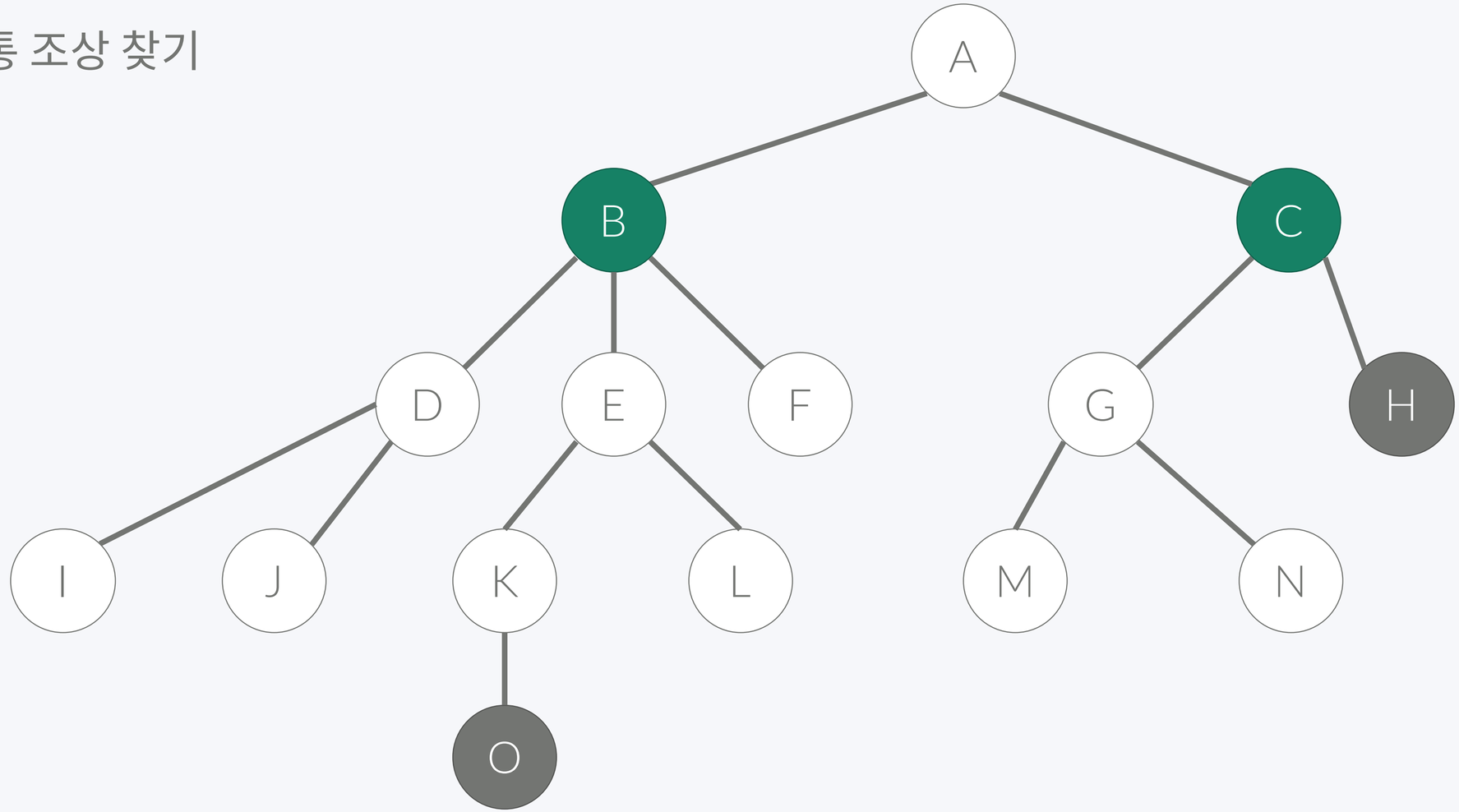
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

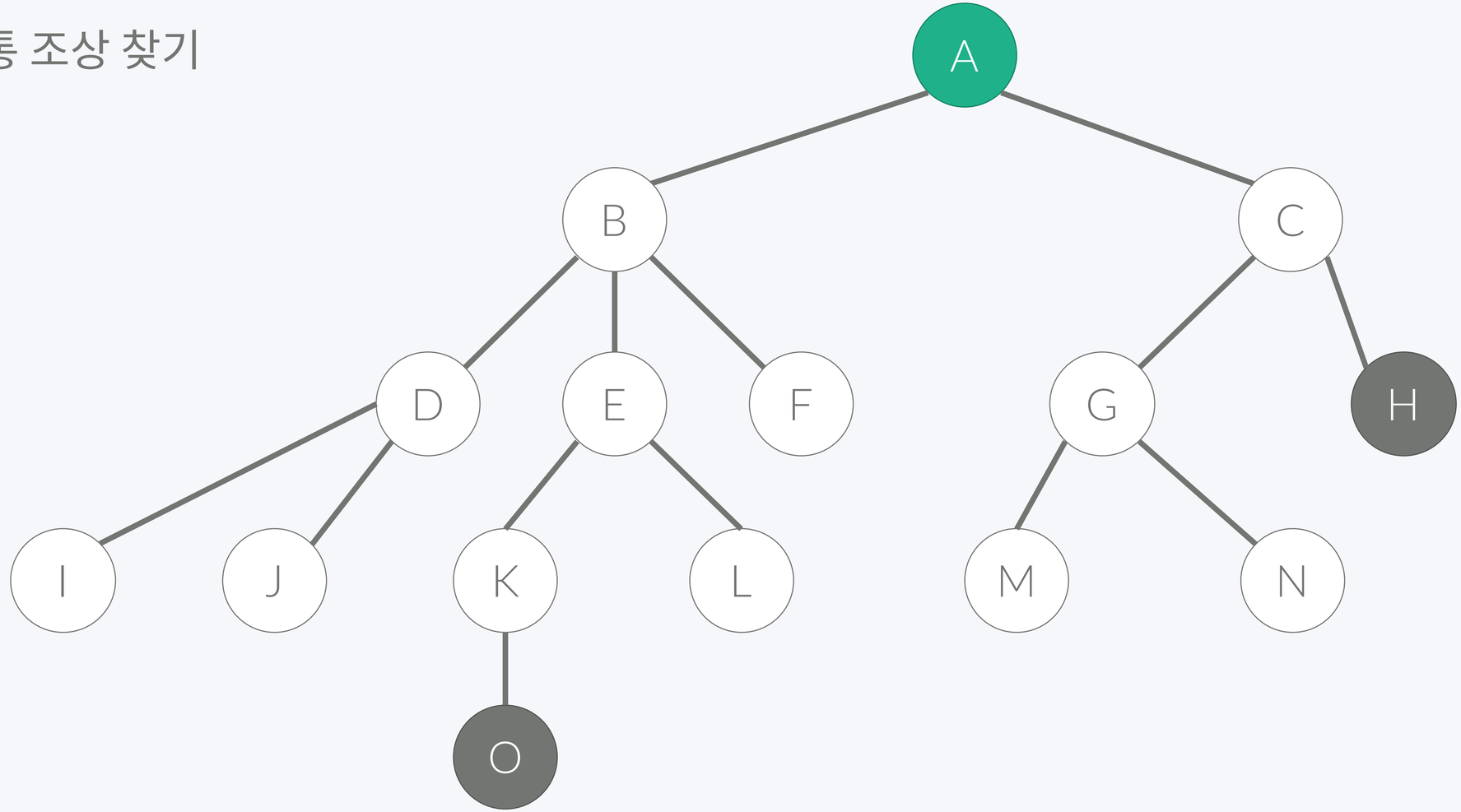
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- 가장 가까운 공통 조상 찾기



LCA

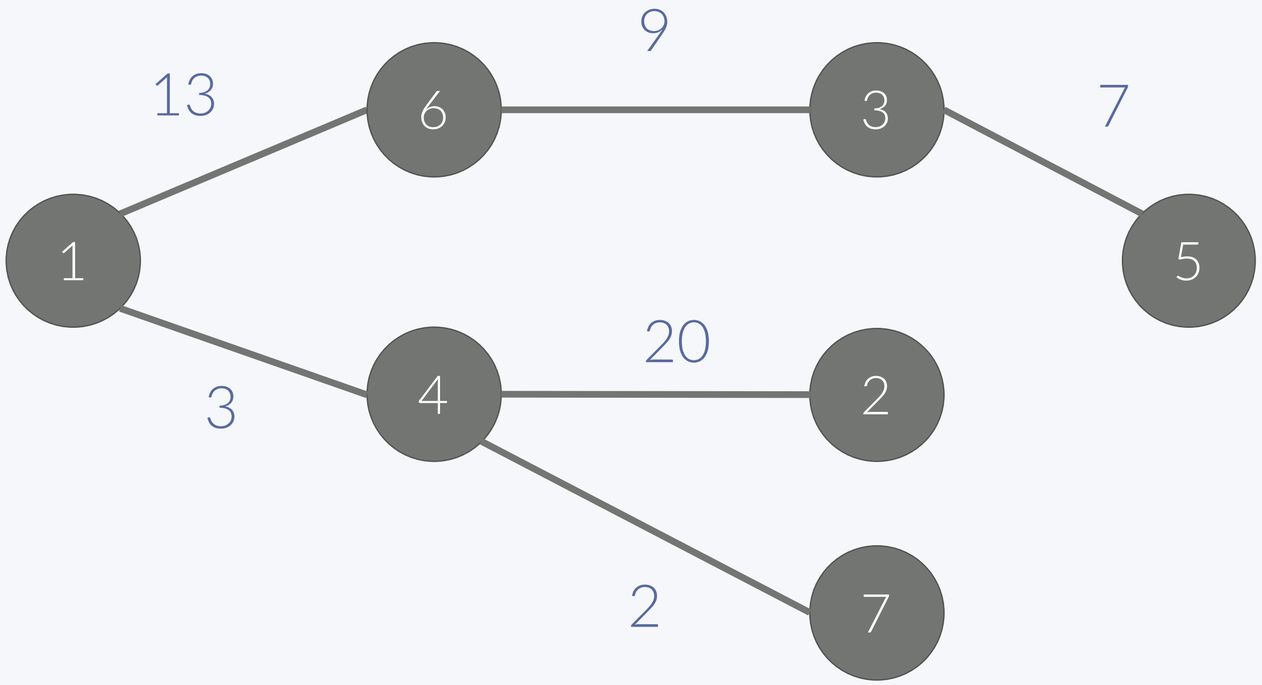
https://www.acmicpc.net/problem/11437

- C/C++: https://gist.github.com/Baekjoon/36771f645013977cfeb1
- Java: https://gist.github.com/Baekjoon/09b029f637354a4a3582

정점들의 거리

https://www.acmicpc.net/problem/1761

- N(2≤N≤40,000)개의 정점으로 이루어진 트리가 주어지고 M(M≤10,000)개의 두 노드 쌍을 입력받을 때 두 노드 사이의 거리 구하기



정점들의 거리

https://www.acmicpc.net/problem/1761

- 트리에서는 모든 정점 쌍 사이의 경로가 1개만 존재하기 때문에
- 어떤 정점 x에서 y로 가는 경로가 곧 최단거리가 된다.

- 거리를 구하는데 필요한 시간: O(N)
- 총 쿼리의 개수: M개
- 시간 복잡도: O(MN)

정점들의 거리

https://www.acmicpc.net/problem/1761

- C/C++: https://gist.github.com/Baekjoon/0a14ffd1c792c49e144e
- C/C++: https://gist.github.com/Baekjoon/c59f726019554fe8a36ab670d9f47ba2
- Java: https://gist.github.com/Baekjoon/b2c4b1b29f518d5d4427

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- 이 방법은 최악의 경우에 시간복잡도가 O(N) 이다.
- Dynamic Programming을 이용해서
- O(lgN) 만에 구할 수 있다.

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

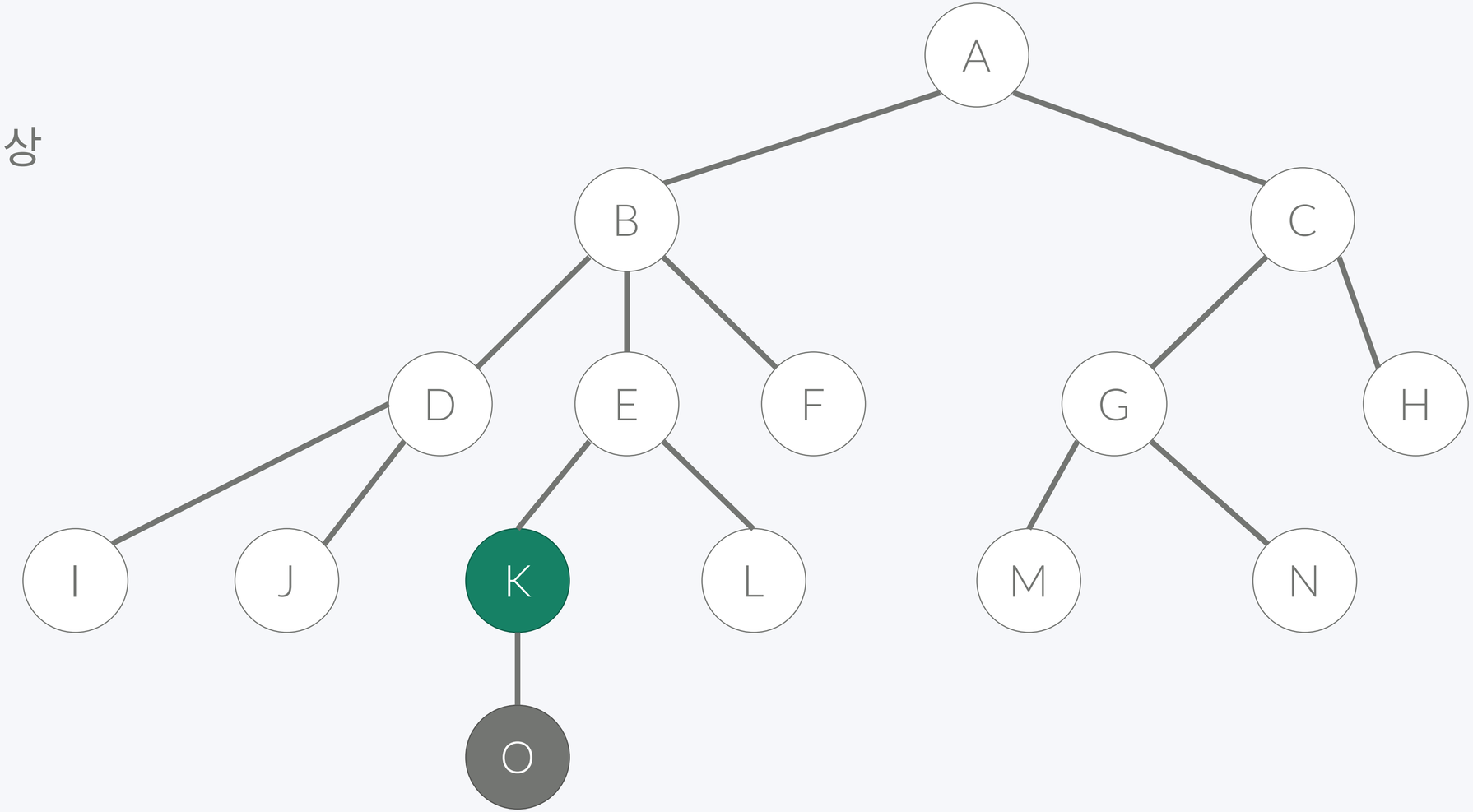
- P[i][j] = 노드 i의 2^j번째 조상

- P[i][0] = Parent[i]
- P[i][j] = P[P[i][j-1]][j-1]

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

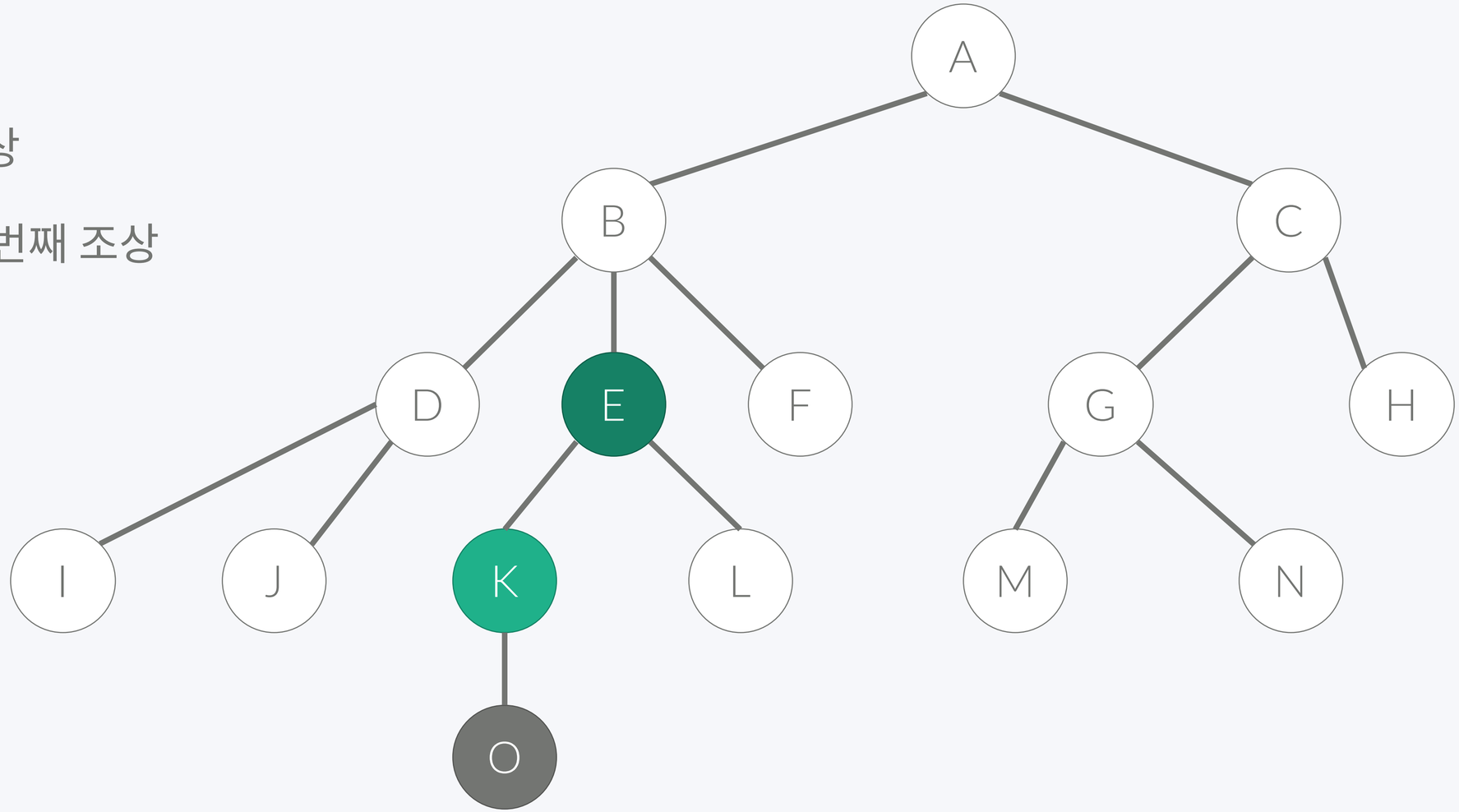
- j = 0
- 2^0 = 1번째 조상



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

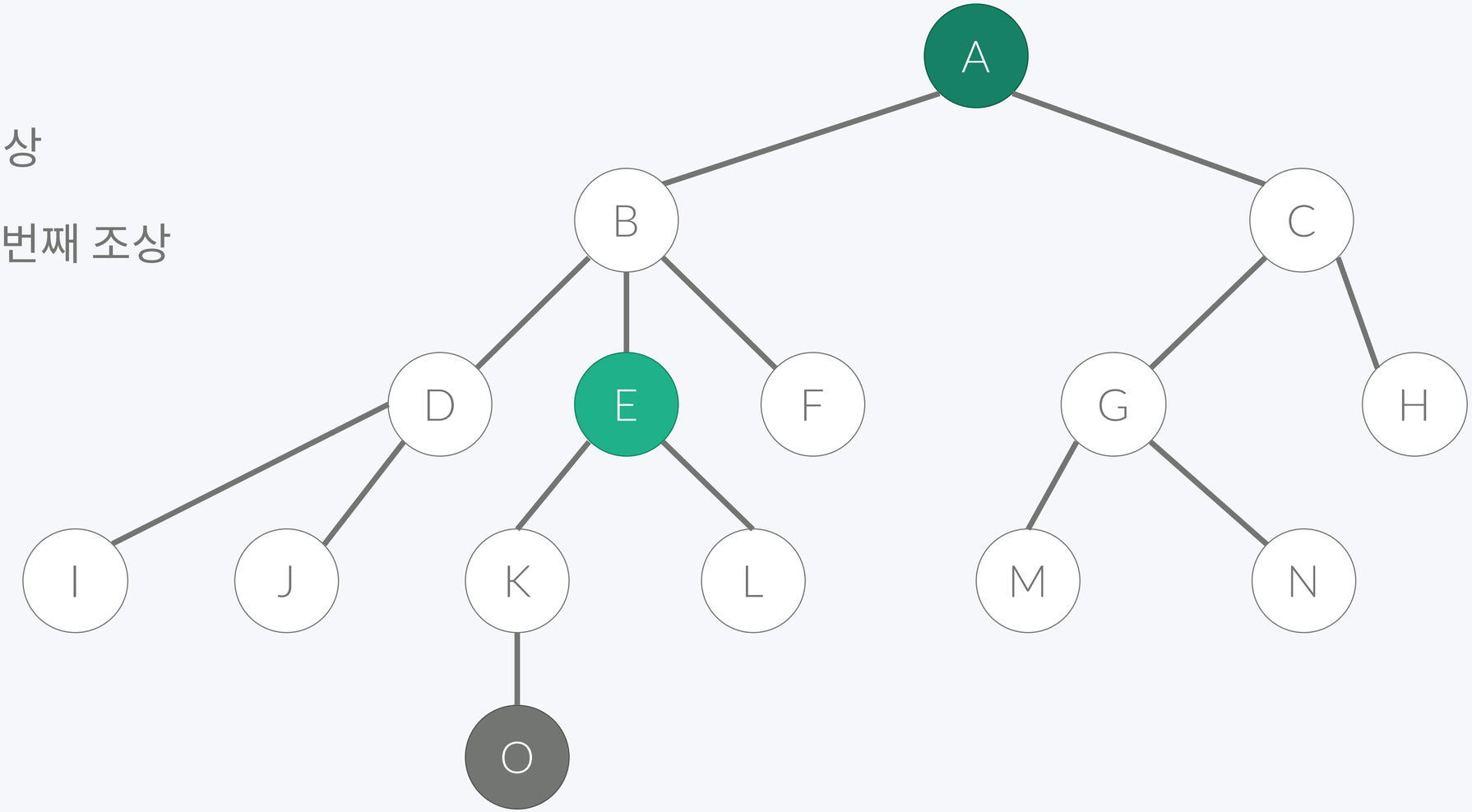
- j = 1
- 2^1= 2번째 조상
- 1번째 조상의 1번째 조상



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- j = 2
- 2^2 = 4번째 조상
- 2번째 조상의 2번째 조상



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

```
for (int i=1; i<=n; i++) {
    p[i][0] = parent[i];
}
```

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

```
for (int j=1; (1<<j) < n; j++) {
    for (int i=1; i<=n; i++) {
        if (p[i][j-1] != 0) {
            p[i][j] = p[p[i][j-1]][j-1];
        }
    }
}
```

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- x와 y의 LCA 구하기

- 두 노드의 레벨이 다르면
- 레벨이 같을 때까지 레벨이 큰 것을 2^k칸 씩 위로 올린다

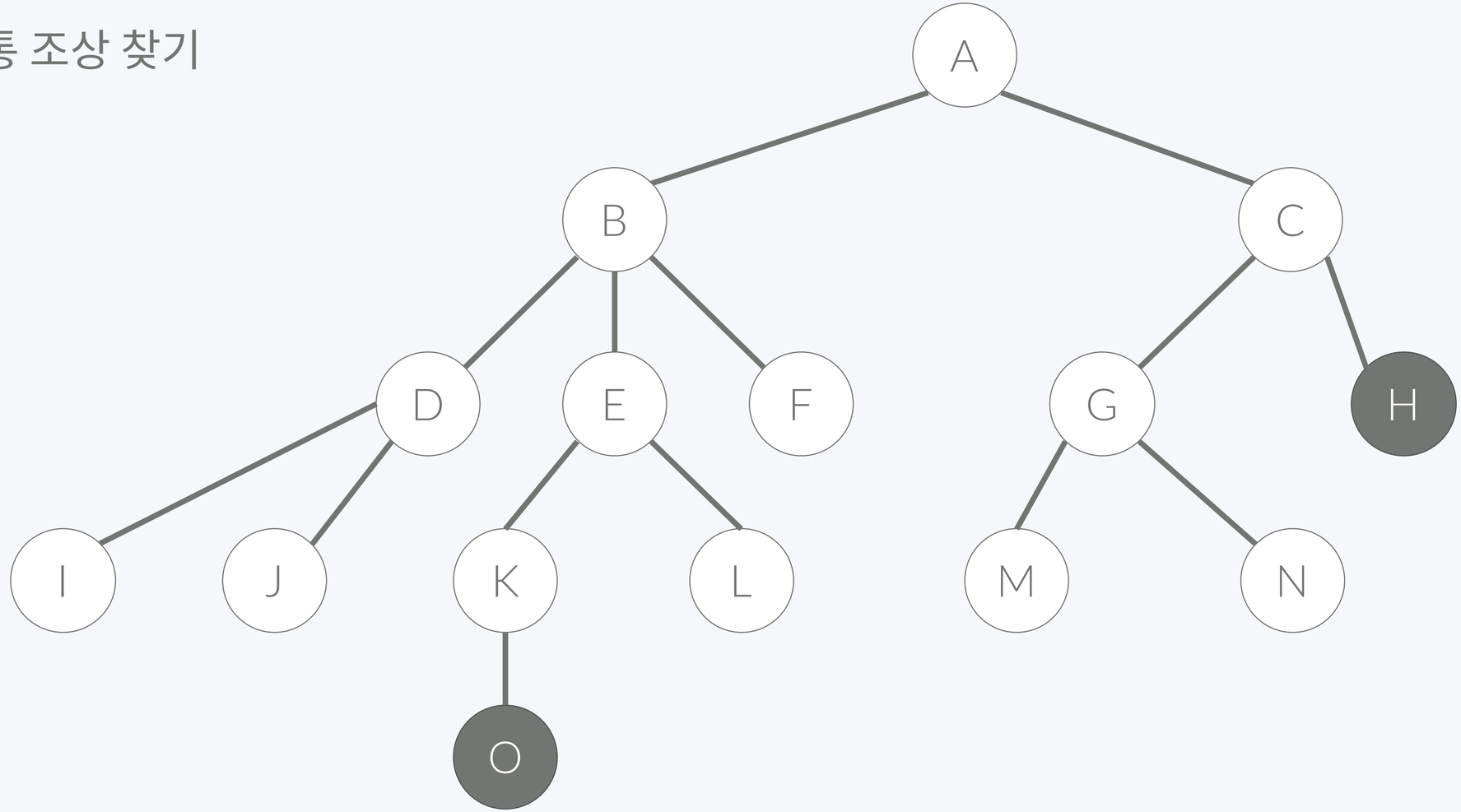
- 두 노드의 레벨이 같아졌으면
- 같은 노드가 되지 않을 때까지 2^k 칸씩 위로 올린다.

- 그 다음 마지막으로 1 칸 올린다.

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

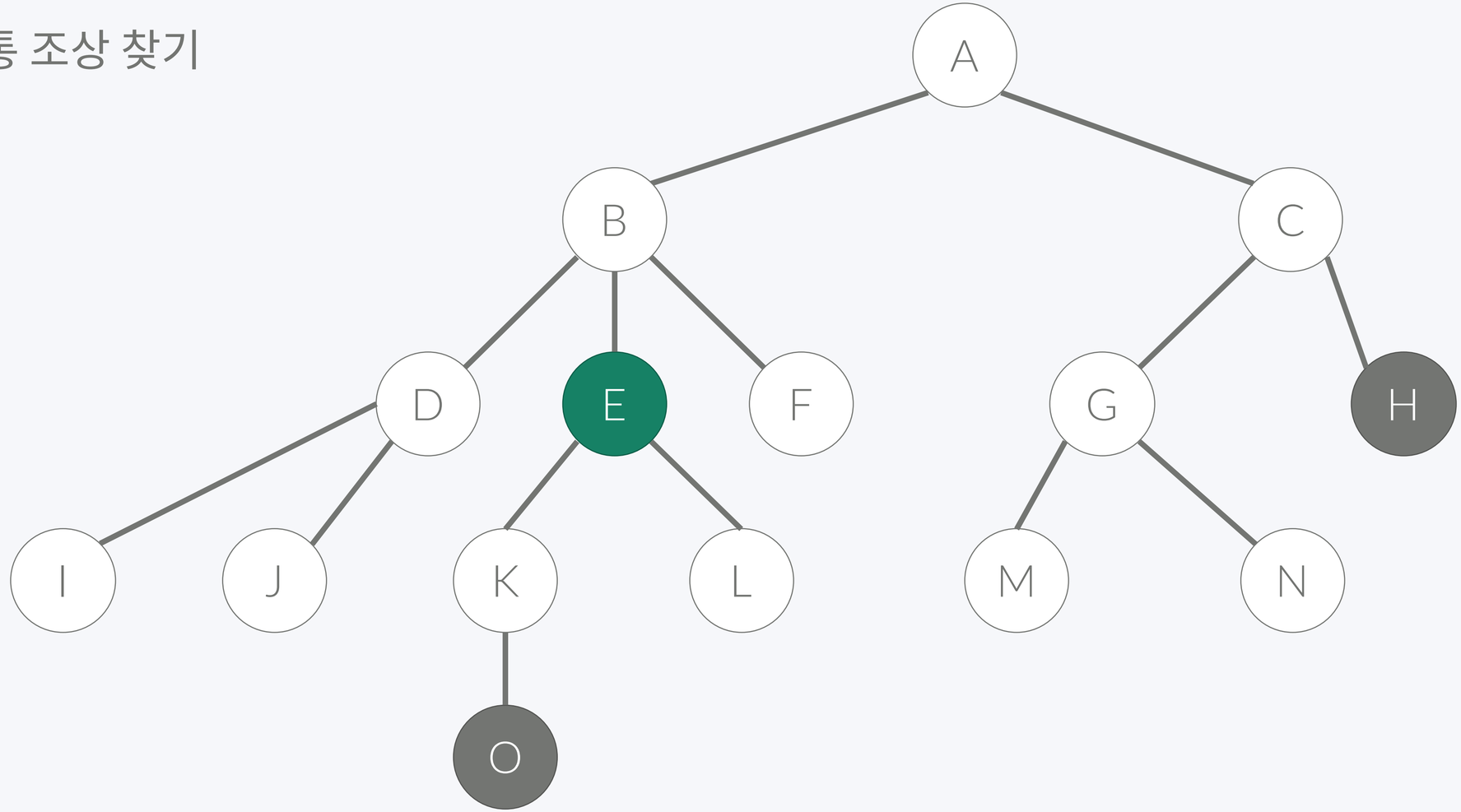
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

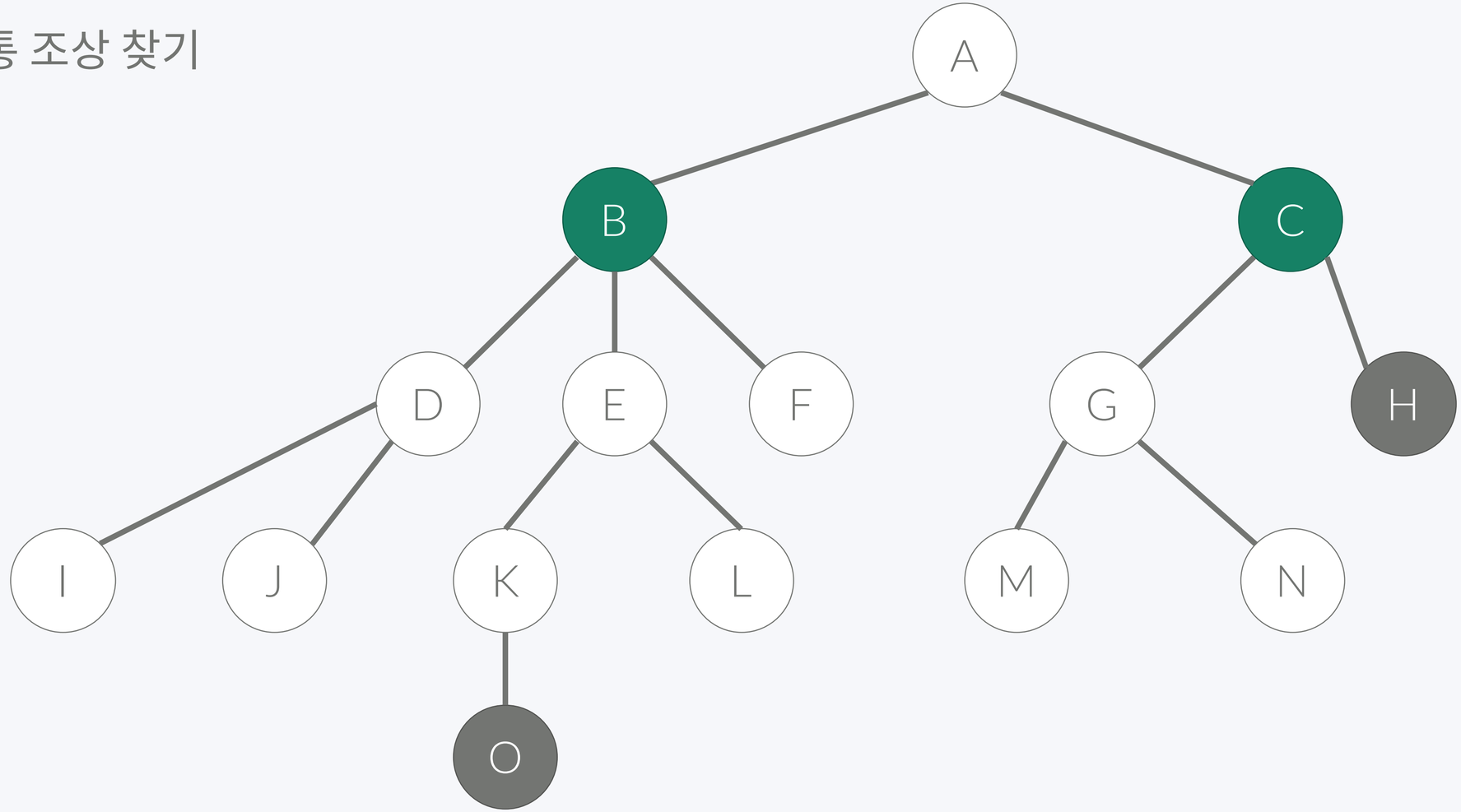
- 가장 가까운 공통 조상 찾기



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- 가장 가까운 공통 조상 찾기



LCA 2

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

- C/C++: https://gist.github.com/Baekjoon/16332082b9536bab2a03
- Java: https://gist.github.com/Baekjoon/808e457beae1abbf3b8e

도로 네트워크

https://www.acmicpc.net/problem/3176

- N개의 도시와 그 도시를 연결하는 N-1개의 도로로 이루어진 도로 네트워크가 있다.
- 모든 도시의 쌍에는 그 도시를 연결하는 유일한 경로가 있고, 각 도로의 길이는 입력으로 주어진다.
- 총 K개의 도시 쌍이 주어진다. 이 때, 두 도시를 연결하는 경로 상에서 가장 짧은 도로의 길이와 가장 긴 도로의 길이를 구하는 프로그램을 작성하시오.

도로 네트워크

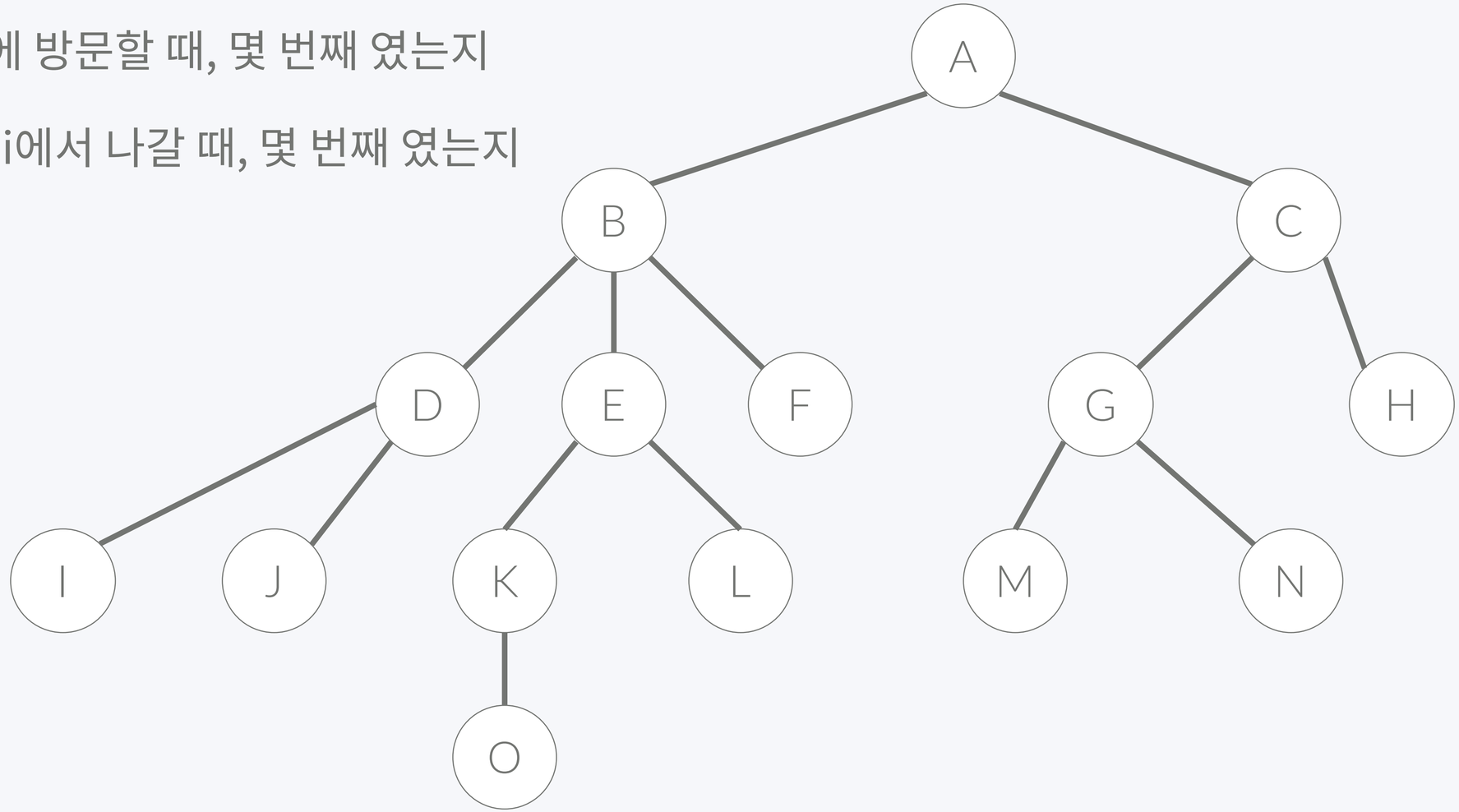
https://www.acmicpc.net/problem/3176

- LCA를 구하는 문제인데
- 배열을 2개 더 만들어야 한다.
- P[i][j] = i의 2^j번째 parent
- len_min[i][j] = i의 2^j번째 parent까지 올라가면서 만나는 모든 도로 중 가장 짧은 것의 길이
- len_max [i][j] = i의 2^j번째 parent까지 올라가면서 만나는 모든 도로 중 가장 긴것의 길이
- C/C++: https://gist.github.com/Baekjoon/4bf3caafe1bdd5fbb8b3

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

- tin[i] = dfs로 i에 방문할 때, 몇 번째 였는지
- tout[i] = dfs로 i에서 나갈 때, 몇 번째 였는지



가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

```
void dfs(int v, int parent) {
    tin[v] = ++timer;
    p[v][0] = parent;
    for (int i=1; i<=l; i++) {
        p[v][i] = p[p[v][i-1]][i-1];
    }
    for (int to : a[v]) {
        if (to != parent) {
            dfs(to, v);
        }
    }
    tout[v] = ++timer;
}
```

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

```
bool upper(int u, int v) {
    return (tin[u] <= tin[v] && tout[u] >= tout[v]);
}
```

가장 가까운 공통 조상 찾기

LCA (Lowest Common Ancestor)

```
int lca(int u, int v) {
    if (upper(u, v)) return u;
    if (upper(v, u)) return v;
    for (int i=l; i>=0; i--) {
        if (!upper(p[u][i], v)) {
            u = p[u][i];
        }
    }
    return p[u][0];
}
```

LCA 2

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

- C/C++: https://gist.github.com/Baekjoon/5143d9de7ad109ef87ba4d788a619353