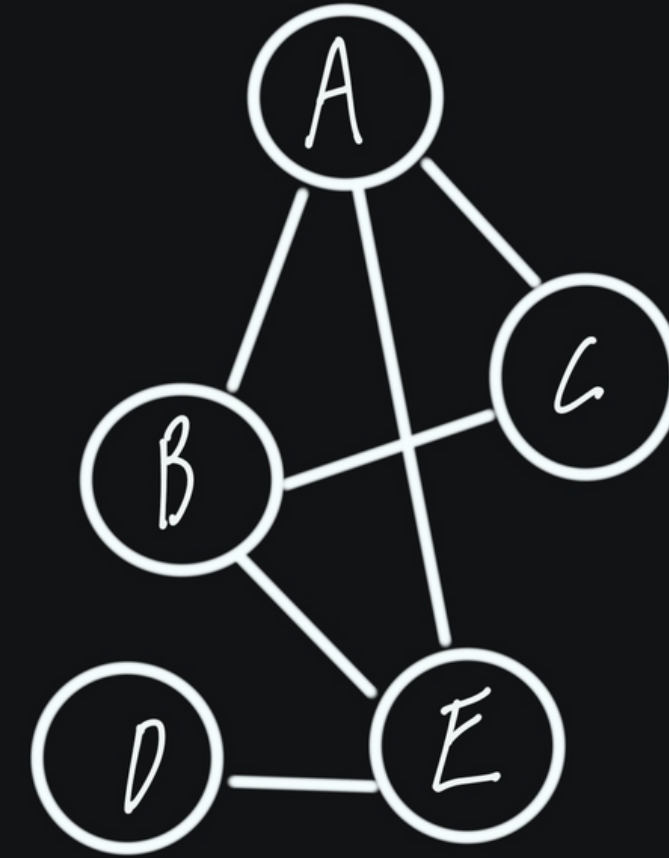


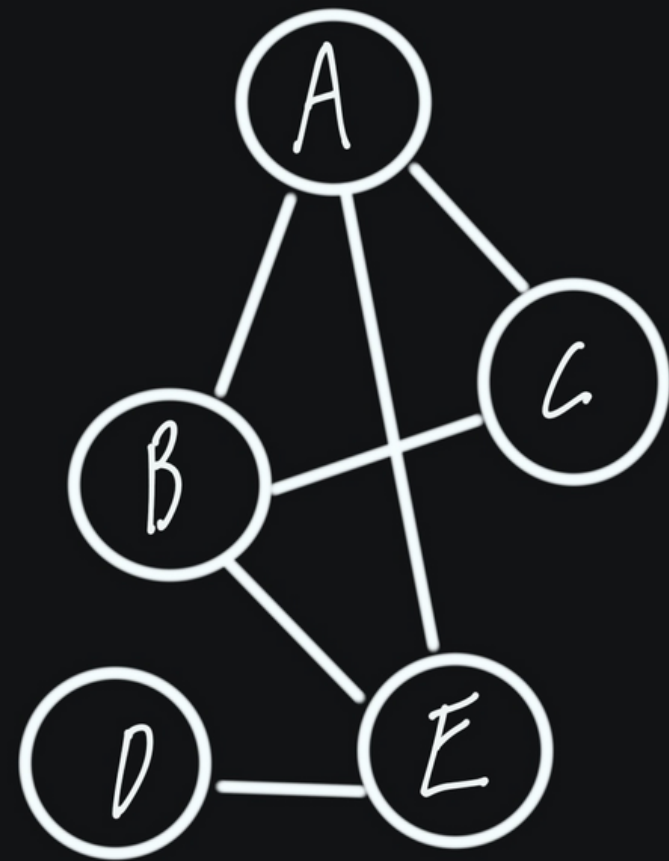
DFS/BFS

$$G = (V, E)$$



V (Vertex) : node의 집합
 E (Edge) : node간 연결 집합

$$V = \{A, B, C, D, E\}$$
$$E = \{(A, B), (A, C), (A, E), (B, A), (B, C), (B, E), (C, A), (C, B), (D, E), (E, A), (E, B), (E, D)\}$$



$V = \{A, B, C, D, E\}$
 $E = \{(A, B), (A, C), (A, E), (B, A), (B, C), (B, E),$
 $(C, A), (C, B), (D, E), (E, A), (E, B), (E, D)\}$

인접 행렬

	A	B	C	D	E
A	0	1	1	0	1
B	1	0	1	0	1
C	1	1	0	0	0
D	0	0	0	0	1
E	1	1	0	1	0

인접 리스트

$A \rightarrow [B, C, E]$

$B \rightarrow [A, C, E]$

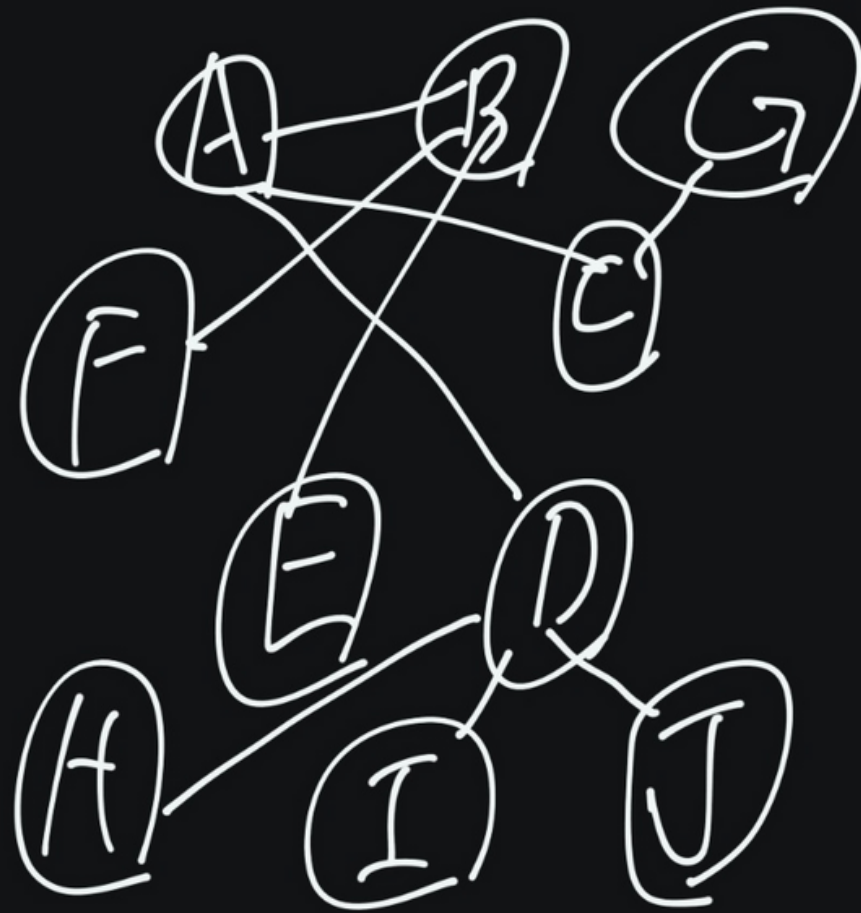
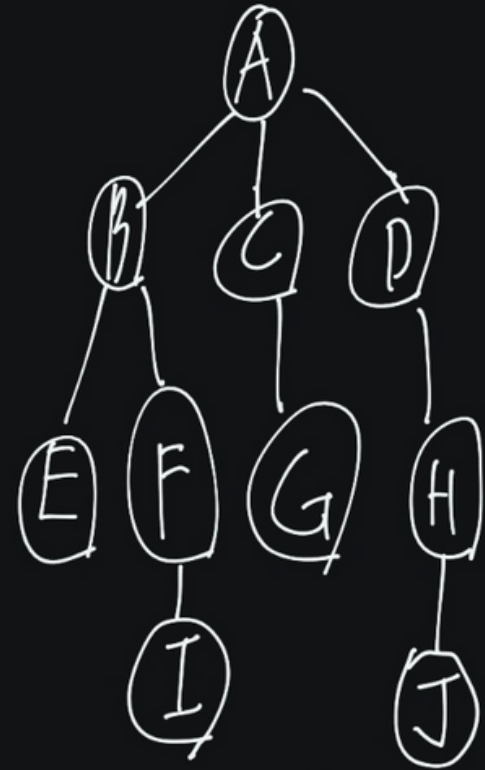
$C \rightarrow [A, B]$

$D \rightarrow [E]$

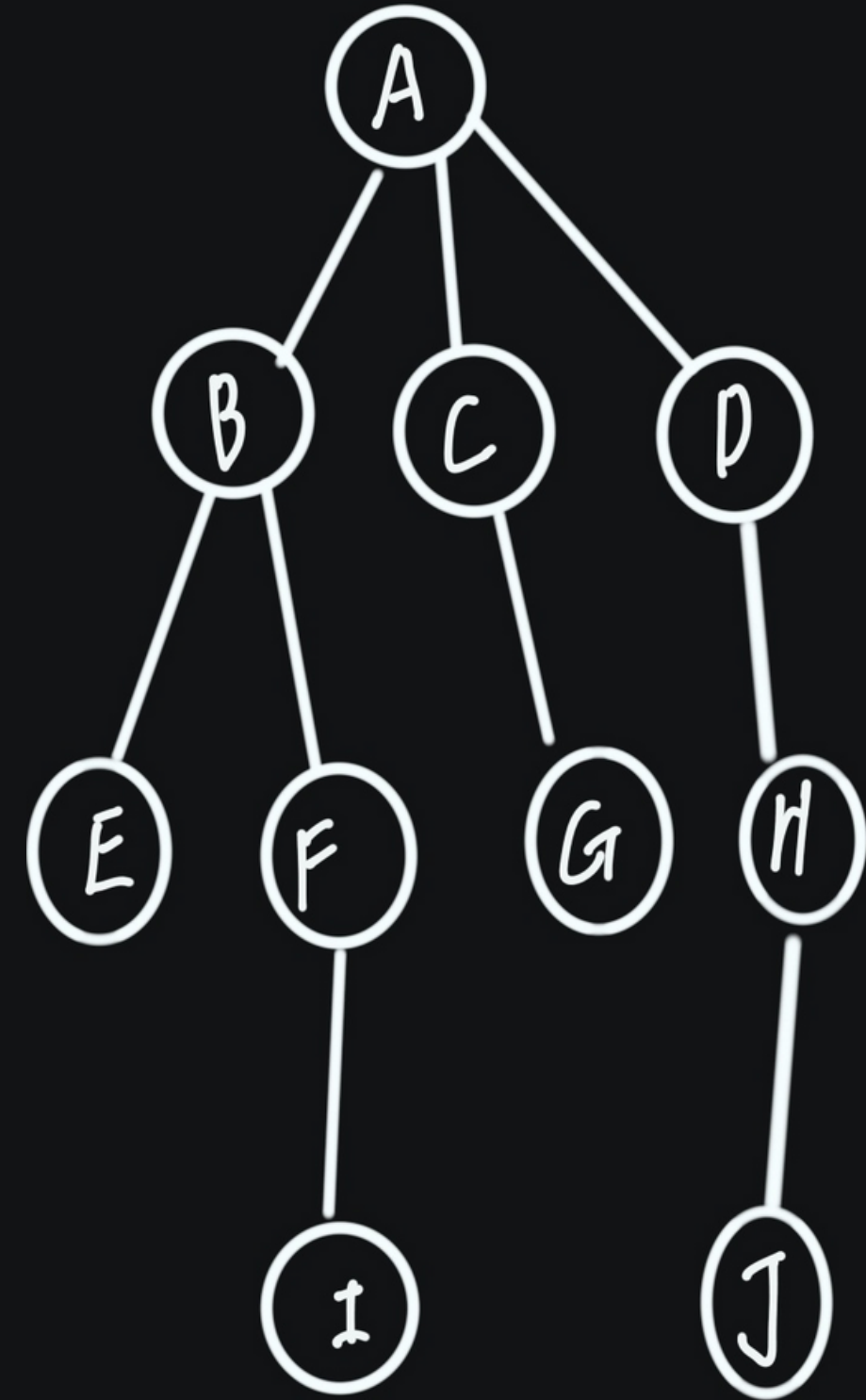
$E \rightarrow [A, B, D]$

이중리스트? 맵?

연결 그래프

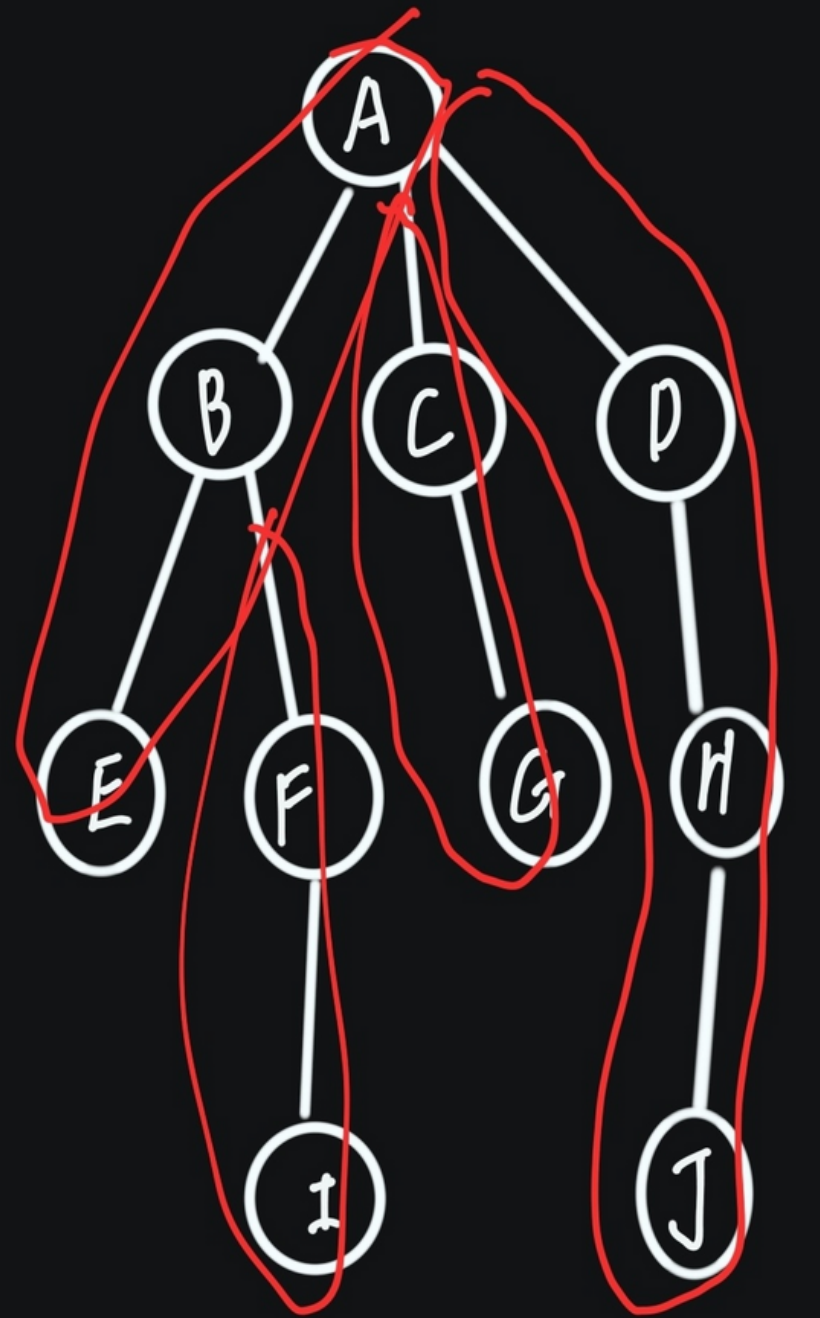


트리



DFS(Depth First Search - 깊이 우선 탐색)

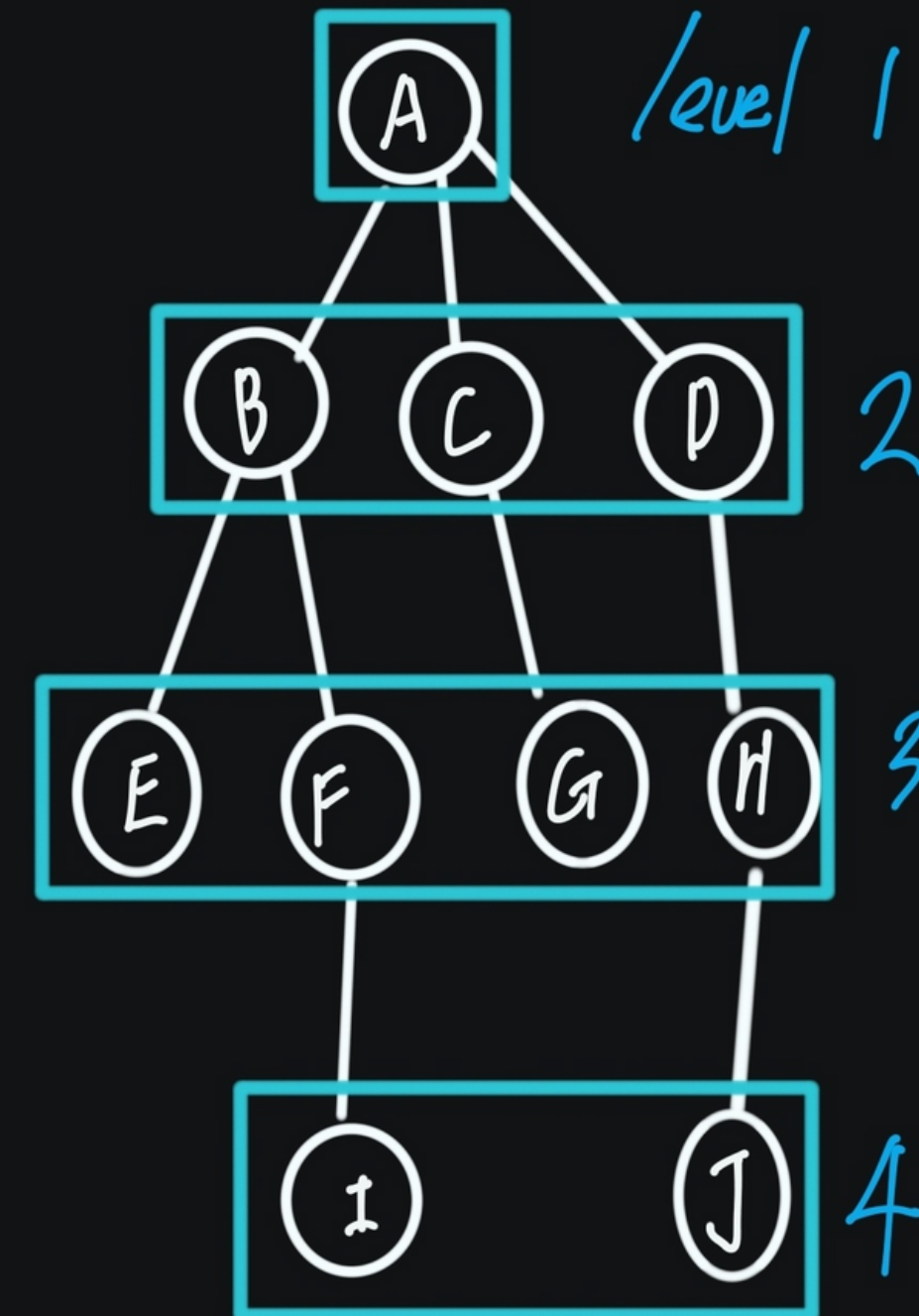
최대한 깊이 내려간 뒤, 더이상 갈 곳이 없을 경우 옆으로 이동



A B C F I C G D H J

BFS(Breath First Search - 너비 우선 탐색)

최대한 넓게 이동한 다음, 더 이상 갈 수 없을 때 아래로 이동



A B C D E F G H I J

각 노드에 대해서,

1) ADJACENT

2) VISITED

Stack

LIFO

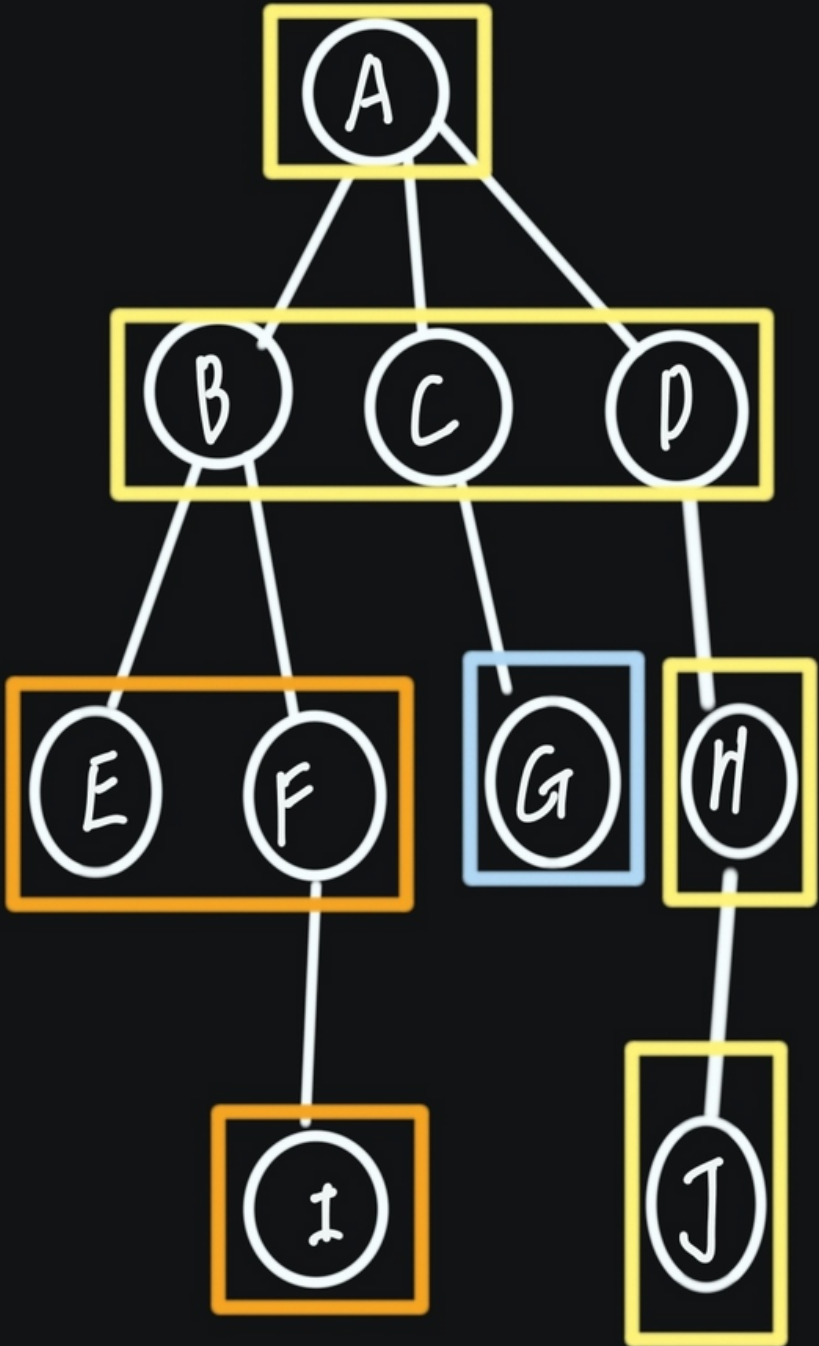


Queue

FIFO



DFS -> 스택



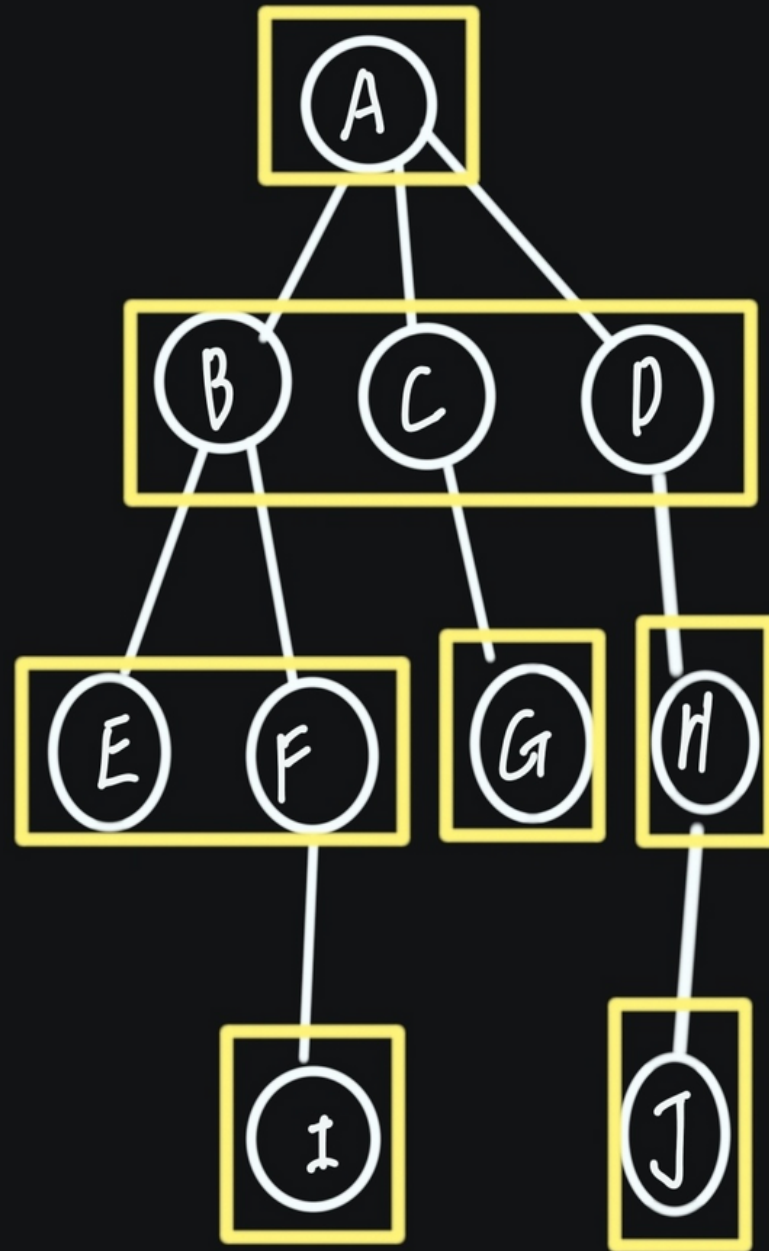
A D H J

C G

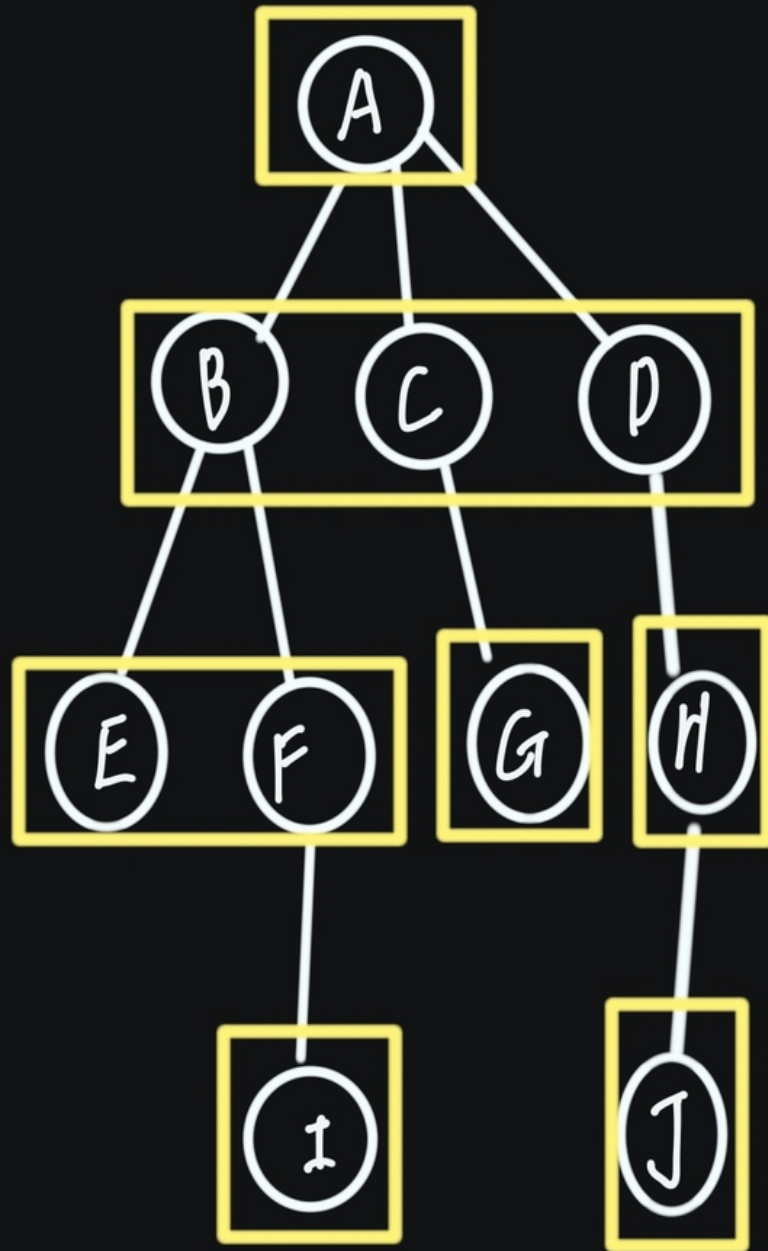
B F I

E

DFS -> 스택 -> 재귀(함수의 스택)



BFS -> 큐



A
BCD
EF
G
H
I
J

활용

1. 네트워크 유형(연결) -> BFS/DFS 둘다 가능
2. 경로탐색 유형 (최단거리, 시간) -> BFS 유리
visited를 level로 저장
3. 경로의 특징을 저장해야 하는 문제 -> DFS
4. 조합문제??

쉬운 문제는 DFS / 어려운 문제는 BFS?
(시간초과 방지)