

2주차 1차시 기본자료 구조(비선형구조)

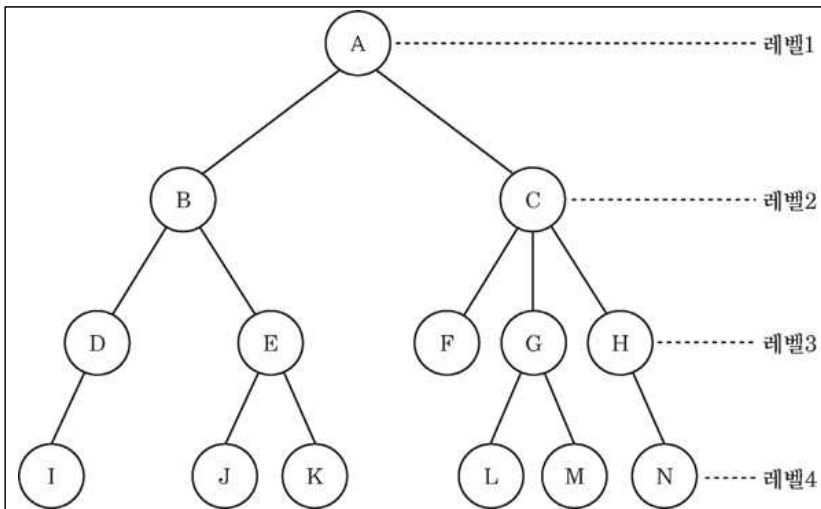
【학습목표】

1. 트리를 이해할 수 있다.
2. 그래프를 이해할 수 있다.

학습내용1 : 트리

* 트리(Tree)란?

- 비선형구조로 노드(Node)와 가지(Branch)로 연결된 그래프의 특수한 형태로 계층구조를 이룸



1. 트리의 기본 용어

① 노드(Node)

- 바로 전 그림과 같이 트리 각 각 알파벳 부분에 해당 하는 것
- 예시 : A B C D E F G H I J K L M N

② 근 노드(Root Node)

- 트리의 맨 위에 노드
- 예시 : A

③ 부모노드(Parent Node)

- 노드에 연결된 이전 레벨의 노드
- 예시 : F G H 의 부모노드 C (J K 의 부모노드 E)

④ 자식노드(Child Node)

- 노드에 연결된 다음 레벨의 노드
- 예시 : B 자식노드 D E

⑤ 형제노드(Brother Node)

- 동일한 부모를 갖는 노드
- 예시 : D와 E는 부모 B로 형제

⑥ 노드의 차수[Degree]

- 노드의 서브 트리수, 혹은 노드의 가지 수
- 예시 : A의 차수=2, D의 차수=1

⑦ 트리의 차수

- 노드들의 차수 중에서 가장 많은 수
- 예시 : C 가 가장 많은 자녀 3 이므로 디그리 = 3

⑧ 단말노드(Terminal Node)

- 차수가 0인 노드를 말하며, 나뭇잎 노드(Leaf Node)라고도 함
- 예시 : 나뭇잎 노드(Leaf Node) = F I J K L M N

⑨ 깊이(Depth, Height)

- 트리에서 노드가 가질 수 있는 최대의 레벨 값
- 예시 : 4

⑩ 레벨(Level)

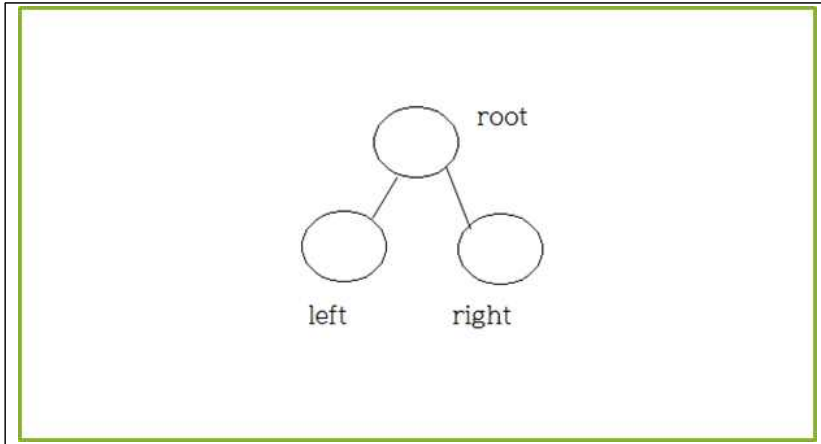
- 루트 노드의 레벨을 1로 가정 했을 때 각 노드들이 속해 있는 깊이
- 예시 : A의 레벨 = 1
B, C의 레벨 = 2
D, E, F, G, H의 레벨 = 3

2. 이진트리(Binary Tree)

* 이진트리란?

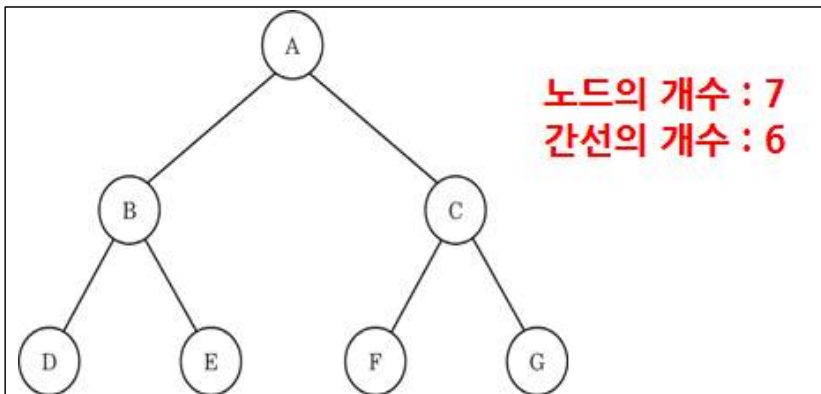
- 트리 중에서 가장 많이 쓰이는 트리가 이진 트리
- 모든 노드가 2개의 서브트리를 가지고 있는 트리를 이진트리 라고 함
- 따라서 이진 트리의 노드에는 최대 2개의 자식 노드가 존재할 수 있고 모든 노드의 차수는 2 이하가 된다. 공집합도 이진 트리가 된다.

1) 이진 트리의 정의

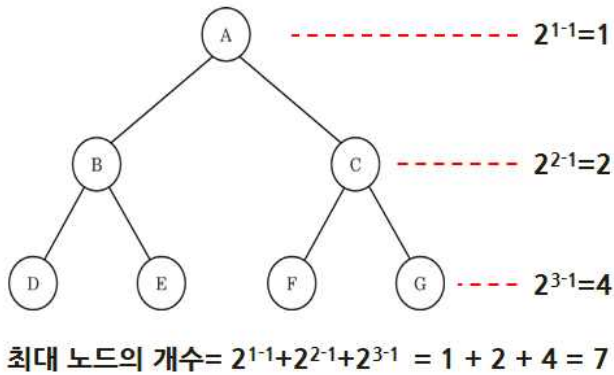
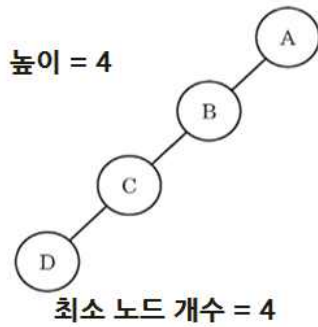


2) 이진트리의 성질

n 개의 노드를 가진 이진트리는 정확하게 $n-1$ 개의 간선을 가진다. 그 이유는 이진트리에서의 노드는 루트를 제외하면 정확하게 하나의 부모노드를 가진다. 그리고 부모와 자식 간에는 정확하게 하나의 간선만이 존재한다. 따라서 간선의 개수는 $n-1$ 이다.



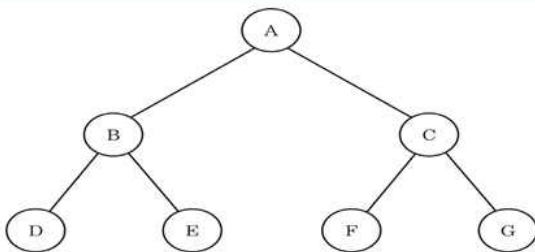
높이가 h 인 이진 트리의 경우, 최소 h 개의 노드를 가지며 최대 2^h-1 개의 높이를 가지게 된다. 그 이유는 한 레벨에 적어도 하나의 노드가 존재해야 하므로 높이가 h 인 이진트리는 적어도 h 개의 노드를 가진다. 또한 하나의 노드는 2^{i-1} 이 된다. 따라서 전체 노드의 개수는 $\sum_{i=1}^h 2^{i-1} = 2^h - 1$ 이 된다.



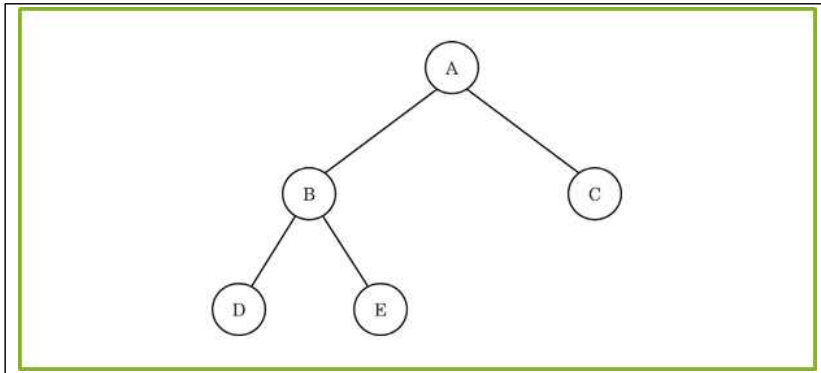
3) 이진트리의 종류

① 정 이진 나무(full binary tree) 포화 이진트리

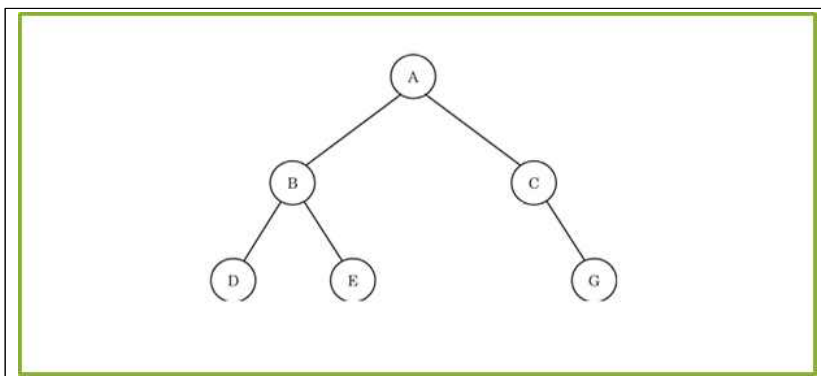
- 노드의 개수 : $2^n - 1$



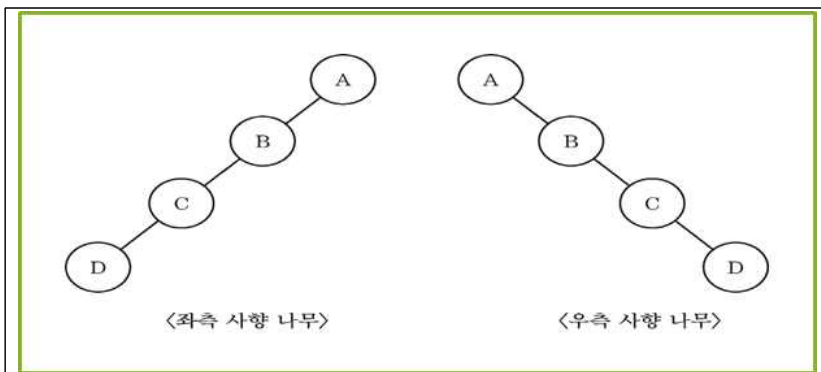
② 전 이진 나무(complete binary tree), 완전 이진트리



③ 기타 이진 트리

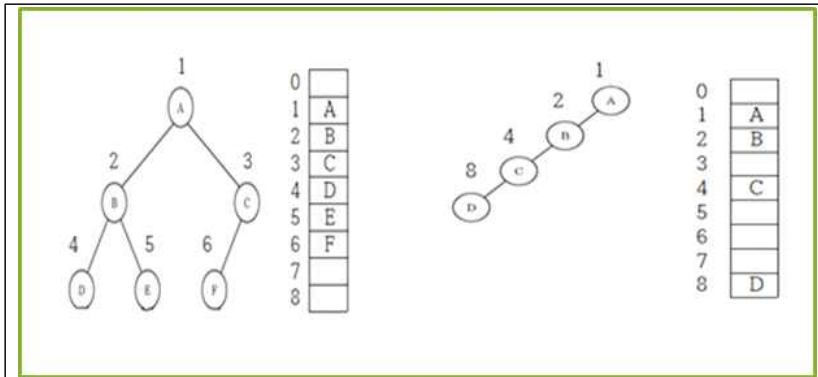


④ 사향 이진 나무(skewed binary tree), 경사이진 트리

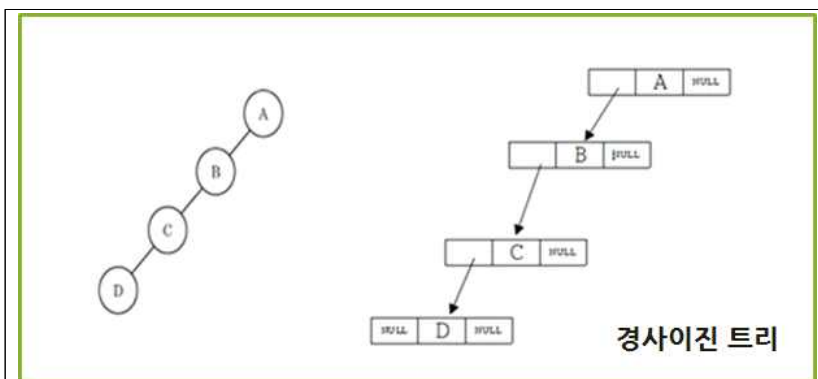
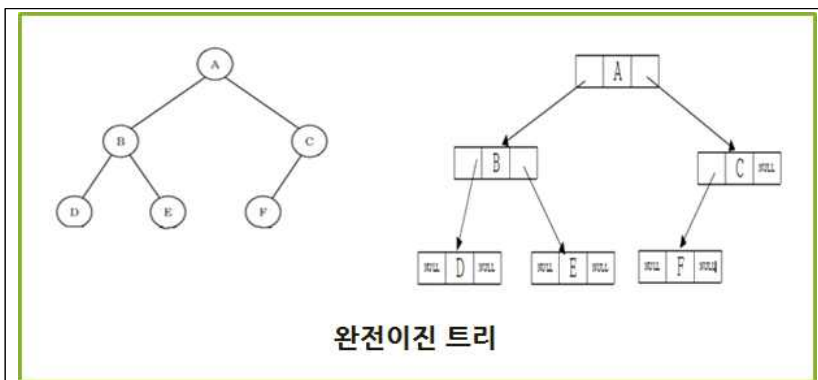
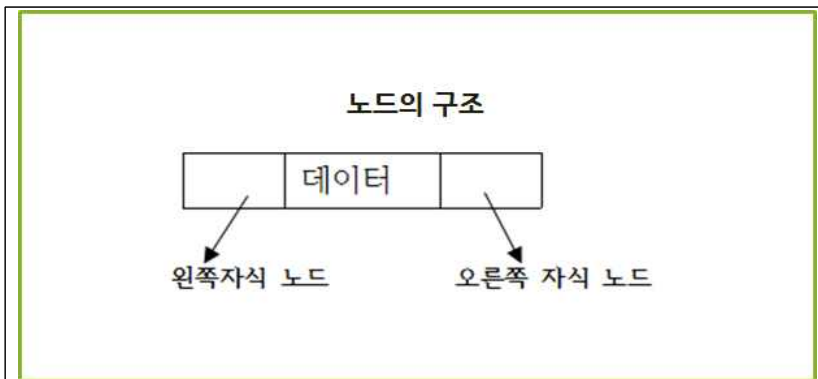


4) 이진트리의 표현

① 배열을 이용하는 방법

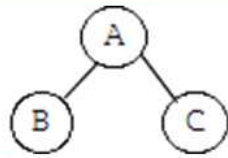


② 링크표현법



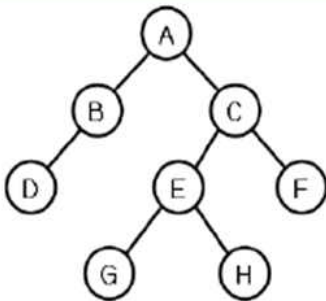
3. 이진 트리의 운행법

* 이진트리의 운행법은 다음 세 가지가 있음



- **Preorder 운행**
: Root → Left → Right 순서대로 운행함(A, B, C)
- **Inorder 운행**
: Left → Root → Right 순서대로 운행함 (B, A, C)
- **Postorder 운행**
: Left → Right → Root 순서대로 운행함 (B, C, A)

* 다음 트리를 Preorder 운행, Inorder 운행, Postorder 운행 순서를 구하라



- **Preorder 운행**
: Root → Left → Right
순서이므로
(A, B, D, C, E, G, H, F)
- **Inorder 운행**
: Left → Root → Right
순서이므로
(D, B, A, G, E, H, C, F)
- **Postorder 운행**
: Left → Right → Root 순서
(D, B, G, H, E, F, C, A)

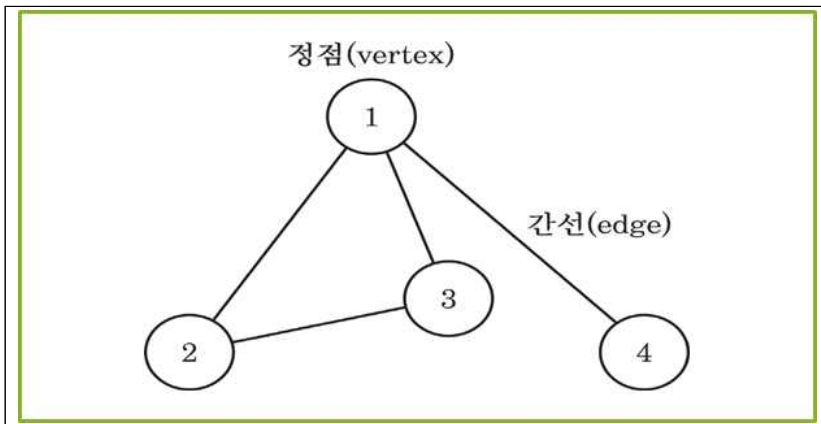
학습내용2 : 그래프

1. 그래프의 정의

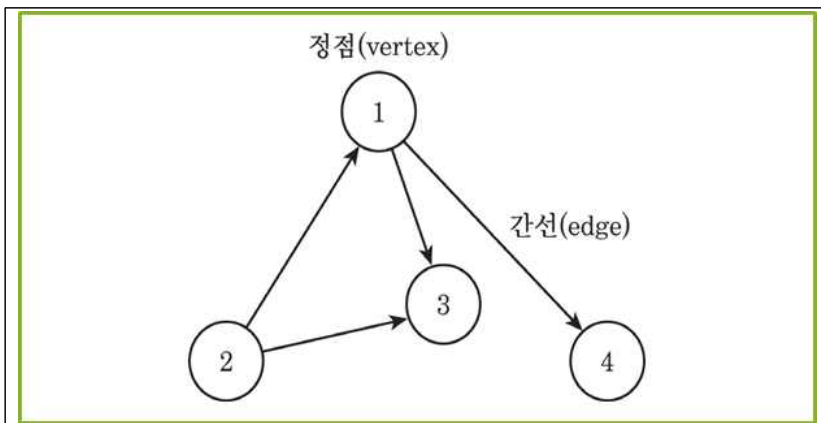
그래프란 각각의 단위 정보를 링크로 연결하여 구조화 시킨 비선형 자료구조로서 무 방향 그래프(Undirected graph)와 방향 그래프(Directed graph), 완전 그래프가 있다. 일반적으로 그래프 하면 보통 무 방향 그래프를 의미한다.

- 정점 (Vertex) : 노드들의 집합
- 간선(Edge) : 정점들 사이의 상호 연결의 집합

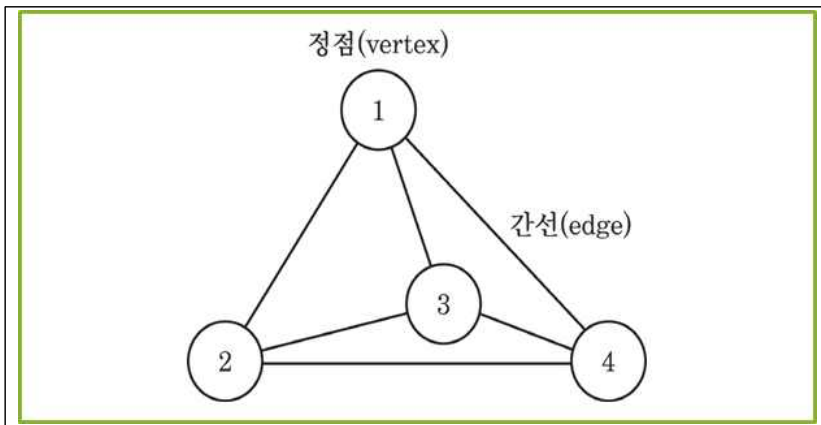
① 무 방향 그래프(Undirected graph)



② 방향 그래프(Directed graph)

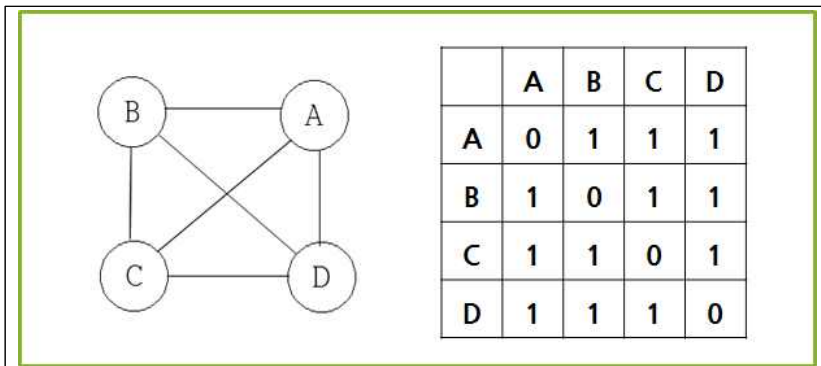


③ 완전 그래프

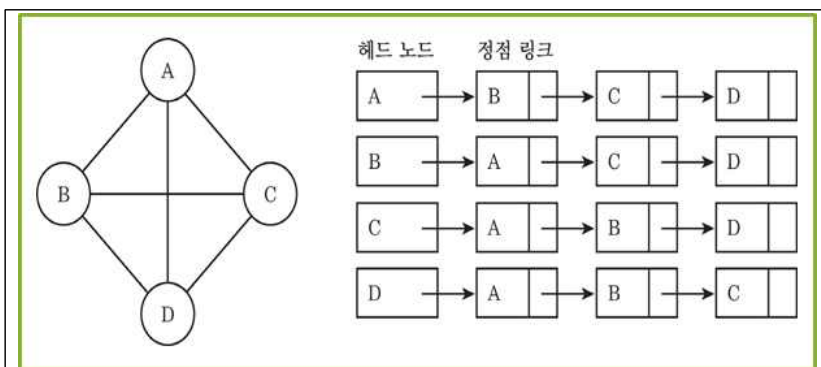


2. 그래프의 표현법

① 인접 행렬(Adjacency Matrix) : 배열을 사용하여 그래프를 표현한 것이다.



② 인접 리스트 : 포인터 형 변수를 사용하여 그래프를 표현한 것이다.



3. 그래프의 순회 알고리즘

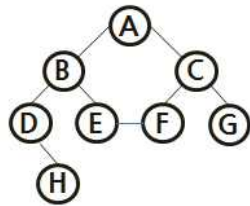
그래프의 모든 정점을 체계적으로 방문하는 것으로 무 방향 그래프의 순회 방법은 깊이 우선 탐색과 너비 우선 탐색이 있다.

① 깊이 우선 탐색(DFS: Depth First Search)

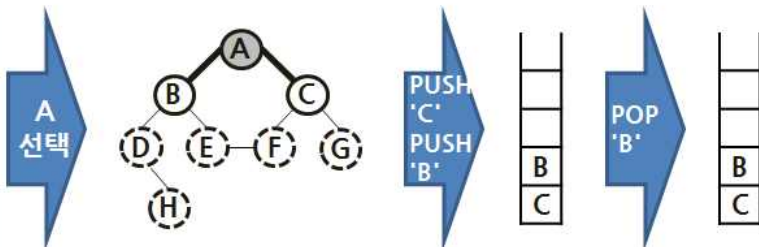
정점을 방문할 때 갈수 있는 데까지 우선 가보다가 더 이상 진행할 수 없으면 거슬러 올라 가면서 아직 가보지 않은 노드가 있으면 그 노드를 따라 갈수 있는데 까지 가보는 방법.

이 방법은 스택을 사용한다.

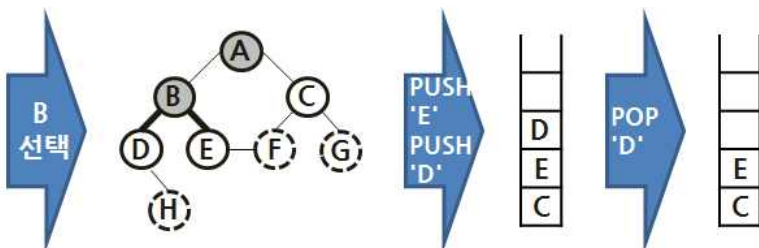
예) 다음 트리를 깊이우선 탐색(DFS)로 탐색 하라



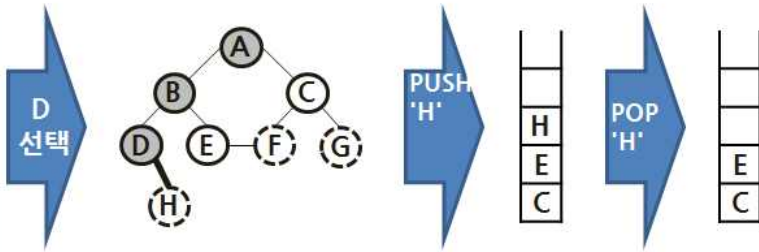
1단계



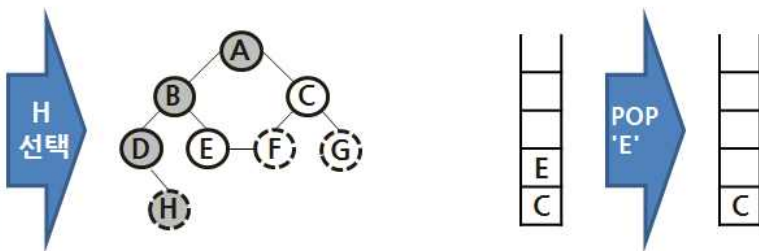
2단계



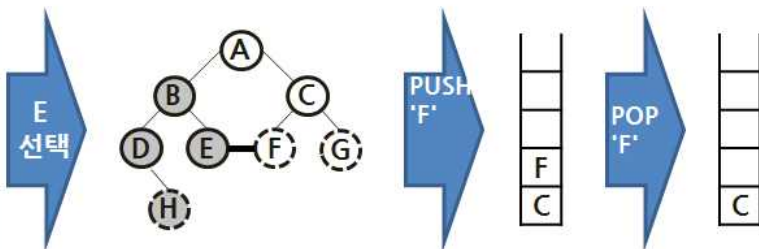
3단계



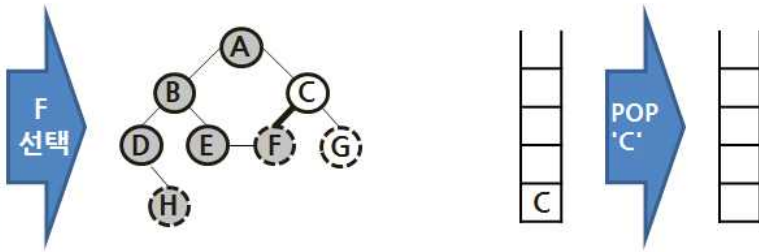
4단계



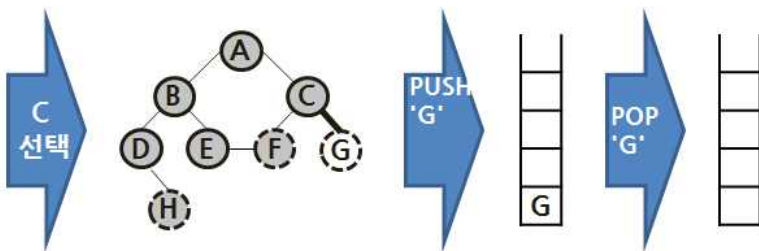
5단계



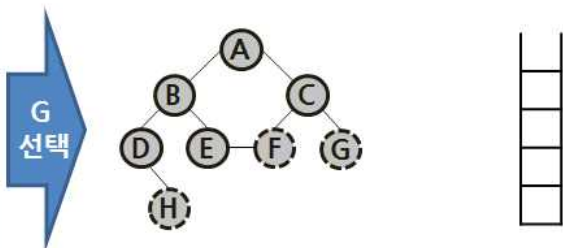
6단계



7단계



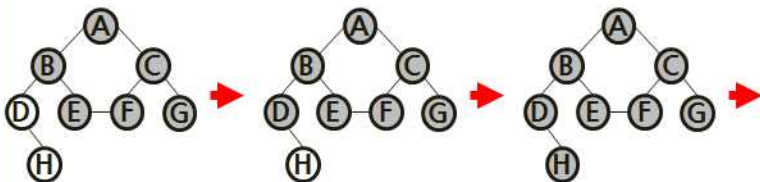
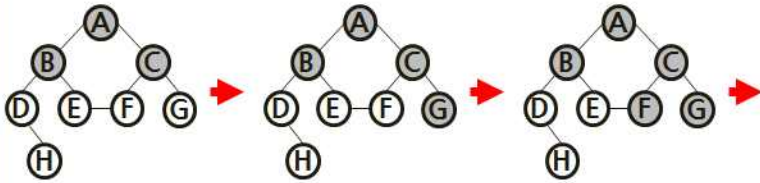
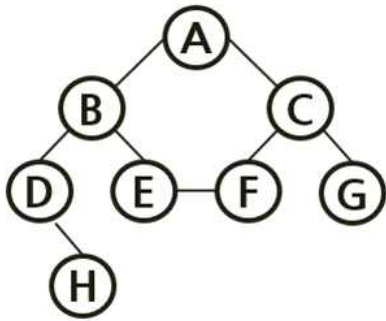
8단계



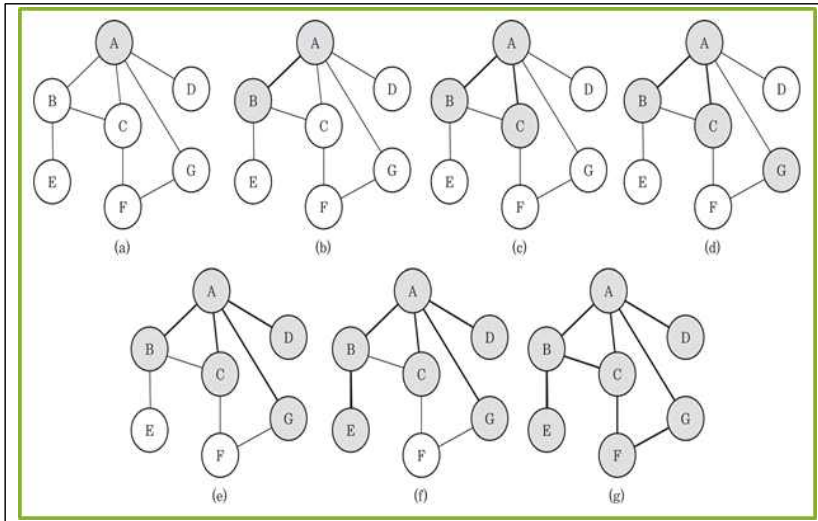
깊이우선 탐색 과정(A, B, D, H, E, F, C, G)이 됨

② 너비 우선 탐색 BFS(Breath First Search)

우선 가깝게 인접한 정점을 모두 방문한 후 그 다음으로 가깝게 인접한 정점을 방문하는 순으로 진행 한다.(주변정점 중에서 오래된 것부터 방문)



* 너비 우선 탐색 과정: A B C G D E F



【학습정리】

1. 나무(Tree)

- 트리(Tree)란 비선형구조로 노드(Node)와 가지(Branch)로 연결된 그래프의 특수한 형태로 계층구조를 이룬다.

2. 이진트리(Binary Tree)

- 트리 중에서 가장 많이 쓰이는 트리가 이진 트리이다. 모든 노드가 2개의 서브트리를 가지고 있는 트리를 이진트리라고 한다. 서브트리는 공집합일 수 있다. 따라서 이진 트리의 노드에는 최대 2개의 자식 노드가 존재할 수 있고 모든 노드의 차수는 2 이하가 된다. 공집합도 이진 트리가 된다. 이진트리는 정이진, 전이진, 기타이진, 사향이진 트리가 있다.

3. 그래프의 정의

- 그래프란 각각의 단위 정보를 링크로 연결하여 구조화 시킨 비선형 자료구조로서 무 방향 그래프(Undirected graph)와 방향 그래프(Directed graph), 완전 그래프가 있다. 일반적으로 그래프 하면 보통 무 방향 그래프를 의미한다.

4. 그래프의 순회 알고리즘

1) 깊이 우선 탐색(DFS: Depth First Search)

- 정점을 방문할 때 갈수 있는 데까지 우선 가보다가 더 이상 진행할 수 없으면 거슬러 올라 가면서 아직 가보지 않은 노드가 있으면 그 노드를 따라 갈수 있는데 까지 가보는 방법.
- 이 방법은 스택을 사용한다.

2) 너비 우선 탐색 BFS(Breach First Search)

- 우선 가깝게 인접한 정점을 모두 방문한 후 그 다음으로 가깝게 인접한 정점을 방문하는 순으로 진행 한다.(주변정점 중에서 오래된 것부터 방문)