

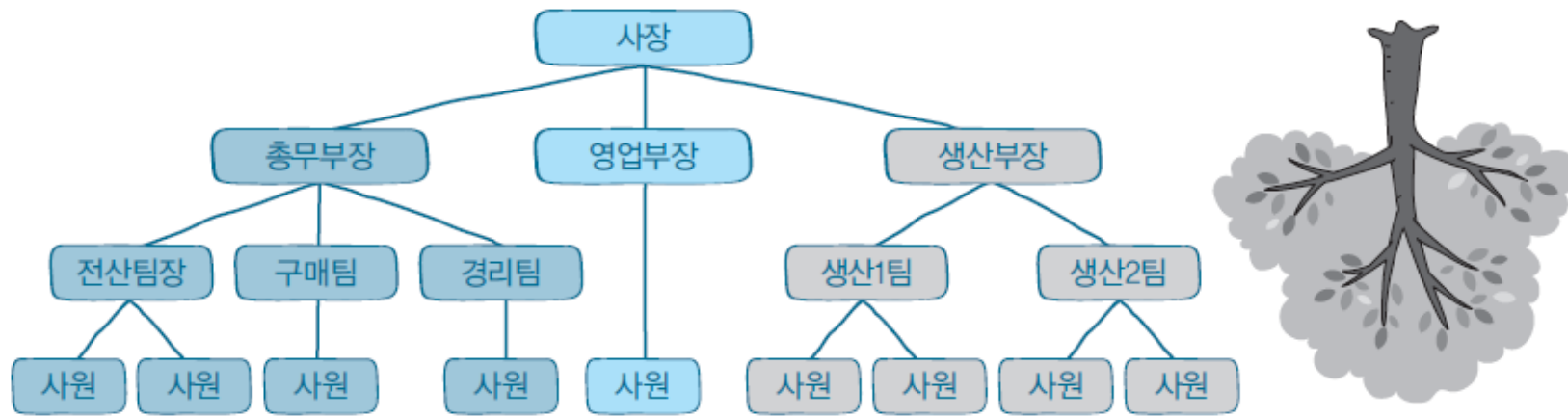


# 데이터 구조

## 9주차: 트리 (TREE)

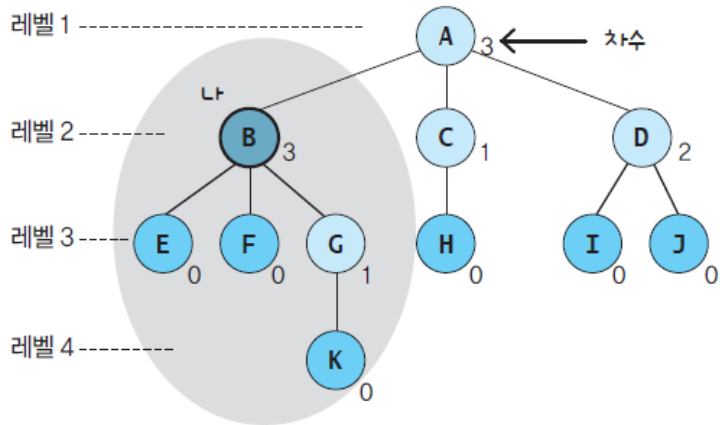
# 트리(TREE) 정의

- 트리는 나무 모양의 자료구조
  - 계층적인 관계를 가진 자료의 표현에 매우 유용
  - 비선형 자료구조, 계층구조



- 활용 예시: 운영체제의 파일시스템, 탐색 트리, 우선순위 큐, 결정 트리 등

# 트리(TREE) 관련 용어

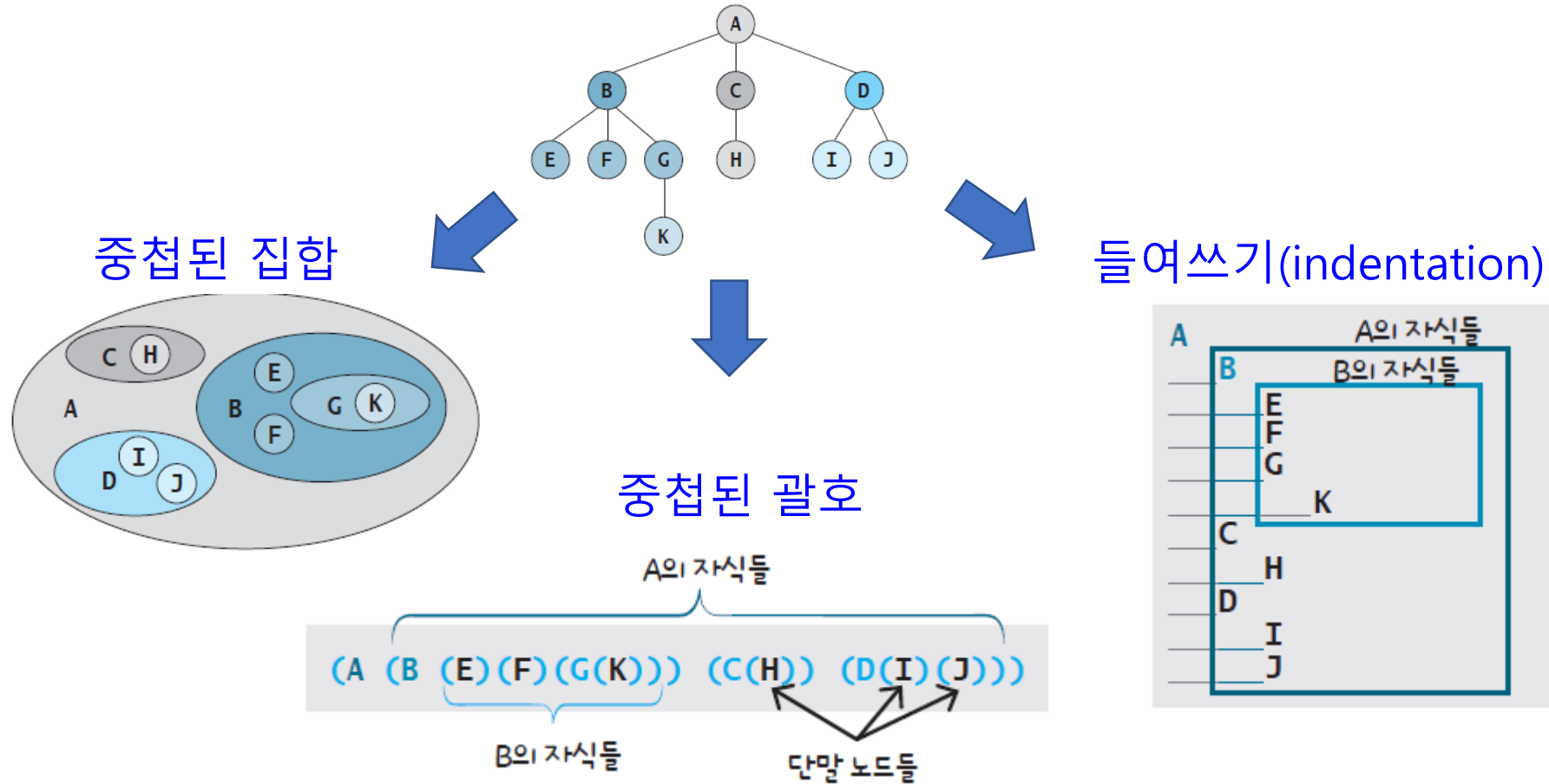


- 루트 노드: A
- B의 부모노드: A
- B의 자식 노드: E, F, G
- B의 자손 노드: E, F, G, K
- K의 조상 노드: G, B, A
- B의 형제 노드: C, D
- B의 차수: 3
- 단말 노드: E, F, K, H, I, J
- 비단말 노드: A, B, C, D, G
- 트리의 높이: 4
- 트리의 차수: 3

용 어	내 용
부모(Parent) 노드	간선으로 직접 연결된 노드 중에 상위 노드
자식(Child) 노드	간선으로 직접 연결된 노드 중에 하위 노드
형제(sibling) 노드	같은 부모 노드를 가진 노드
조상(ancestor)노드	어떤 노드에서 루트 노드까지의 경로상에 있는 모든 노드
자손(descendent)노드	어떤 노드 하위에 연결된 모든 노드
단말(terminal, leaf)노드	자식 노드가 없는 노드, 자식이 있으면 비단말 노드
노드의 차수(degree)	노드가 가지고 있는 자식의 수, 단말 노드는 항상 0
트리의 차수	트리에 포함된 모든 노드의 차수 중에서 가장 큰 수
레벨(level)	트리의 각 층에 번호를 매기는 것, 루트 노드의 레벨은 1이고, 한 층씩 내려갈 수록 레벨은 1씩 증가
트리의 높이(height)	트리가 가지고 있는 최대 레벨

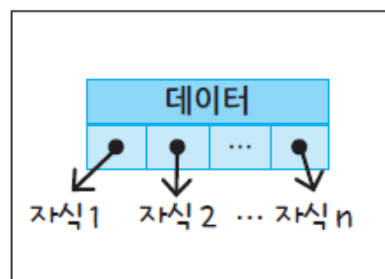
# 트리(TREE) 표기 방법

- 노드와 간선의 연결 관계

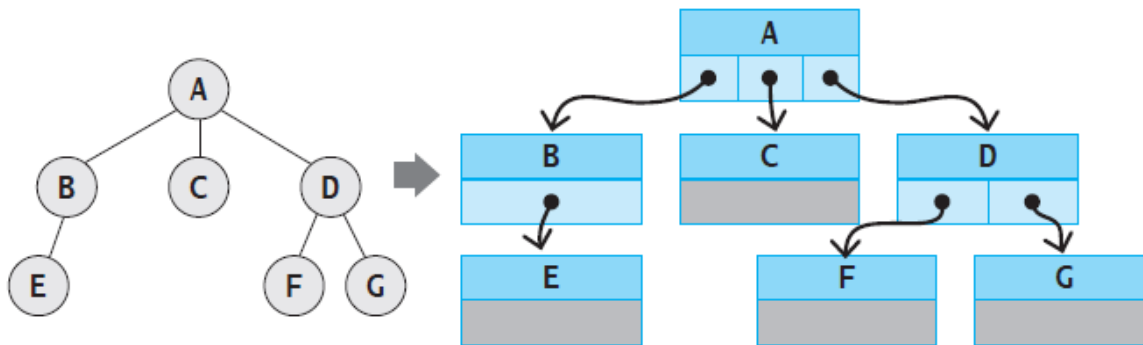


# 트리(TREE) 표현법

- 방법1: N-링크표현
  - 자식의 개수에 제한이 없는 트리 (general TREE)

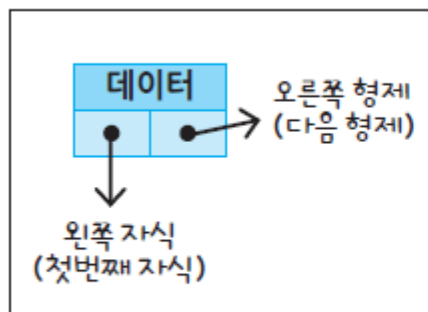


N-링크 노드의 구조

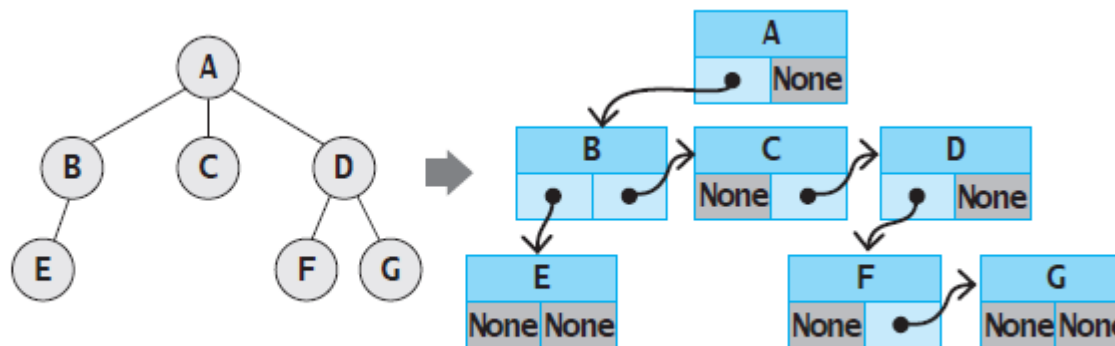


N-링크 표현의 예

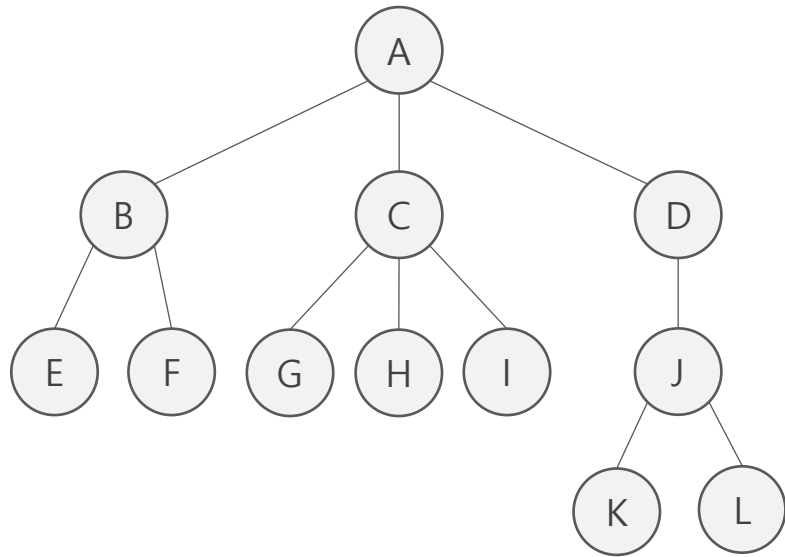
- 방법2: 왼쪽 자식 – 오른쪽 형제



왼쪽 자식-오른쪽 형제 표현의 예



왼쪽 자식-오른쪽 형제 표현의 예



문제1. 왼쪽 트리에서 다음을 구하세요

- (a) 루트노드
- (b) 노드 J의 부모노드
- (c) 노드 G의 형제노드
- (d) 노드 C의 차수
- (e) 트리의 높이
- (f) 트리의 차수

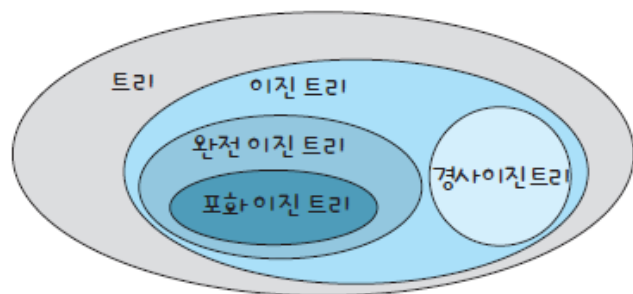
문제2. 왼쪽 트리를 다음의 방법으로 각각 표기하세요.

- (a) 중첩된 집합으로 표기
- (b) 중첩된 괄호로 표기

# 이진 트리 (Binary TREE)

- 모든 노드가 **최대 2개**의 자식만을 가질 수 있는 트리
  - 모든 노드의 차수가 2 이하로 제한
  - 자식 노드에도 순서가 존재
  - 컴퓨터 분야에서 널리 활용되는 기본적인 자료 구조
  - 데이터의 구조적인 관계를 잘 반영
  - 효율적인 삽입과 탐색 가능
  - 일반적인 TREE에 비해 계층적인 관계를 가지는 모든 자료형을 표현하기에는 부족
- 활용예시
  - 이진 탐색트리 (Binary search tree)
  - 우선순위 큐를 효과적으로 구현하는 힙 트리(heap tree) – 컴퓨터 프로세스 과정의 중요한 알고리즘 중 하나
  - 수식을 트리 형태로 표현하여 계산하는 수식 트리

# 이진 트리 (Binary TREE) 종류

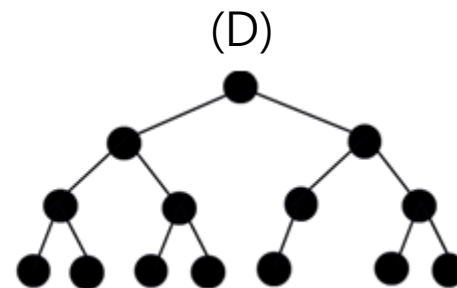
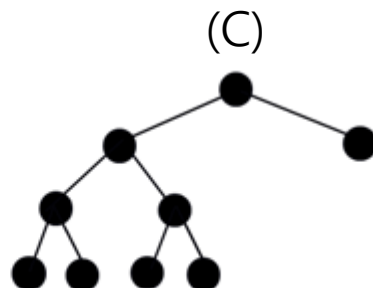
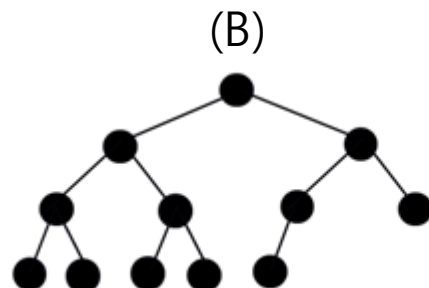
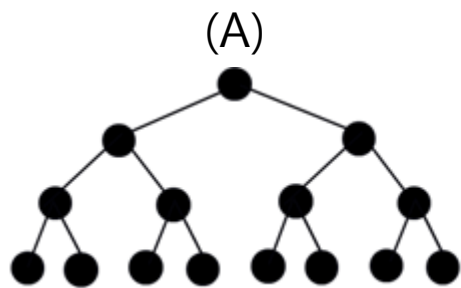


- 포화 이진 트리 (full binary tree)
  - 트리의 각 레벨에 노드가 꽉 차 있는 이진 트리

$$\text{전체 노드 개수} : 2^{1-1} + 2^{2-1} + 2^{3-1} + \dots + 2^{k-1} = \sum_{i=0}^{k-1} 2^i = 2^k - 1$$

QUIZ

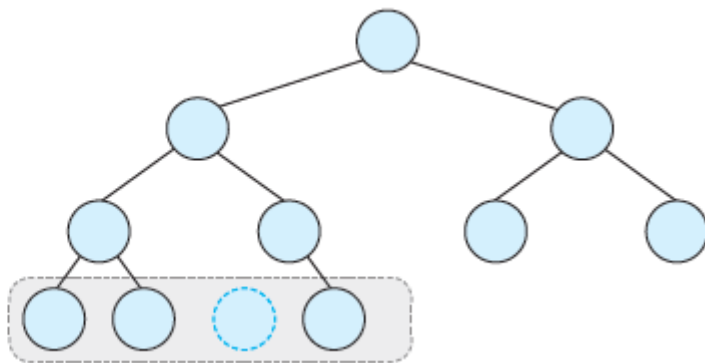
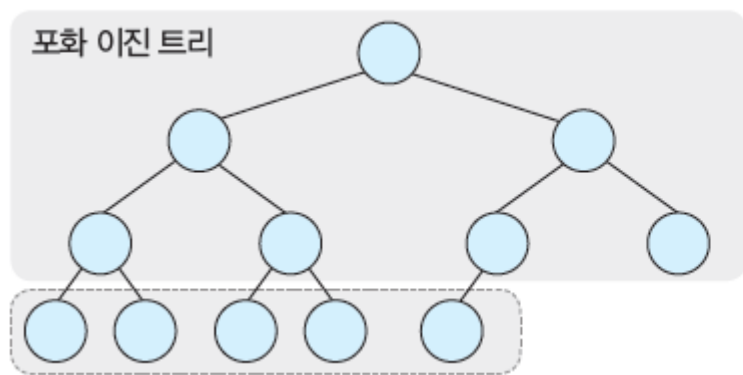
아래의 트리 중 포화 이진 트리는?





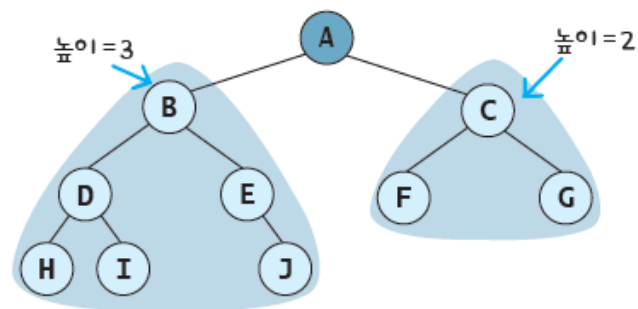
# 이진 트리 (Binary TREE) 종류

- 완전 이진 트리 (Complete binary tree)
  - 마지막 레벨을 제외한 각 레벨이 노드들로 꽉 차 있는 있는 트리를 말하며,  
마지막 레벨에서는 왼쪽부터 오른쪽으로 노드가 순서대로 채워져 있는 이진 트리
  - 마지막 레벨에서는 노드가 꽉 차 있지 않아도 되지만 중간에 빈 곳이 있으면 안 됨
  - "포화 이진 트리는 항상 완전 이진 트리"는 성립,  
"완전 이진 트리는 항상 포화 이진 트리"는 성립되지 않음

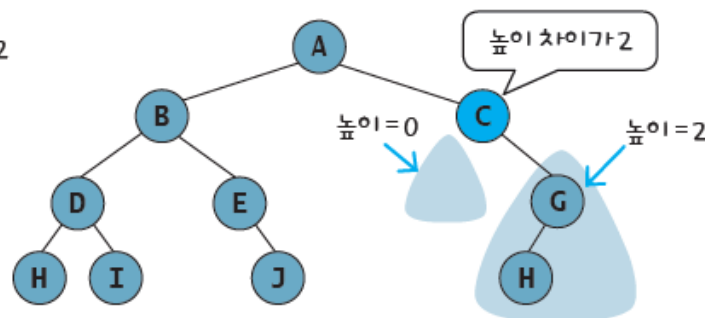


# 이진 트리 (Binary TREE) 종류

- 균형 이진 트리 (balanced binary tree)
  - 높이 균형 이진 트리 (height-balanced binary tree)
  - 모든 노드에서 좌우 서브 트리의 높이 차이가 1 이하인 트리를 말하며, 높이 차이가 1 초과할 경우 경사트리

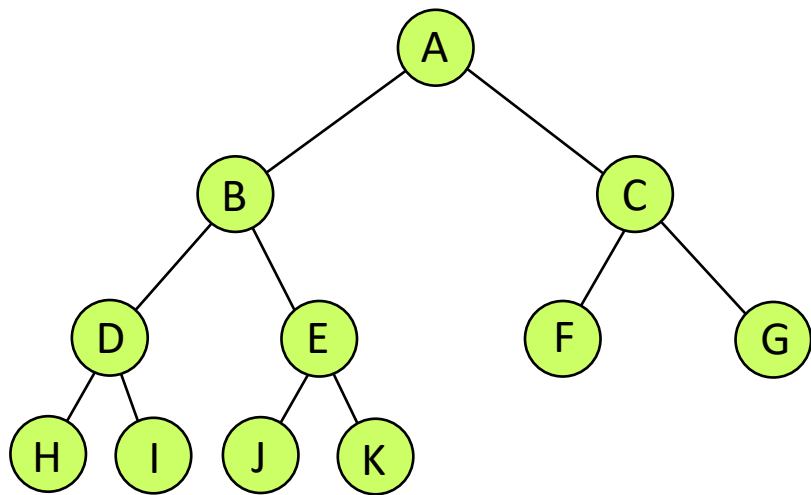


(a) 모든 노드의 좌우 서브 트리 높이 차이가 1 이하임 → **균형 이진 트리**



(b) **노드 C**의 좌우 서브 트리 높이 차이가 2임(1 초과) → **균형 이진 트리가 아님**

# 이진 트리 (Binary TREE) 와 배열 자료형의 관계



0	1	2	3	4	5	6	7	8	9	10	11	12
	A	B	C	D	E	F	G	H	I	J	K	

- 이진 트리의 특성을 활용한 배열 자료형에 저장할 경우 탐색 및 활용이 용이함
- 배열의 첫 인덱스는 건너뛰고, 두번째 인덱스 부터 저장
- 루트 노드부터 각 레벨로 내려오고, 각 레벨은 왼쪽에서 오른쪽으로 순차적으로 저장

# 이진 트리 (Binary TREE) 와 배열 자료형의 관계

0	1	2	3	4	5	6	7	8	9	10	11	12
	A	B	C	D	E	F	G	H	I	J	K	

- 노드  $i$ 의 부모 노드 인덱스 =  $i/2$
- 노드  $i$ 의 왼쪽 자식 노드 인덱스 =  $2i$
- 노드  $i$ 의 오른쪽 자식 노드 인덱스 =  $2i+1$

파이썬에서는 나눗셈 연산자가 /와 //로 구분되어 있습니다. 정수 나눗셈을 위해서는  $i//2$ 를 써야합니다.

## QUIZ

문제 1. "D"의 부모 노드는?

문제 2. "D"의 왼쪽 자식노드와 오른쪽 자식노드는?