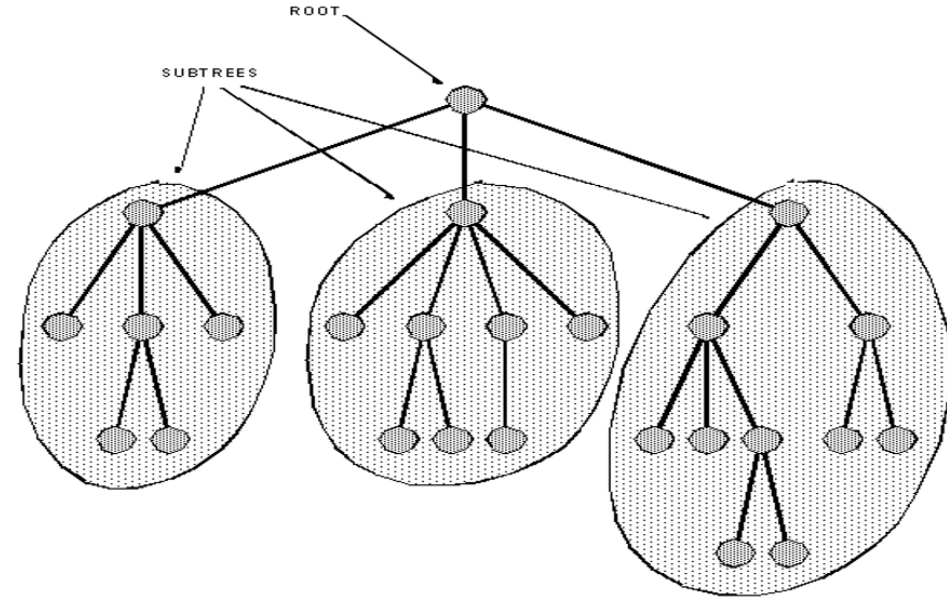
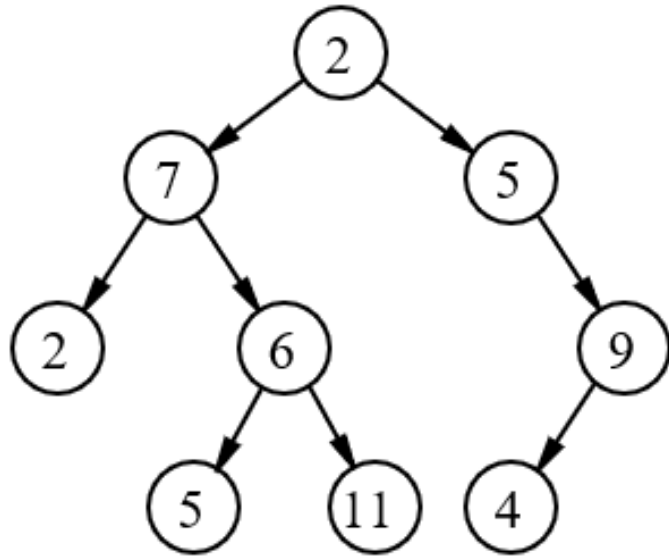


트리



• 트리

어떤 속성을 만족하는 노드와 방향 간선의 집합

하나의 노드가 루트를 가리키고 있는 사이클이 없는 그래프

순환 정의 -> 루트노드와 서브트리의 집합으로 구성된다

서로 다른 두 노드를 잇는 길이 하나뿐인 그래프를 트리라고 부른다.

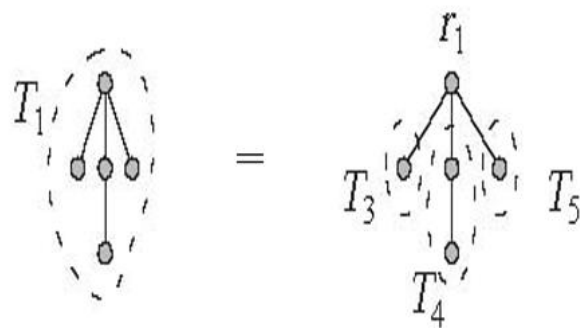
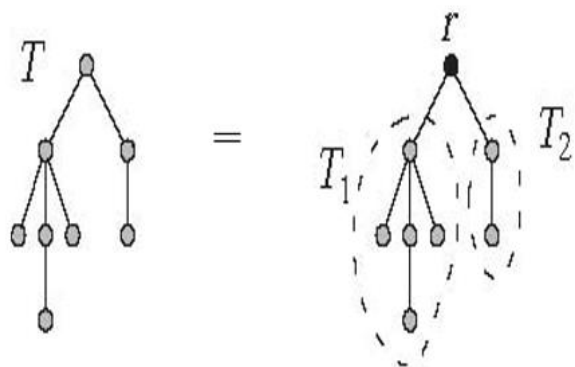
1개 이상의 노드를 갖는 집합.

트리구조는 계층적 구조를 표현하는데 가장 효율적인 구조!

11.3 트리의 순환 정의

◆ 정의

- 트리는 공집합이거나 r 이 노드이고 S 가 서로 분리된 트리의 집합일 때, 어떤 트리도 r 을 포함하지 않는 쌍 (r, S) 이다. 노드 r 을 트리의 루트라고 부르고 S 에 있는 트리를 서브트리 (subtree)라고 부른다.



단독트리이다.
모든 단독 트리는 S 가
공집합일 때 쌍 (r, S) 로 볼 수
있으므로 트리이다.

트리와 그래프의 차이?

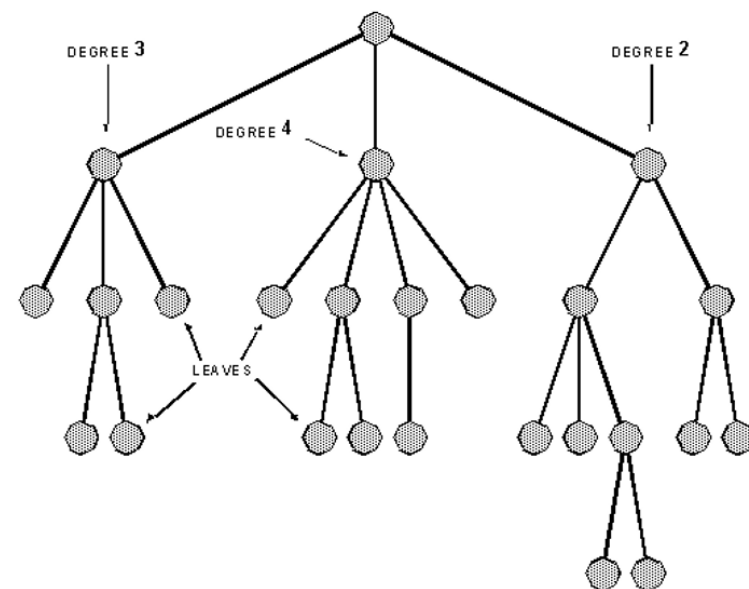
- 트리는 노드 자신을 가리키는 loof가 존재하지 않음.
(가장 큰 차이점으로, **트리는 사이클이 없음**)

- 트리는 부모에서 자식으로 향하는 단방향, 그래프는 양방향이 가능하고 가중치를 가질 수 있음.

트리의 특성

- 차수(degree)

: 하위 트리 집합의 크기. 즉, **자식의 수**. 리프노드는 차수가 0
트리의 차수는 노드들의 차수 중 최대 차수



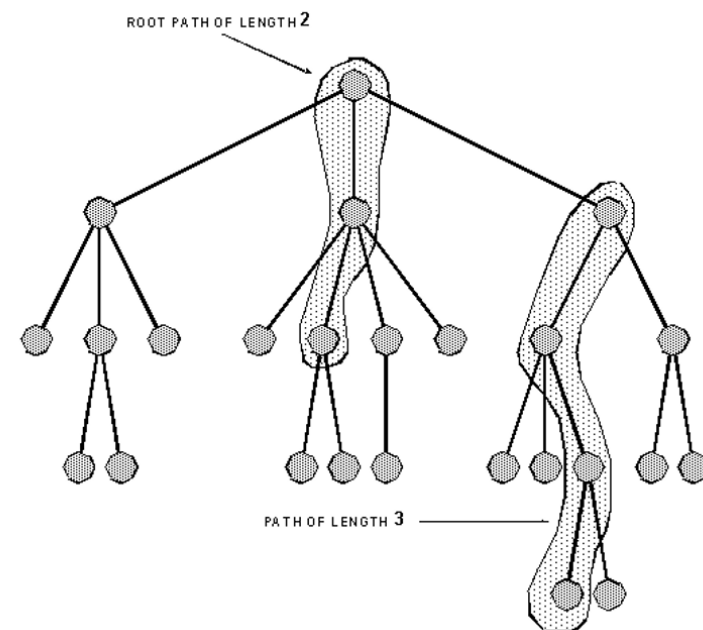
- 경로

- 루트경로: 루트~해당노드까지의 유일한 경로
- 루트-리프 경로: 리프노드에 이르는 루트 경로
- **경로의 길이**: 부모-자식 쌍의 개수. **노드의 개수-1**

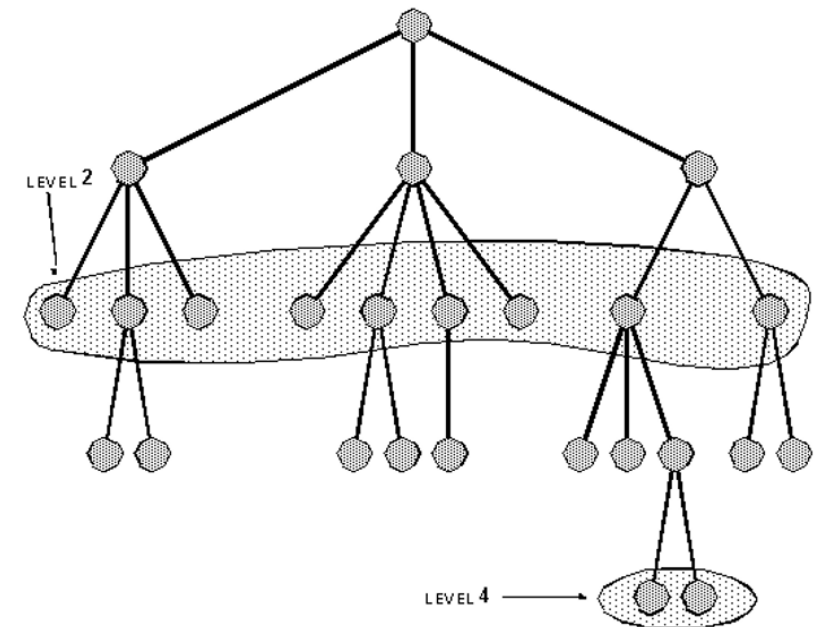
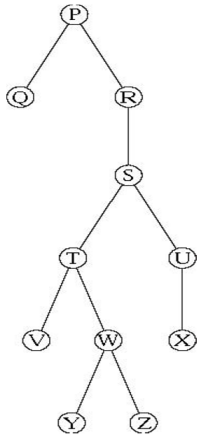
- 트리의 크기: 노드의 개수

- 공백 트리: 크기가 0인 유일한 트리
- 단독 트리: 크기가 1인 트리

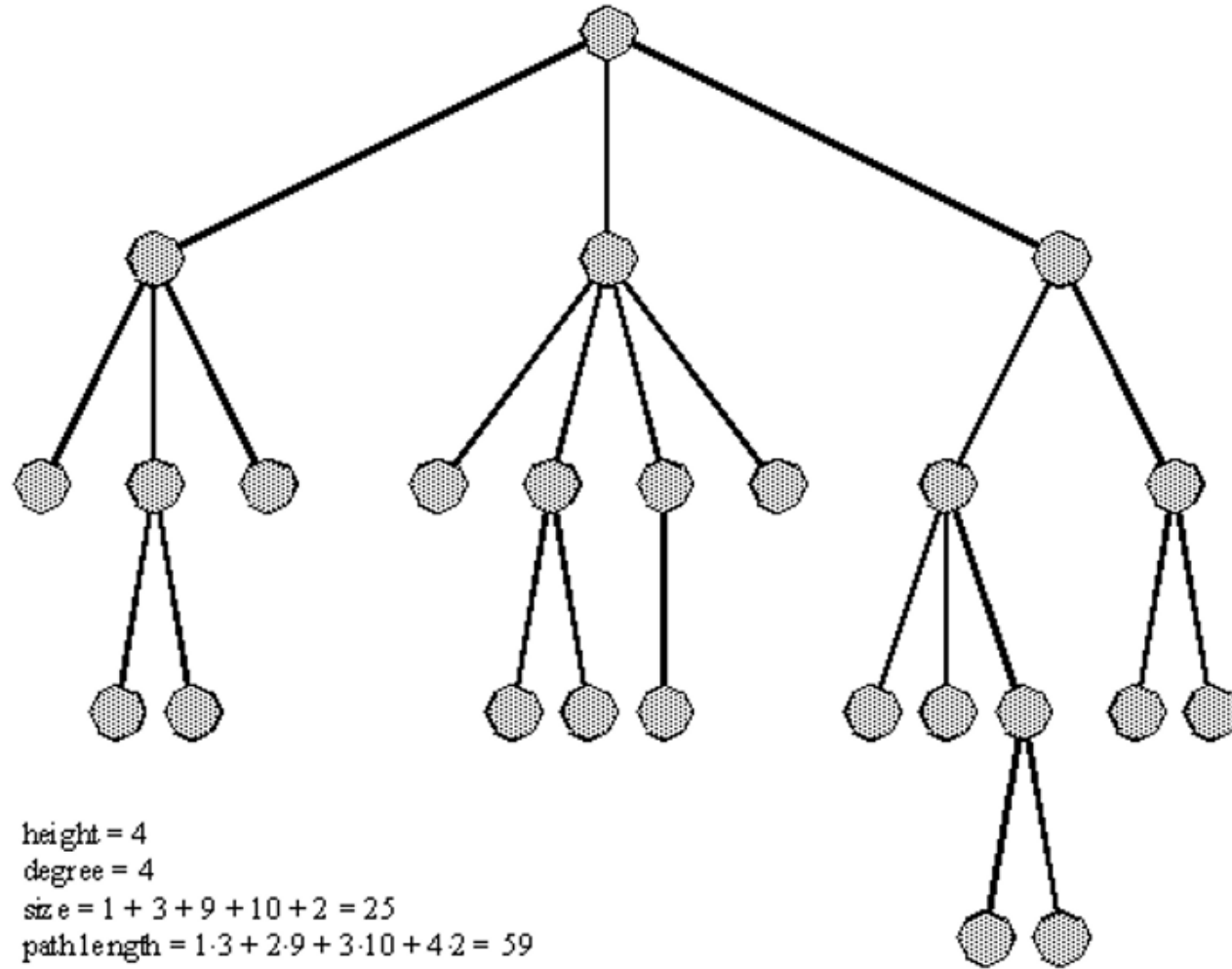
- 서브트리와 슈퍼트리



- 트리의 높이: 가장 루트 경로의 길이 = 가장 루트-리프 경로의 길이
- 노드의 깊이: 노드의 루트 경로의 길이
- 레벨: 주어진 깊이에 있는 모든 노드로 구성. **루트는 레벨0**
- 트리의 경로 길이(path length): 트리에 있는 모든 노드에 대한 깊이의 합 $0*1 + 1*2 \dots = 32$
- 트리의 너비: 최대 레벨 크기
- 내부 노드: 리프가 아닌 노드
- 외부 노드: 리프 노드가 데이터가 없는 더미 노드로 사용될 경우
- 내부 경로 길이: 내부 노드에만 적용되는 경로 길이
- 외부 경로 길이: 어떤 레벨에서 외부 노드의 개수와 레벨을 곱한 값으로 주어지는 가중치의 합계



경로 길이의 예: 허프만 코드 트리



포화 트리 (full tree)

: 모든 리프가 같은 레벨에 있고, 모든 노드가 동일한 차수를 가지는 트리.

◆ 포화 트리의 크기

– The size n of a full tree of degree d and height h

$$n = \frac{d^{h+1} - 1}{d - 1}$$

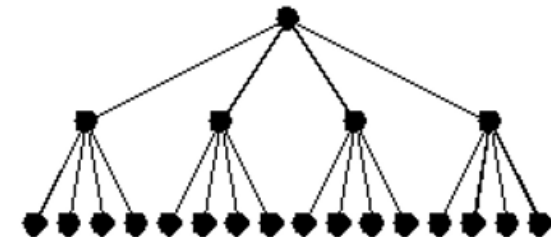
◆ 포화 트리의 크기에 대한 범위(차수:d, 높이:h)

$$h + d \leq n \leq \frac{d^{h+1} - 1}{d - 1}$$

각 노드의 차수 = 트리의 차수

각각의 차수와 높이를 가지는 포화 트리는 유일

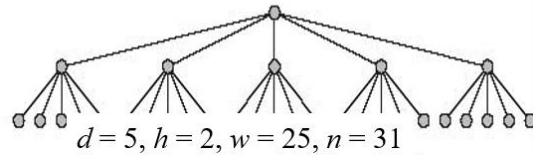
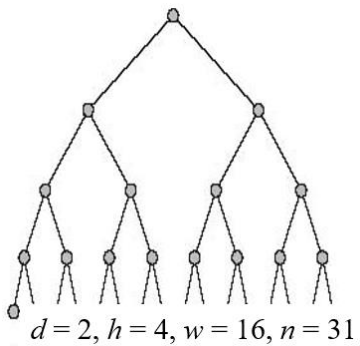
A full tree of degree $d = 4$ and height $h = 2$:



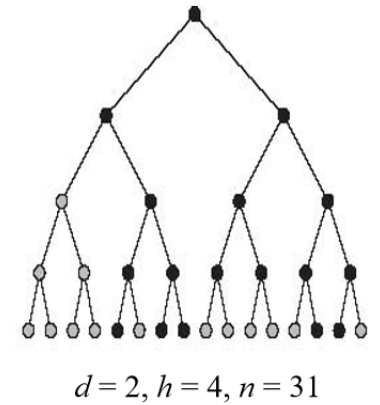
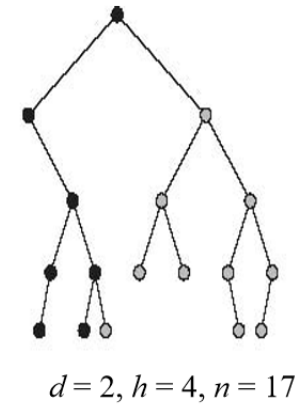
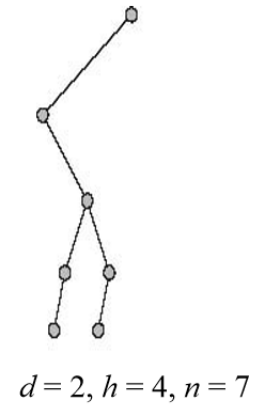
$$n = \frac{d^{h+1} - 1}{d - 1} = \frac{(4)^{(2)+1} - 1}{(4) - 1} = \frac{4^3 - 1}{3} = \frac{64 - 1}{3} = 21$$

- 각각의 차수와 높이를 가지는 포화 트리는 유일

포화(full) 트리



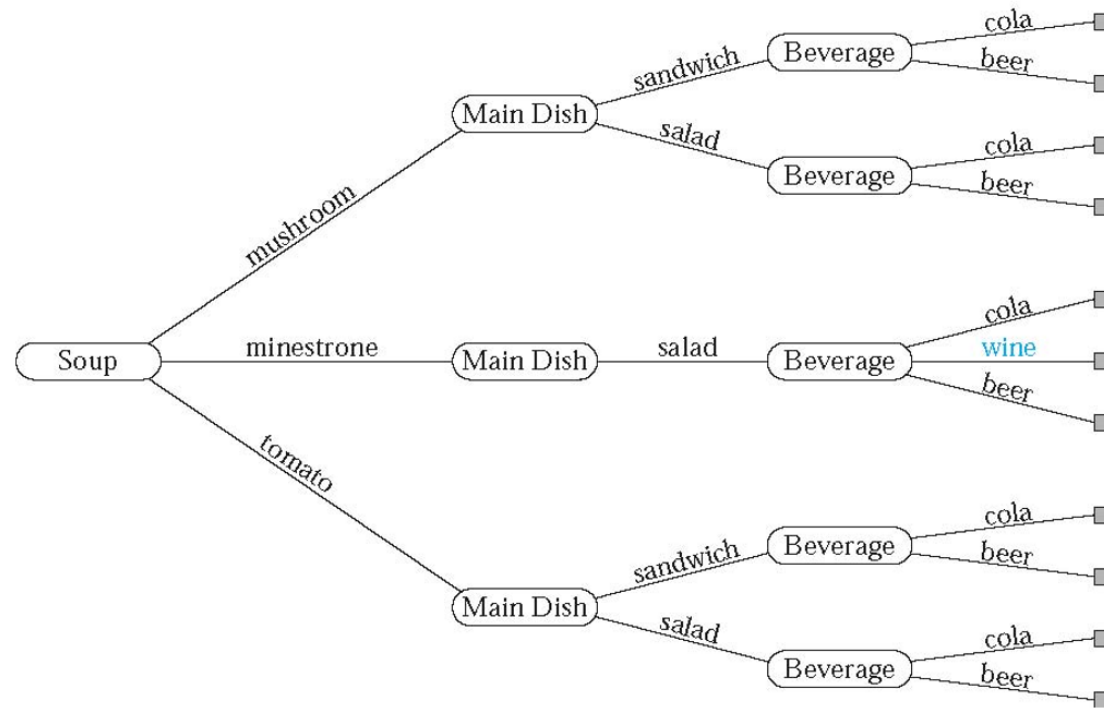
차수가 2이고 높이가 4인 트리



결정 트리(decision tree)

: 여러가지 대안들의 시퀀스를 분석하는데 사용되는 무순서트리

식사 주문에 사용되는 다른 결정 트리



순서 트리(ordered tree)

- 서브트리의 시퀀스를 가지고 있음



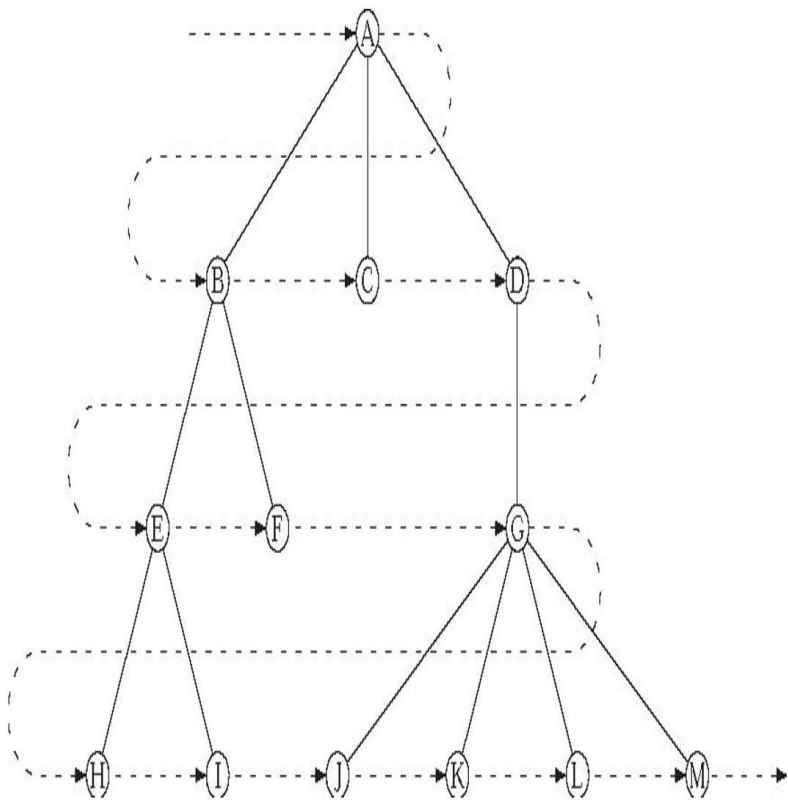
- 순서트리 \neq 정렬트리

- 순서 트리를 위한 순회 알고리즘
 - 레벨 순서(level order) 순회
 - 전위(preorder)순회 – 루트 노드 우선
 - 후위(postoreder)순회 – 루트 노드 나중

모두 $O(n)$ 시간에 실행된다.

레벨 순서 순회

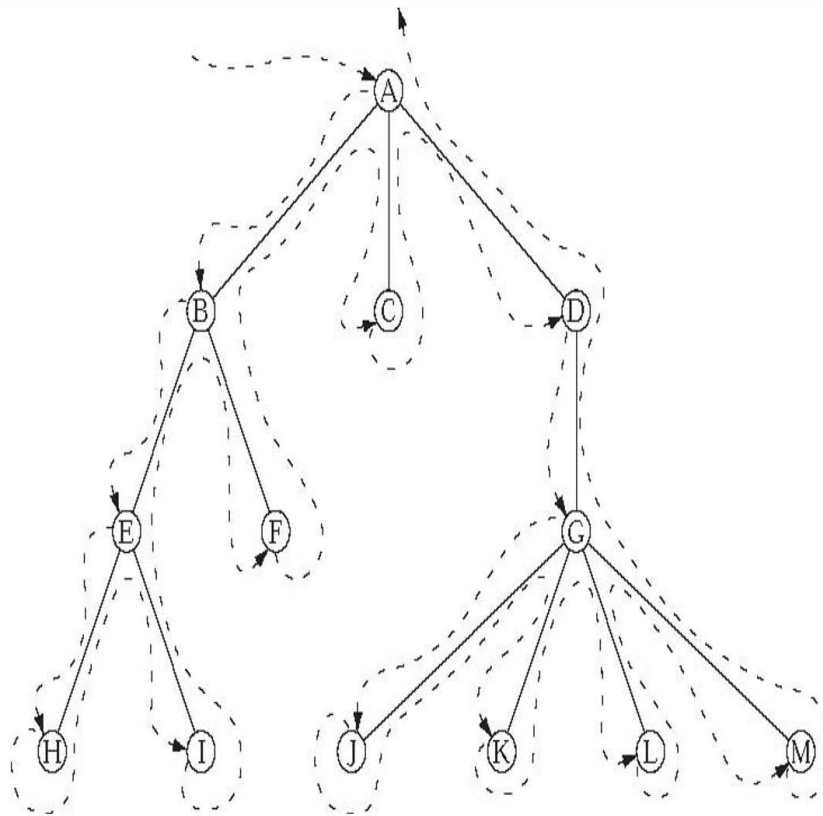
레벨 순서 순회



큐를 사용

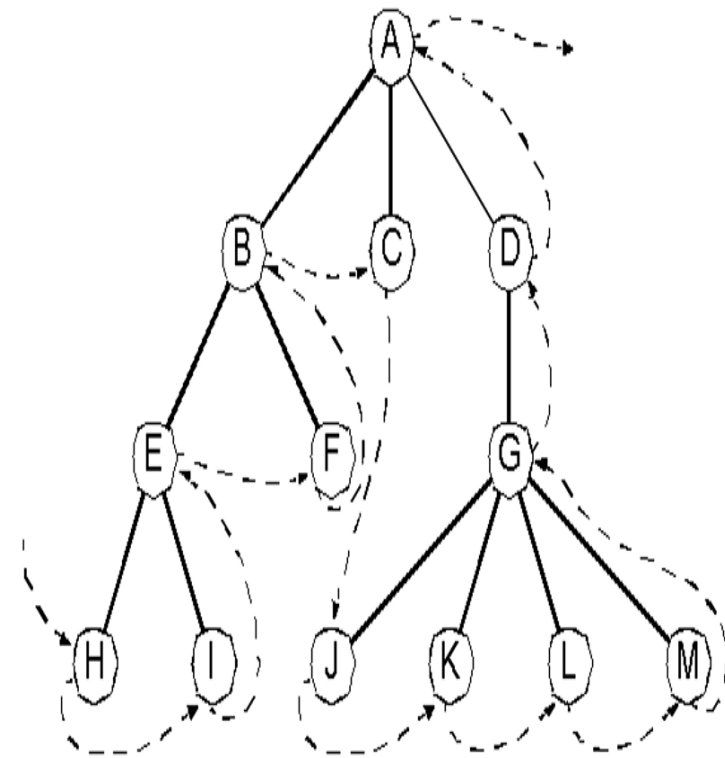
전위 순회

전위 순회



스택 사용

후위 순회



Postorder traversal

스택 사용

- **선형화(linearize):** 트리를 메모리나 디스크에 저장하는 방법. 위의 3가지 순회 방법이 배열이나 연결리스트에 저장할 수 있도록 해줌. 이러한 과정을 **직렬화(serialization)**이라고 부름
- 자연 매핑: 레벨 순서 순회를 사용하여 선형화 하는 과정(부모와 자식의 배열 인덱스를 계산하는데 간단한 공식을 사용할 수 있기 때문)

완전 순서 트리(complete ordered tree)

: 최하위 레벨의 노드 중 오른쪽 몇 개의 원소가 없는 것을 제외하고는 포화 트리와 같은 순서 트리

