

3주차 - 트리

검색



# 자료구조 학술회

게임소프트웨어 전공 3학년 고병진

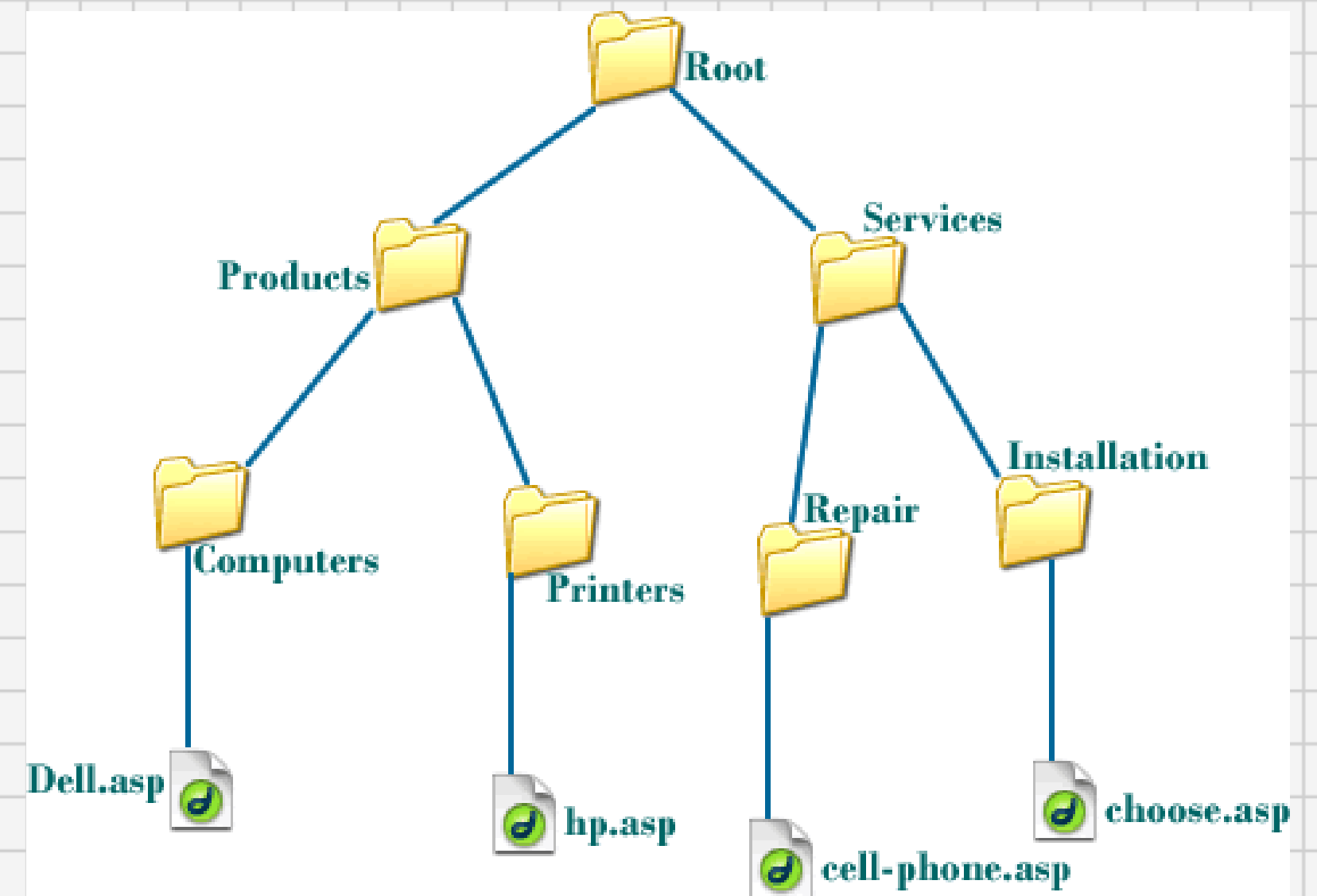
# 트리(Tree)란?



- 계층적 관계를 표현하는 자료구조이다.

## 특징

- 비선형 구조
- 재귀적: Root + Sub Tree
- 계층적: Parent + Child
- 생각보다 일상 생활에서 많이 사용함



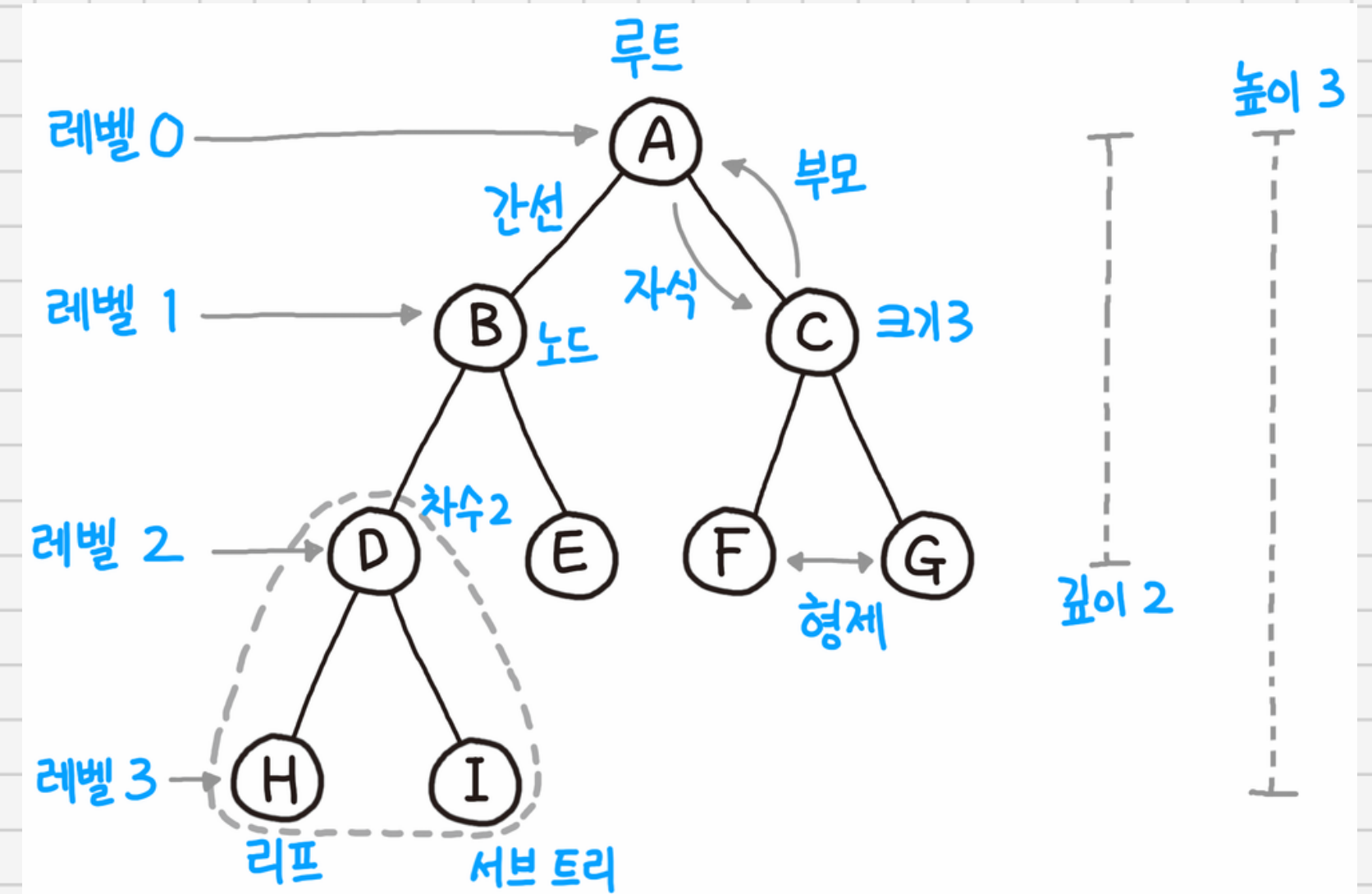
# 트리(Tree)란?



× □ -

## 용어

- 노드(node): A, B, C, D 같은 트리 구성요소
- 간선(edge): 노드와 노드를 연결하는 연결선
- 루트 노드(root node): 트리 구조에서 A같은 최상위 노드
- 단말 노드(terminal node): 아래로 다른 노드가 없는 노드 (E, F, G, H, I)
- 내부 노드(internal node): 단말 노드를 제외한 모든 노드 (A, B, C, D)



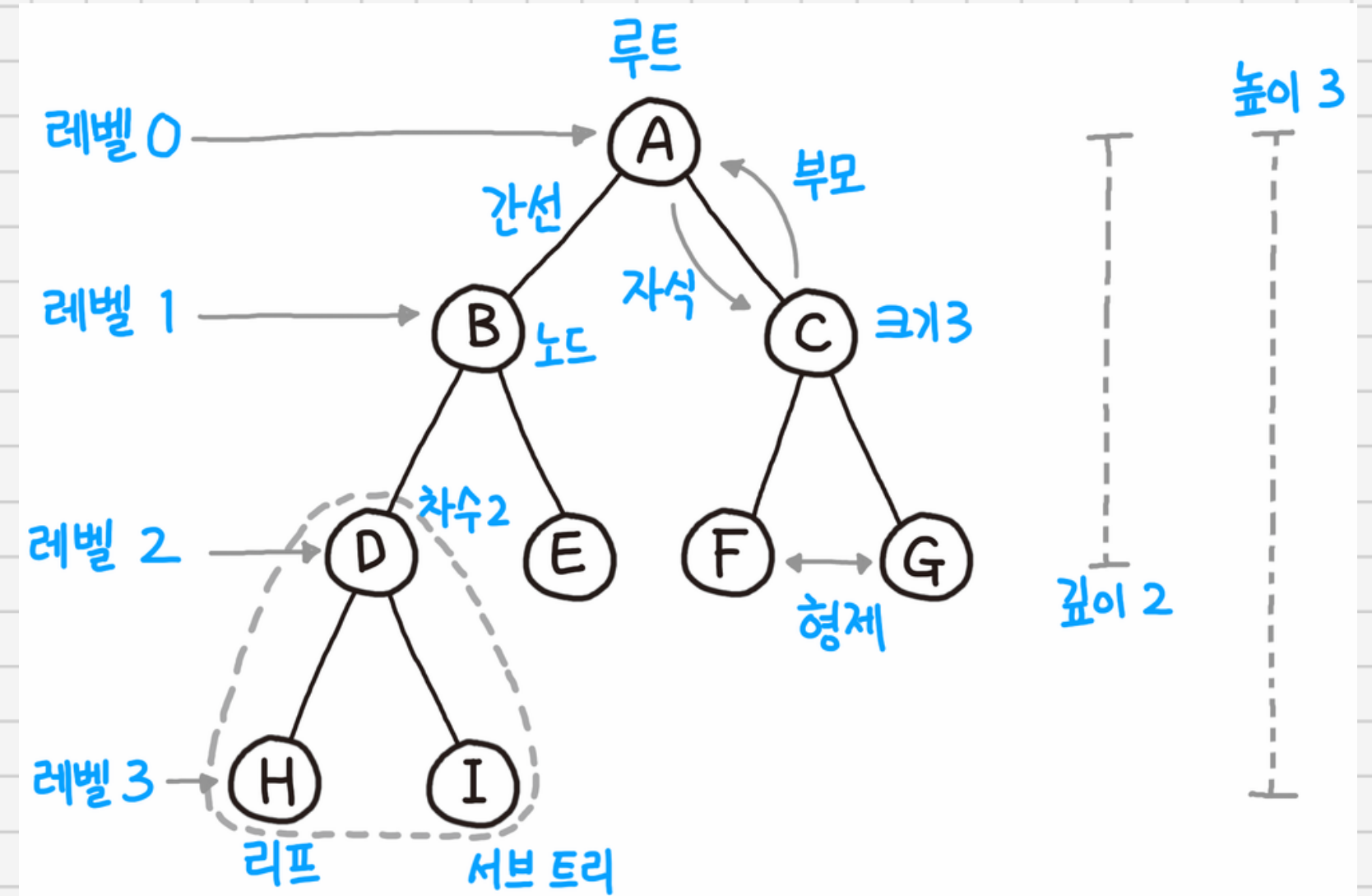
# 트리(Tree)란?



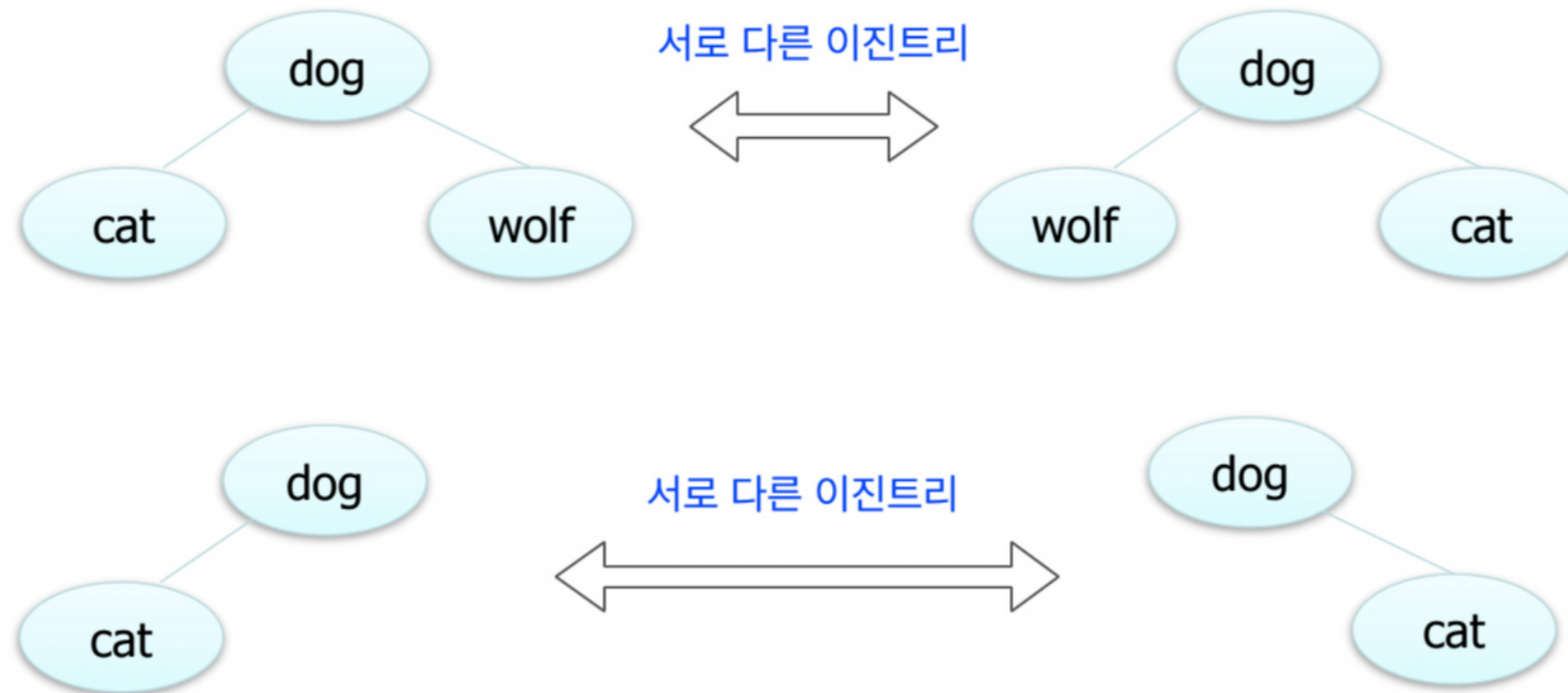
× □ -

## 용어

- Depth: 루트 노드에서 임의의 노드 x까지 경로의 edge의 수 (루트에서 노드까지)
- Height: 임의의 노드 x로부터 단말(leaf) 노드까지 가장 긴 경로의 edge의 수
- 레벨(level): depth를 가지고 있는 노드를 묶어서 레벨로 표현

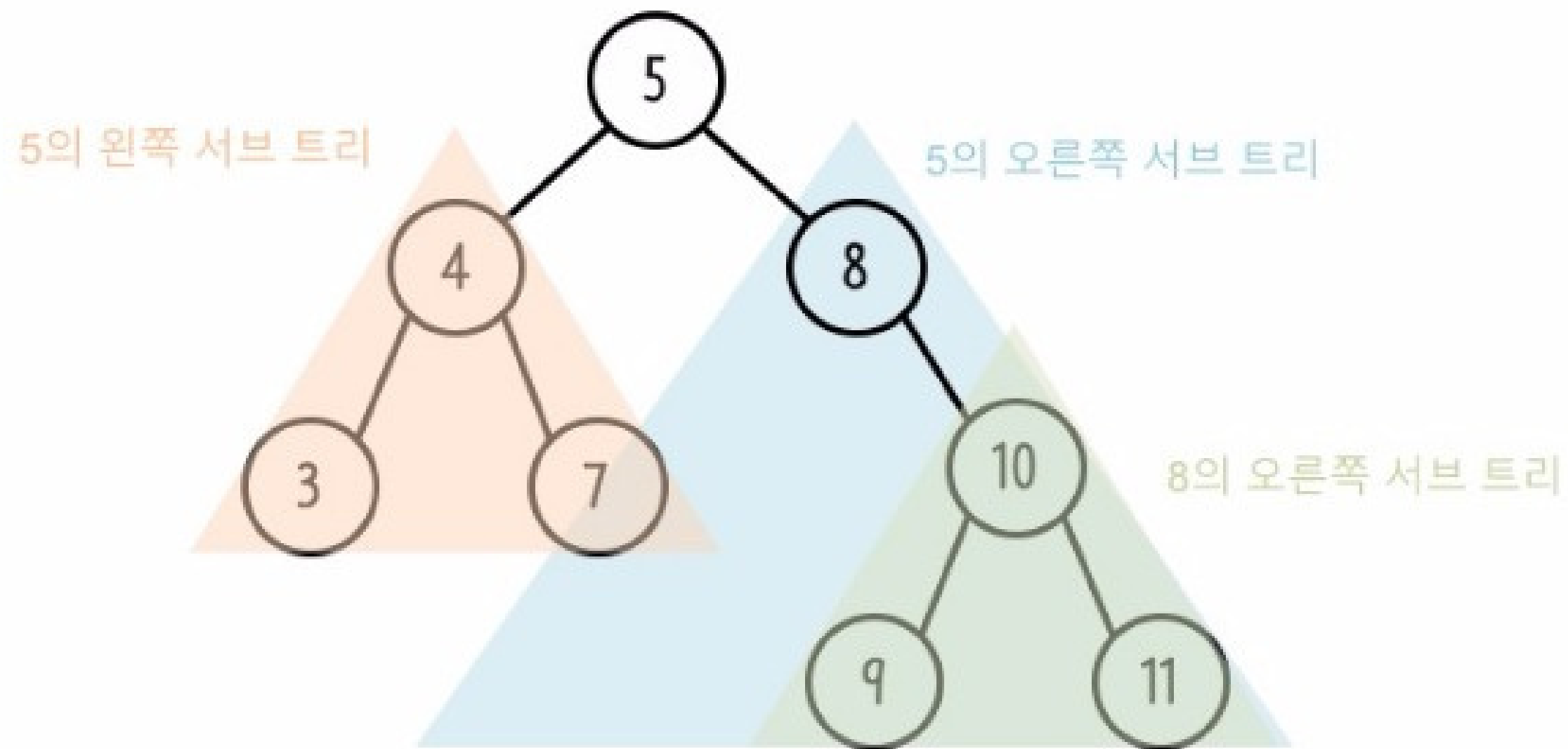


# 이진 트리(Binary Tree)란?



- 이진 트리에서 각 노드는 최대 2개의 자식을 가짐.
- 각각의 자식 노드는 자신이 부모의 왼쪽 자식인지 오른쪽 자식인지가 지정됨.

## 서브 트리(Sub Tree)란?



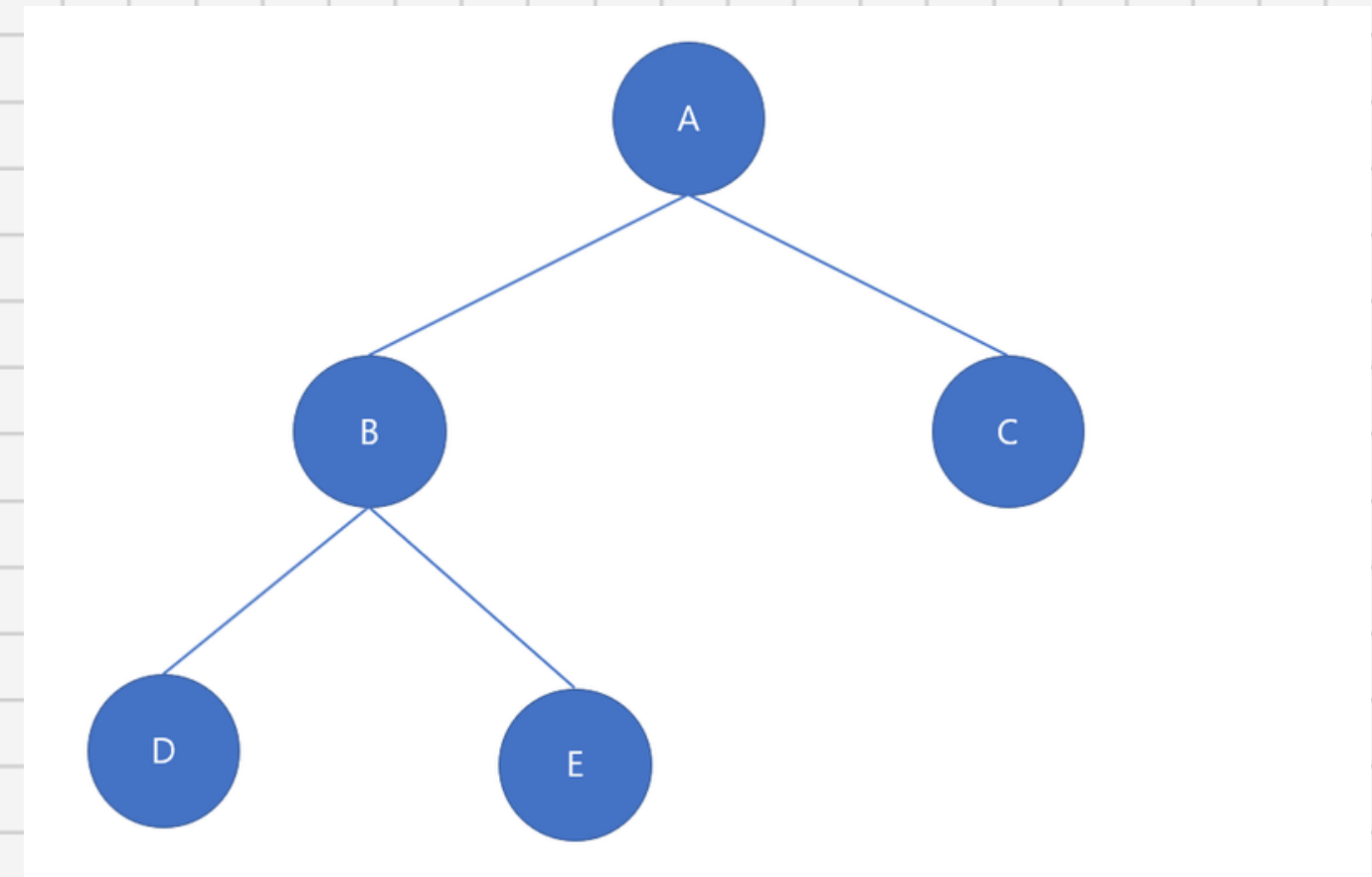
## 이진 트리의 종류?



✕ □ —

### 정 이진 트리(Full Binary Tree)

- 각 내부 노드가 두 개의 자식 노드를 가지고 있는 트리
- 홀수 개의 자식 노드를 가질수 없음 (자식이 없거나 2개)



## 이진 트리의 종류?



✕ □ —

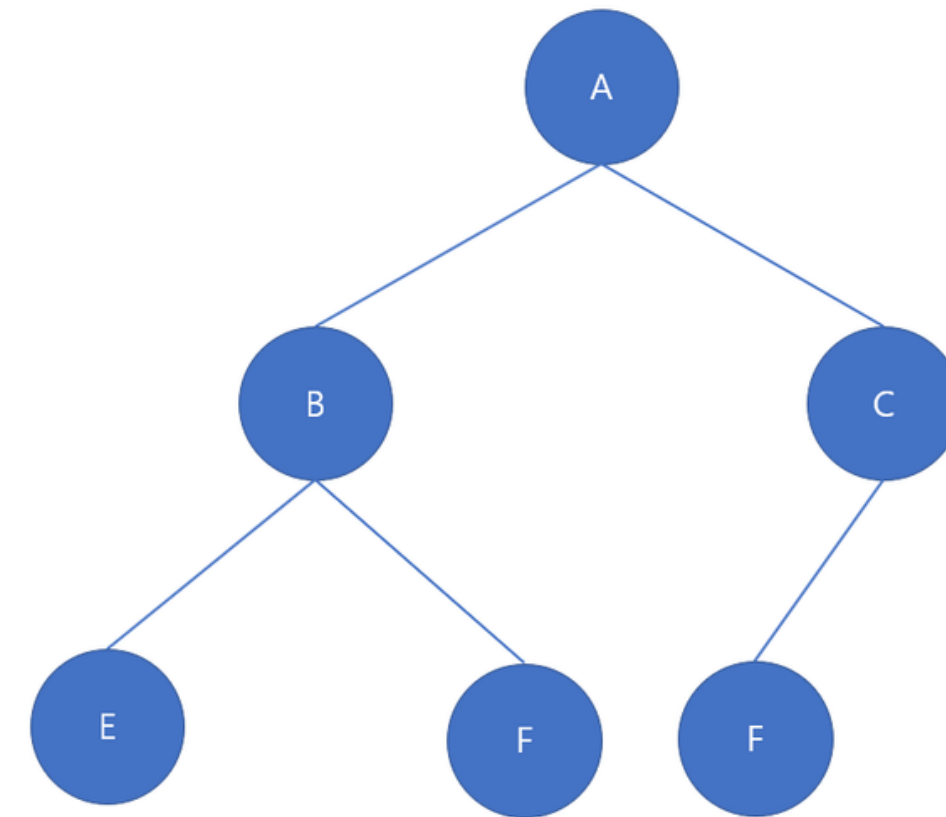
### 완전 이진 트리(Complete Binary Tree)

- 부모, 왼쪽 자식, 오른쪽 자식 순서대로 채워지는 트리
- 마지막 레벨을 제외하고 모든 노드가 가득 차 있어야 하며, 마지막 레벨의 노드는 전부 차 있지 않아도 되지만 왼쪽이 채워져야 함

<



>





## 이진 트리의 종류?



✕ □ -

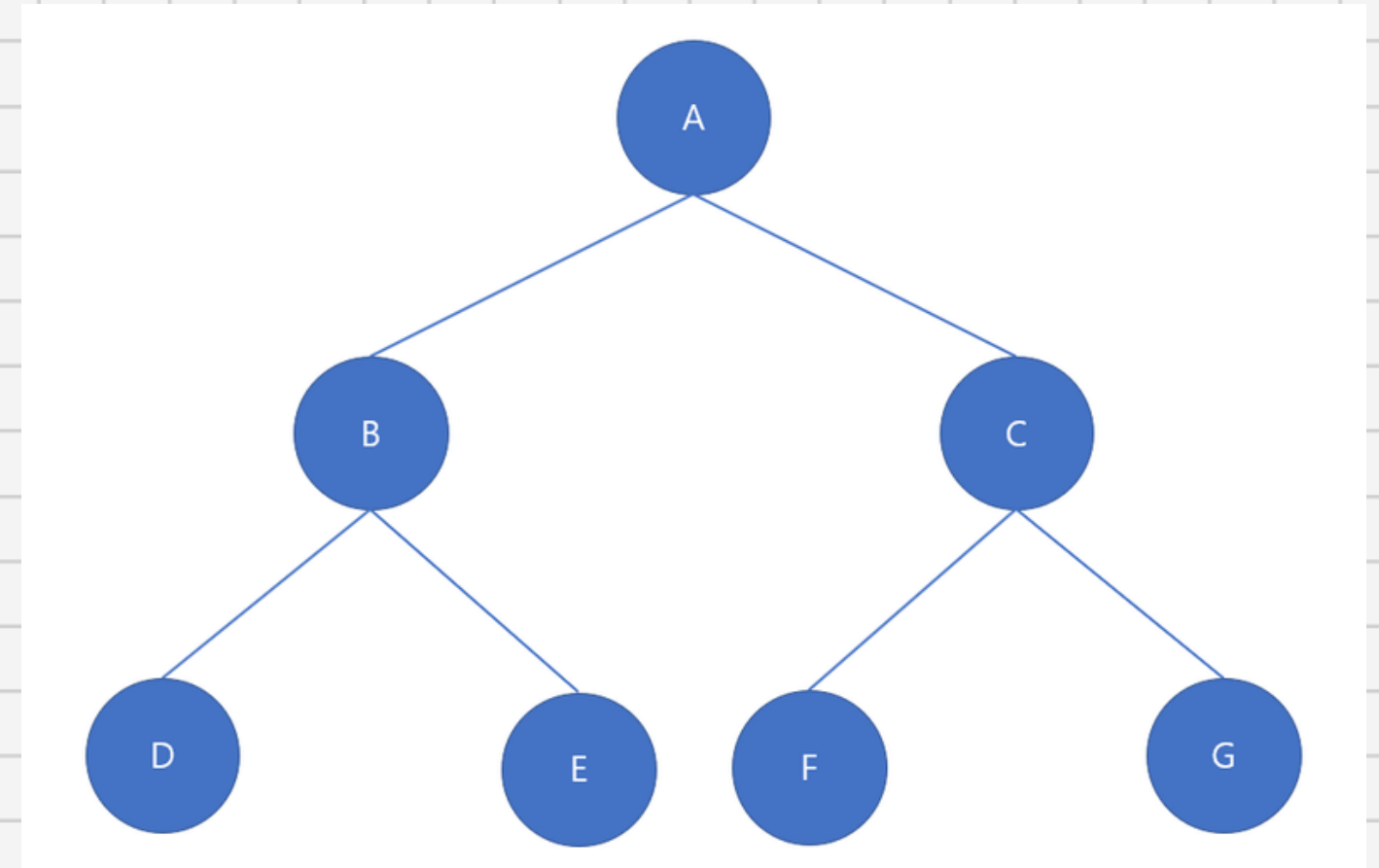
### 균형 이진 트리(Balanced Binary Tree)

- 왼쪽 자식, 오른쪽 자식 노드의 갯수가 정확하게 일치해야 하 필요는 없지만 지나치게 한쪽으로 치우치지 않은 트리구조
- 트리가 한 쪽으로 치우쳐져 있는 경우, 시간 복잡도가 악화되므로 균형잡힌 트리를 만드는 것이 주안점

<



>



## 시간 복잡도?



	Array (Unsorted)	Linked List	Array (Sorted)	Binary Search Tree
Search(x)	$O(N)$	$O(N)$	$O(\log N)$	$O(\log N)$
Insert(x)	$O(N)$	$O(N)$	$O(N)$	$O(\log N)$
Remove(x)	$O(N)$	$O(N)$	$O(N)$	$O(\log N)$

- insert의 경우, 임의의 위치 또는 노드에 삽입한다고 가정
- 단일 연결 리스트의 경우 특정 노드를 삭제하기 위해서는 삭제할 노드의 앞 노드를 찾아야함
- 양방향 리스트의 경우 삭제할 노드 주소를 알고 있다면  $O(1)$

## 이진 트리의 순회 방법?



✕ □ -

### 전위 순회(Preorder Traversal)

- 루트 노드를 먼저 탐색

### 중위 순회(Inorder Traversal)

- 루트 노드를 중간 탐색

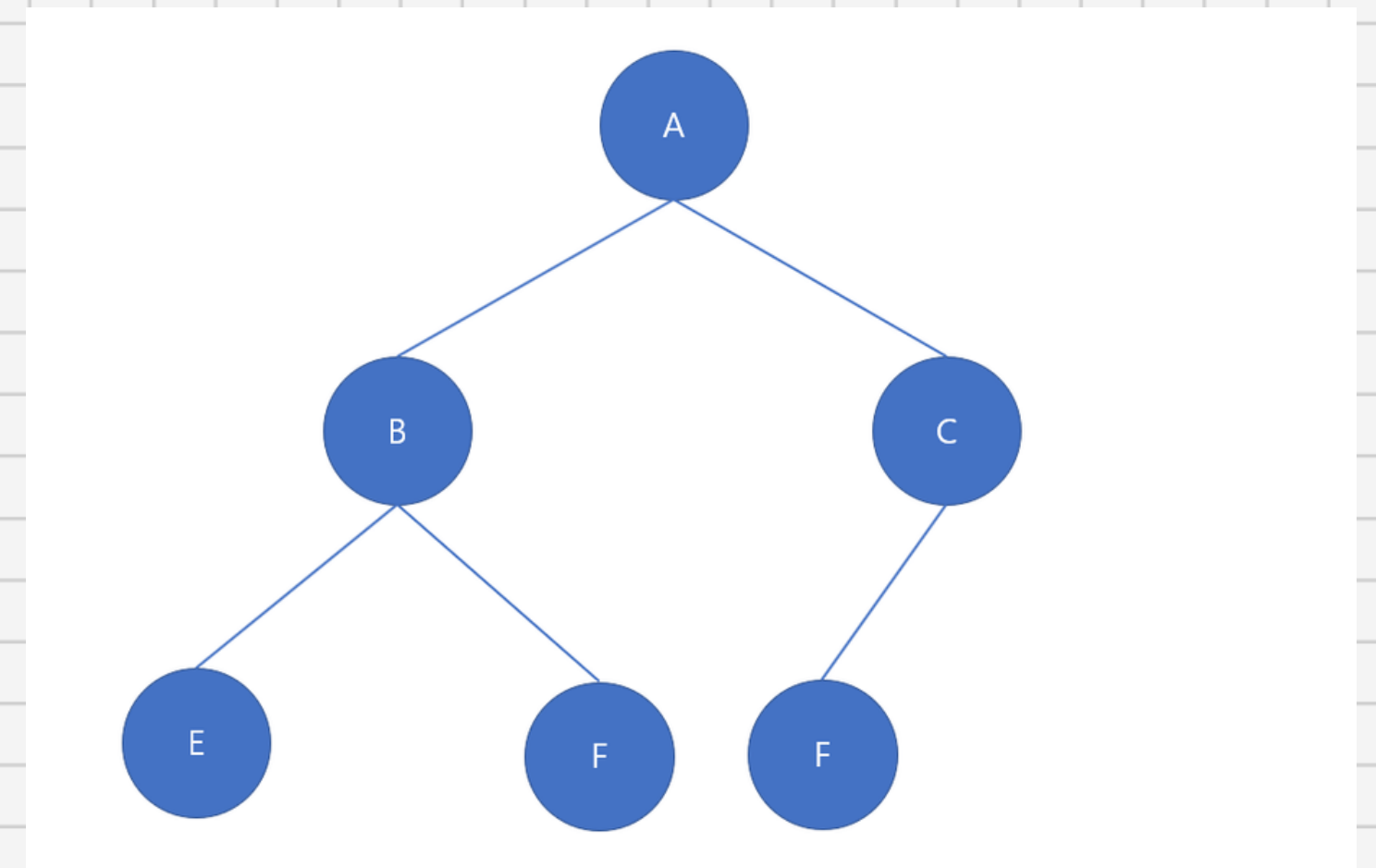
### 후위 순회(Postorder Traversal)

- 루트 노드를 마지막에 탐색

<



>



이진 트리의 순회 방법?

