



2020 학년도 1 학기

컴퓨터 정보과

자료구조(Data Structures)

담당교수 : 김주현

제 5 주차 / 제 1 차시

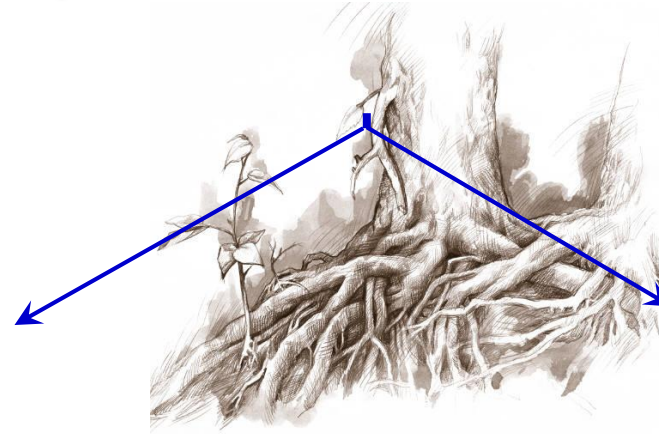


Tree

- 원소들 간에 1:多 관계를 가지는 비선형 자료구조
- 원소들 간에 계층관계를 가지는 계층형 자료구조
- 상위 원소에서 하위 원소로 내려가면서 확장되는 트리(나무)모양의 구조
- 트리는 DAG(Directed Acyclic Graphs, 방향성이 있는 비순환 그래프)의 한 종류
- 루트에서 어떤 노드로 가는 경로는 유일
- 한 개의 루트 노드만이 존재하며 모든 자식 노드는 한 개의 부모 노드만을 가짐



하나의 줄기에서 가지로 뻗어나가면서
확장되는 구조



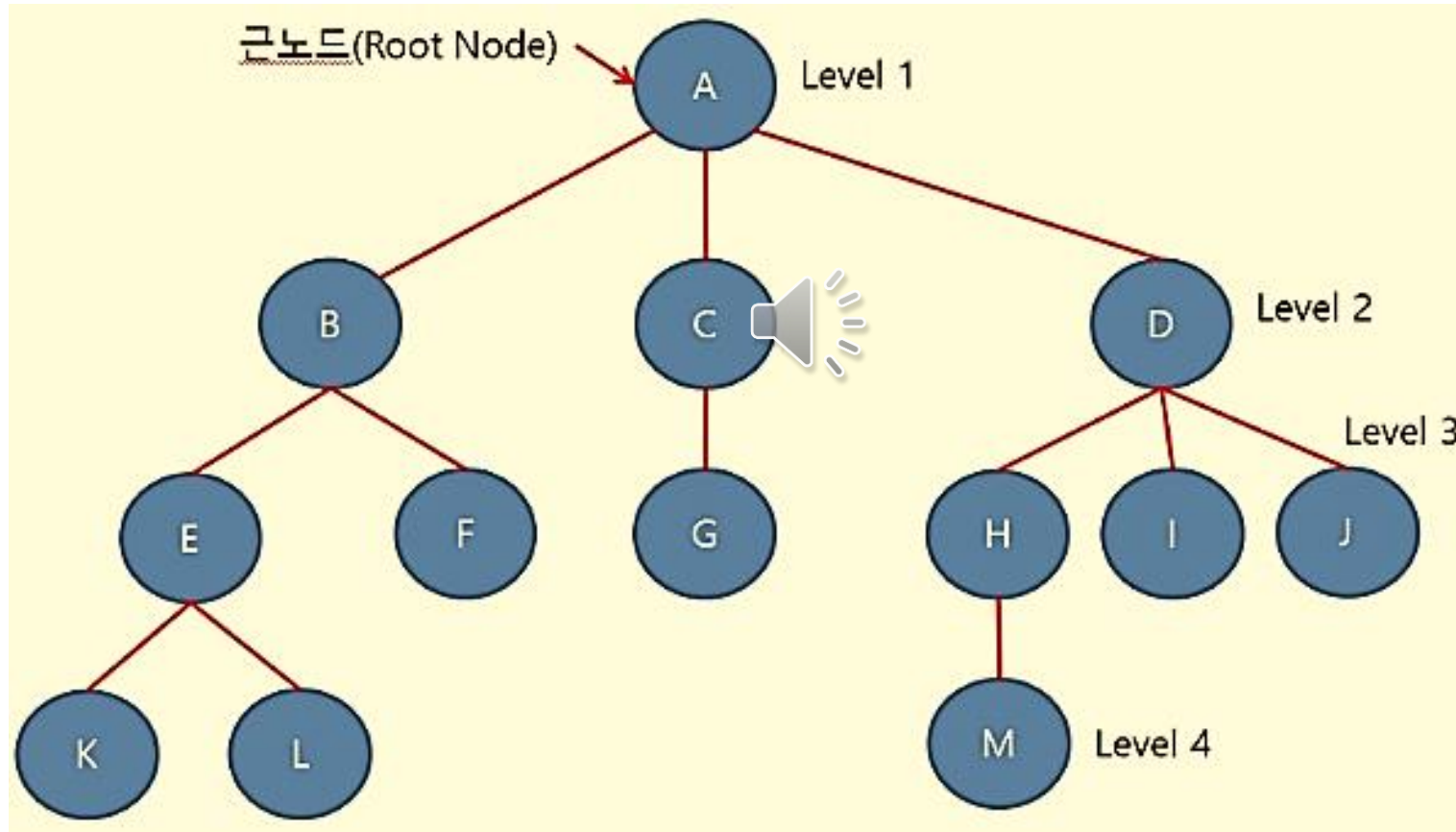
하나의 그루터기에서 뿌리로 뻗어나가면서
확장되는 구조

응용

- 가족관계나 조직이나 기관의 계층구조
- 컴퓨터 운영체제의 파일 시스템
- 자바 클래스 계층구조 등
- 트리는 일반적인 트리와 이진트리(Binary Tree)로 구분
- 다양한 탐색트리(Search Tree), 힙(Heap) 자료구조, 컴파일러의 수식을 위한 구문트리(Syntax Tree) 등의 기본이 되는 자료구조로서 광범위하게 응용
- etc



트리(Tree) 관련용어



트리(Tree) 관련용어

노드(Node) : 트리의 기본요소로서 자료 항목과 다른 항목에 대한 가지(Branch)를 합친 것

EX : A, B, C, D, E, F, G, H, I, J, K, L, M

근 노드(Root Node) : 트리의 맨 위에 있는 노드

EX : A

디그리(Degree, 차수) : 각 노드에서 뻗어나온 가지의 수

EX : A=3 , B=2, C=1

단말 노드(Terminal Node) : 자식이 하나도 없는 노드

EX : K, L, F, G, M, I, J

비단말 노드(Non-Terminal Node) : 자식이 하나라도 있는 노드, 즉 Degree가 0이 아닌 노드

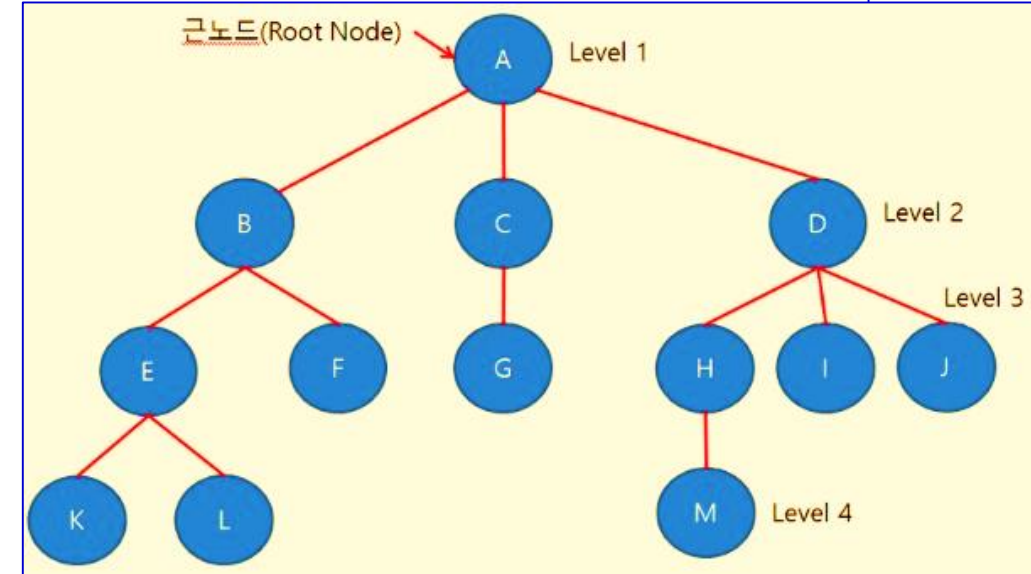
EX : A, B, C, D, E, H

조상 노드(Ancestors Node) : 임의의 노드에서 루트 노드에 이르는 경로상에 있는 노드들

EX : M의 조상 노드는 H, D, A

자식 노드(Child Node) : 어떤 노드에 연결된 다음 레벨의 노드들

EX : D의 자식노드는 : H, I, J



트리(Tree) 관련용어

부모 노드(Parent Node): 어떤 노드에 연결된 이전 레벨의 노드들
EX: E,F의 부모노드는 B

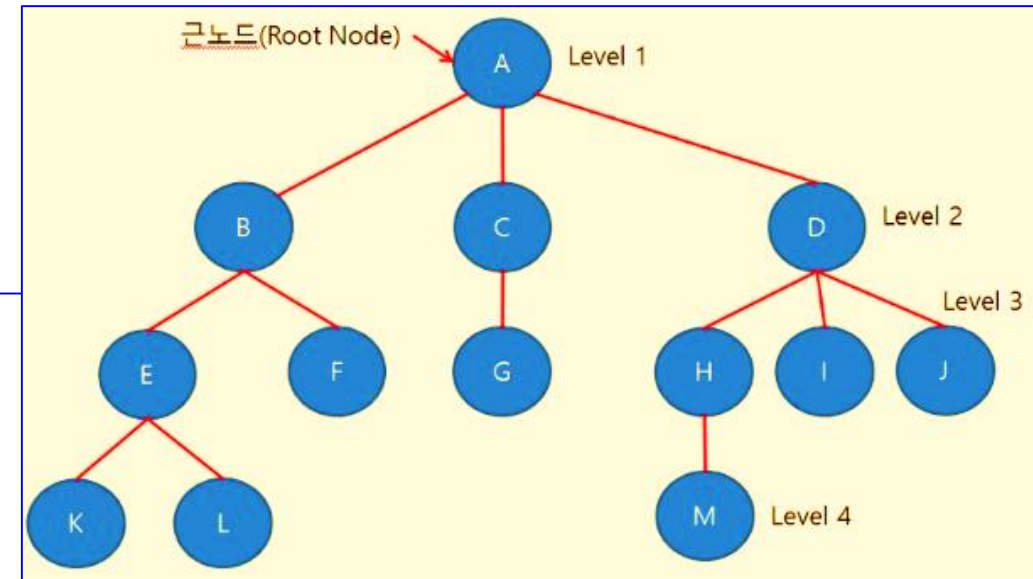
형제 노드(Brother Node, Sibling): 동일한 부모를 갖는 노드들
EX: H의 형제 노드는 I, J

Level or Depth: 루트 노드의 **Level**을 1로 한 후 자손노드들의 단계,
EX: H의 레벨은 3

높이(Height): Tree에서 노드가 가질 수 있는 최대의 레벨
EX: 위 트리의 높이는 4

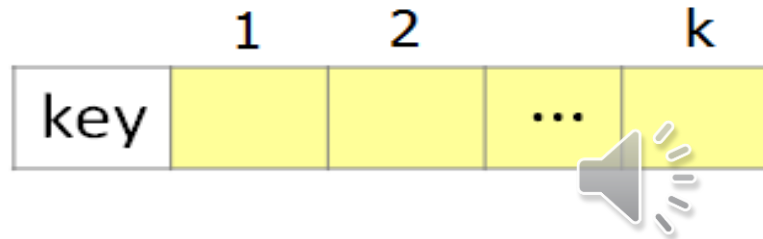
숲(Forest): 여러 개의 트리가 모여있는 것
EX: 위 트리에서 근 노드 **A**를 제거하면 **B,C,D**를 근 노드로 하는 세 개의 트리가 있는 숲이 생긴다.

트리의 차수(degree): 노드들이 차수 중에서 가장 많은 수
EX: 노드 **A**나 **D**가 세 개의 차수를 가지므로 위 트리의 차수는 3이다.



Tree

- 일반적인 트리를 메모리에 저장하려면 각 노드에 키(혹은 데이터)와 자식 수만큼의 레퍼런스 필요
- 트리의 차수가 k라면, k개의 레퍼런스 필드를 다음과 같이 선언



- N개의 노드가 있는 최대 차수가 k인 트리

$$\text{null 레퍼런스 수} = Nk - (N-1) = N(k-1) + 1$$

Nk = 총 레퍼런스의 수

$(N-1)$ = 트리에서 부모-자식을 연결하는 레퍼런스 수

- k가 클수록 **메모리의 낭비**가 심해지는 것은 물론 트리를 탐색하는 과정에서 null 레퍼런스를 확인해야 하므로 시간적으로도 **매우 비효율적**

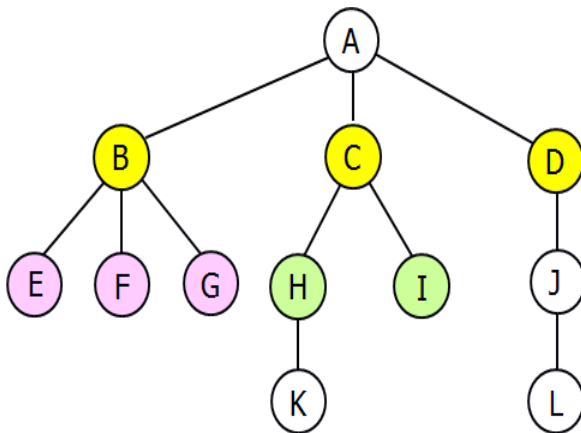


왼쪽자식 - 오른쪽 형제(left child- right sibling)

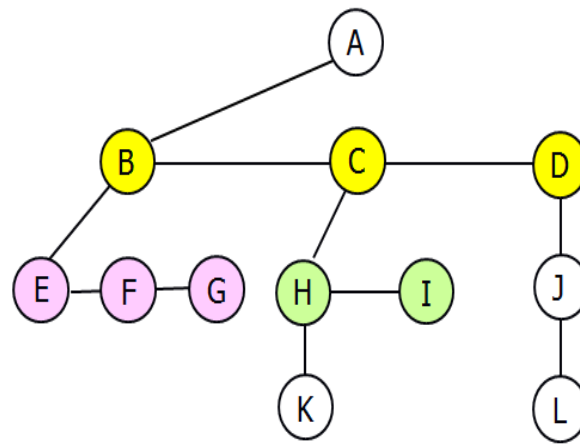
노드의 왼쪽 자식과 왼쪽 자식의 오른쪽 형제노드를 가리키는 2개의 레퍼런스만을 사용

key	
왼쪽 자식	오른쪽 형제

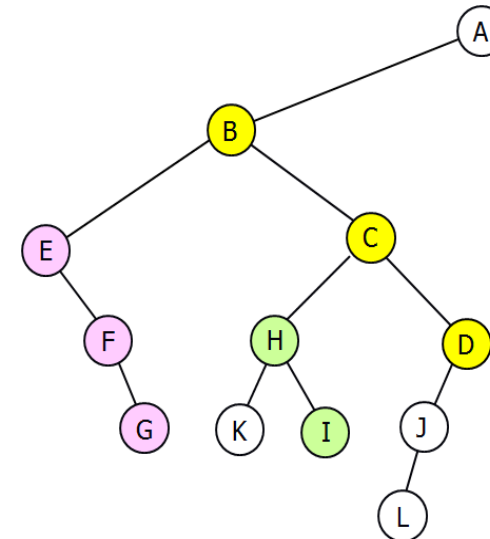
(a)의 트리를 왼쪽자식-오른쪽형제 표현으로 변환하면, (b)의 트리를 얻으며, (c)는 (b)의 트리를 45° 시계 방향으로 회전시킨 것



(a)



(b)



(c)



Reference

- https://ko.wikipedia.org/wiki/%ED%8A%B8%EB%A6%AC_%EA%B5%AC%EC%A1%B0https://jsieun73.tistory.com/26
- <https://coding-factory.tistory.com/231>
- <https://gmlwjd9405.github.io/2018/08/12/data-structure-tree.html>
- 자바로 배우는 쉬운 자료구조, 이지영, 한빛아카데미
- 자바와 함께하는 자료구조의 이해, 양성봉, 생능출판

언제 어디서나 즐^공 열^공, 진공하세요.

감사합니다

