

05

트리

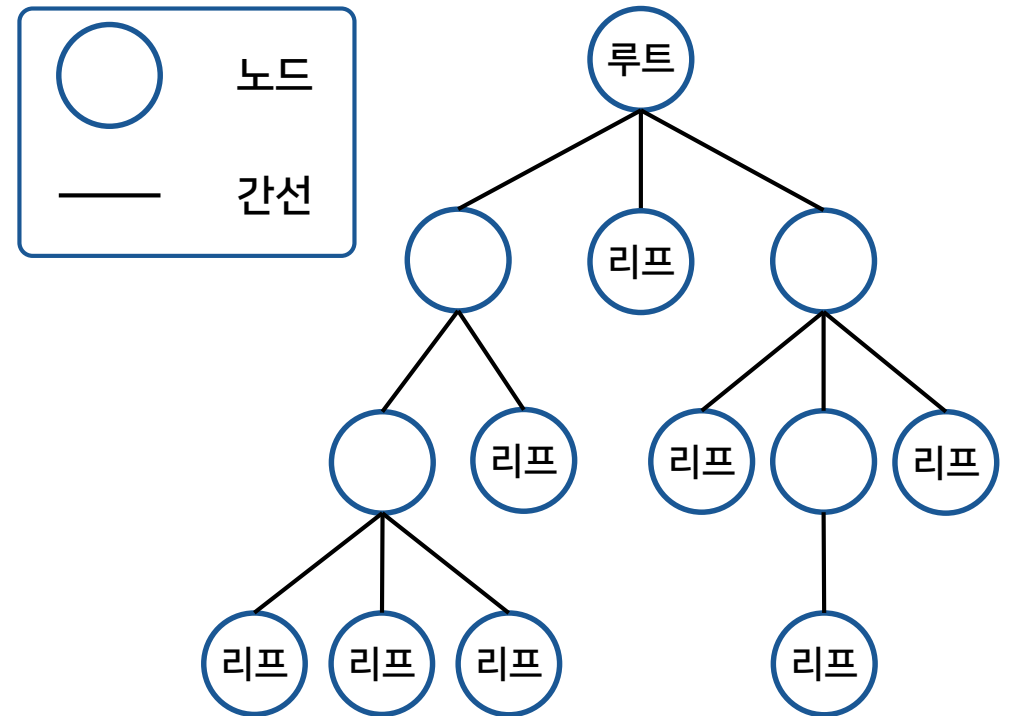
01 트리

02 완전 이진 트리

03 완전 이진 트리 탐색

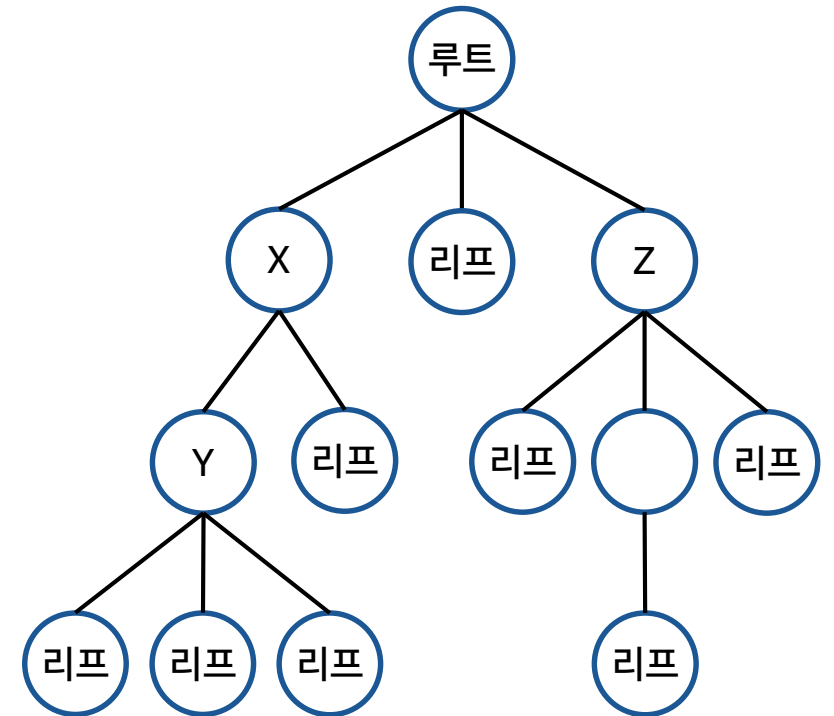
트리

- 트리를 구성하는 요소는 노드(Node)와 간선(Edge)으로 이루어져 있다.
- 각각의 노드는 간선으로 연결되어 있다.
- 루트 노드(Root node)
 - 트리의 가장 윗부분의 노드
- 리프 노드(Leaf node)
 - 자식이 없는 노드



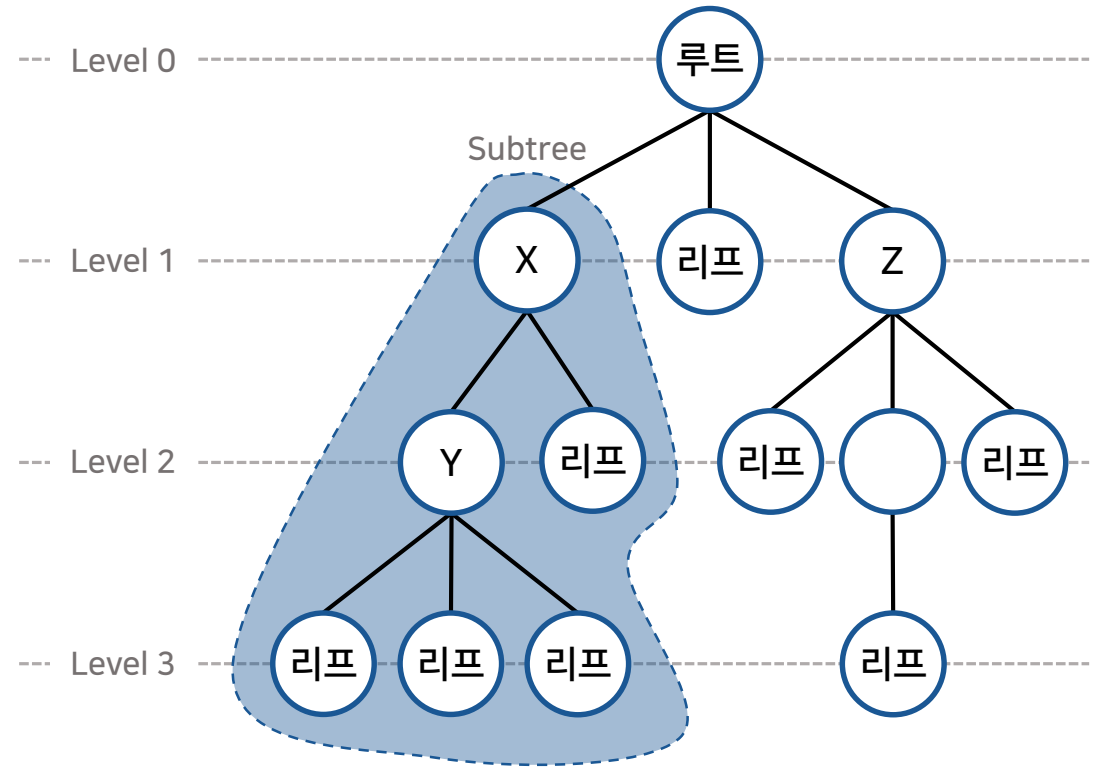
트리

- 자식 노드(Child node)
 - 어떤 노드의 간선으로 연결된 아래쪽 노드
 - Y 노드는 X 노드의 자식 노드
- 부모 노드(Parent node)
 - 어떤 노드의 간선으로 연결된 위쪽 노드
 - X 노드는 Y 노드의 부모 노드
- 형제 노드(Sibling node)
 - 부모가 같은 노드
 - Z 노드는 X 노드의 형제 노드



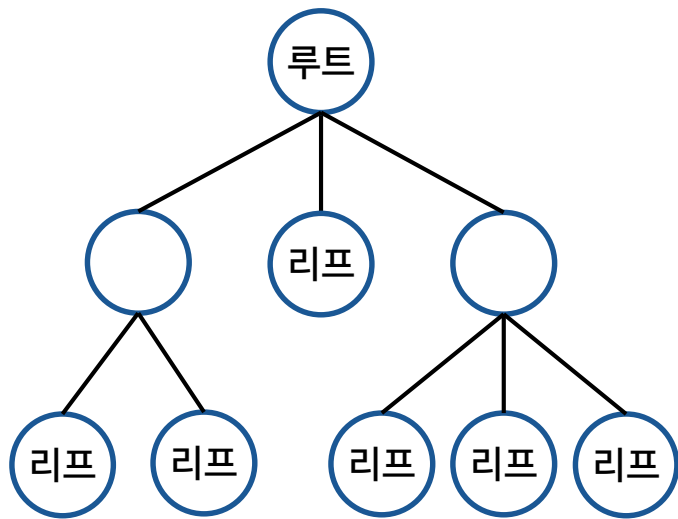
트리

- 레벨(Level)
 - 루트로부터 얼마나 떨어져 있는 지를 나타낸 값
 - X, Z 노드의 레벨 : 1
- 차수(Degree)
 - 노드가 갖는 자식의 수
 - Y 노드의 차수 : 3
- 깊이(Depth) 혹은 높이(Height)
 - 루트에서 가장 멀리 떨어진 리프까지의 거리
 - 트리의 깊이 : 3
- 서브 트리(Subtree)
 - 트리 안에서 다시 어떤 노드를 루트로 정하고 그 자손으로 이루어진 트리

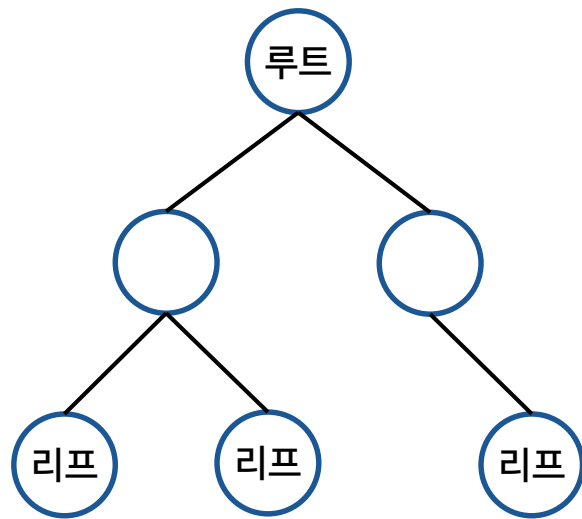


이진 트리

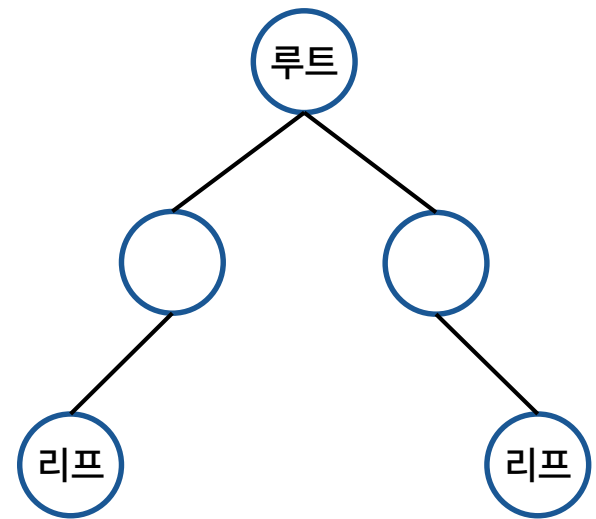
- 각 노드가 최대 두 개의 자식(왼쪽, 오른쪽 자식)을 갖는 트리
- 두 자식 중 한쪽이 없거나 둘 다 없는 경우도 포함



이진 트리 X



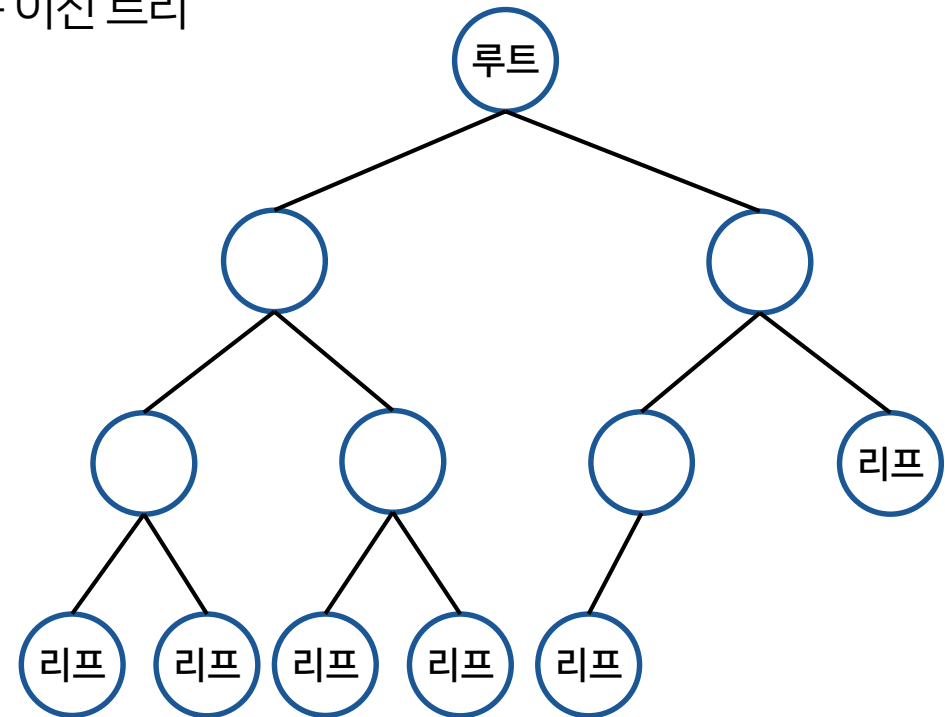
이진 트리



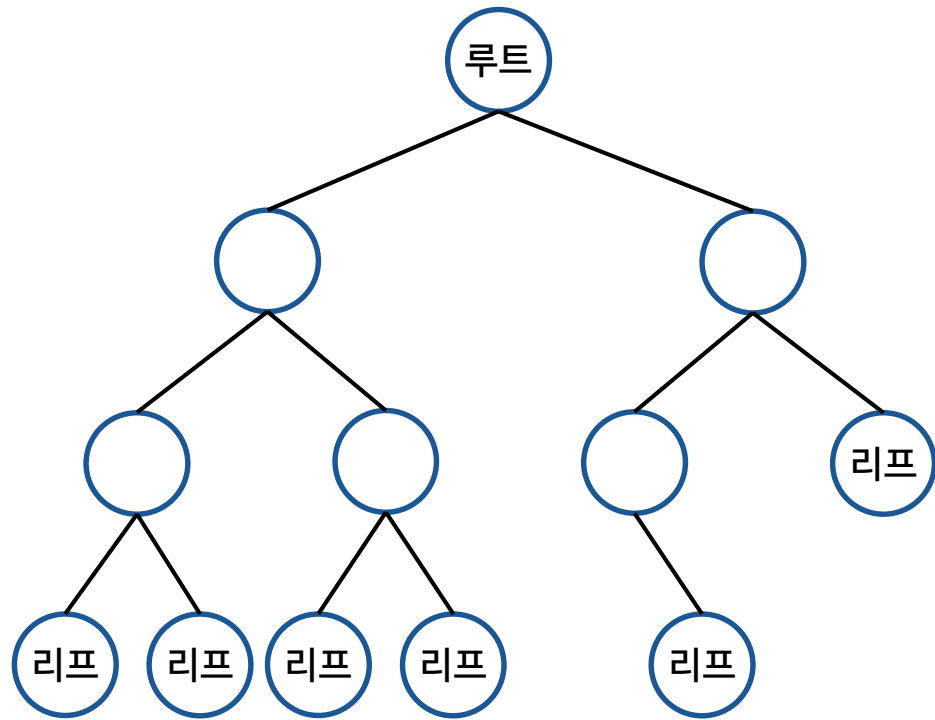
이진 트리

완전 이진 트리

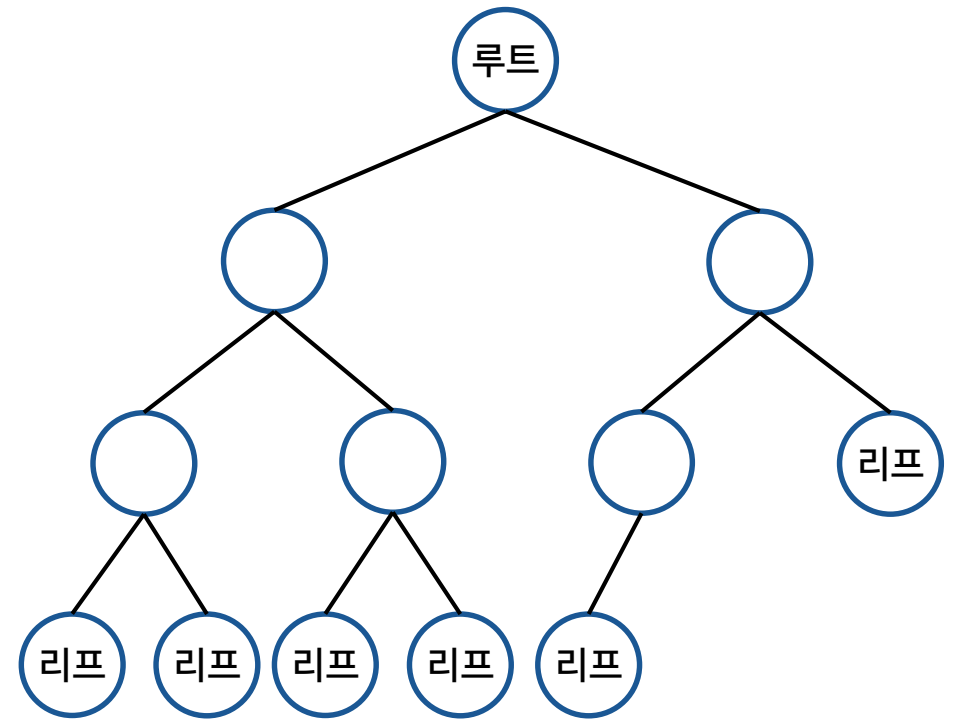
- 루트에서 아래쪽 레벨로 내려가는 노드가 빠짐없이 채워져 있고,
- 같은 레벨에서 왼쪽에서 오른쪽으로 노드가 빠짐없이 채워져 있는 이진 트리
 - 즉, 마지막 레벨을 제외한 모든 레벨의 노드가 완전히 채워져 있고,
 - 마지막 레벨의 노드는 왼쪽에서 오른쪽으로 채워져 있는 이진 트리



완전 이진 트리



완전 이진 트리 X

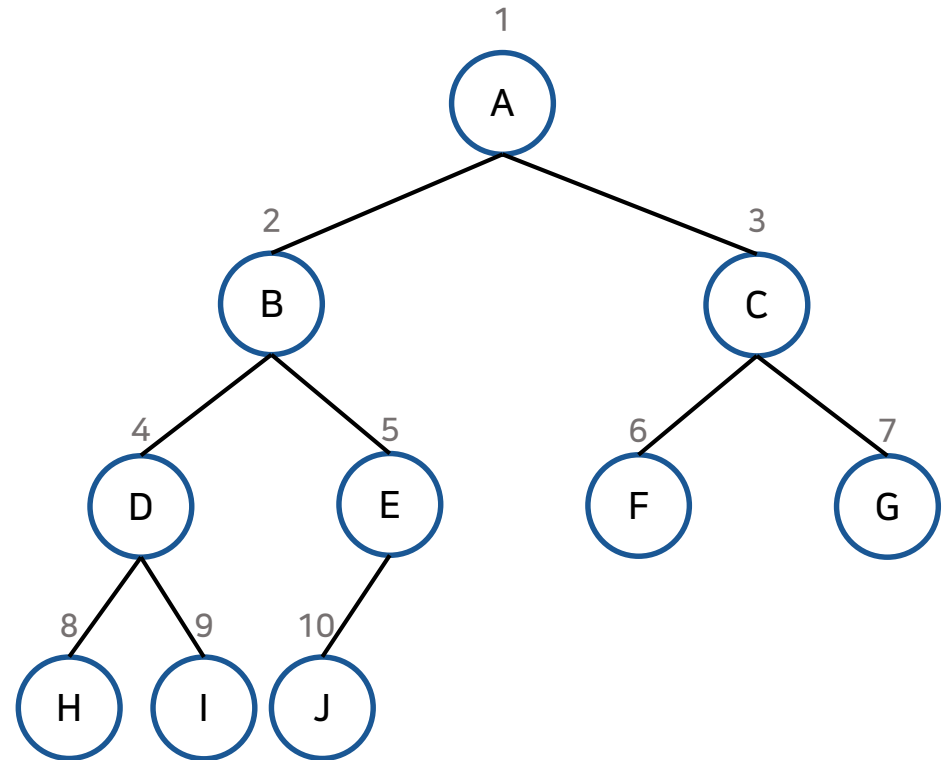


완전 이진 트리

완전 이진 트리

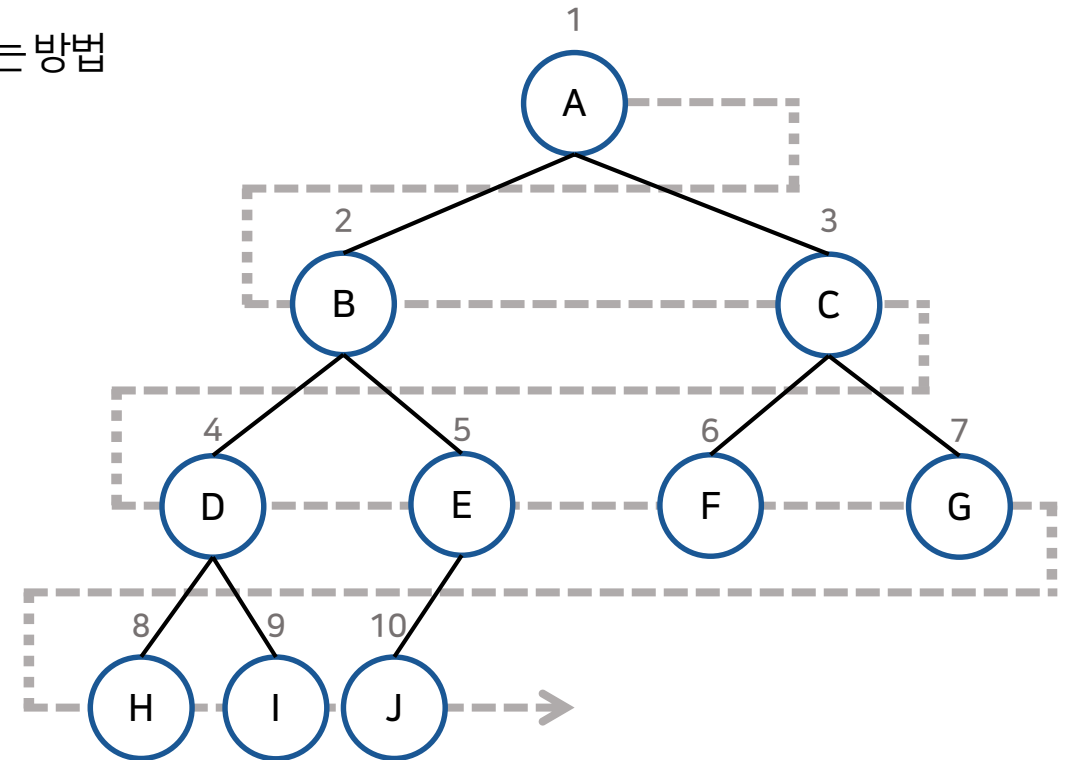
- 완전 이진 트리는 1차원 배열로 구현 가능
 - 인덱스 1부터 시작하는 배열에 순차적으로 값 저장
 - i 번째 노드의 자식 노드의 인덱스는 $2*i$, $2*i+1$
 - i 번째 노드의 부모 노드의 인덱스는 $i/2$

	A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9	10



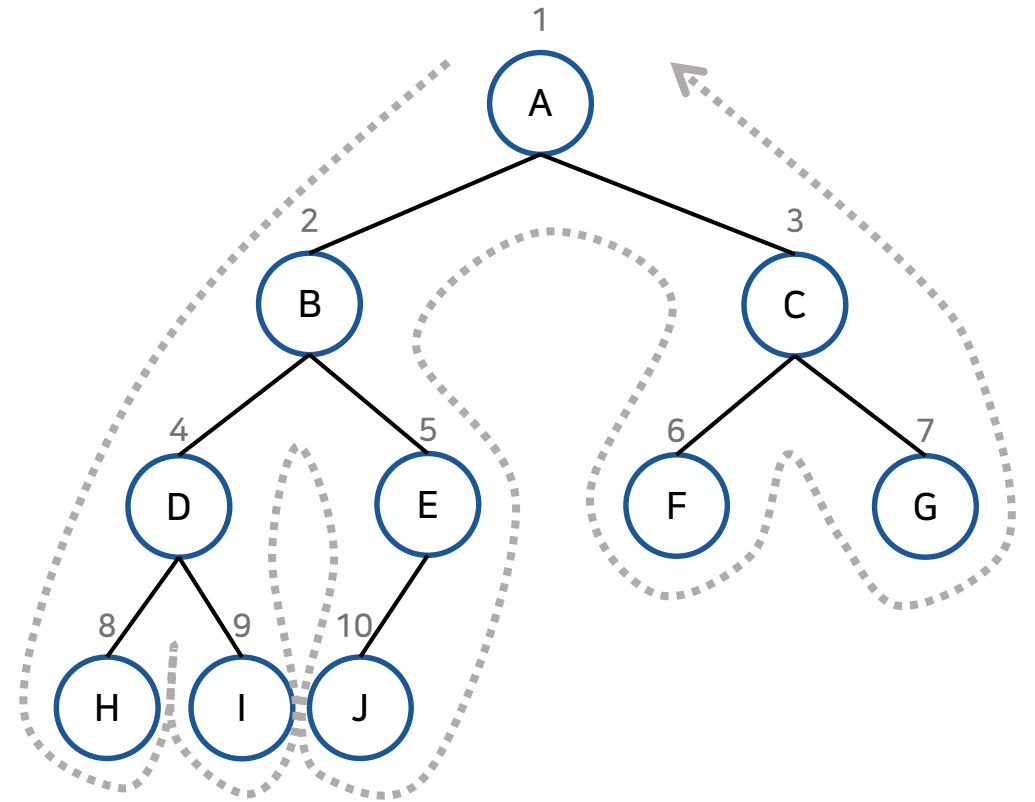
완전 이진 트리 탐색

- 너비 우선 탐색(Breadth-First Search)
 - 낮은 레벨에서 시작해 왼쪽에서 오른쪽 방향으로 노드를 탐색하는 방법
 - $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow J$



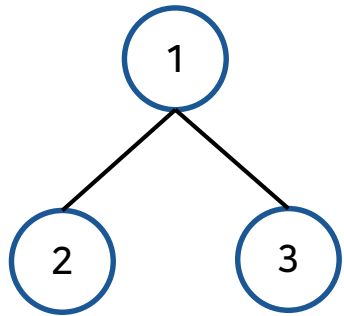
완전 이진 트리 탐색

- 깊이 우선 탐색(Depth-First Search)
 - 리프 노드까지 아래로 내려가면서 탐색하는 방법
 - 리프 노드에 도달하면 부모 노드로 돌아간다.
 - $A \rightarrow B \rightarrow D \rightarrow H \rightarrow I \rightarrow E \rightarrow J \rightarrow C \rightarrow F \rightarrow G$

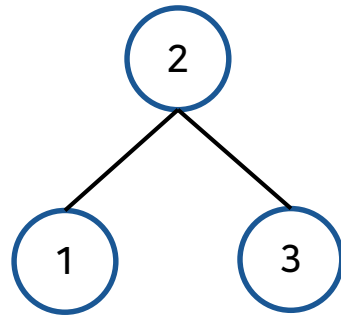


완전 이진 트리 탐색

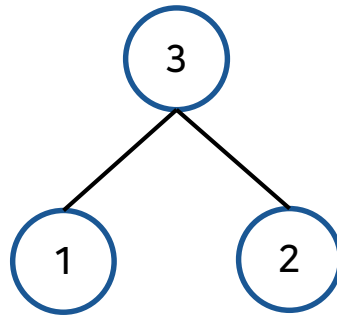
- 깊이 우선 탐색(Depth-First Search)
 - 부모, 자식 노드의 방문 순서에 따라 전위, 중위, 후위 순회 탐색



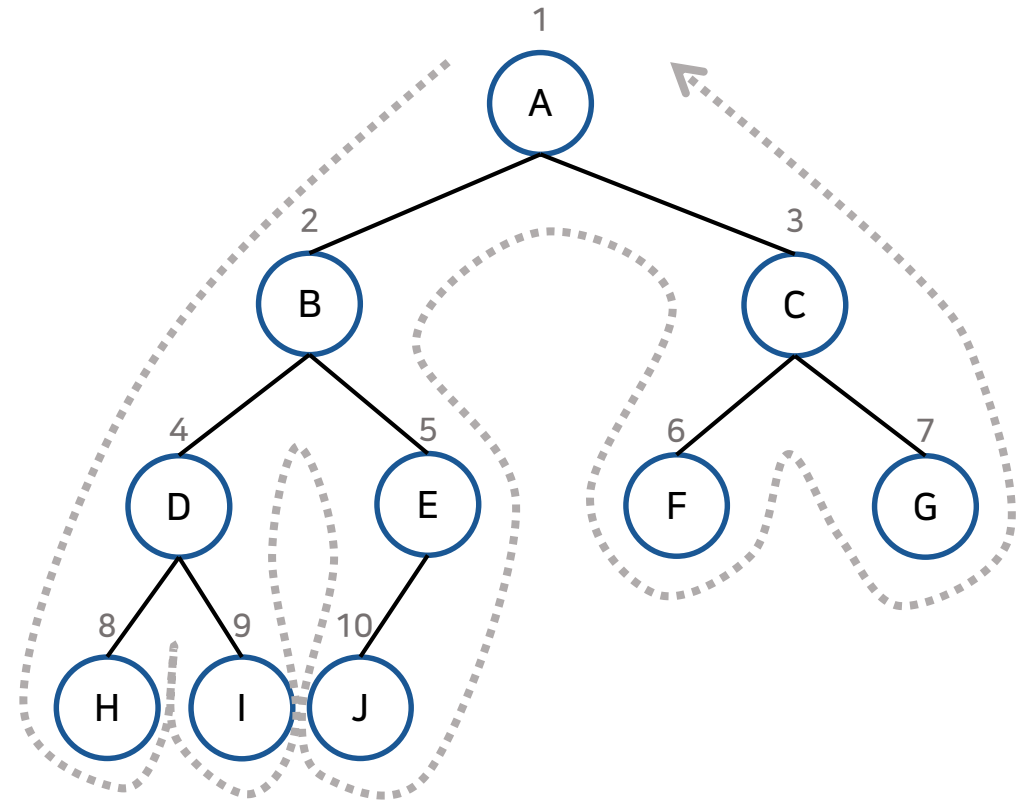
전위 순회



중위 순회

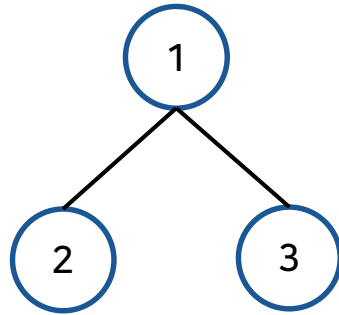


후위 순회



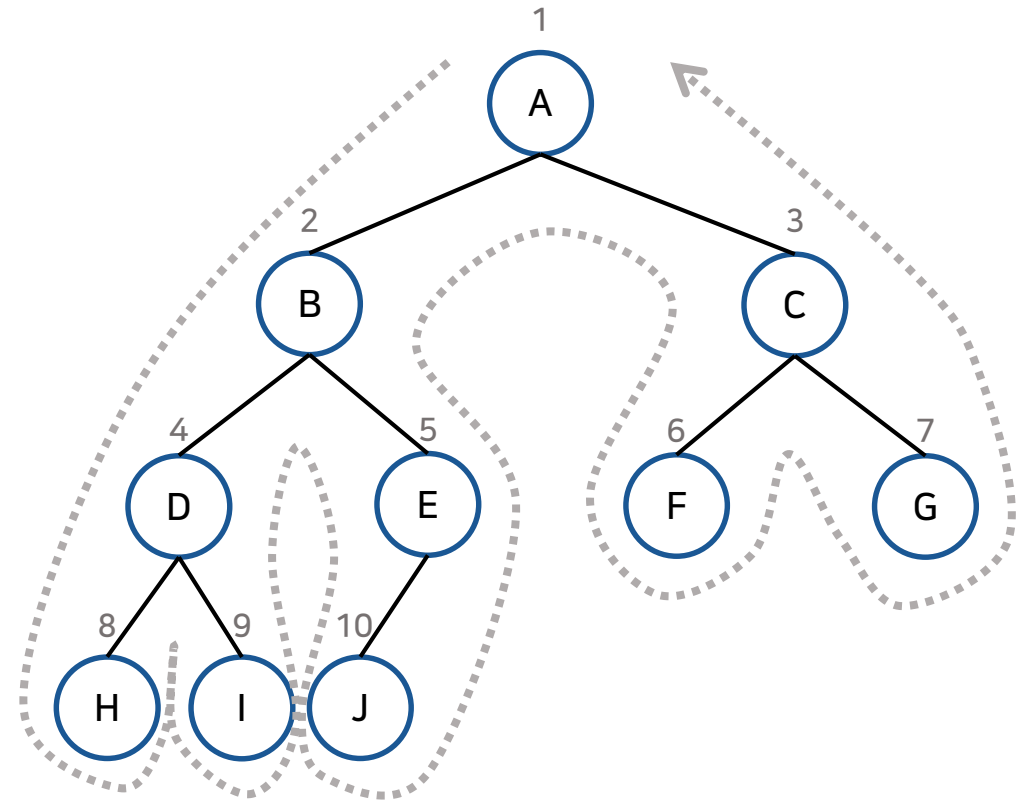
완전 이진 트리 탐색

- 깊이 우선 탐색(Depth-First Search)
 - 전위(Pre-order) 순회
 - 부모 노드 → 왼쪽 자식 노드 → 오른쪽 자식 노드 순서로 탐색



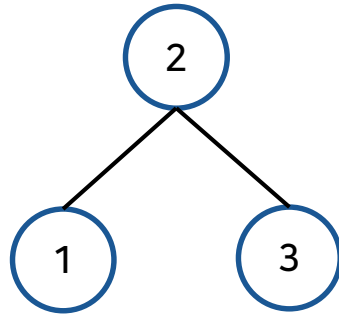
전위 순회

- $A \rightarrow B \rightarrow D \rightarrow H \rightarrow I \rightarrow E \rightarrow J \rightarrow C \rightarrow F \rightarrow G$



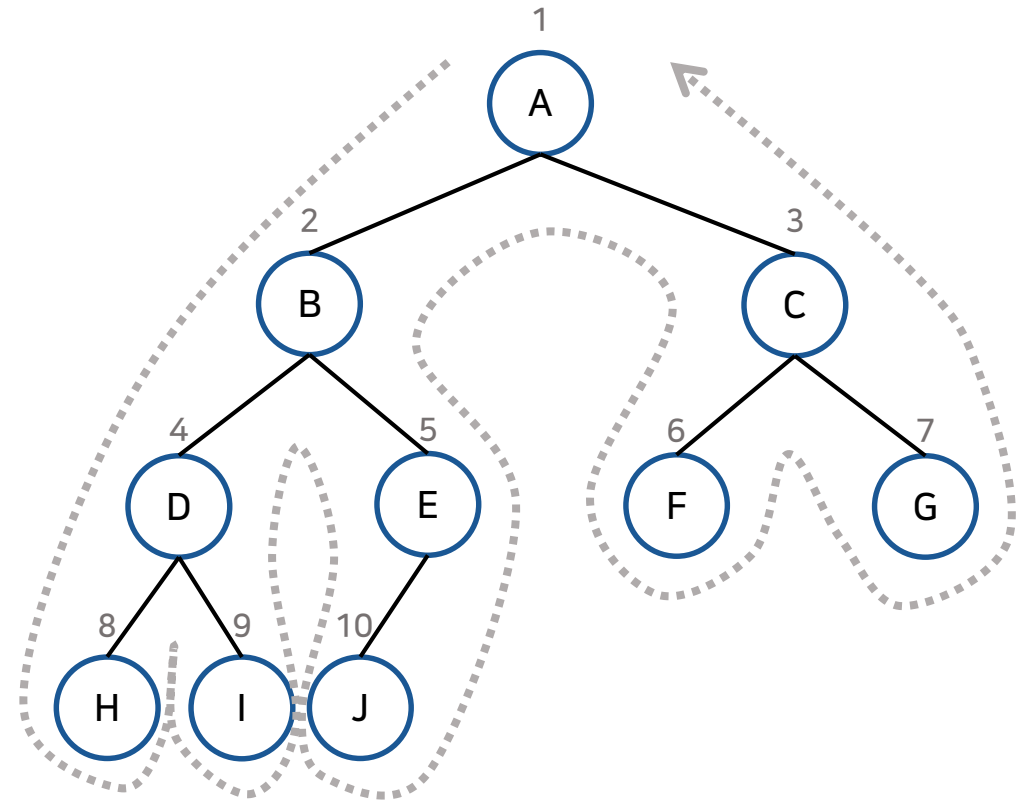
완전 이진 트리 탐색

- 깊이 우선 탐색(Depth-First Search)
 - 중위(In-order) 순회
 - 왼쪽 자식 노드 → 부모 노드 → 오른쪽 자식 노드 순서로 탐색



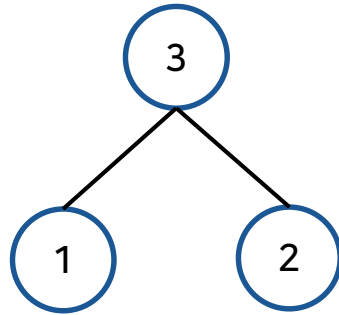
중위 순회

- $H \rightarrow D \rightarrow I \rightarrow B \rightarrow J \rightarrow E \rightarrow A \rightarrow F \rightarrow C \rightarrow G$



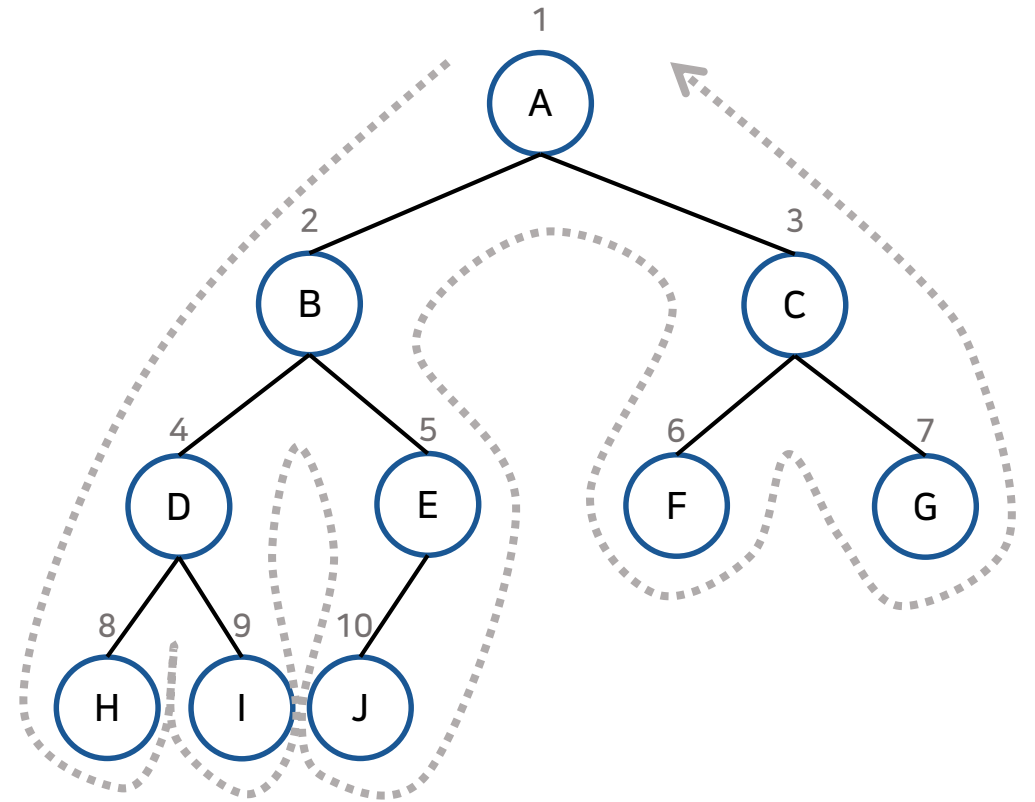
완전 이진 트리 탐색

- 깊이 우선 탐색(Depth-First Search)
 - 후위(Post-order) 순회
 - 왼쪽 자식 노드 → 오른쪽 자식 노드 → 부모 노드 순서로 탐색



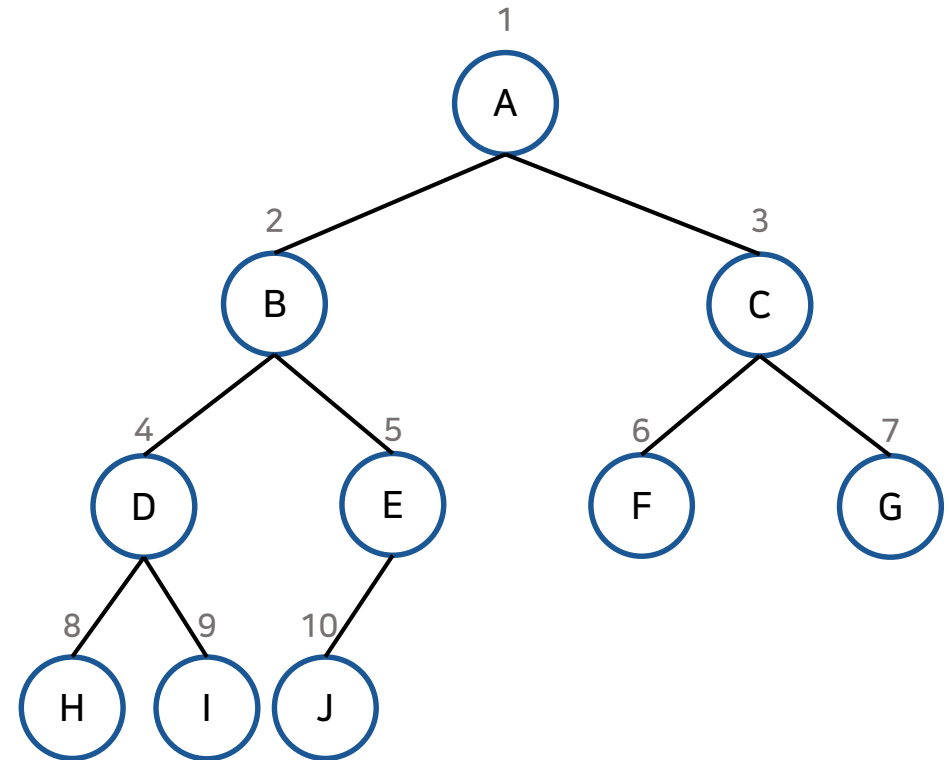
후위 순회

- $H \rightarrow I \rightarrow D \rightarrow J \rightarrow E \rightarrow B \rightarrow F \rightarrow G \rightarrow C \rightarrow A$



연습문제

- 배열을 이용하여 오른쪽 트리를 구현하세요.
 - 배열의 크기를 입력 받아 트리를 구현하세요.
 - 깊이 우선 탐색(DFS) 메소드를 구현하세요.
 - 전위, 중위, 후위 탐색 구현



추가문제

- SW Expert Academy - 1233 사칙연산 유효성 검사