

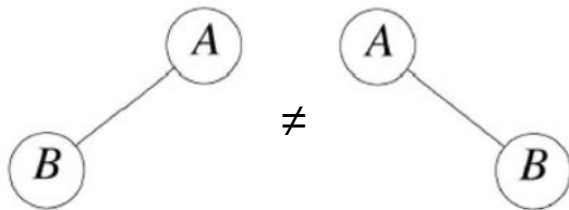
이진트리

# 이진트리

: 모든 내부 노드의 차수가 최대 2인 순서트리

모든 내부 노드가 왼쪽 서브트리와 오른쪽 서브트리를 가지고 있음

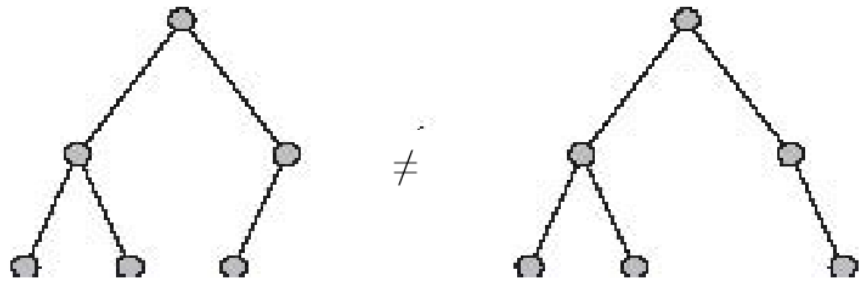
- 이진트리는 반드시 왼쪽 서브트리와 오른쪽 서브트리를 가져야함. 서브트리는 모두 공백일 수 있음.
- 자식의 순서 구별



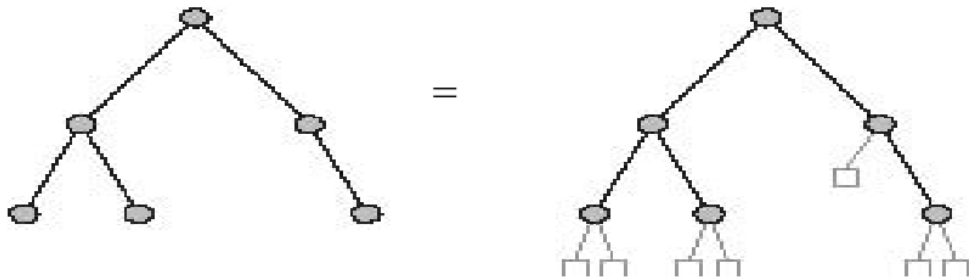
다른 트리들은

트리를 연결리스트로 나타내기 위해서는 서브트리 개수만큼의 링크 필드가 필요 => 메모리 낭비  
=> 차수가 2인 트리 표현하여 절약

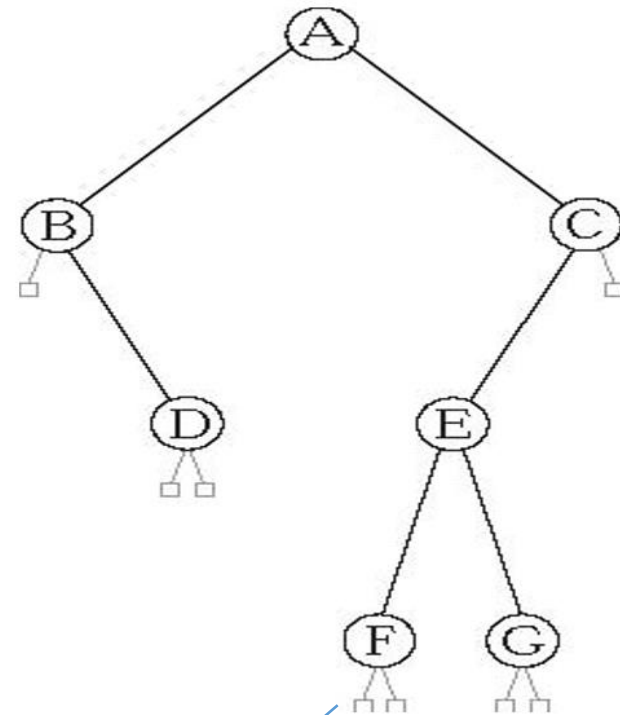
## 확장 이진트리



서로 다른 이진트리



서로 같은 이진트리



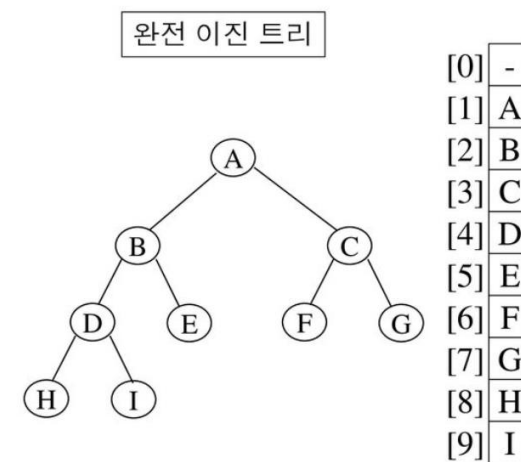
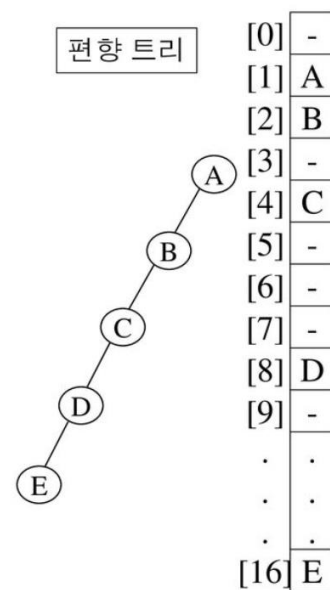
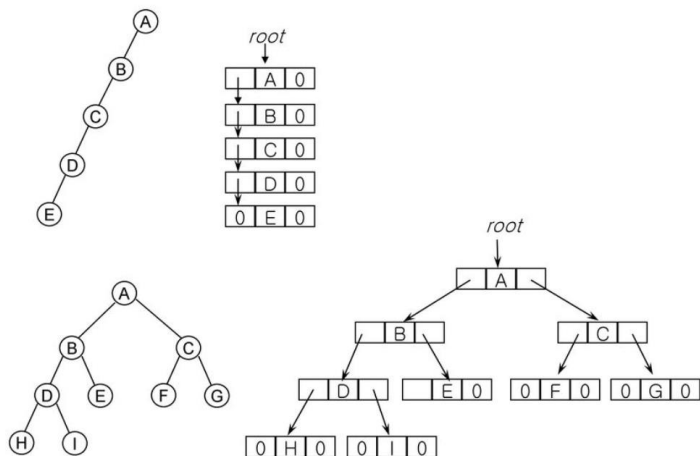
NIL 노드  
공백 서브트리  
높이 계산 시 무시

## > 배열로 구현

- 깊이가  $k$ 인 완전 이진 트리는  $2^{k-1}$ 의 노드를 저장할 수 있는 공간 필요
- 편향 이진 트리인 경우  $2^{k-1}$ 개 중  $k$ 개만 사용하므로 공간 낭비 -> 연결리스트 사용
- 삽입, 삭제 시 노드 저장위치 변경해야 -> 연결리스트 사용



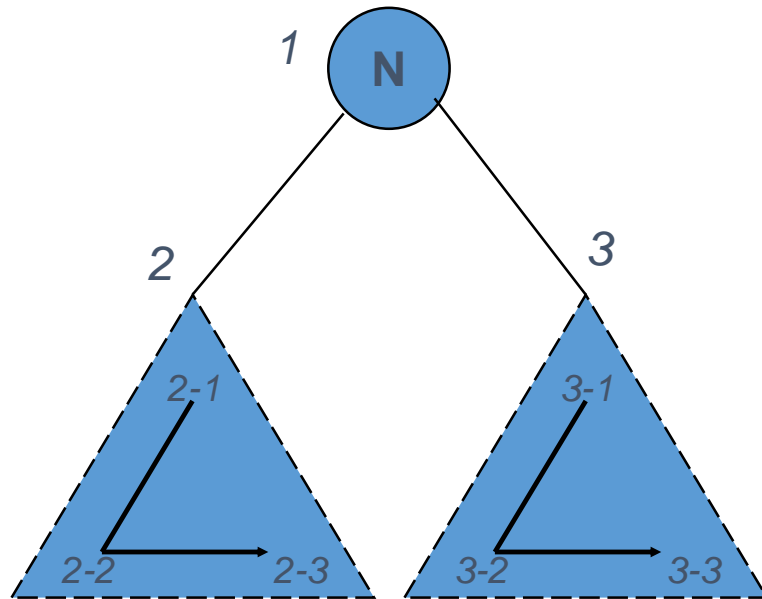
## > 연결리스트로 구현



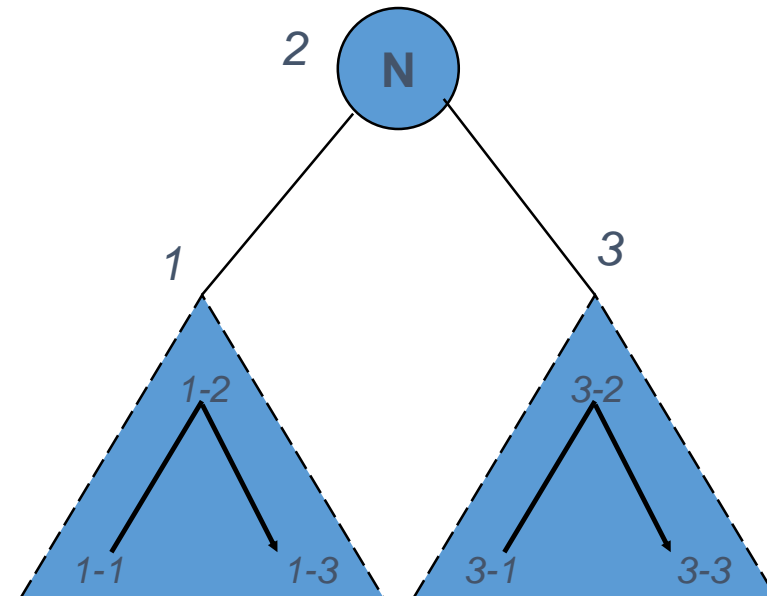
## 이진 트리 순회 알고리즘

- 레벨순서순회
- 순환 전위 순회(Preorder traversal)
  - 루트 방문 -> 왼쪽 서브트리 전위순회 -> 오른쪽 서브트리 전위순회
- 순환 후위 순회(Postorder traversal)
  - 왼쪽 서브트리 후위순회 -> 오른쪽 서브트리 후위순회 -> 루트 방문
- 순환 중위 순회(Inorder traversal)
  - 왼쪽 서브트리 중위순회 -> 루트 방문 -> 오른쪽 서브트리 중위 순회

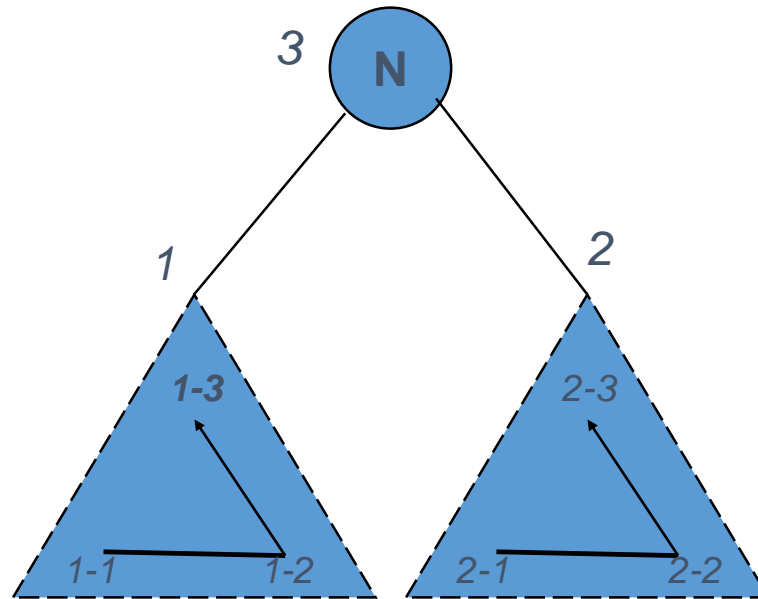
## <전위 운행>



## <중위 운행>



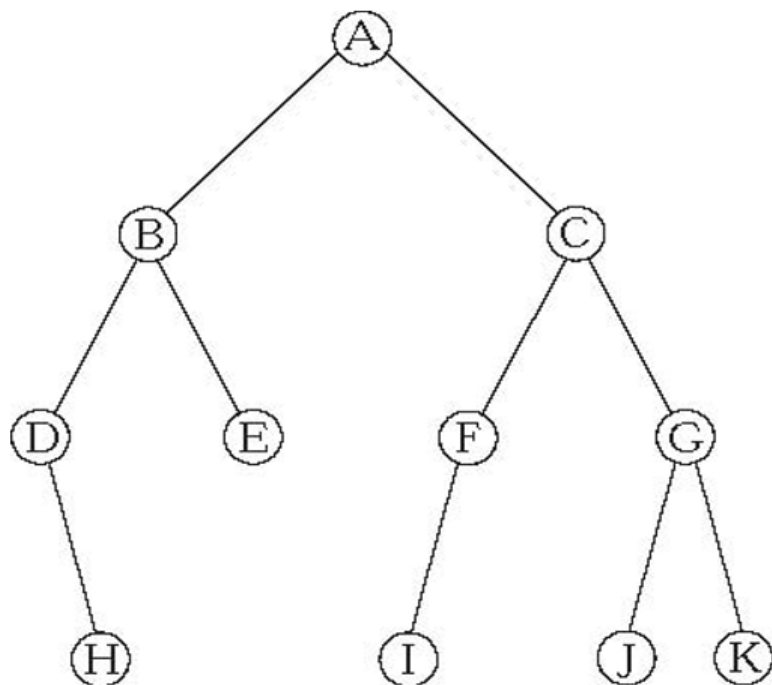
## <후위 운행>



# 이진 트리 순회의 예

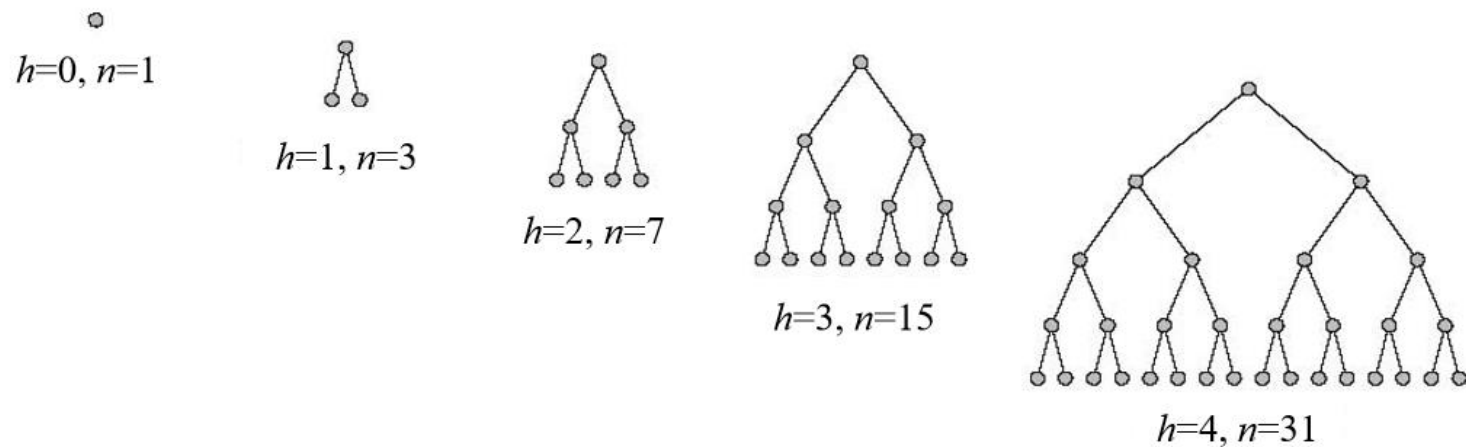
## ◆ 네 가지 순회

- 전위 순회 : A, B, D, H, E, C, F, I, G, J, K
- 중위 순회 : D, H, B, E, A, I, F, C, J, G, K
- 후위 순회 : H, D, E, B, I, F, J, K, G, C, A
- 레벨 순서 순회 : A, B, C, D, E, F, G, H, I, J, K





## 포화 이진 트리



### ◆ 포화 이진 트리의 크기

$$\underline{n = 2^{h+1} - 1}$$

공백 트리는 높이  $h=-1$  인 포화 이진 트리

### ◆ 이진 트리의 높이

$$\lfloor \lg n \rfloor \leq h$$

### ◆ 이진 트리의 크기에 대한 범위

$$h+1 \leq n \leq 2^{h+1} - 1$$

## 카탈로니아 수 (Catalan number)

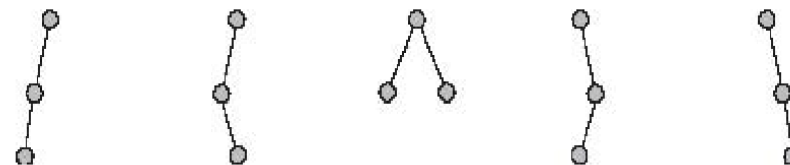
: 크기가 같은 이진 트리의 개수를 구하는 공식

$$C_n = \frac{(2n)!}{n!(n+1)!}$$

◆ 크기가 2인 2 개의 이진 트리

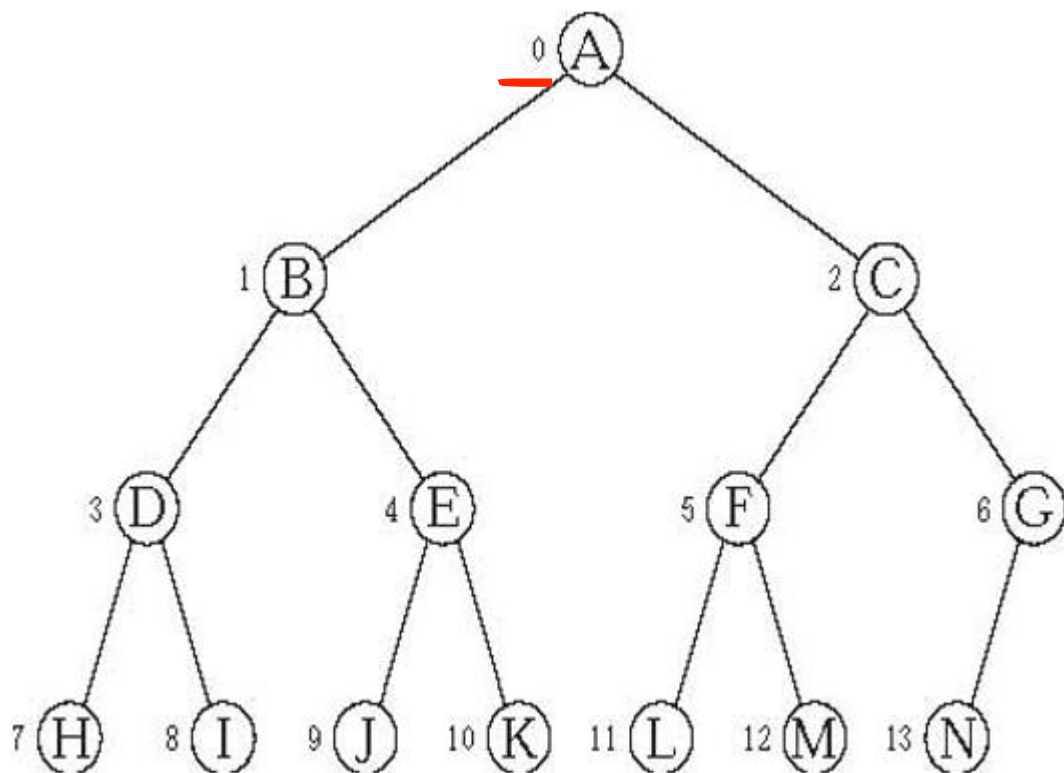


◆ 크기가 3인 5 개의 이진 트리



## 완전 이진 트리

마지막 레벨을 제외하고 모든 노드가 채워진 이진 트리.



부모 인덱스:  $(i-1)/2$

자식 인덱스:  $2i+1, 2i+2$

크기가  $n$ 일 때, 리프는  $n/2 \sim n-1$ 까지의 인덱스

내부 노드의 개수:  $n/2$ 개

리프의 개수:  $(n+1)/2$ 개

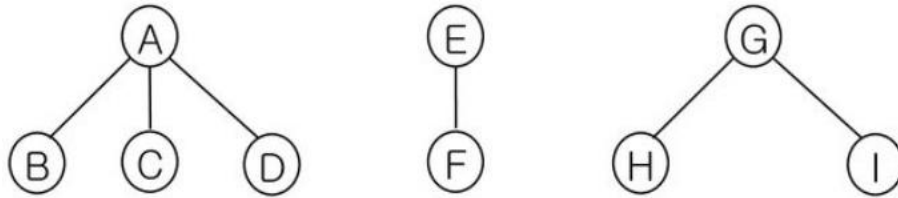
## 포리스트 (forest)

### 트리의 집합

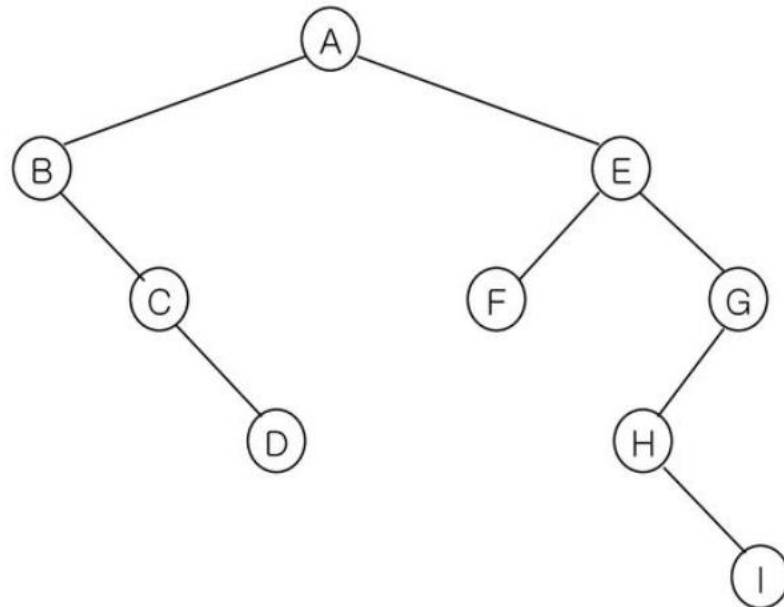
- 순서 포리스트: 순서 트리의 시이퀀스

- 하나의 이진 트리로 포리스트를 표현

EX) ◆ 세 개의 트리로 구성된 포리스트



◆ 위 그림의 이진 트리 표현



자식은 왼쪽 자식으로  
형제는 오른쪽 자식으로 매핑