

그림

(V, E)

트리 1

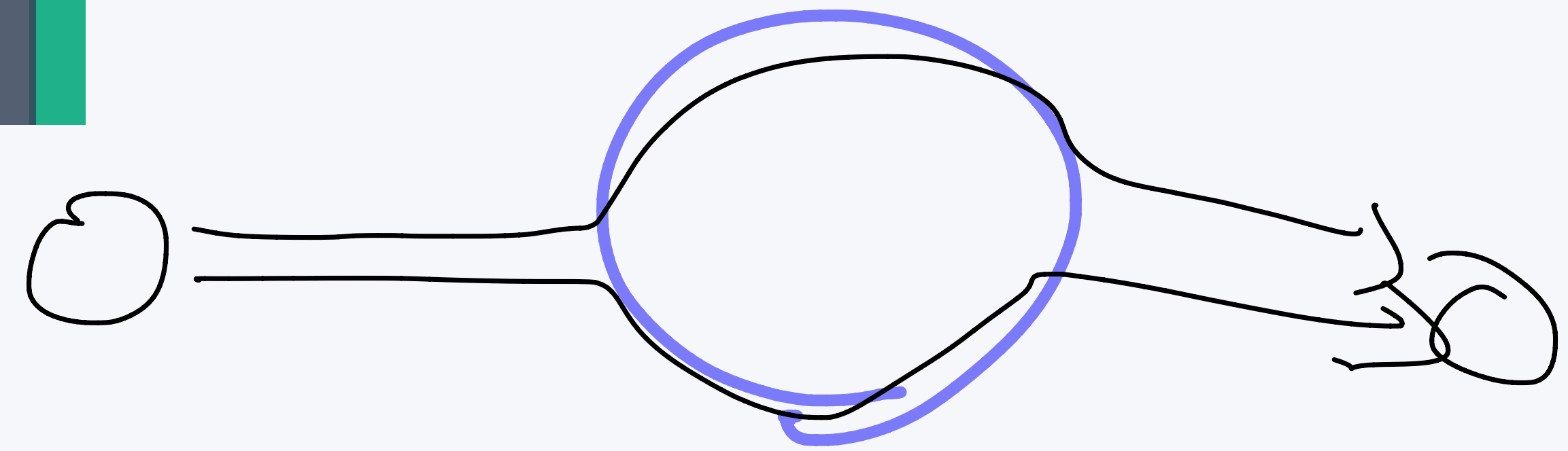
최백준 choi@startlink.io



$$\Sigma \hookleftarrow N \quad \Sigma \xrightarrow{M} N \downarrow$$

$\frac{\Sigma}{1}$ 2+2=4 u, v 2+3=5 124

트리

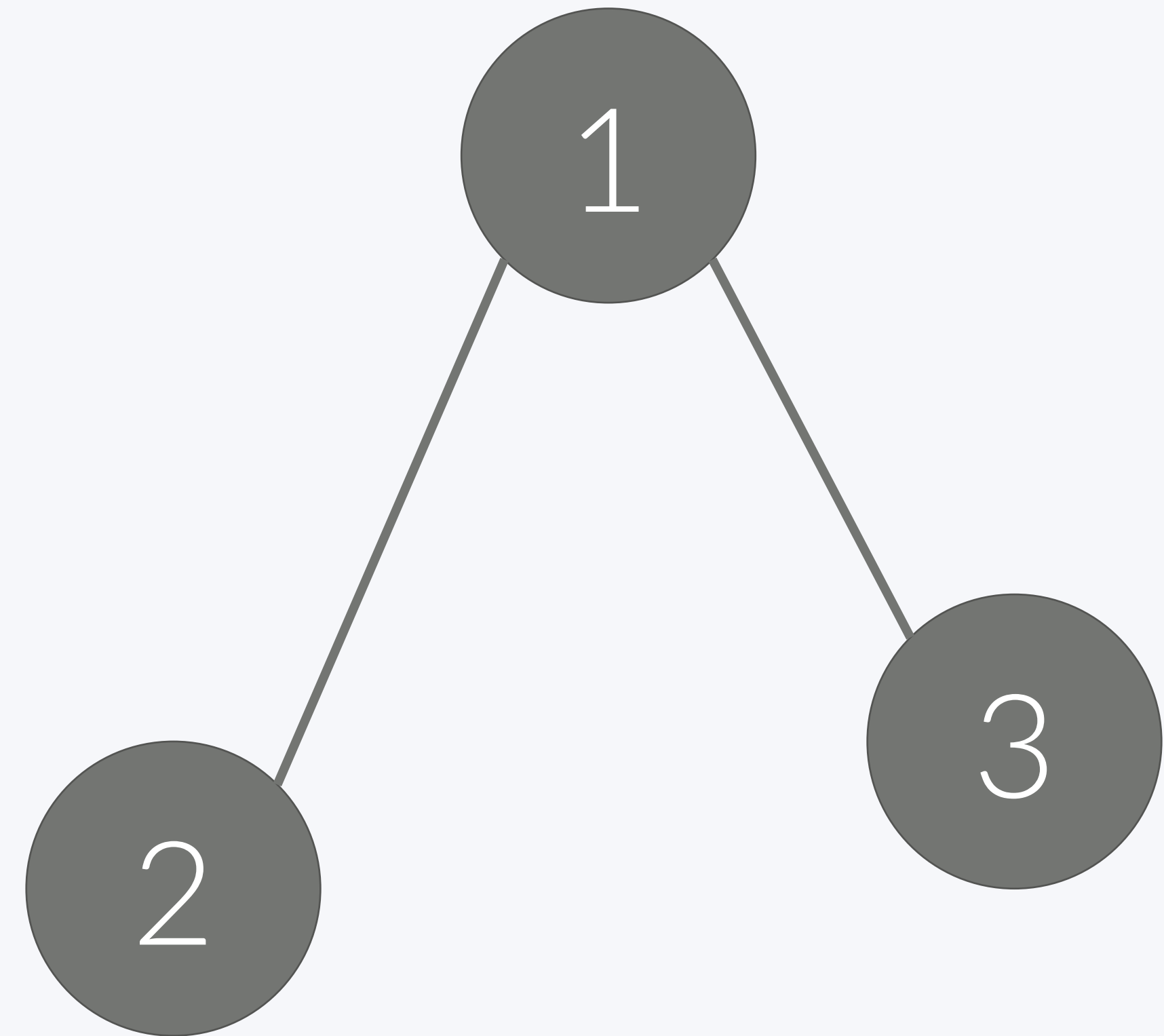


트리

Tree

- 자료구조의 일종
- 사이클이 없는 그래프
- 정점의 개수: V
- 간선의 개수: $V-1$

사이클이 없는 연결 그래프
방향 \times

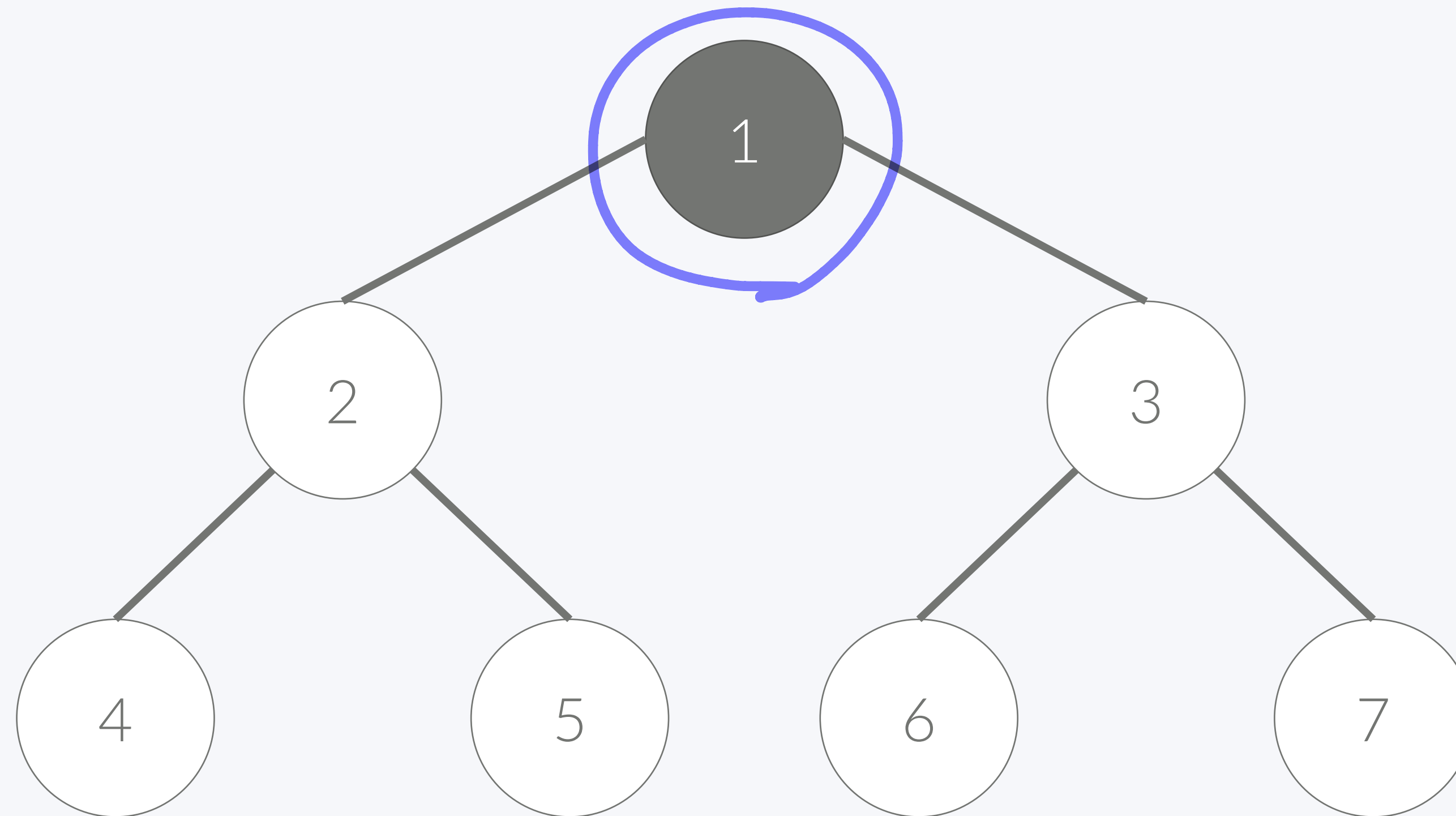


루트 있는 트리

Rooted Tree

- 루트가 있는 트리

- 1번이 루트이다

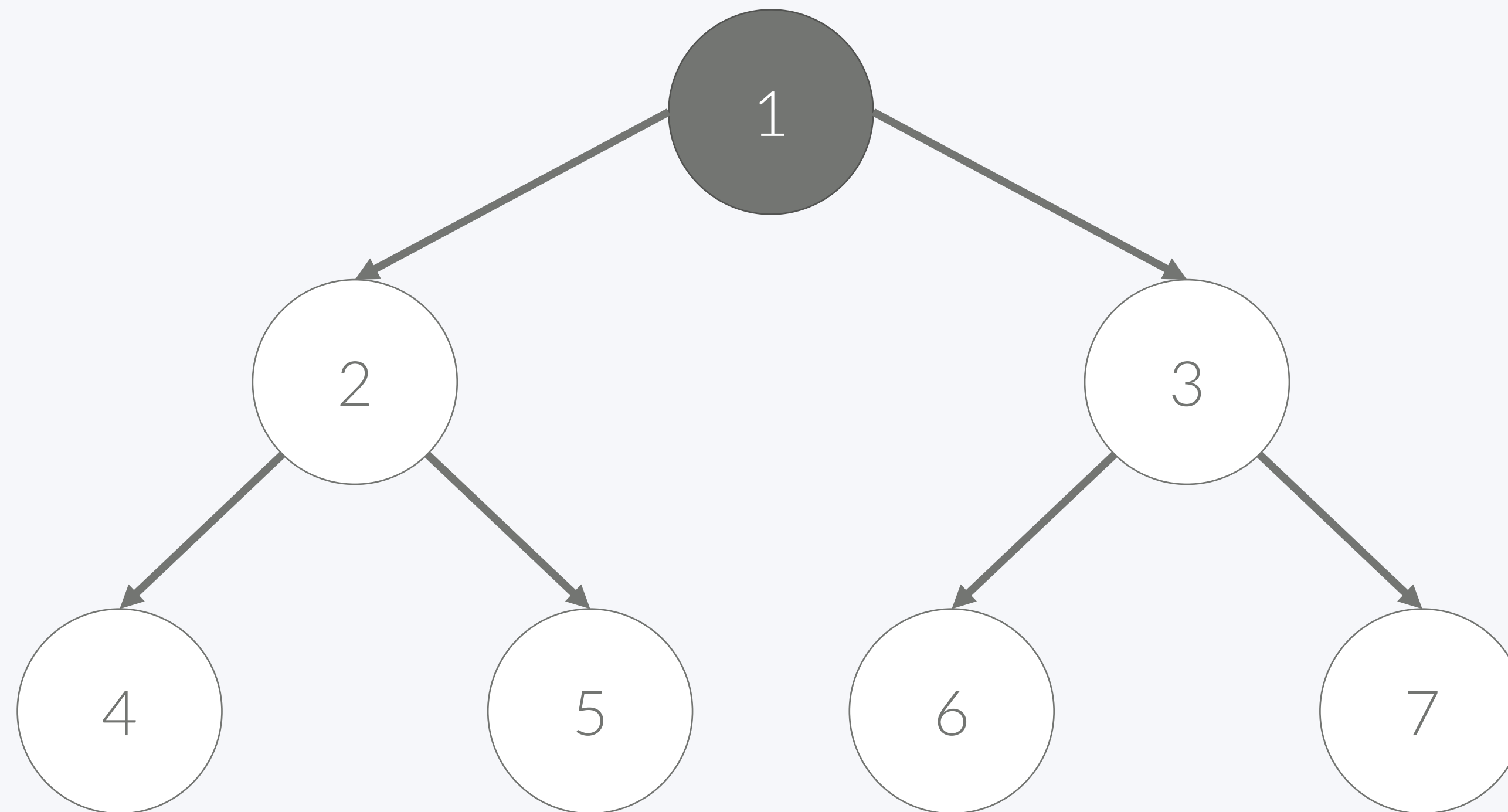


루트 있는 트리

Rooted Tree

5

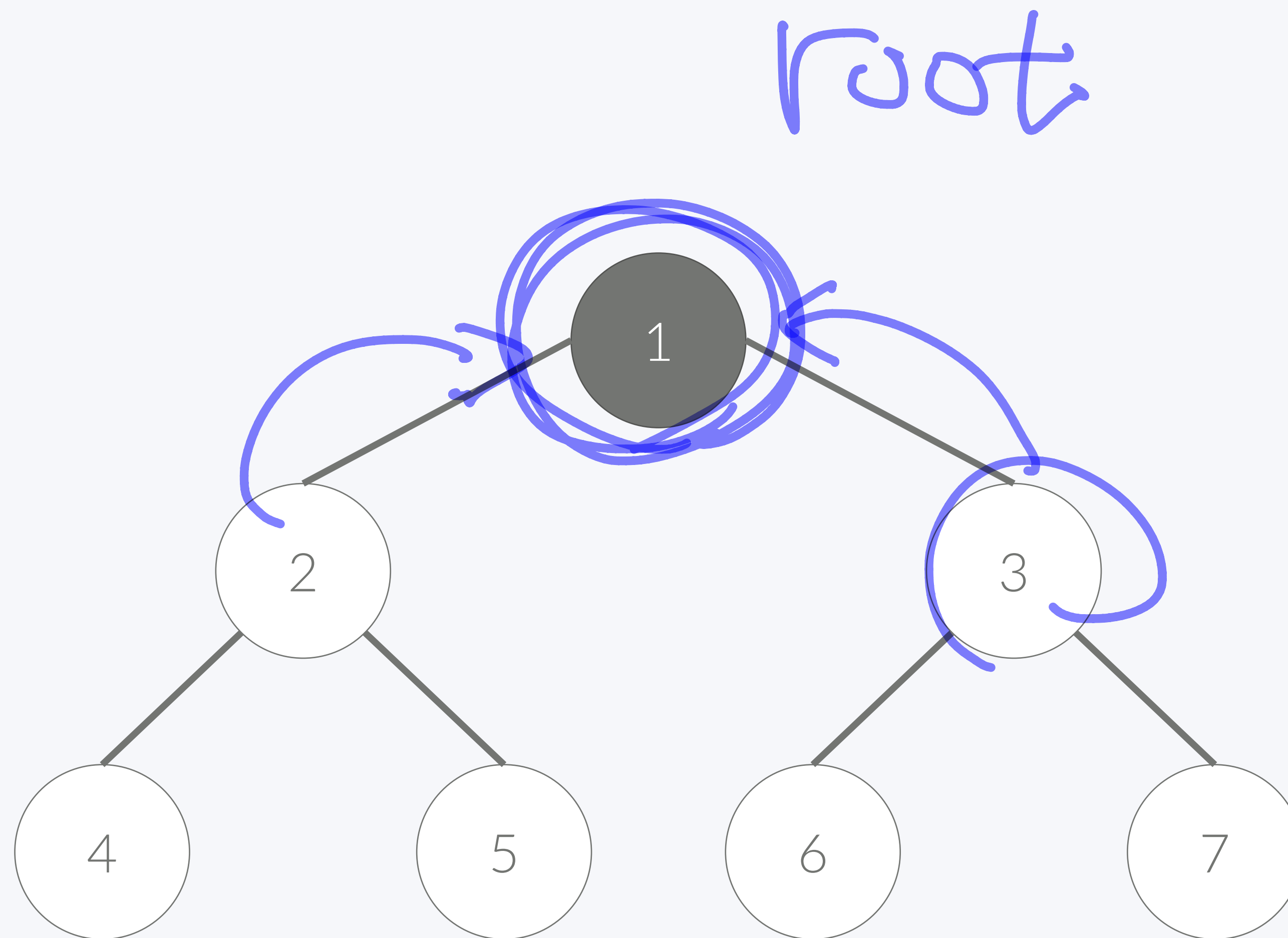
- 루트 부터 아래로 방향을 정할 수 있다



부모

Parent

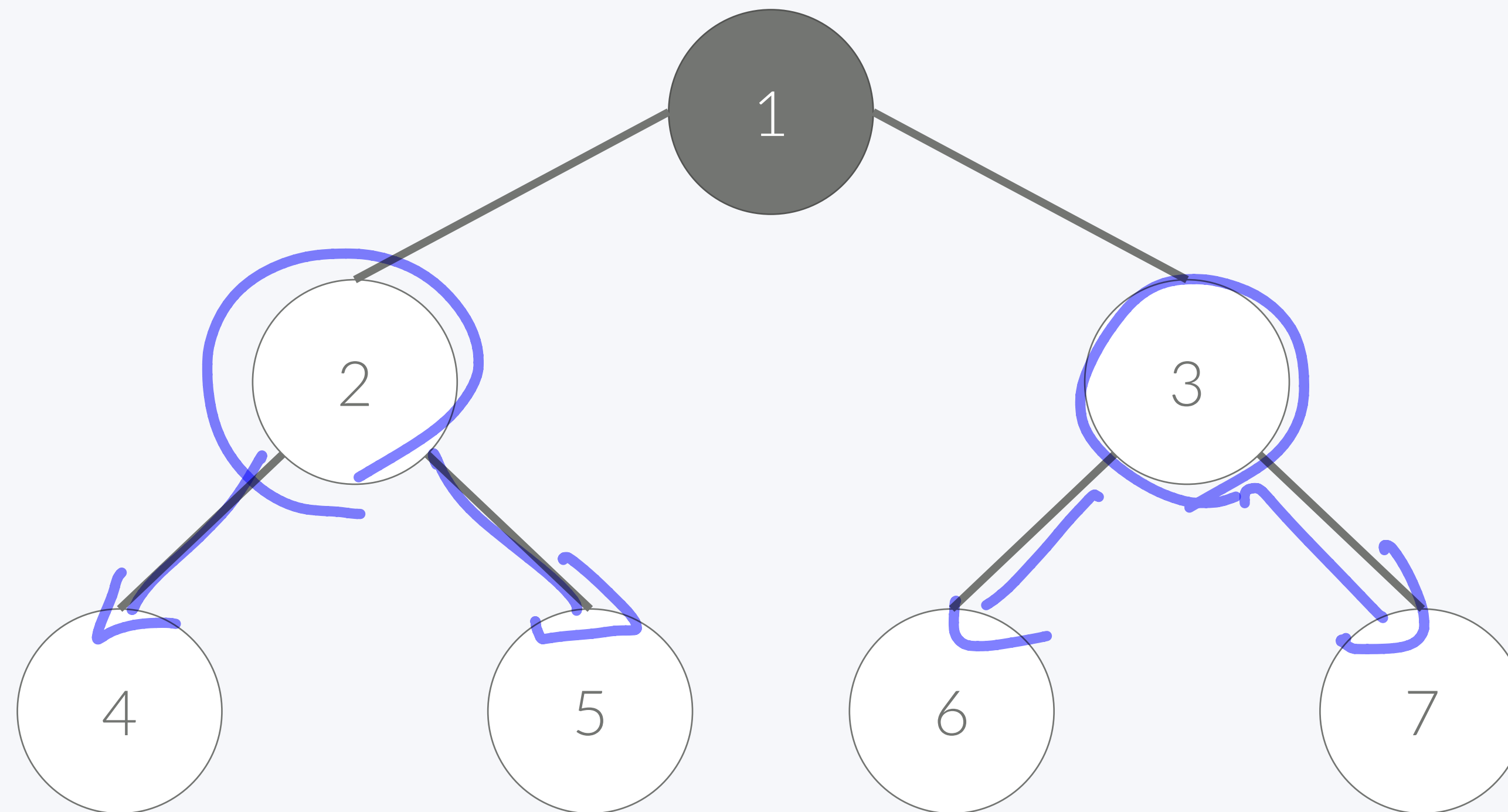
- 1은 2의 부모
- 2는 4의 부모



자식

Children

- 2는 1의 자식
- 4는 2의 자식
- 3의 자식: 6, 7

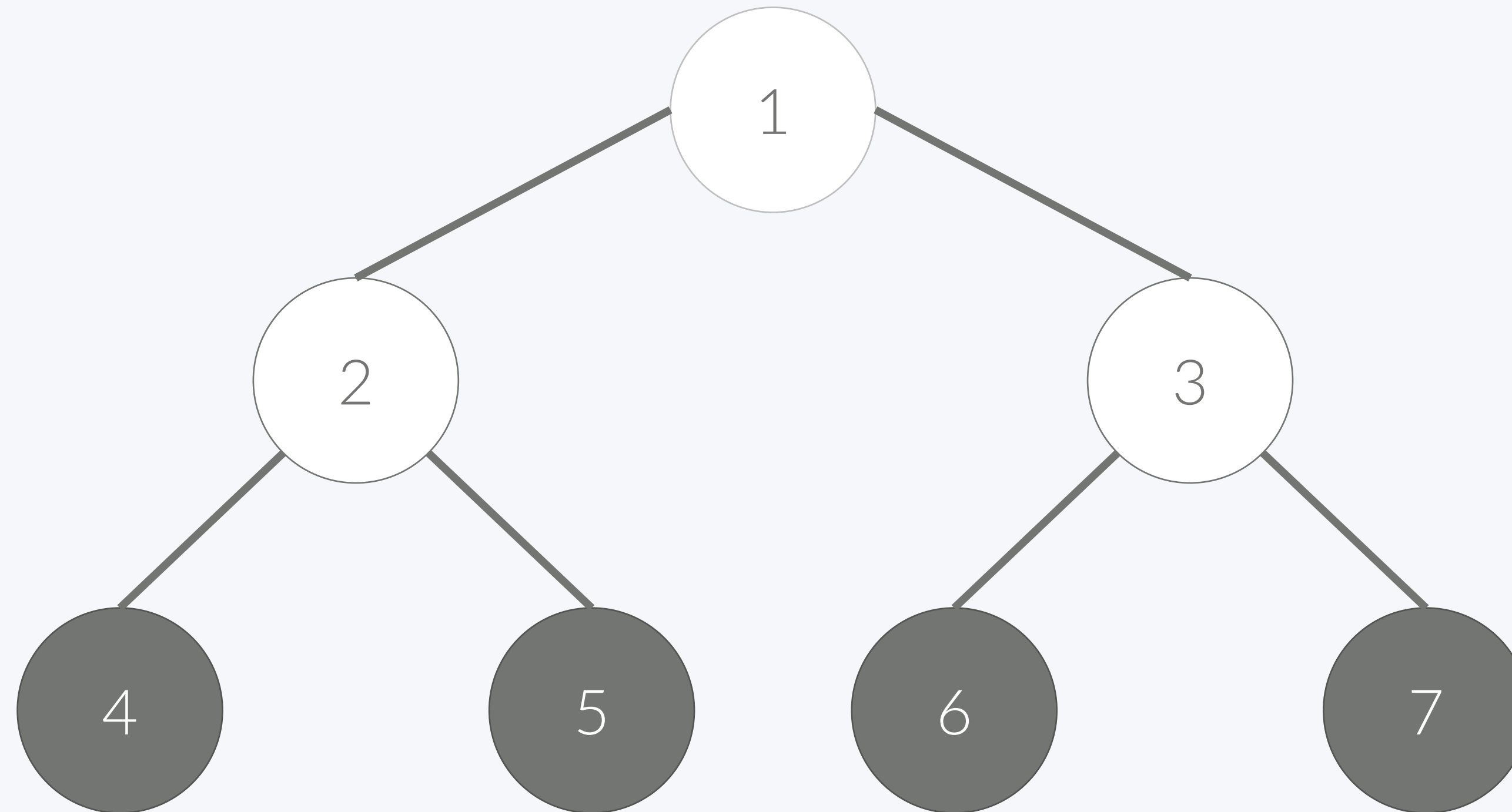


단말 정점

Leaf Node

- 4, 5, 6, 7

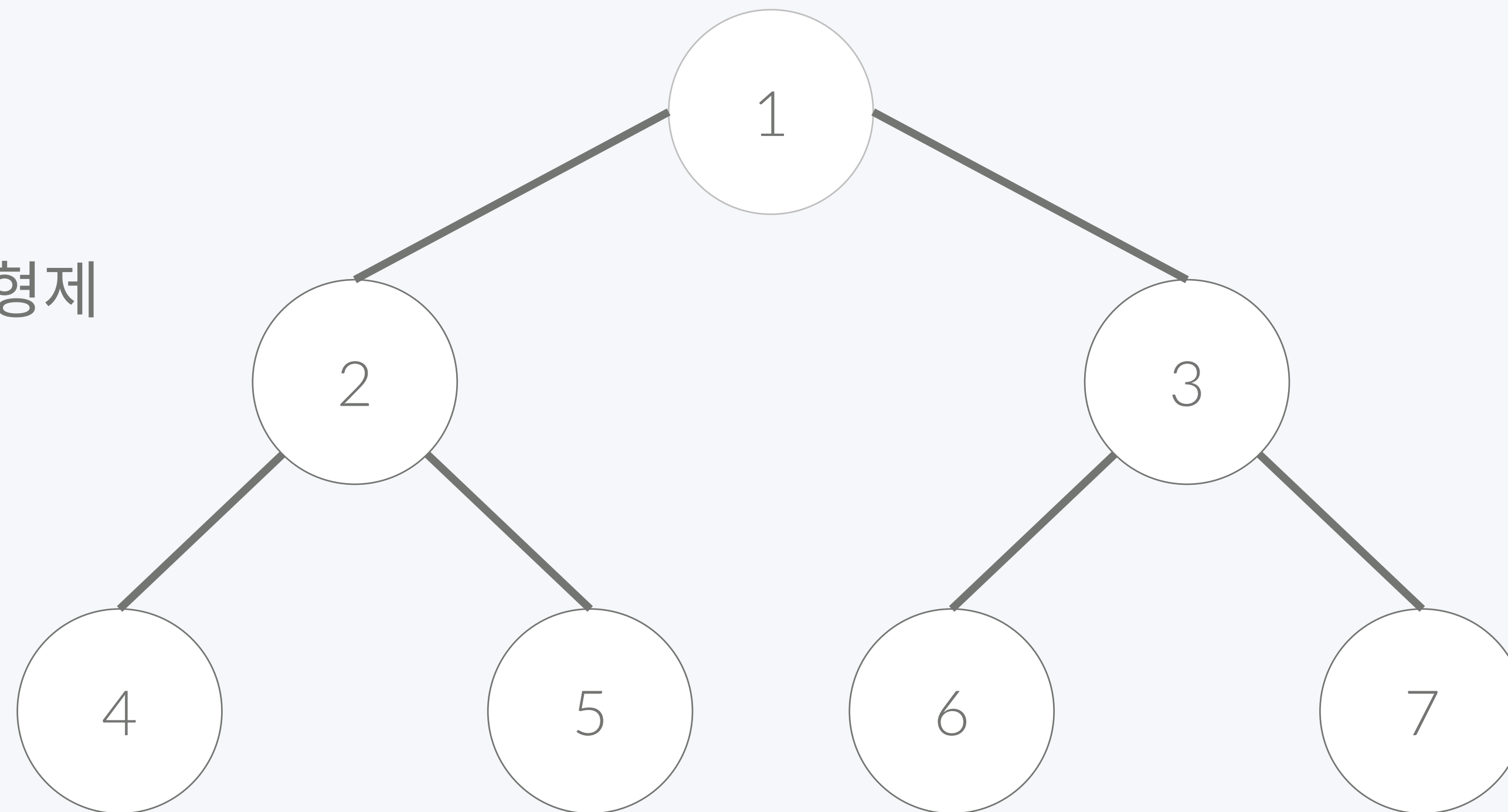
terminal node



형제

Sibling

- 4와 5는 형제
- 6과 7은 형제
- 2와 3도 형제
- 같은 부모를 가지면 형제

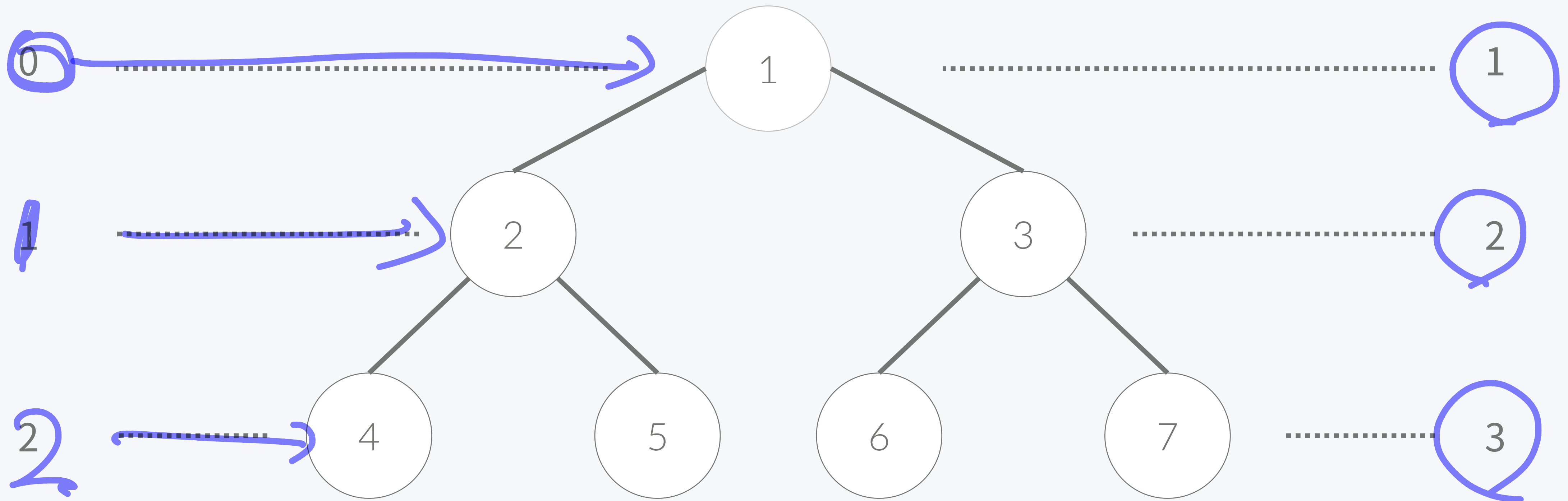


깊이

Depth

10

- 루트에서 부터 거리 (루트의 깊이를 0으로 하는 경우와 1로 하는 경우가 있다)

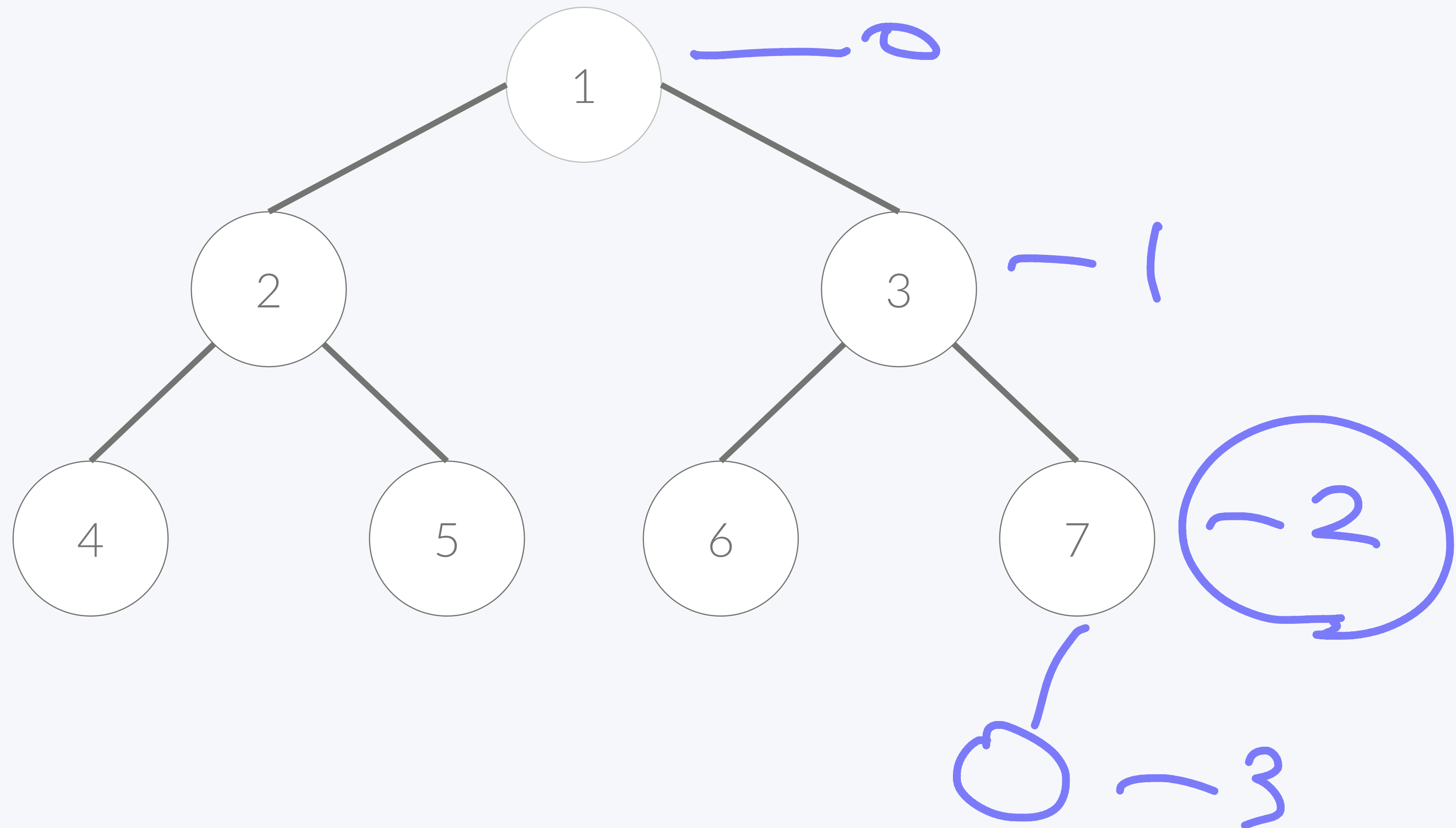


높이

Height

11

- 깊이 중 가장 큰 값 2 또는 3

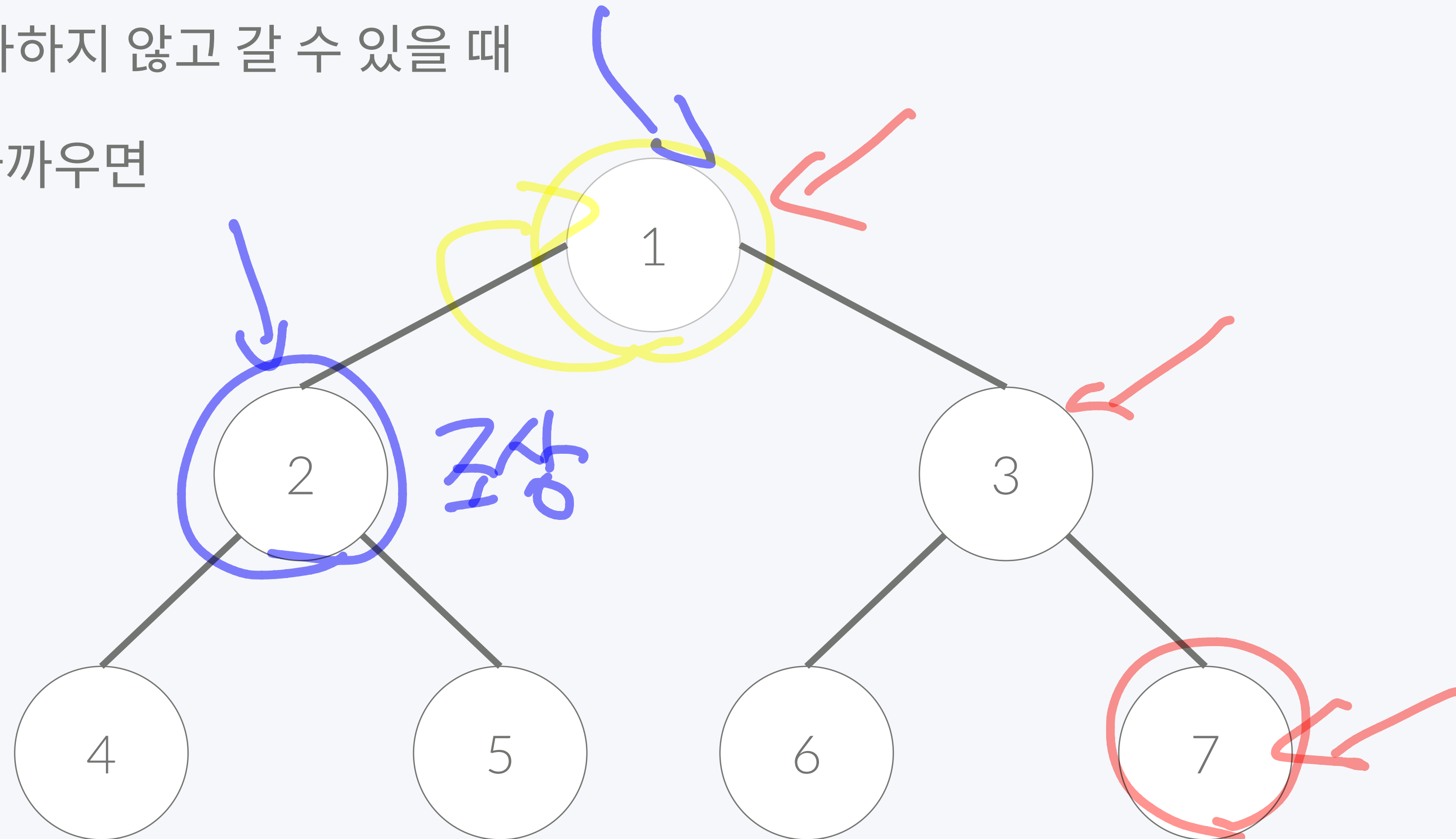


조상, 자손

Ancestor, Descendent

12

- $p \rightarrow q$ 로 루트를 통과하지 않고 갈 수 있을 때
- p 가 q 보다 루트에 가까우면
- p 는 q 의 조상
- q 는 p 의 자손

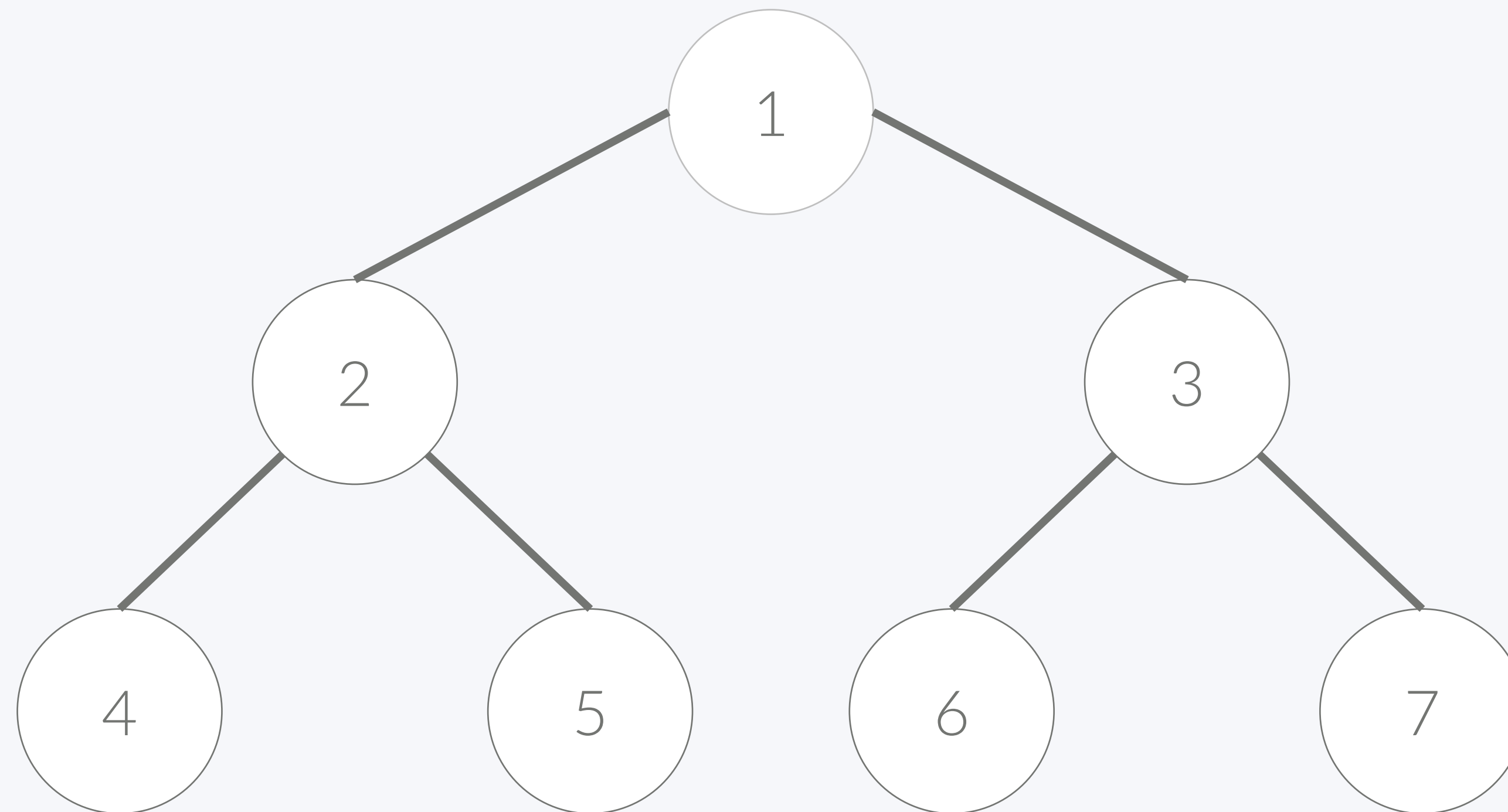


이진 트리

Binary Tree

13

- 자식을 최대 2개만 가지고 있는 트리

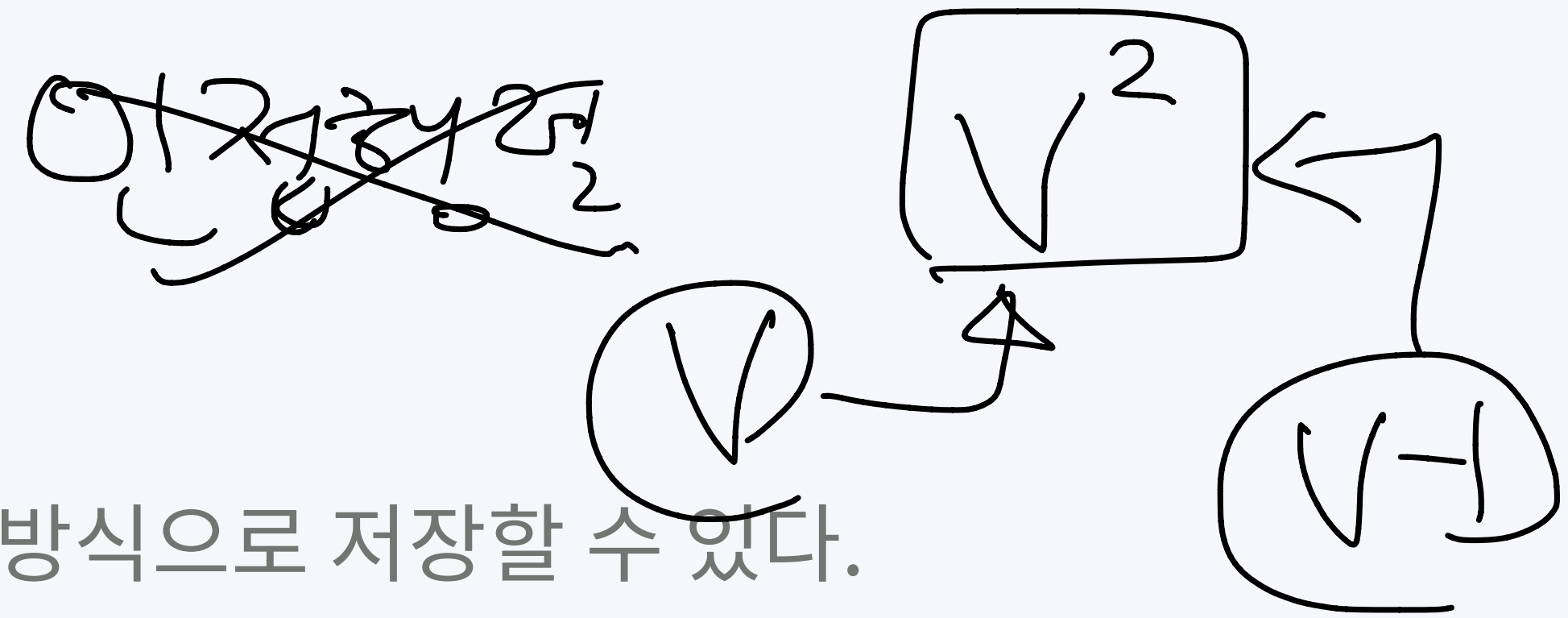


트리의 표현

트리의 표현

Representation of Tree

이진트리



15

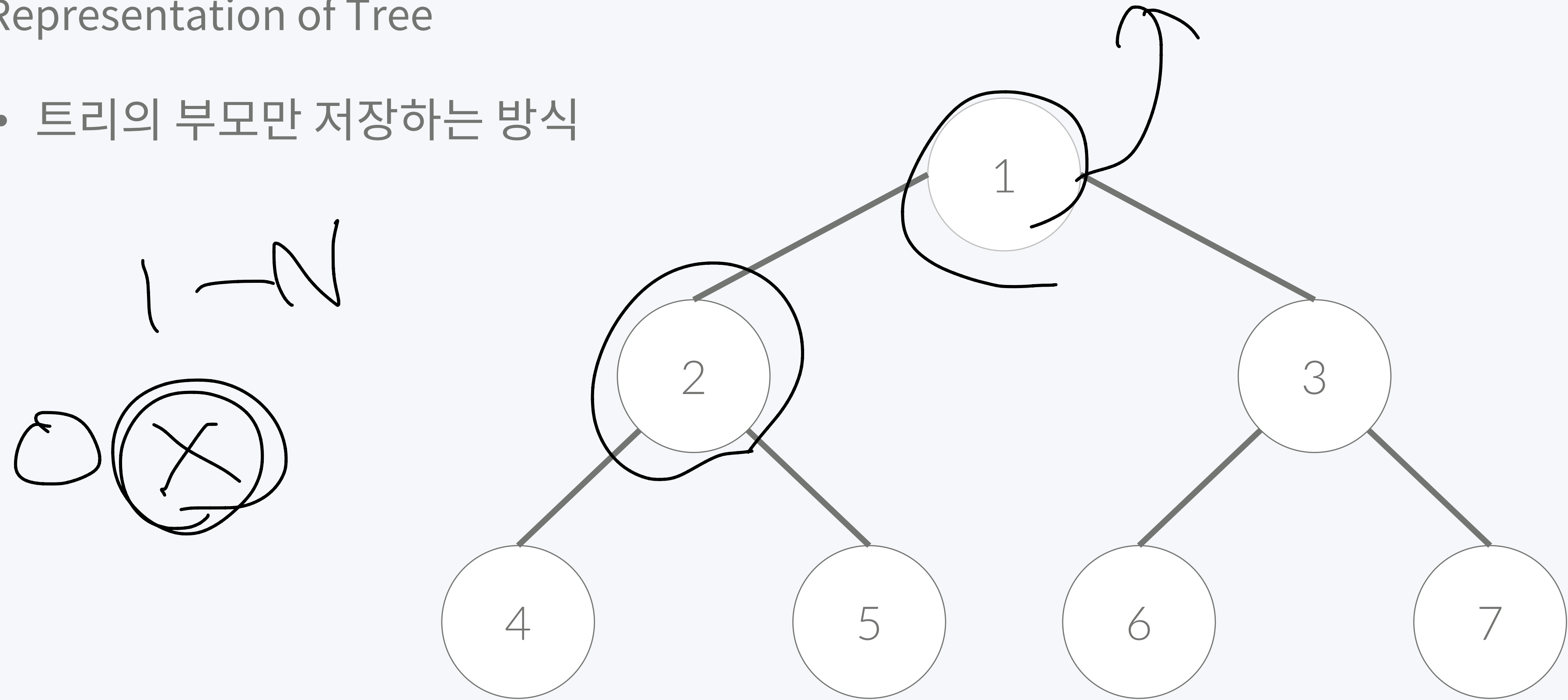
- 트리는 그래프이기 때문에, 그래프의 표현과 같은 방식으로 저장할 수 있다.
- 또는
- 트리의 모든 노드는 부모를 하나 또는 0개만 가지기 때문에 부모만 저장하는 방식으로 저장할 수 있다
- 부모가 0개인 경우는 트리의 루트인데, 이 경우 부모를 -1이나 0으로 처리하는 방식을 사용한다

트리의 표현

Representation of Tree

- 트리의 부모만 저장하는 방식

Union Find

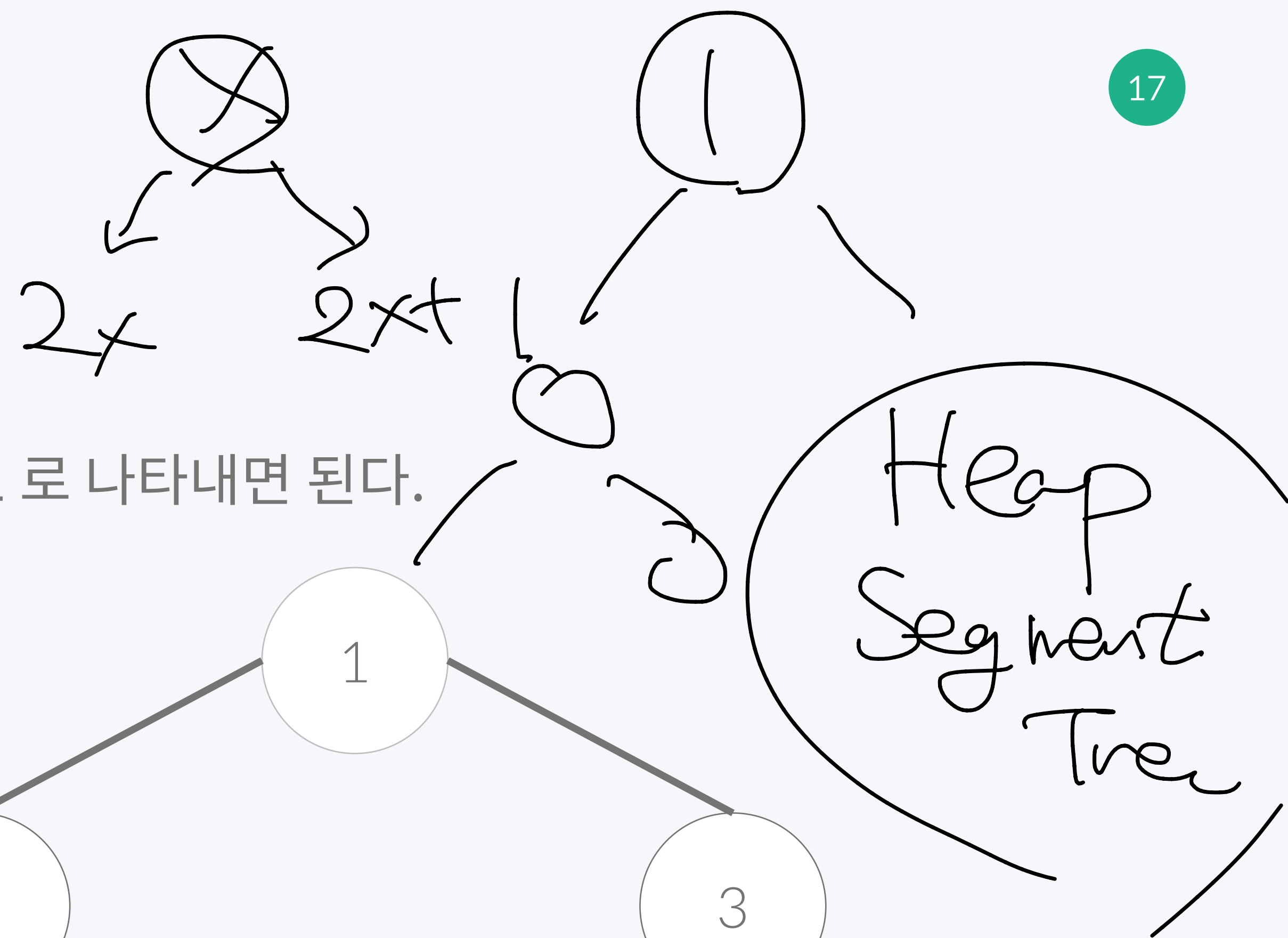
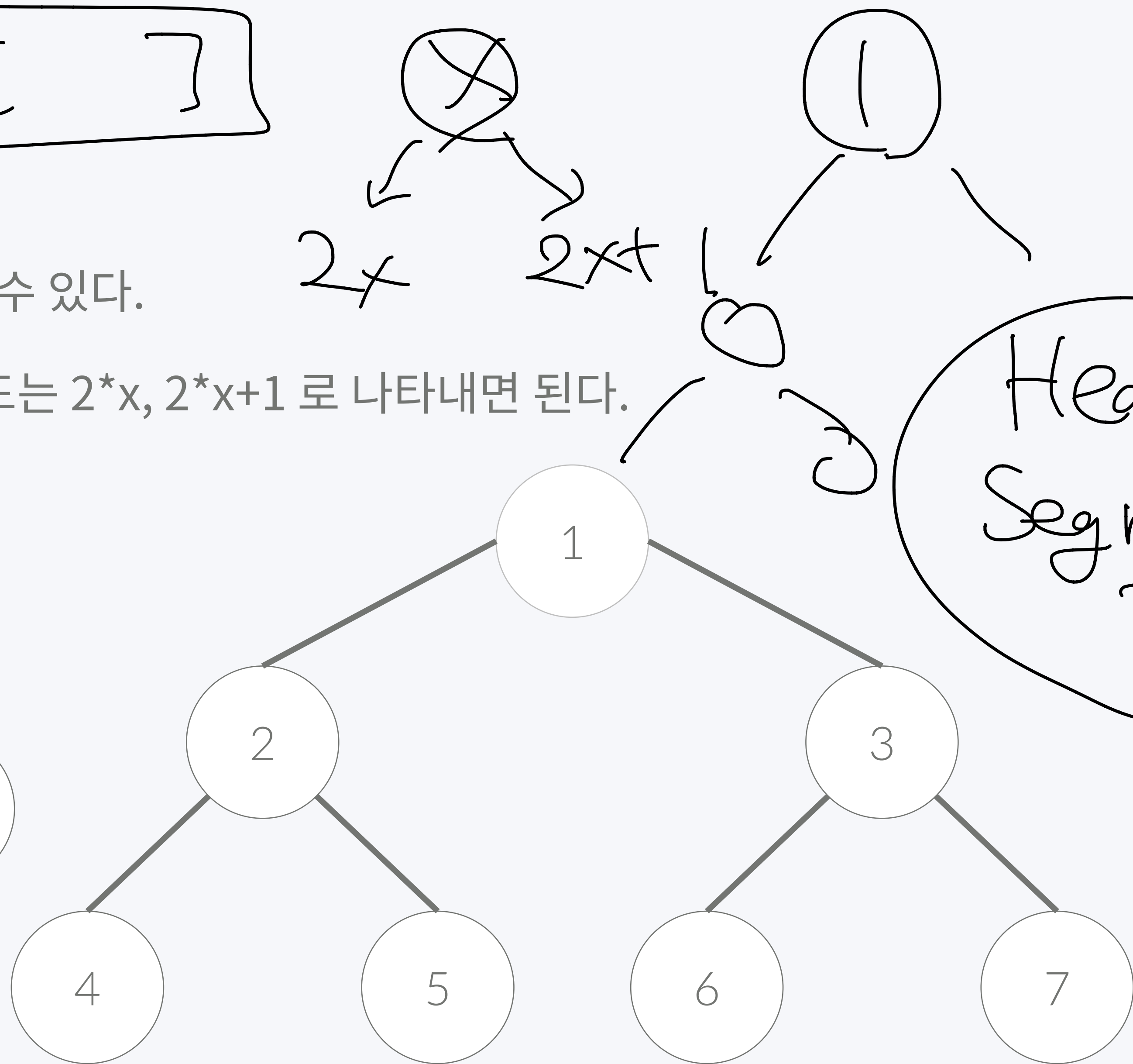
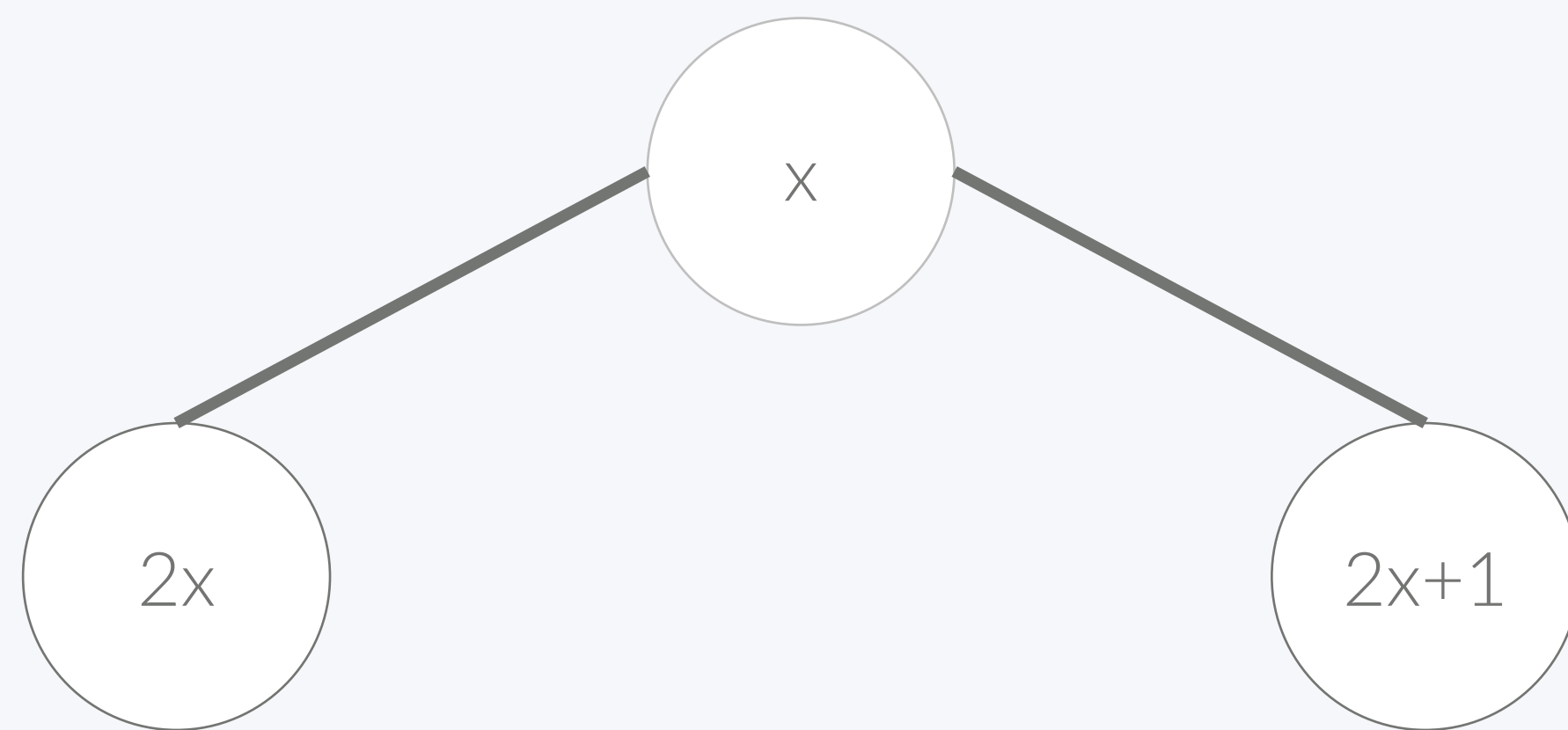


i	1	2	3	4	5	6	7
parent[i]	0	1	1	2	2	3	3

트리의 표현

Representation of Tree

- 이진 트리의 경우에는 배열로 표현할 수 있다.
- 부모의 노드가 x 인 경우에 자식의 노드는 $2x$, $2x+1$ 로 나타내면 된다.



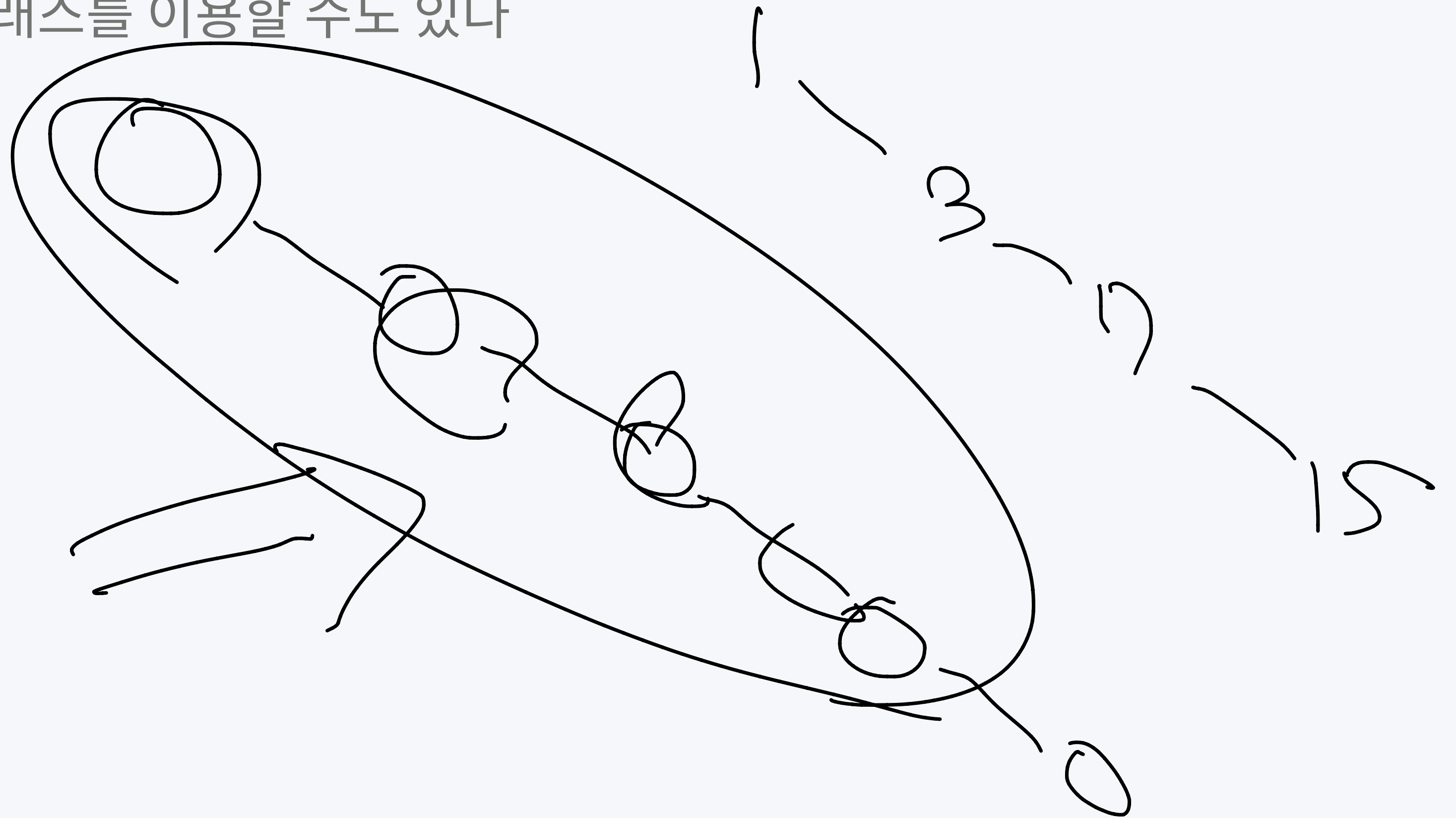
트리의 표현

Representation of Tree

- 이진 트리의 경우에는 구조체나 클래스를 이용할 수도 있다

```
struct Node {  
    Node *left;  
    Node *right;  
}
```

BST



트리의 순회

DFS, BFS

- > pre order
- => in order
- post order

지
좌
우

트리의 순회

Tree Traversal

- 트리의 모든 노드를 방문하는 순서이다.
- 그래프의 경우에는 DFS와 BFS가 있었다
- 트리에서도 위의 두 방법을 사용할 수 있다.
- DFS는 아래와 같이 3가지 출력 순서가 있다
- 프리오더
- 인오더
- 포스트오더
- 세 방법의 차이는 노드 방문 처리를 언제 할 것인가이다.

이진트리

프리: 노드 (L) (R)

인: (L) 노드 (R)

포스트: (L) (R) 노드

노드 ① ^ ② ^ ③ 노드

트리의 순회

Tree Traversal

- 프리오더
 - 노드 방문
 - 왼쪽 자식 노드를 루트로 하는 서브 트리 프리오더
 - 오른쪽 자식 노드를 루트로 하는 서브 트리 프리오더
- 인오더
- 포스트오더

트리의 순회

Tree Traversal

22

- 프리오더
- 인오더
 - 왼쪽 자식 노드를 루트로 하는 서브 트리 인오더
 - 노드 방문
 - 오른쪽 자식 노드를 루트로 하는 서브 트리 인오더
- 포스트오더

트리의 순회

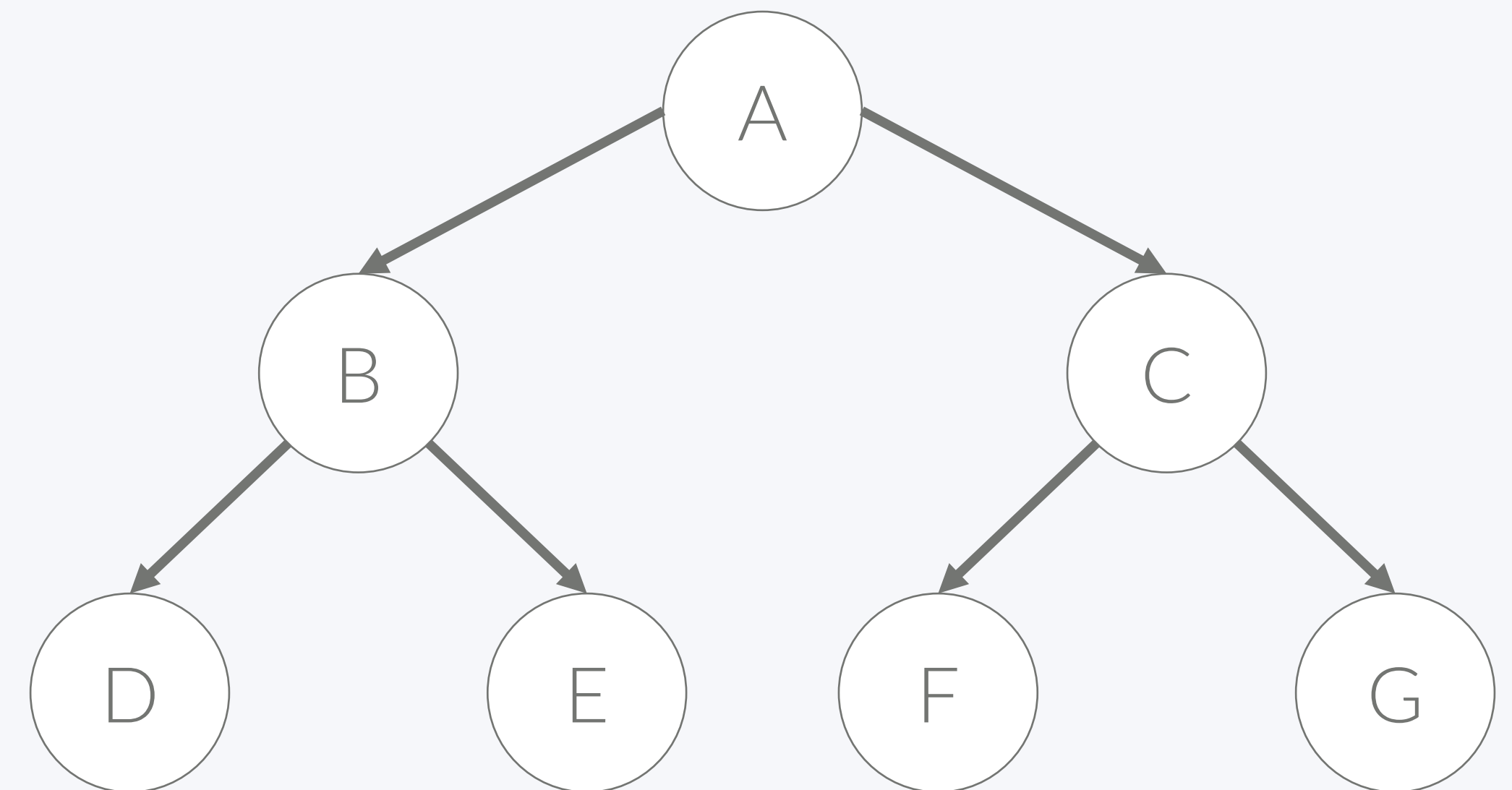
Tree Traversal

- 프리오더
- 인오더
- 포스트오더
 - 왼쪽 자식 노드를 루트로 하는 서브 트리 포스트오더
 - 오른쪽 자식 노드를 루트로 하는 서브 트리 포스트오더
 - 노드 방문

프리오더

Pre-order

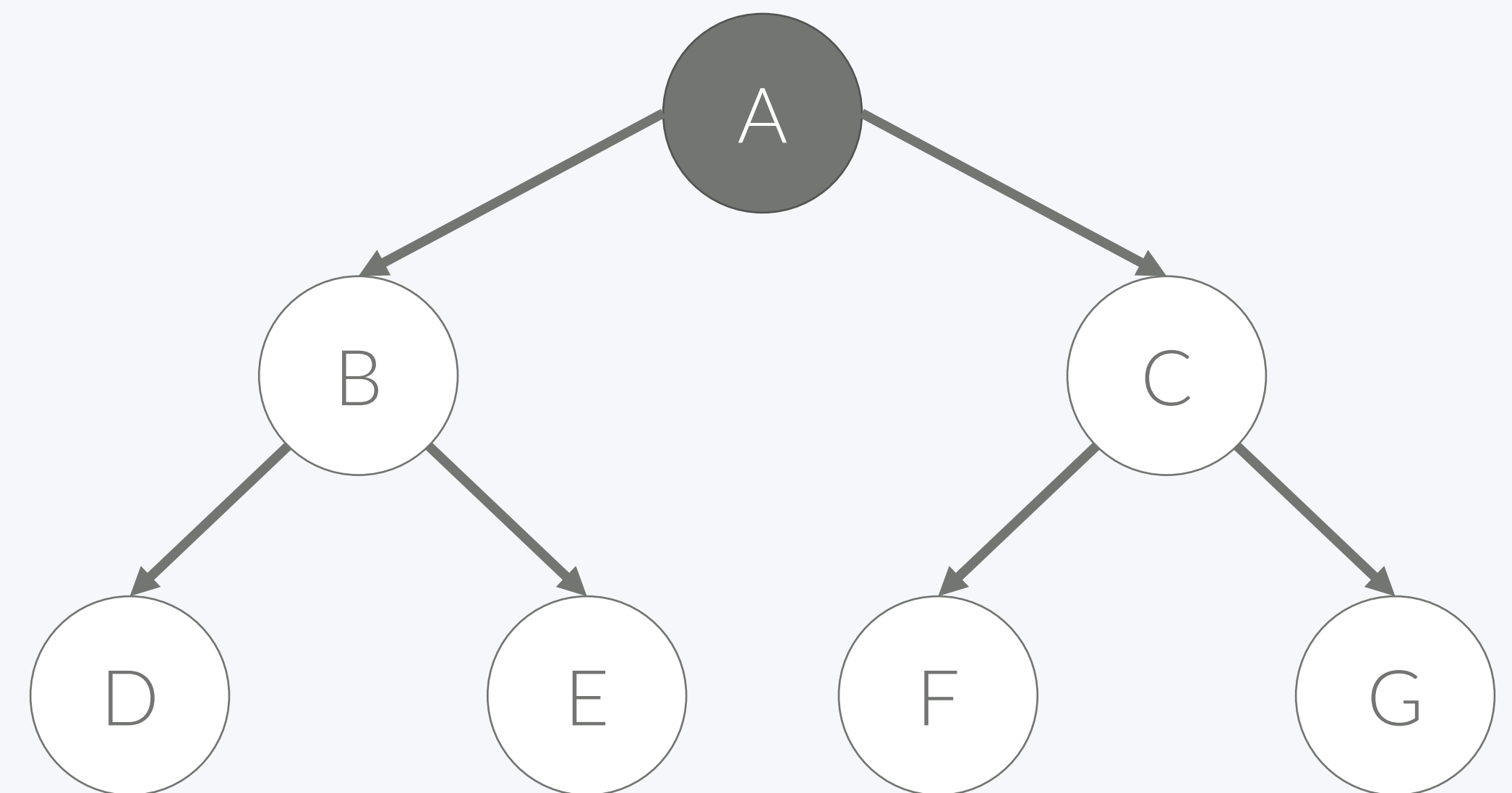
- 노드 방문
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

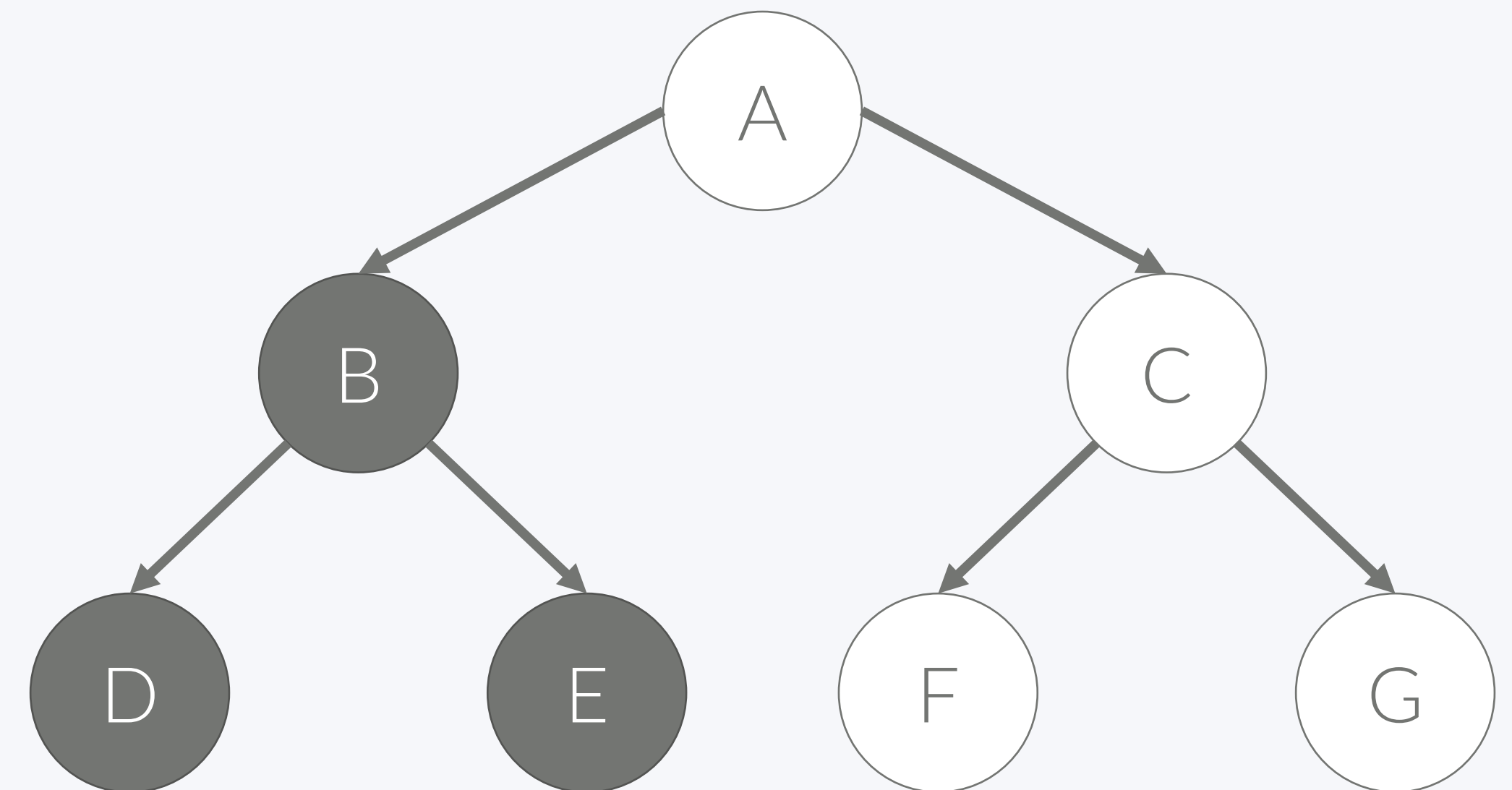
- 노드 방문
 - A
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더

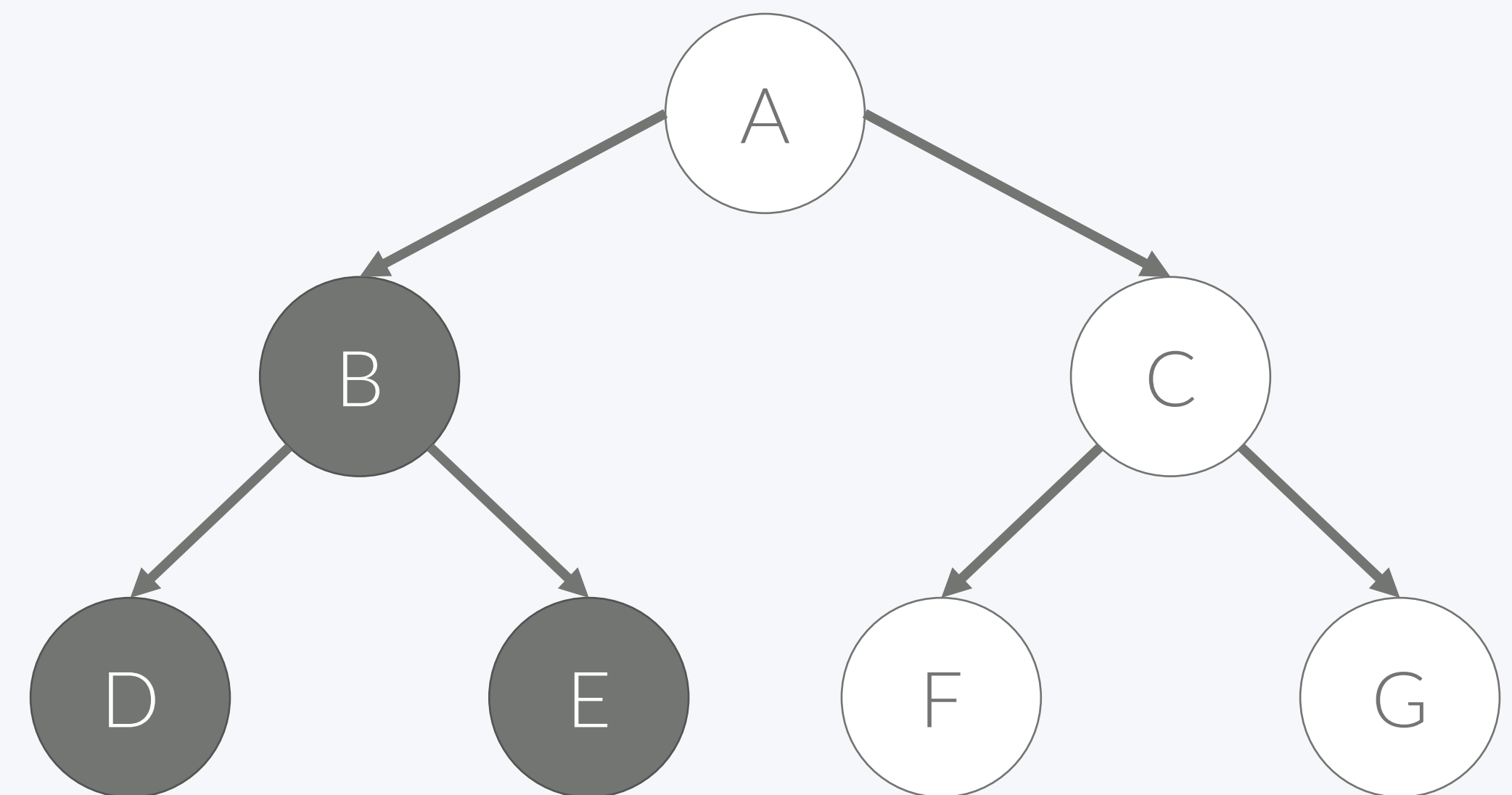


프리오더

27

Pre-order

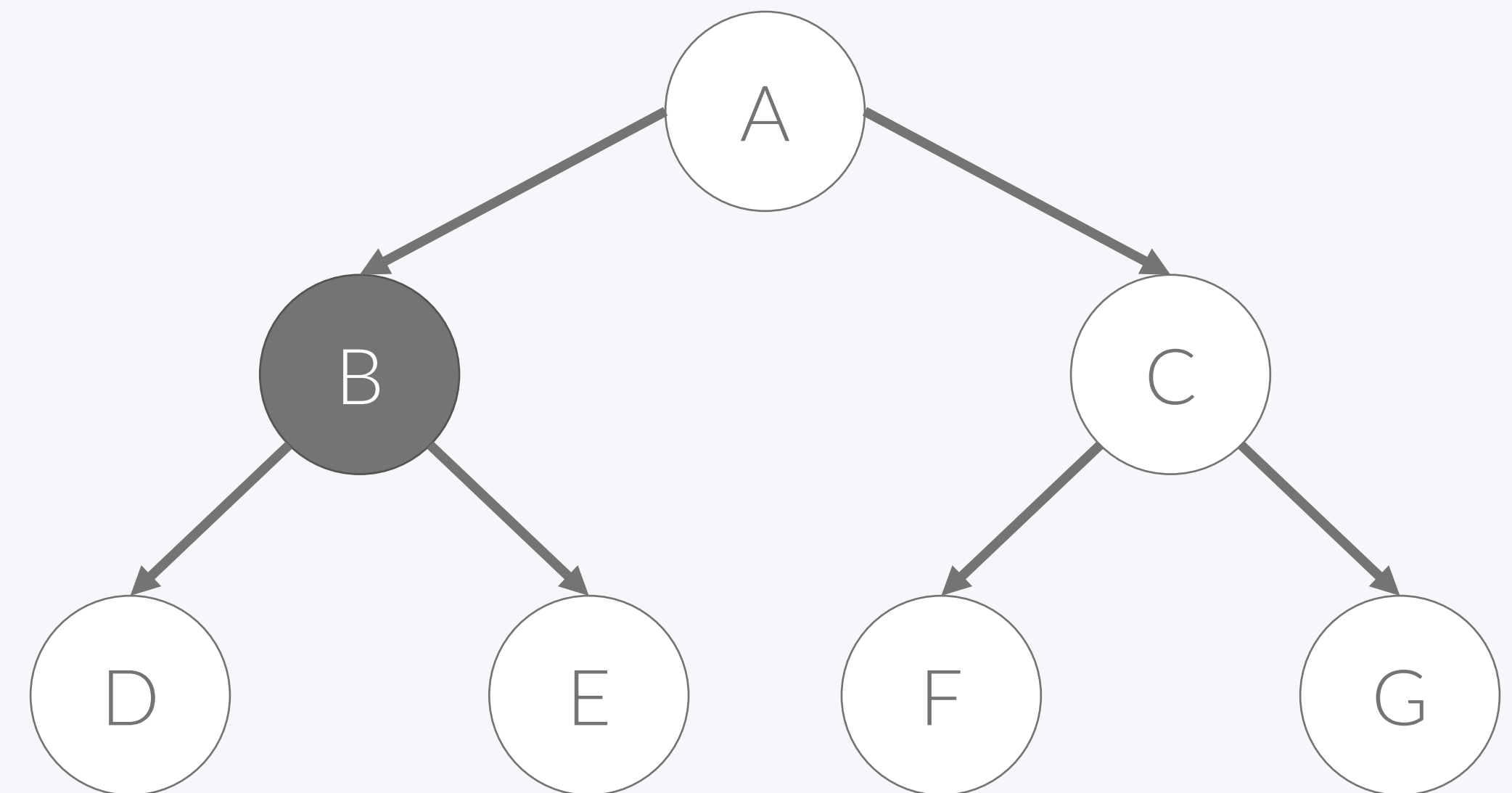
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

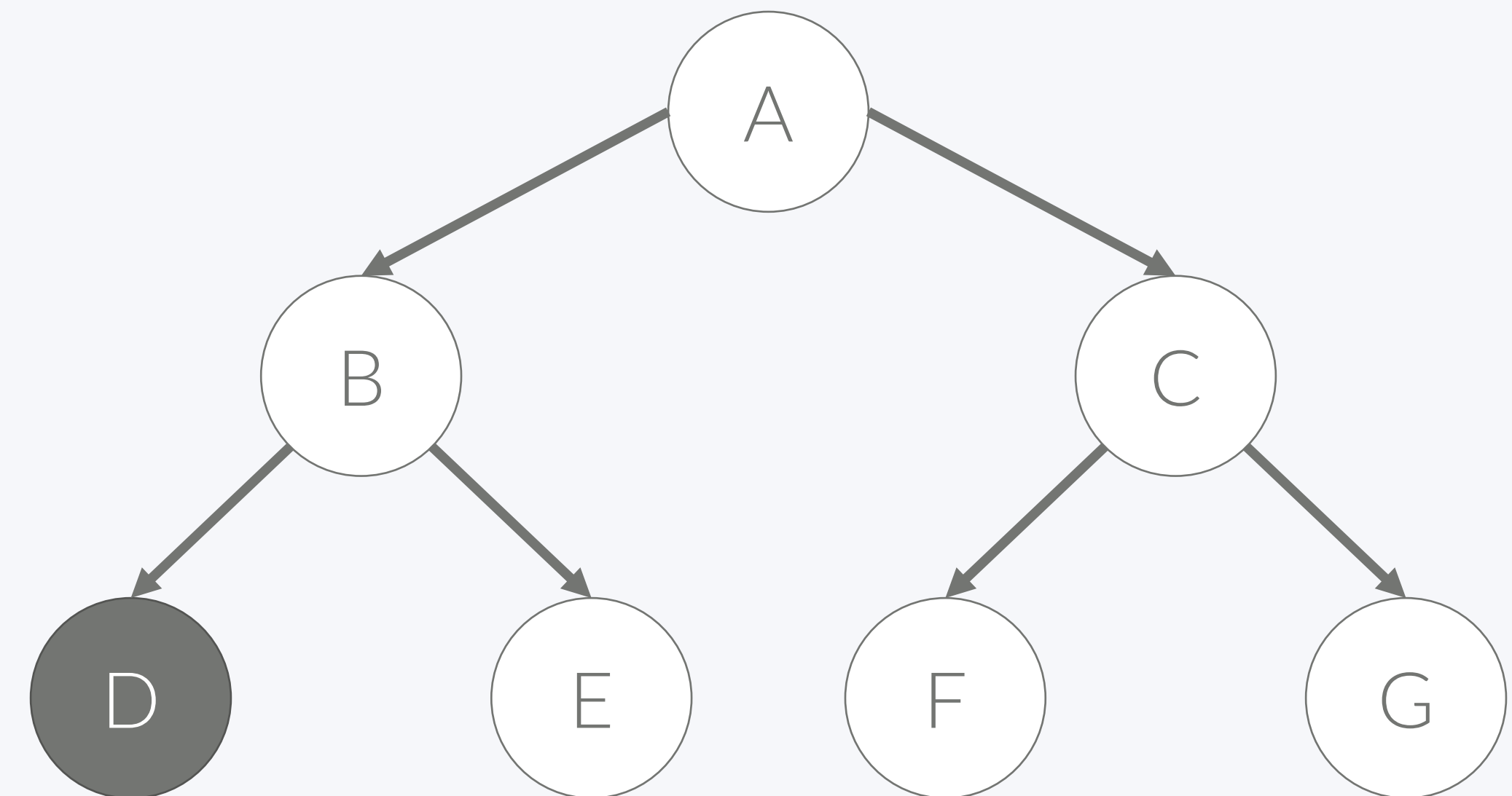
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

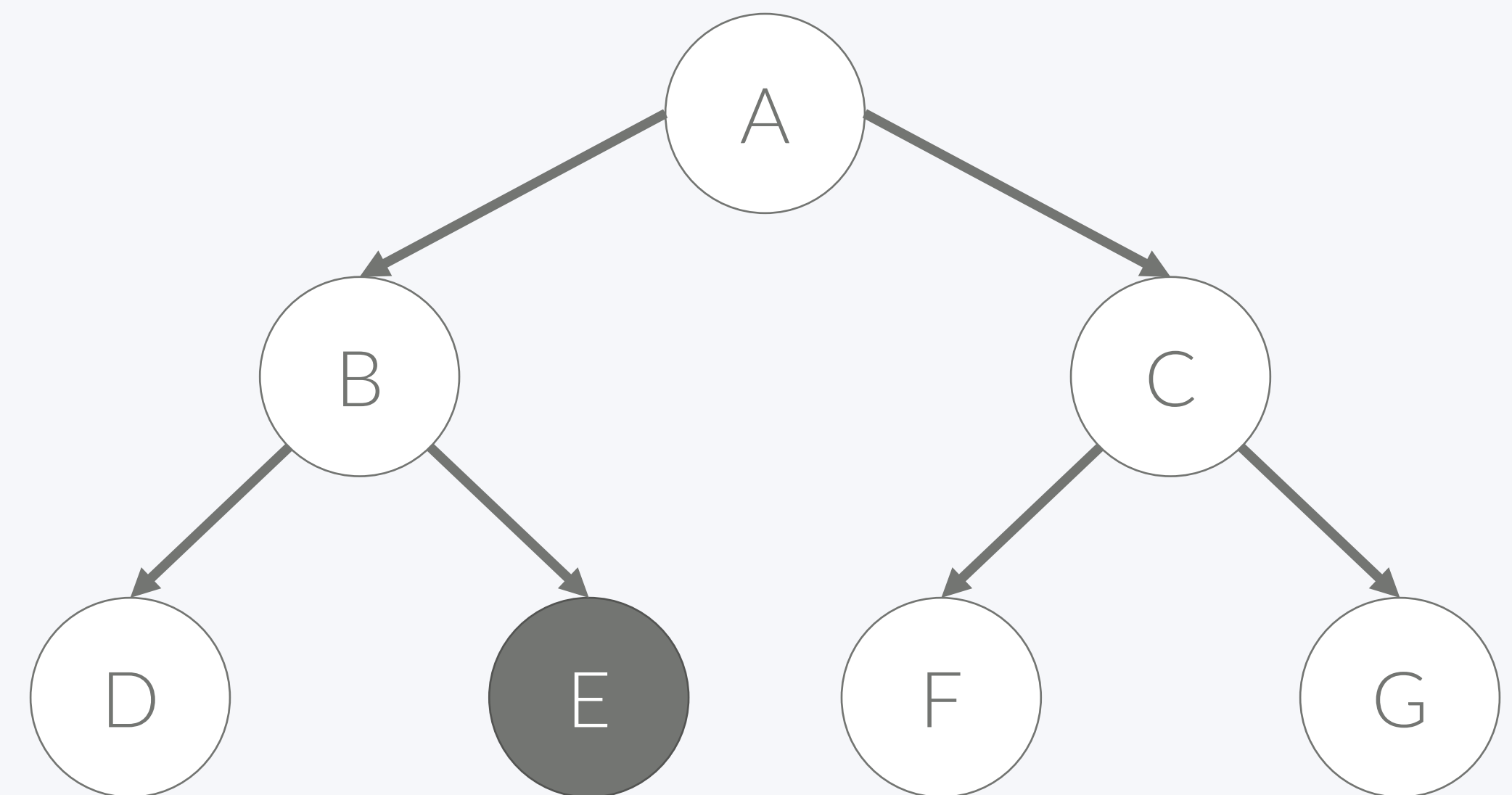
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
 - 오른쪽 자식 프리오더
 - E
- 오른쪽 자식 프리오더

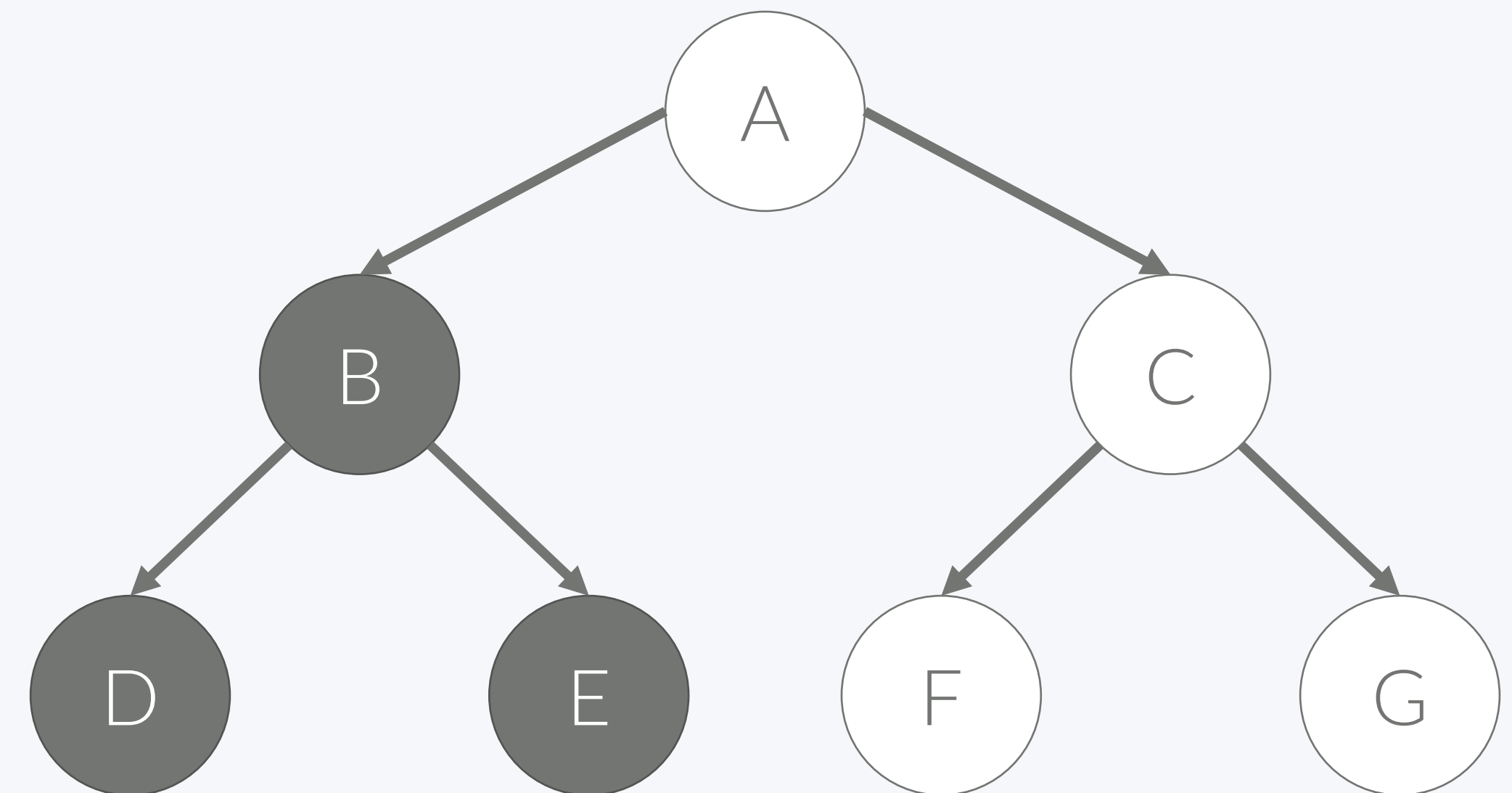


프리오더

31

Pre-order

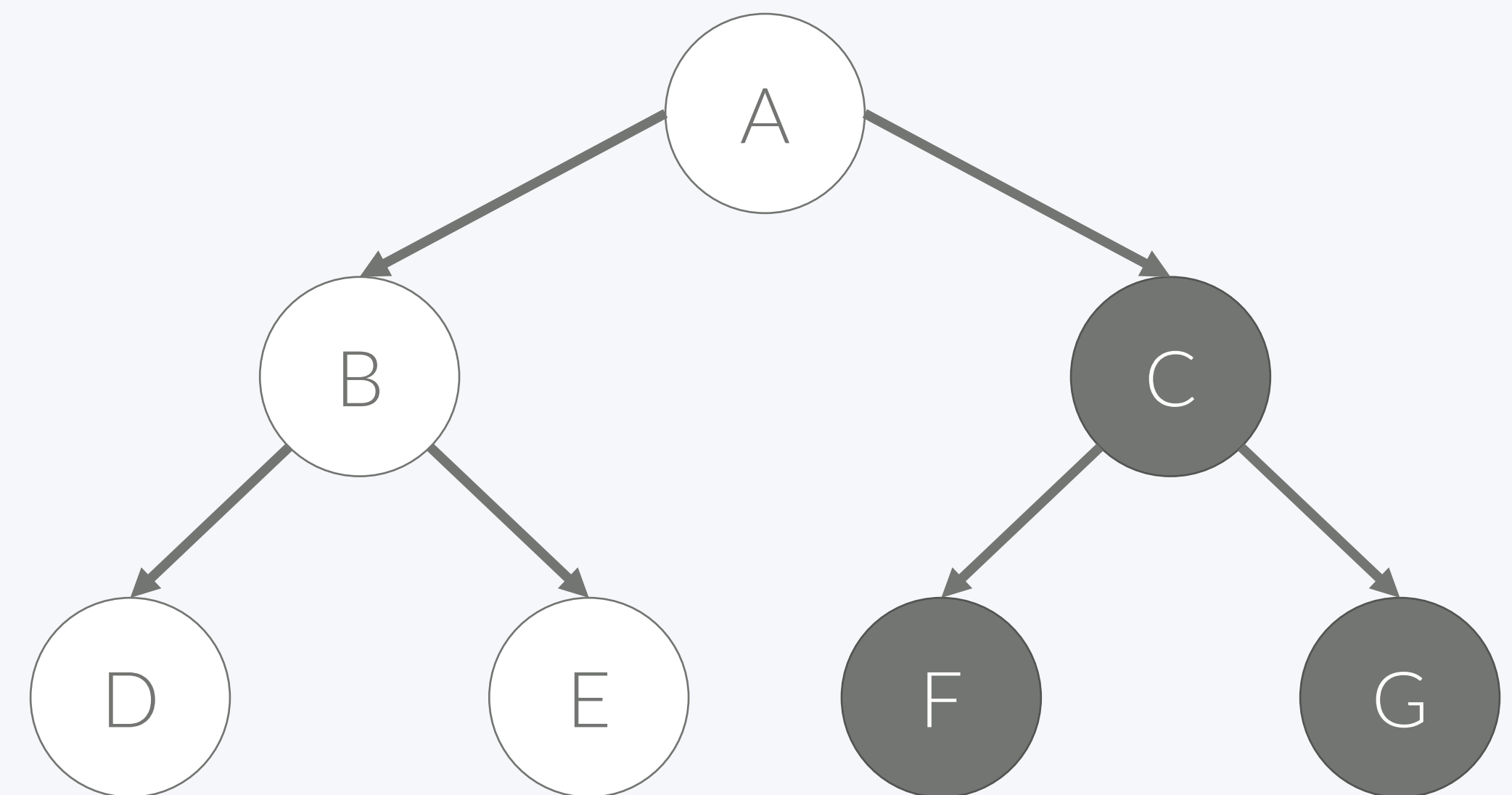
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더



프리오더

Pre-order

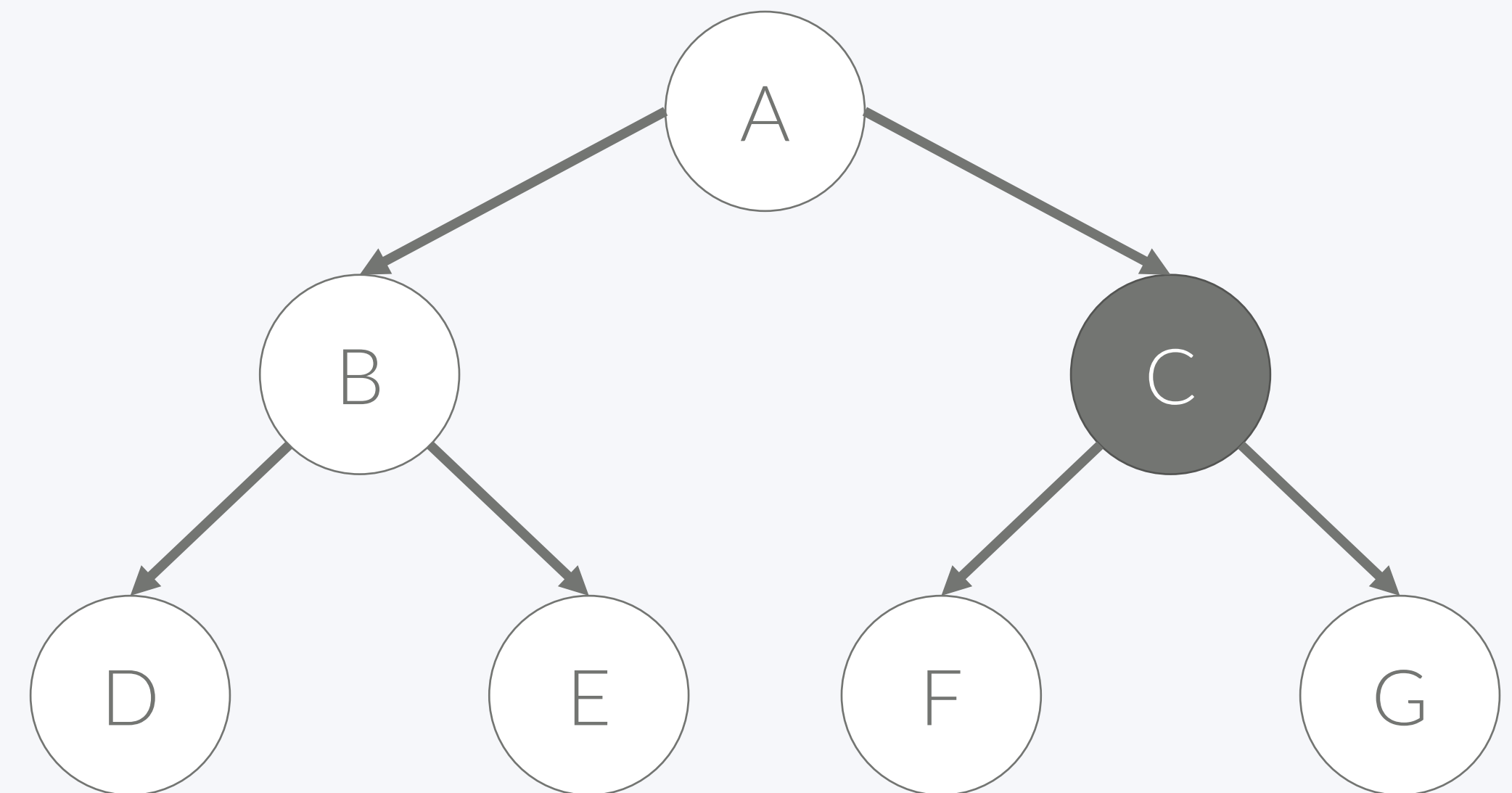
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더



프리오더

Pre-order

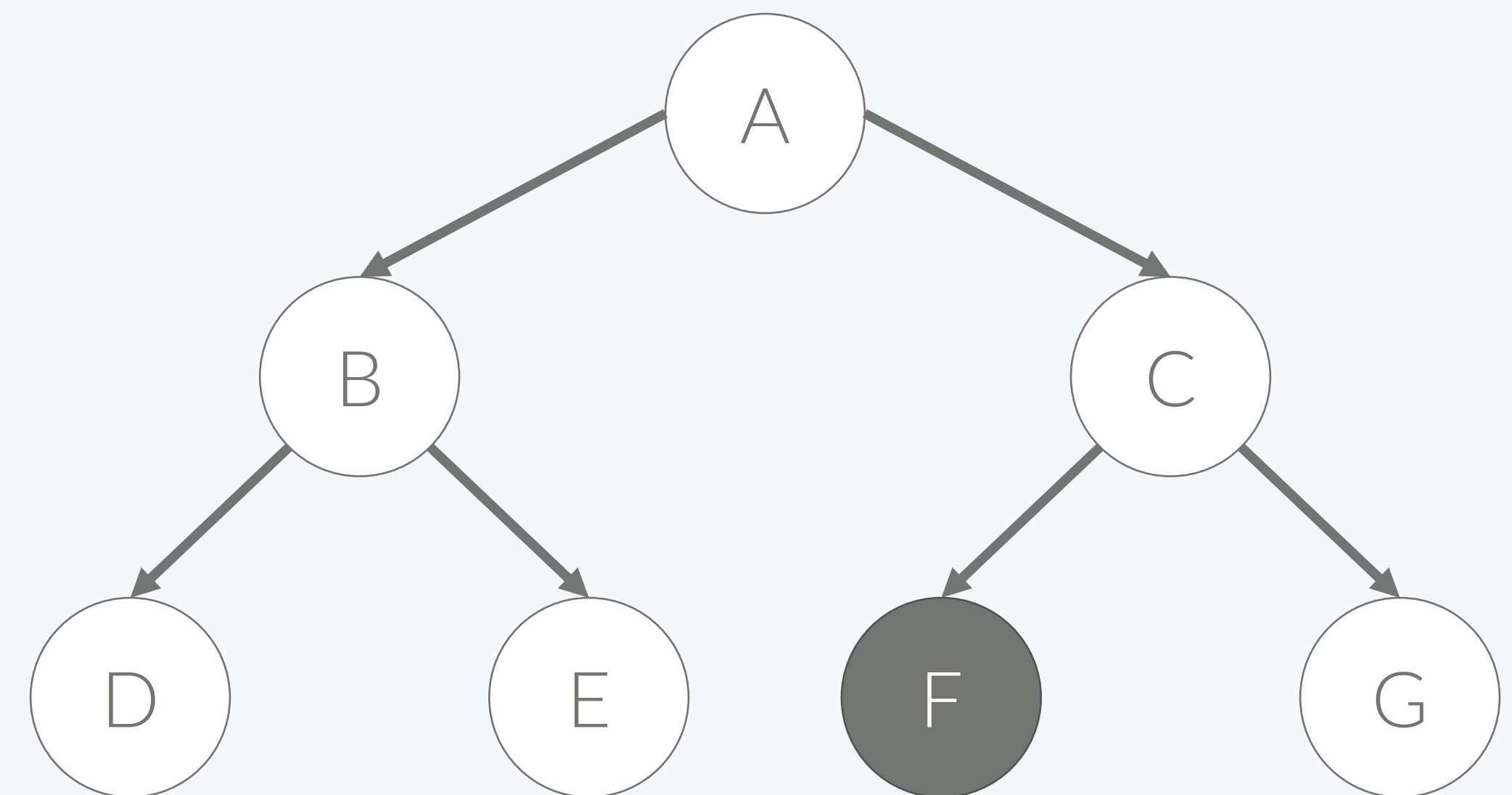
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더



프리오더

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - F
 - 오른쪽 자식 프리오더

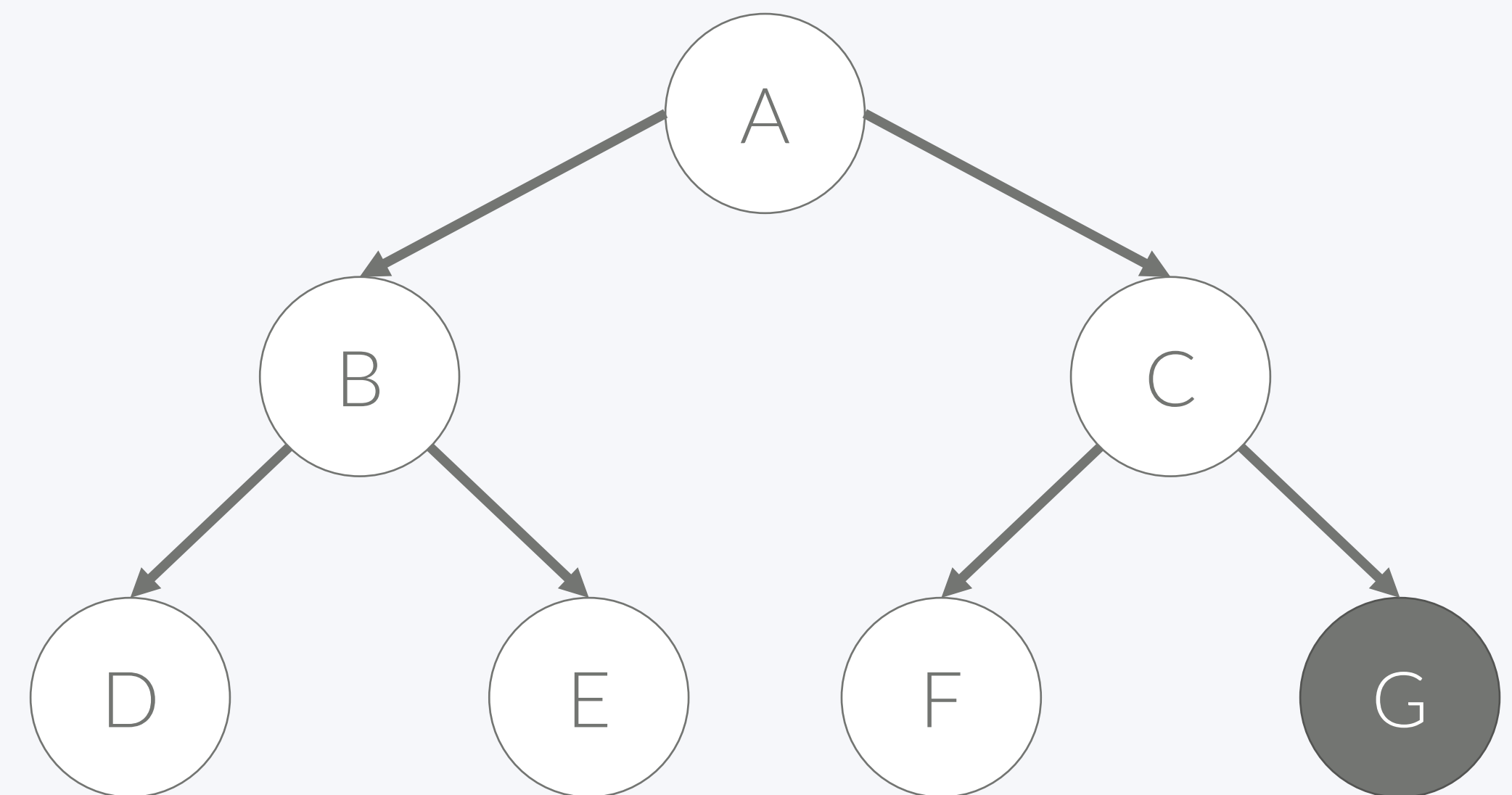


프리오더

35

Pre-order

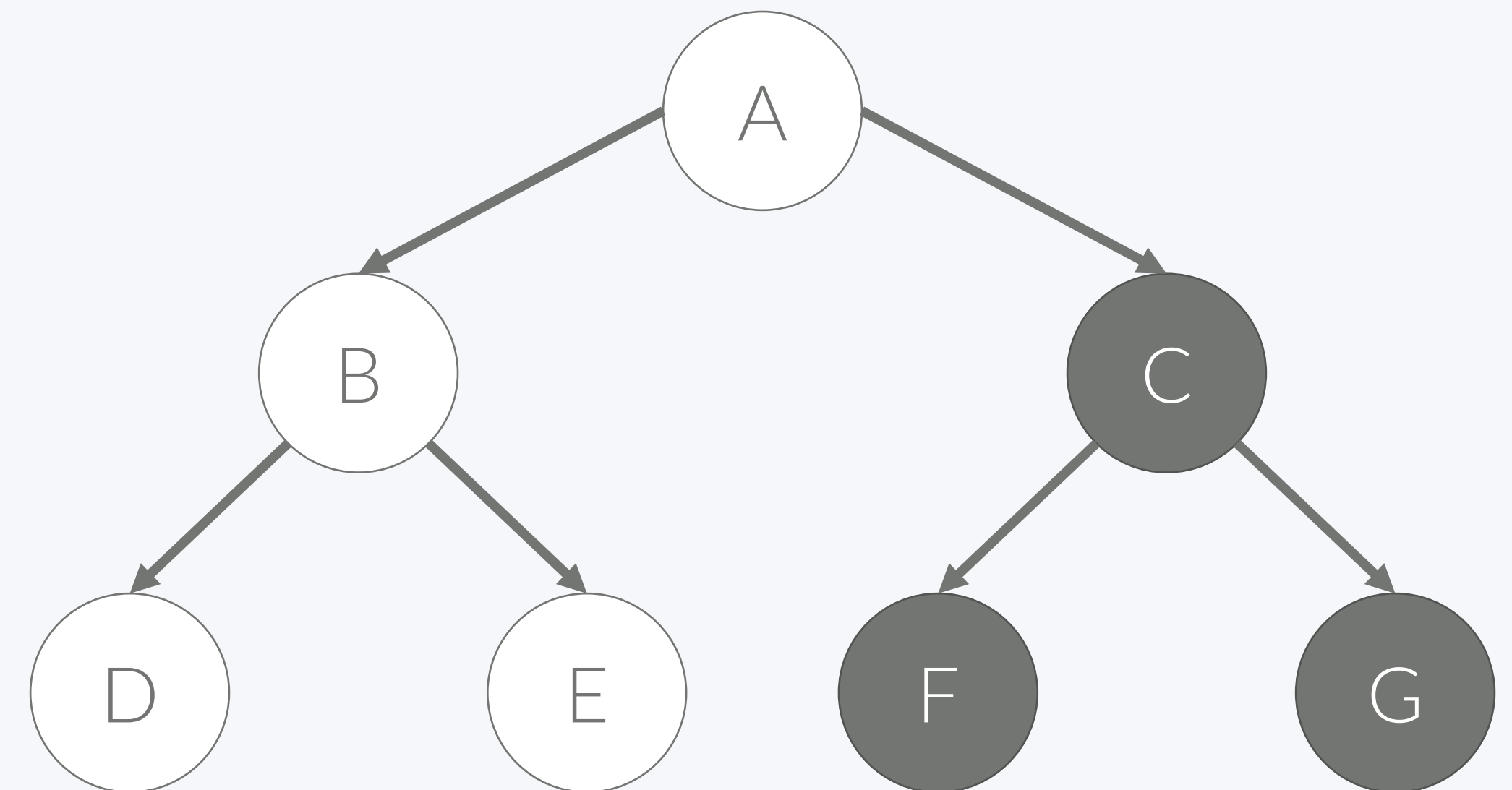
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - F
 - 오른쪽 자식 프리오더
 - G



프리오더

Pre-order

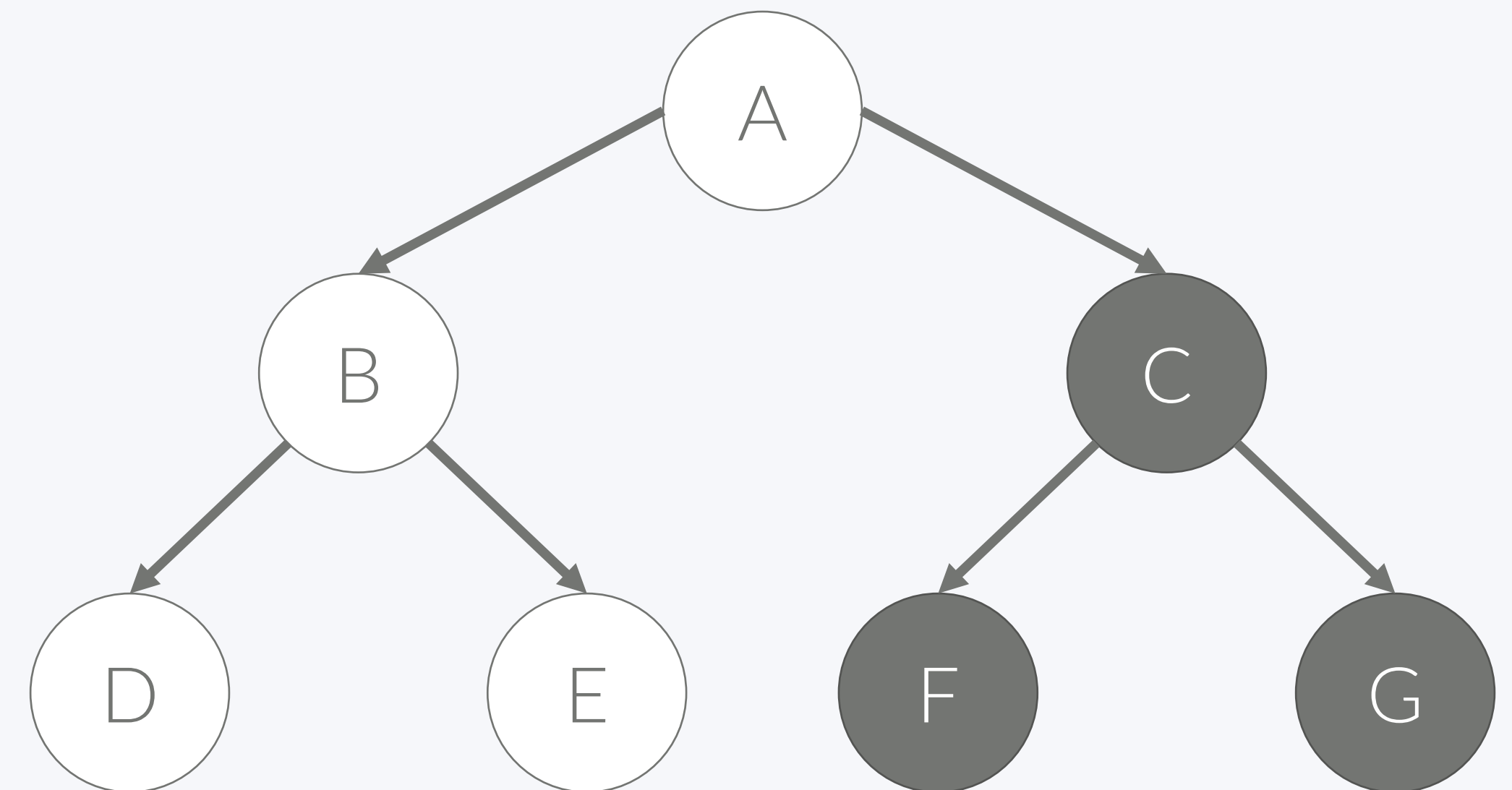
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - CFG



프리오더

Pre-order

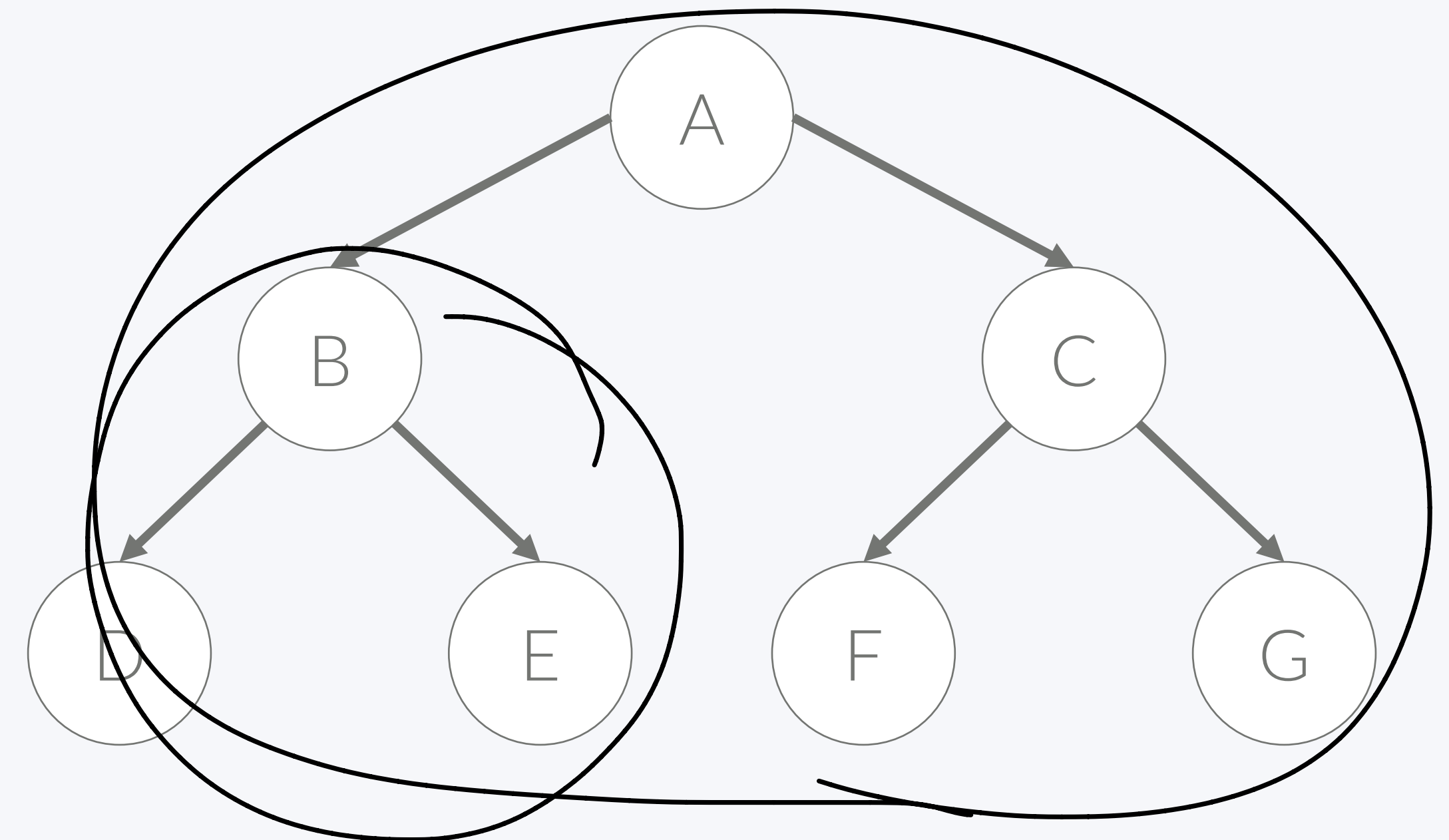
- ABDECFG
- 그래프의 DFS이 순서와 같다



인오더

In-order

- 왼쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

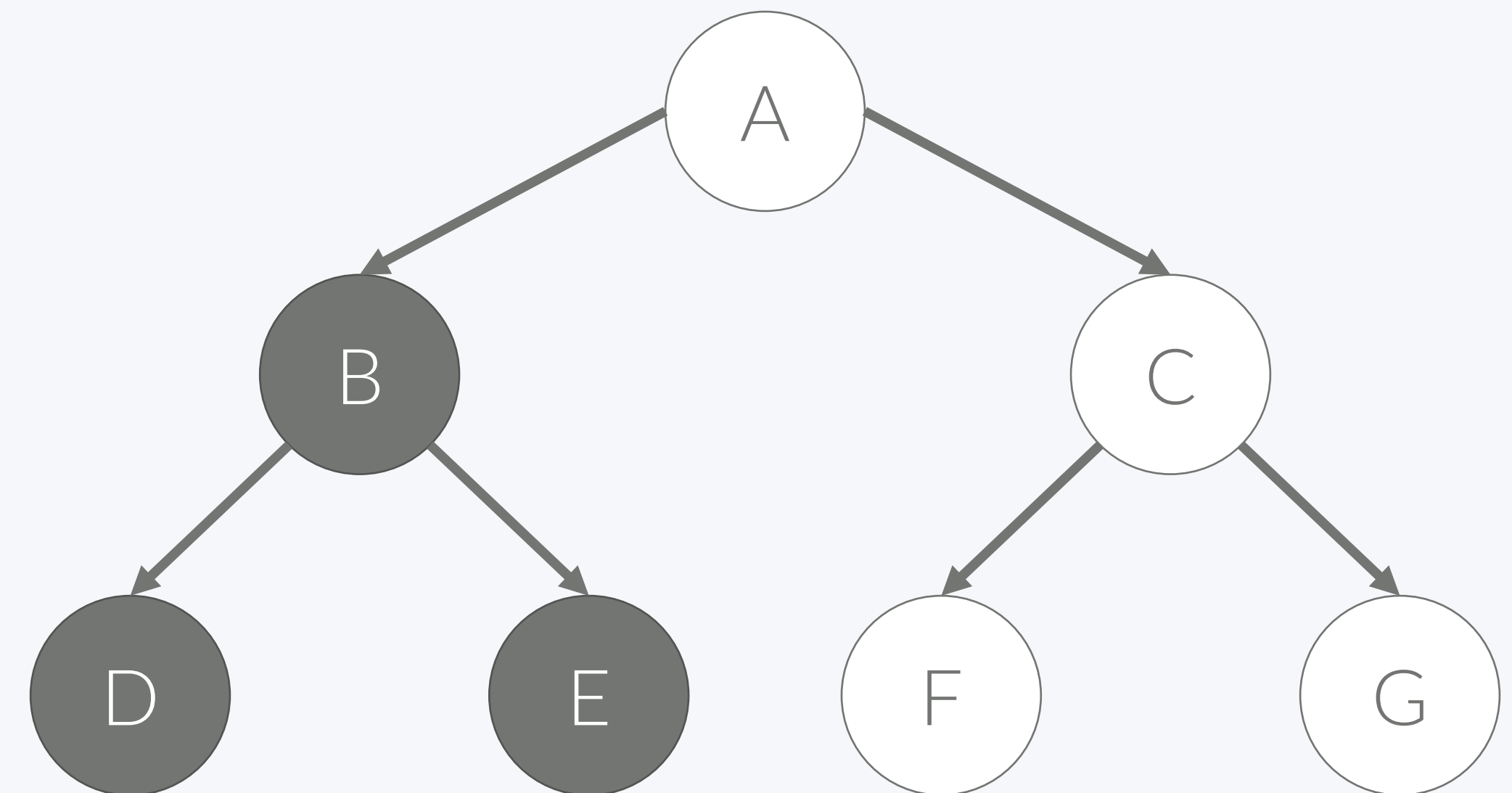


인오더

In-order

39

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - 노드 방문
 - 오른쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

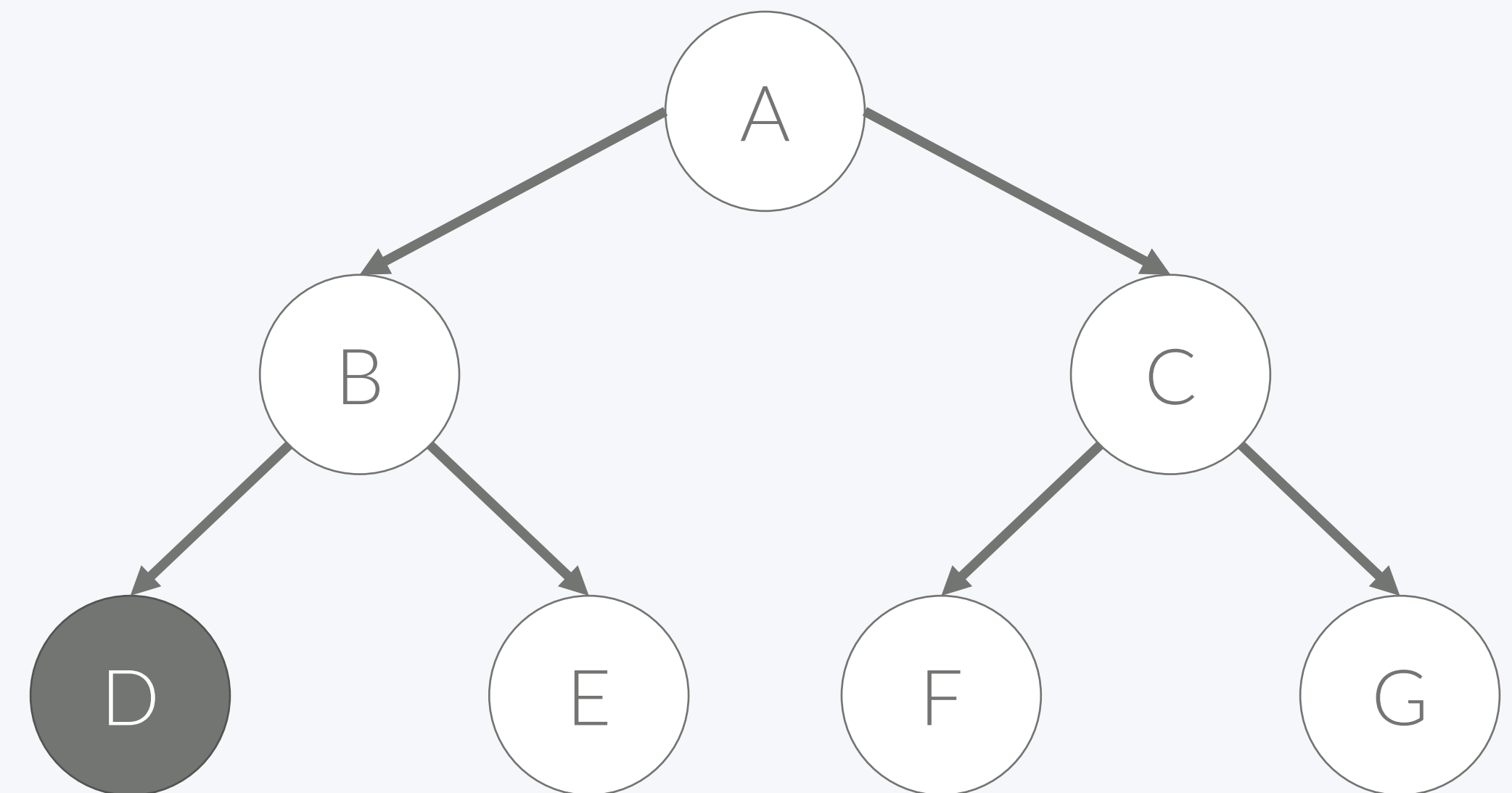


인오더

In-order

40

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
 - 노드 방문
 - 오른쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

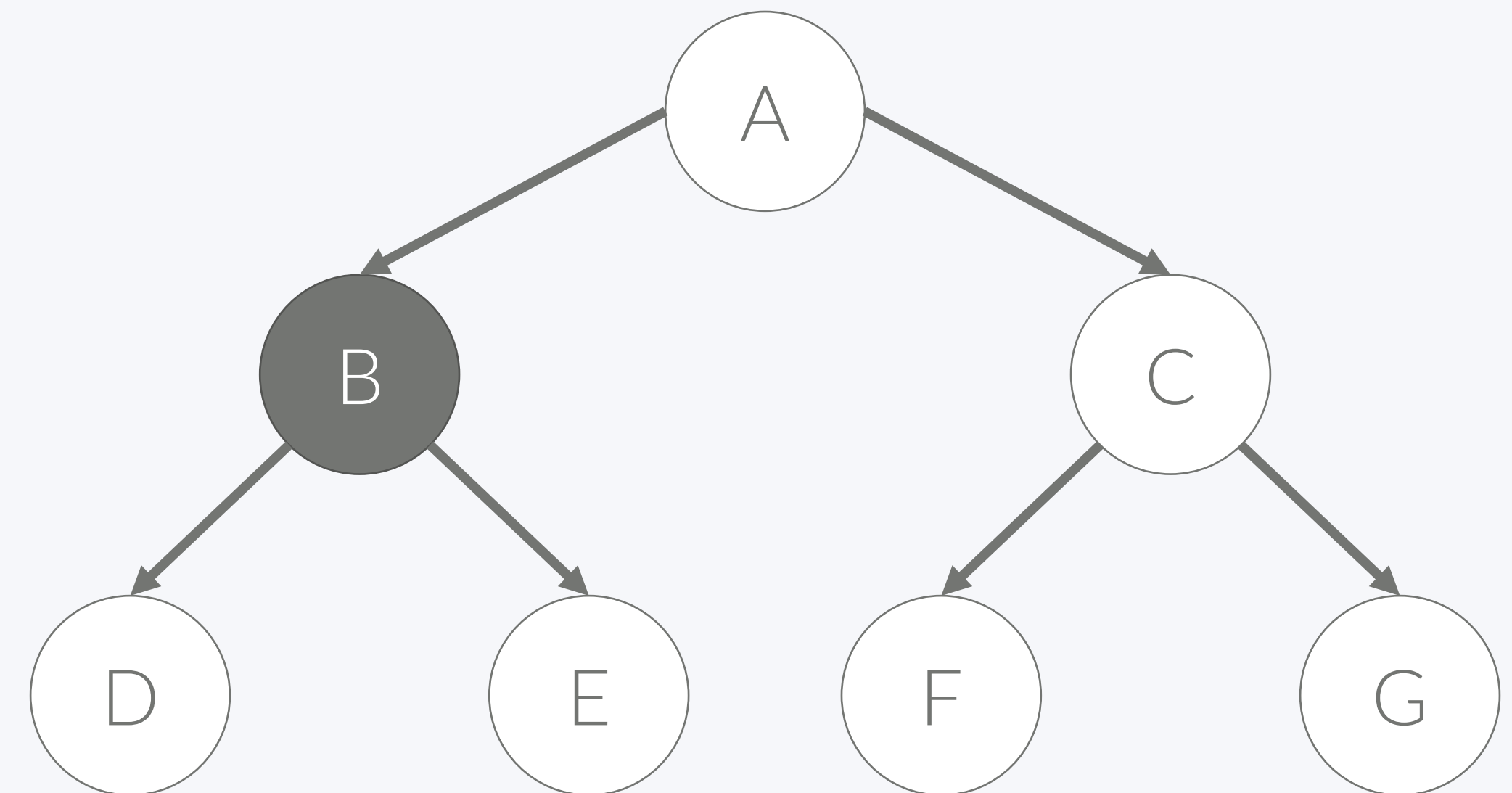


인오더

In-order

41

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
- 노드 방문
 - B
- 오른쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

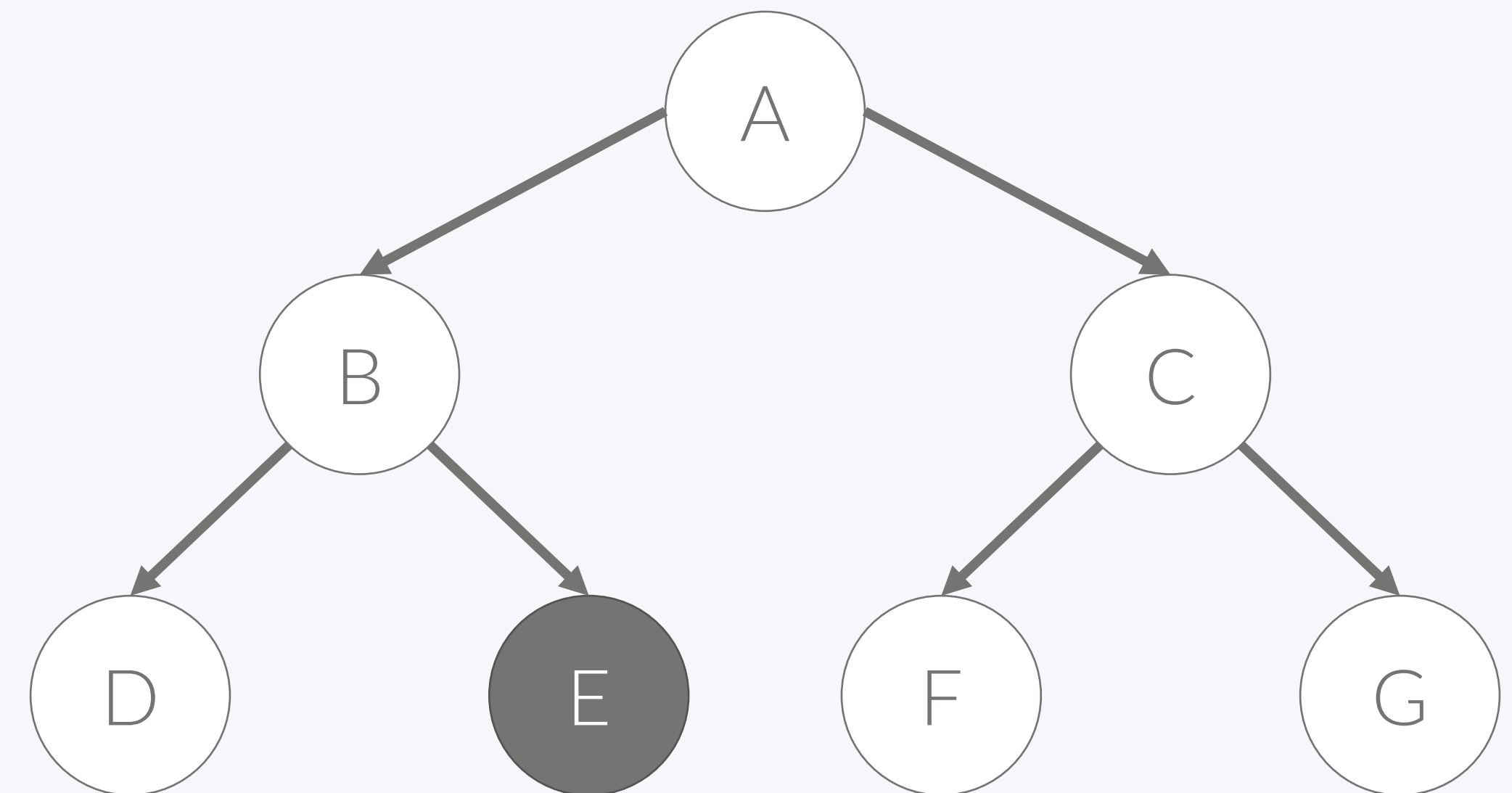


인오더

In-order

42

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
 - 노드 방문
 - B
 - 오른쪽 자식 인오더
 - E
- 노드 방문
- 오른쪽 자식 인오더

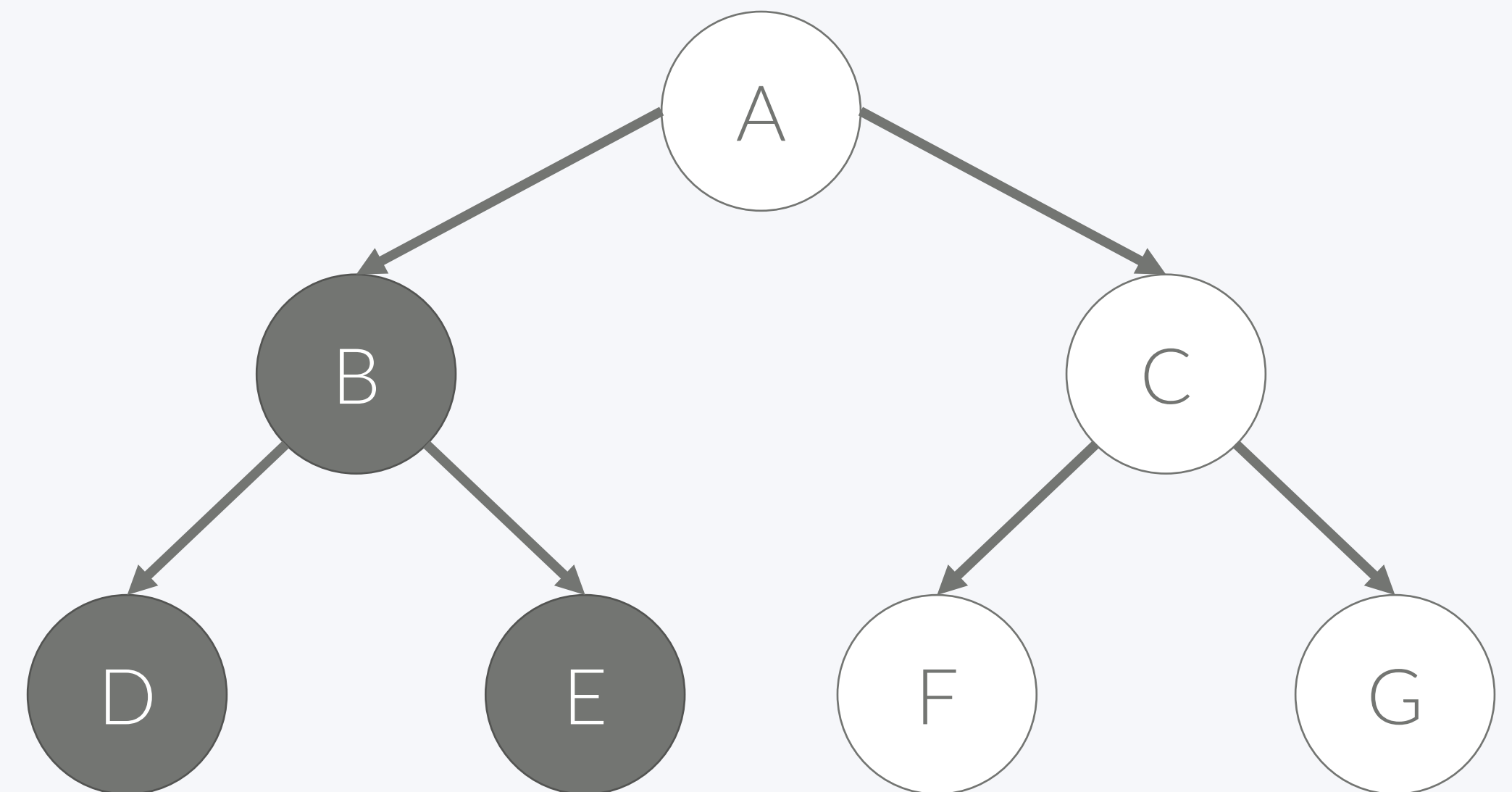


인오더

In-order

43

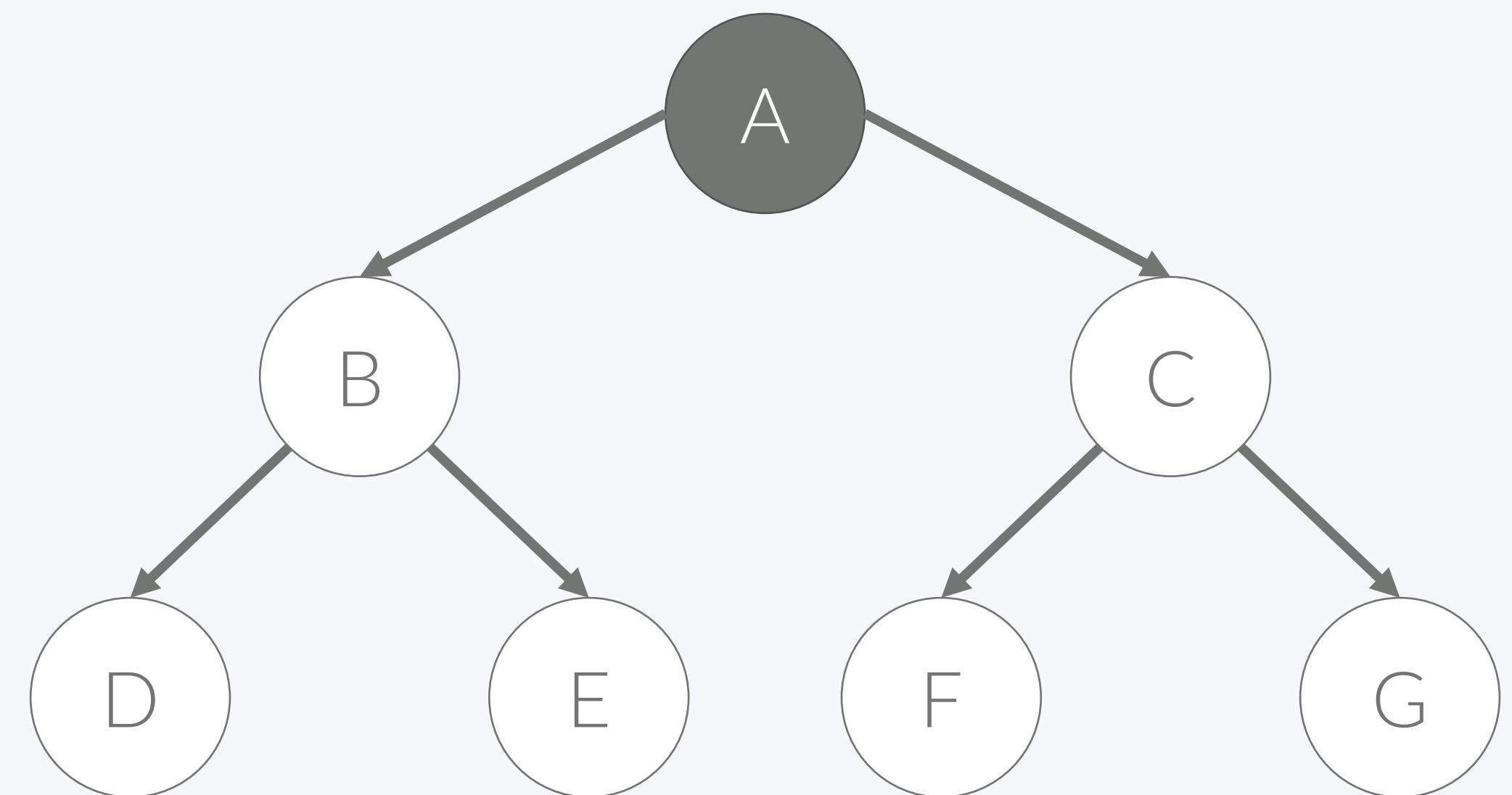
- 왼쪽 자식 인오더
 - DBE
- 노드 방문
- 오른쪽 자식 인오더



인오더

In-order

- 왼쪽 자식 인오더
 - DBE
- **노드 방문**
 - A
- 오른쪽 자식 인오더

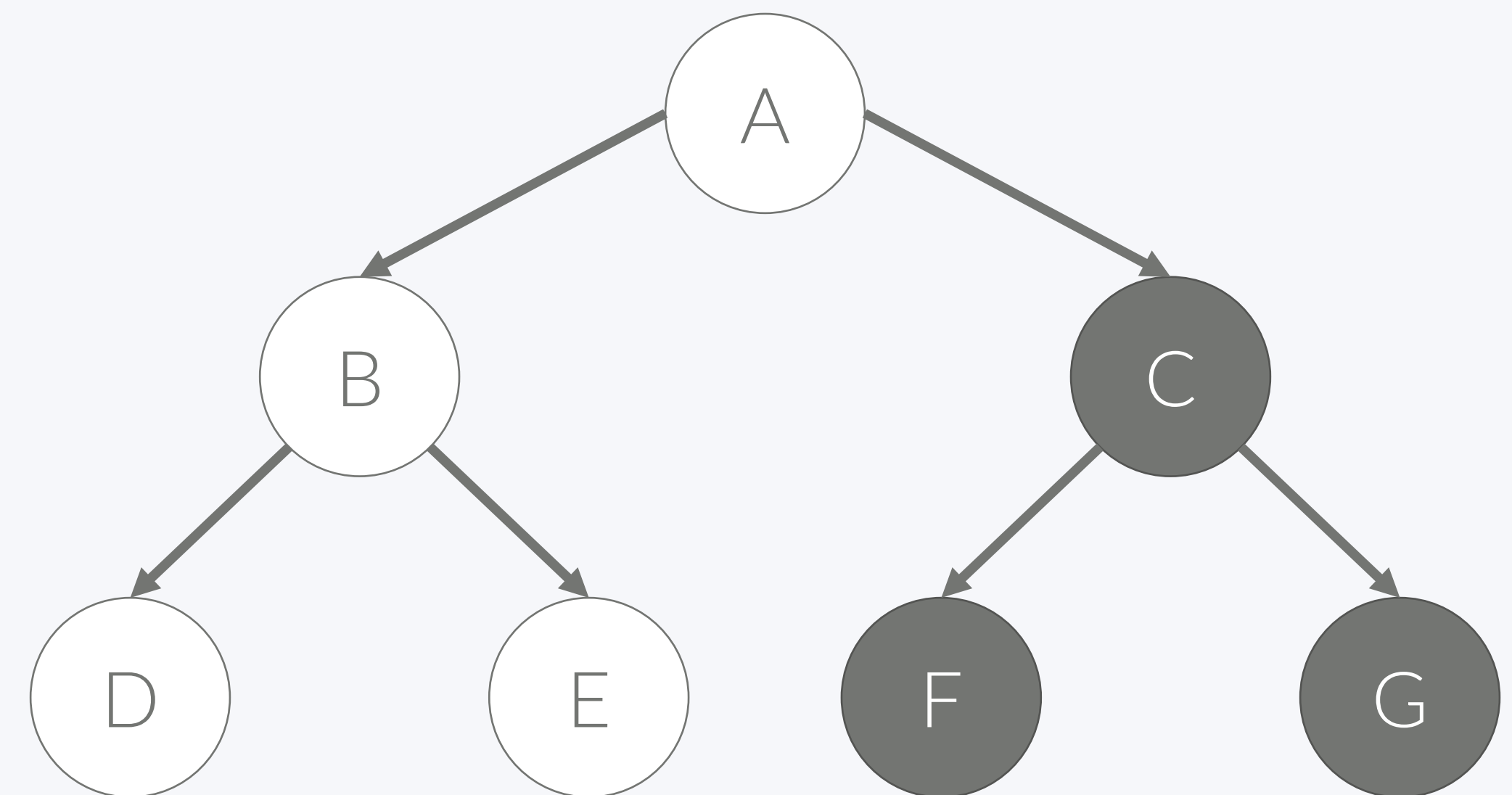


인오더

In-order

45

- 왼쪽 자식 인오더
 - DBE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - 노드 방문
 - 오른쪽 자식 인오더

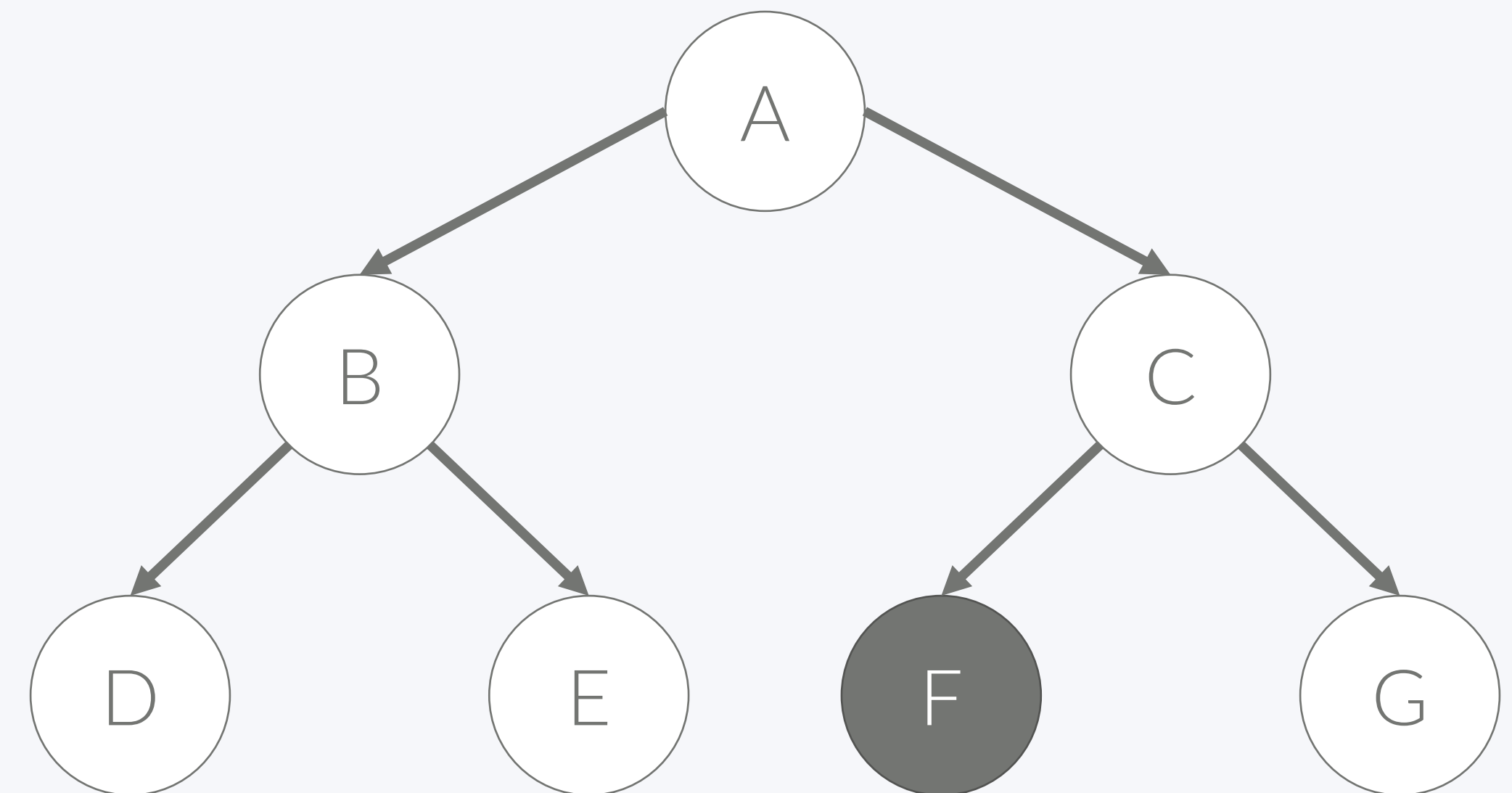


인오더

In-order

46

- 왼쪽 자식 인오더
 - DBE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - F
 - 노드 방문
 - 오른쪽 자식 인오더

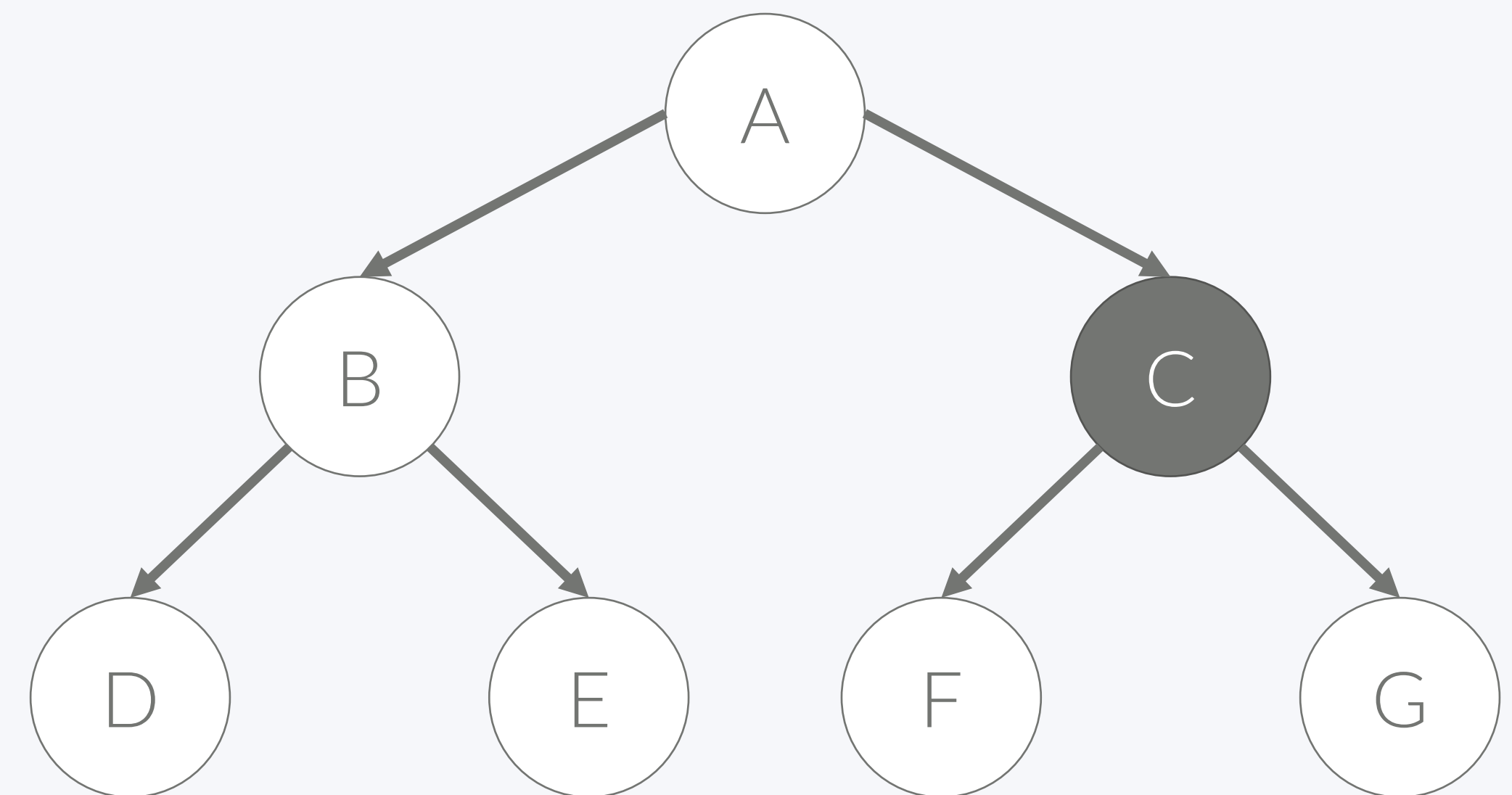


인오더

47

In-order

- 왼쪽 자식 인오더
 - DBE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - F
 - 노드 방문
 - C
 - 오른쪽 자식 인오더

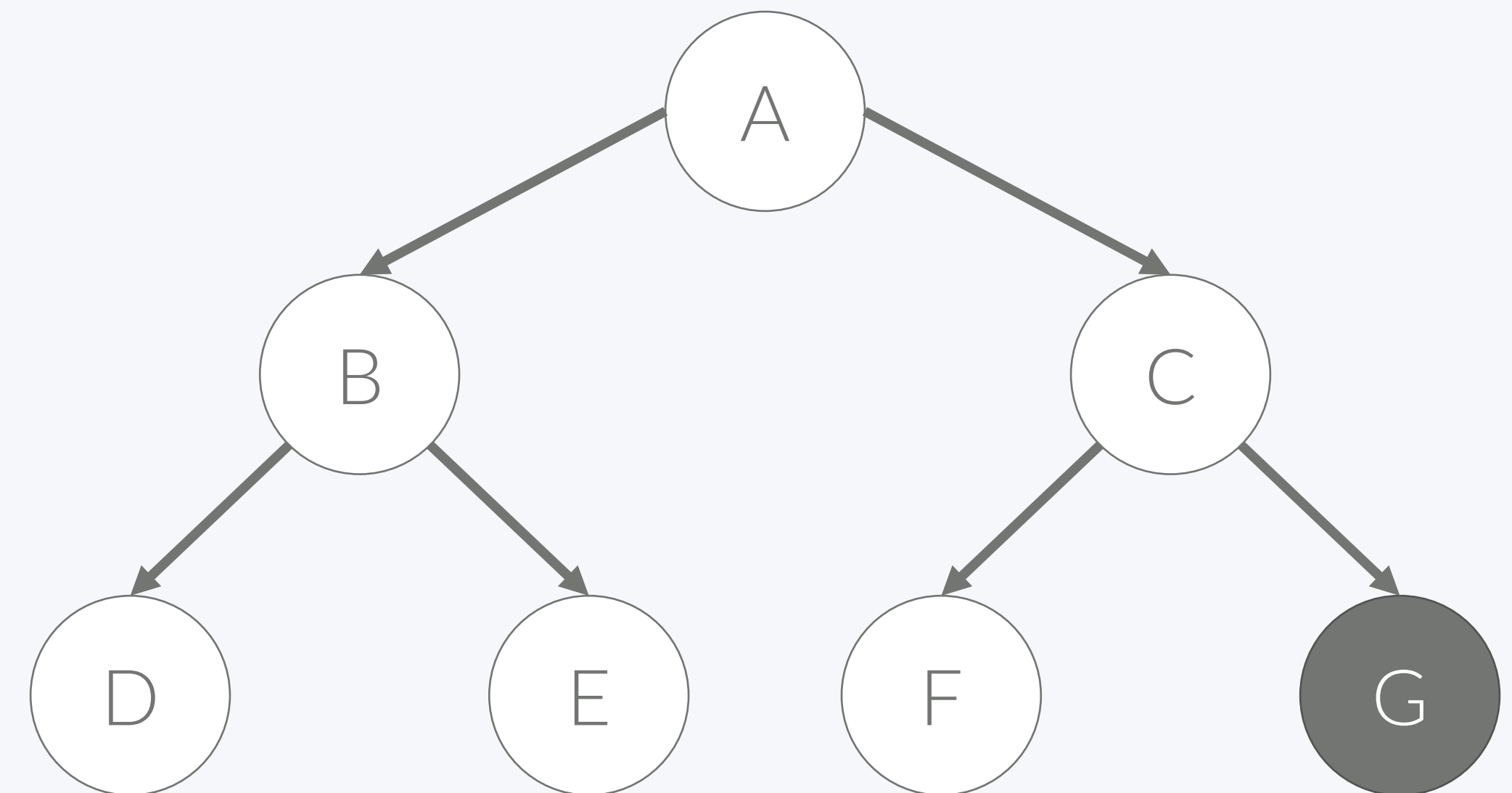


인오더

48

In-order

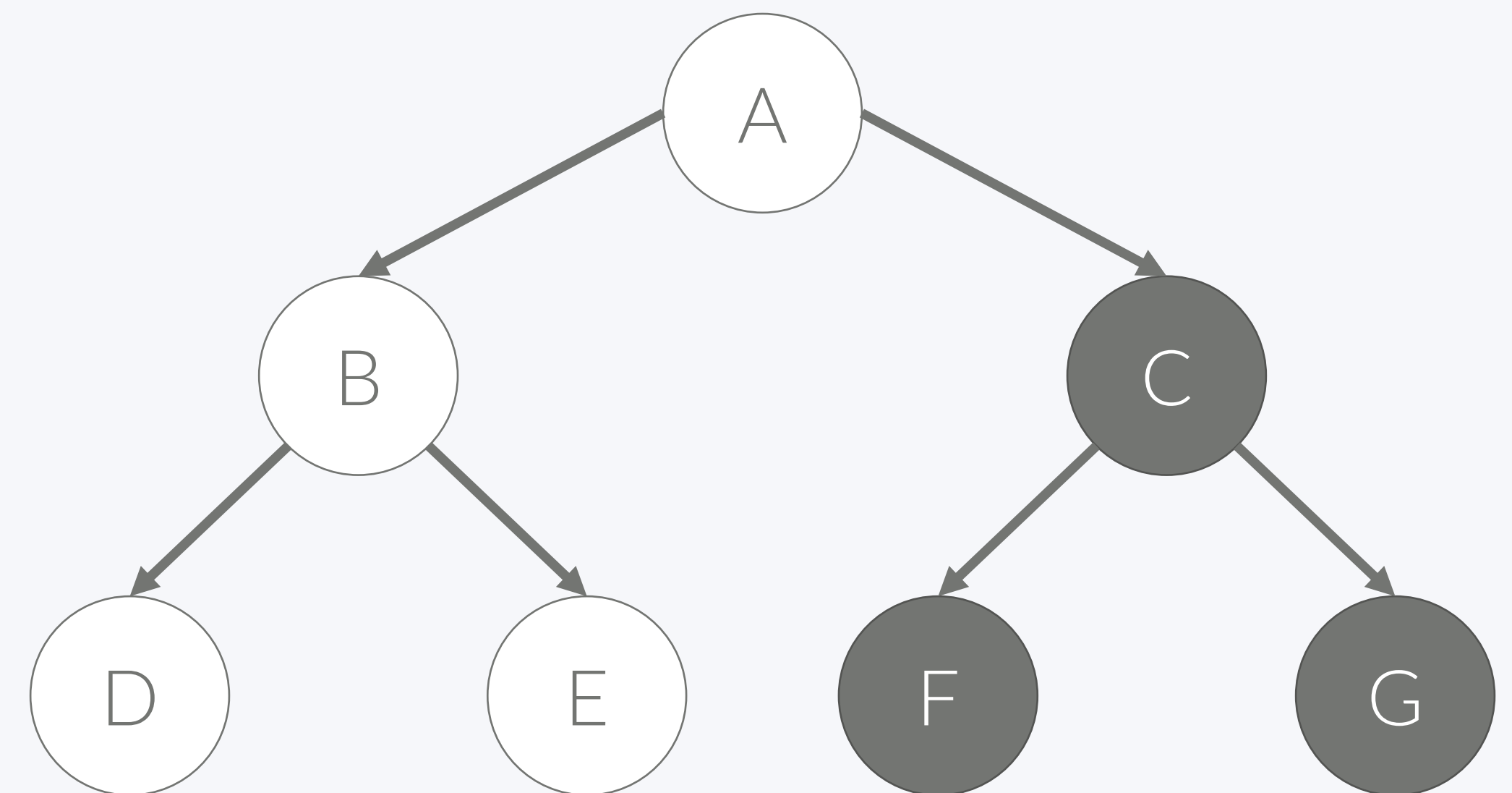
- 왼쪽 자식 인오더
 - DBE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - F
 - 노드 방문
 - C
 - 오른쪽 자식 인오더
 - G



인오더

In-order

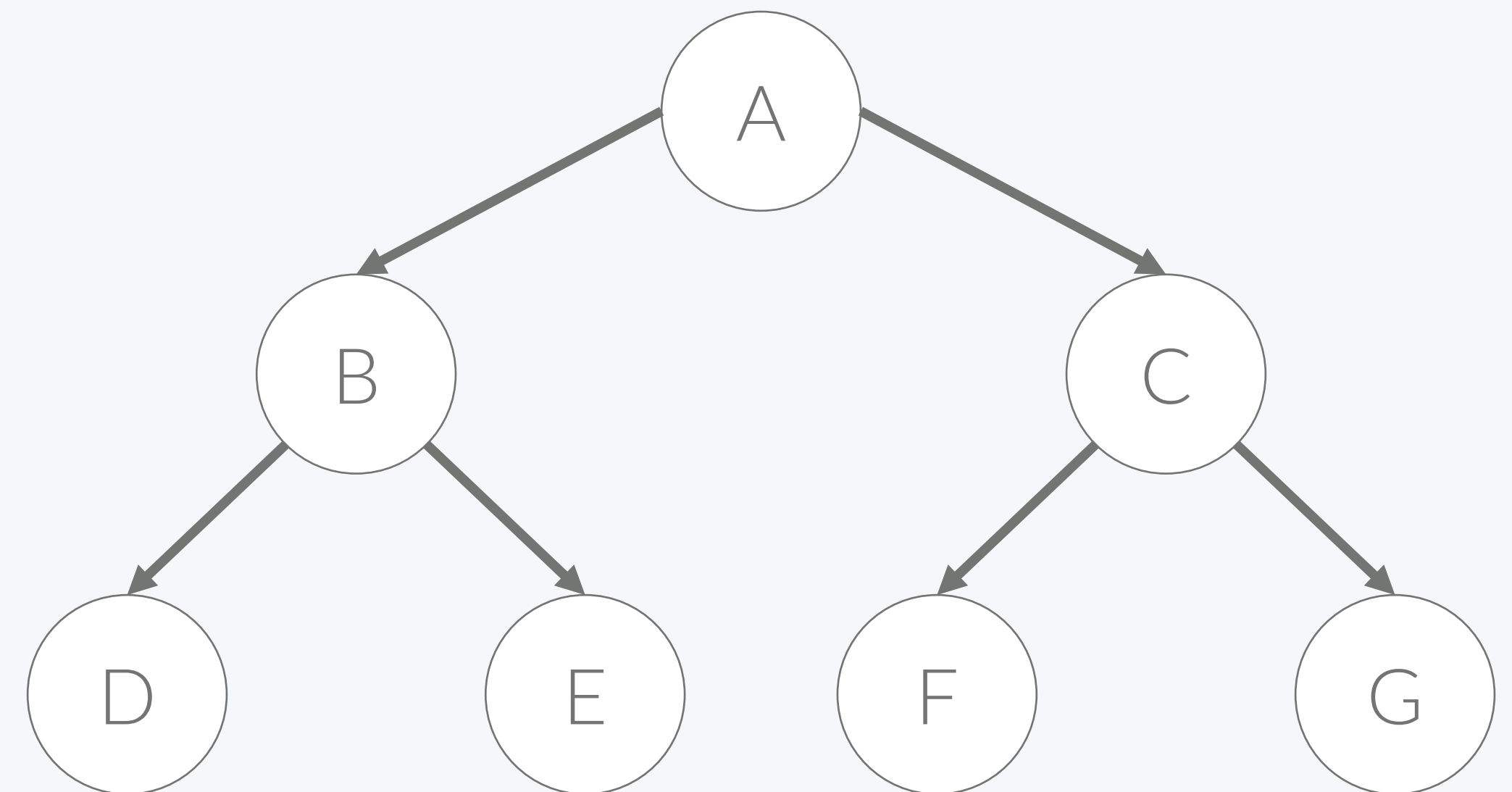
- 왼쪽 자식 인오더
 - DBE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - FCG



인오더

In-order

- DBEAFCG

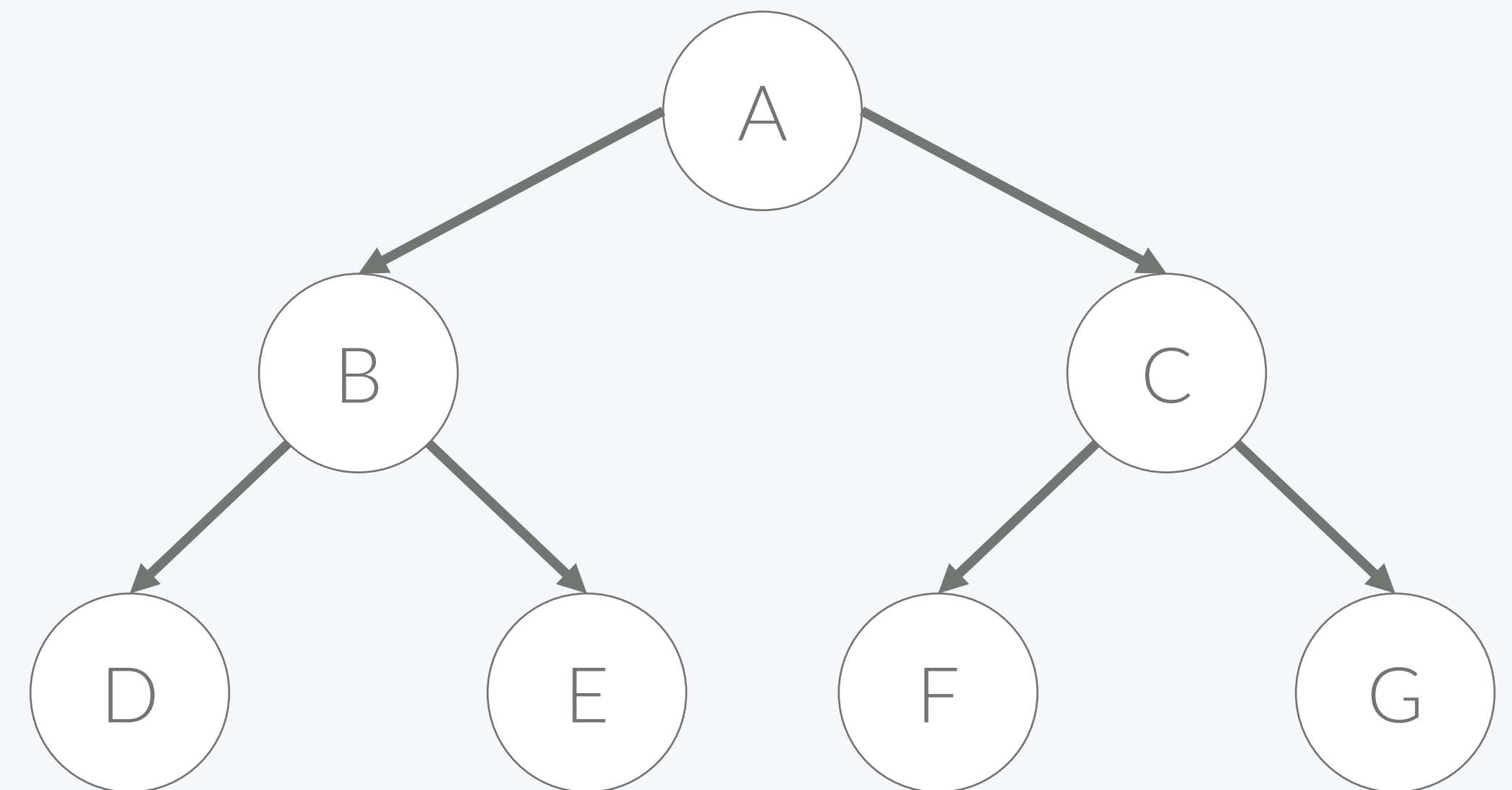


포스트오더

51

Postorder

- 왼쪽 자식 포스트오더
- 오른쪽 자식 포스트오더
- 노드 방문

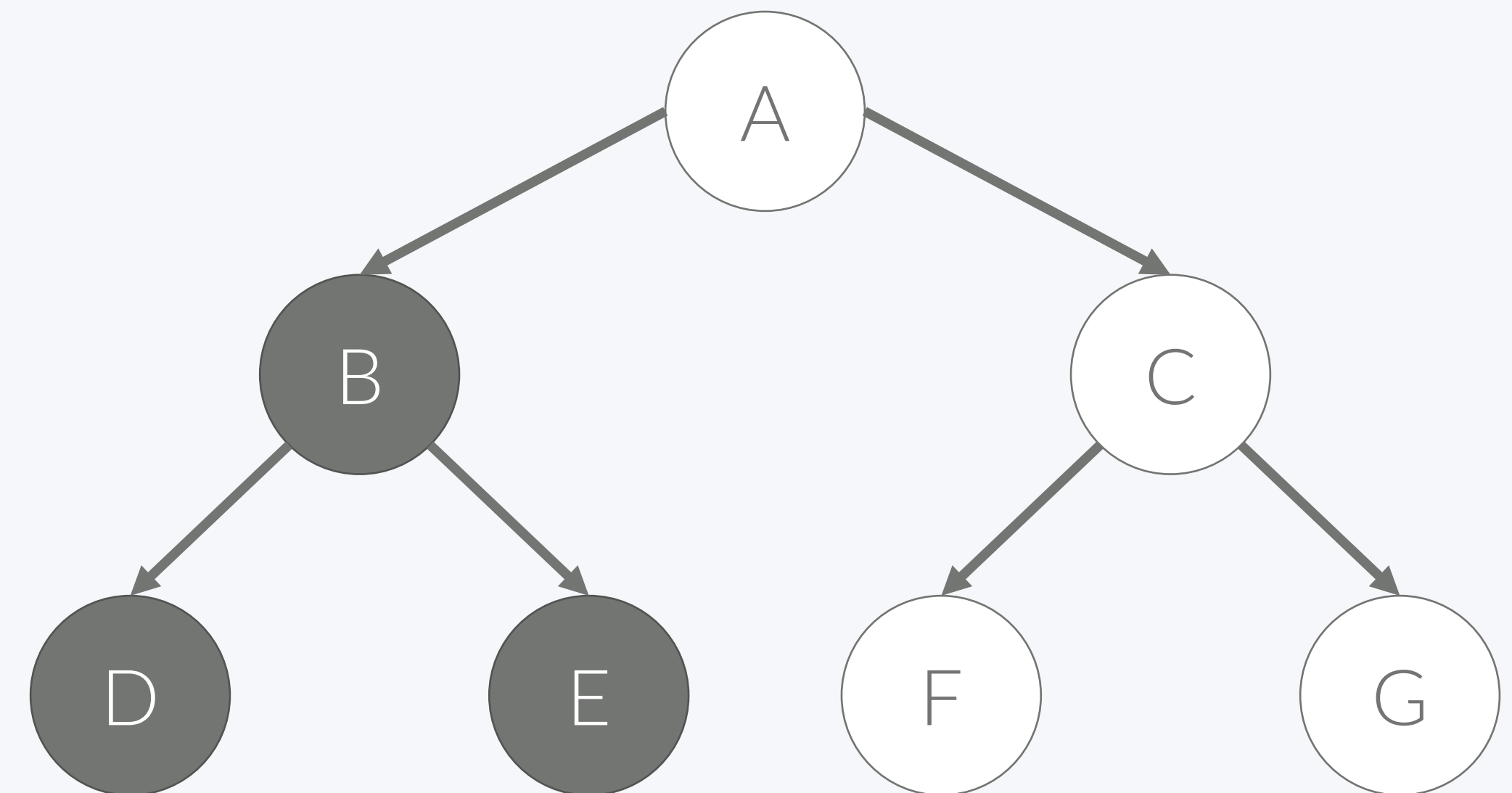


포스트오더

52

Postorder

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - 오른쪽 자식 포스트오더
 - 노드 방문
- 오른쪽 자식 포스트오더
- 노드 방문

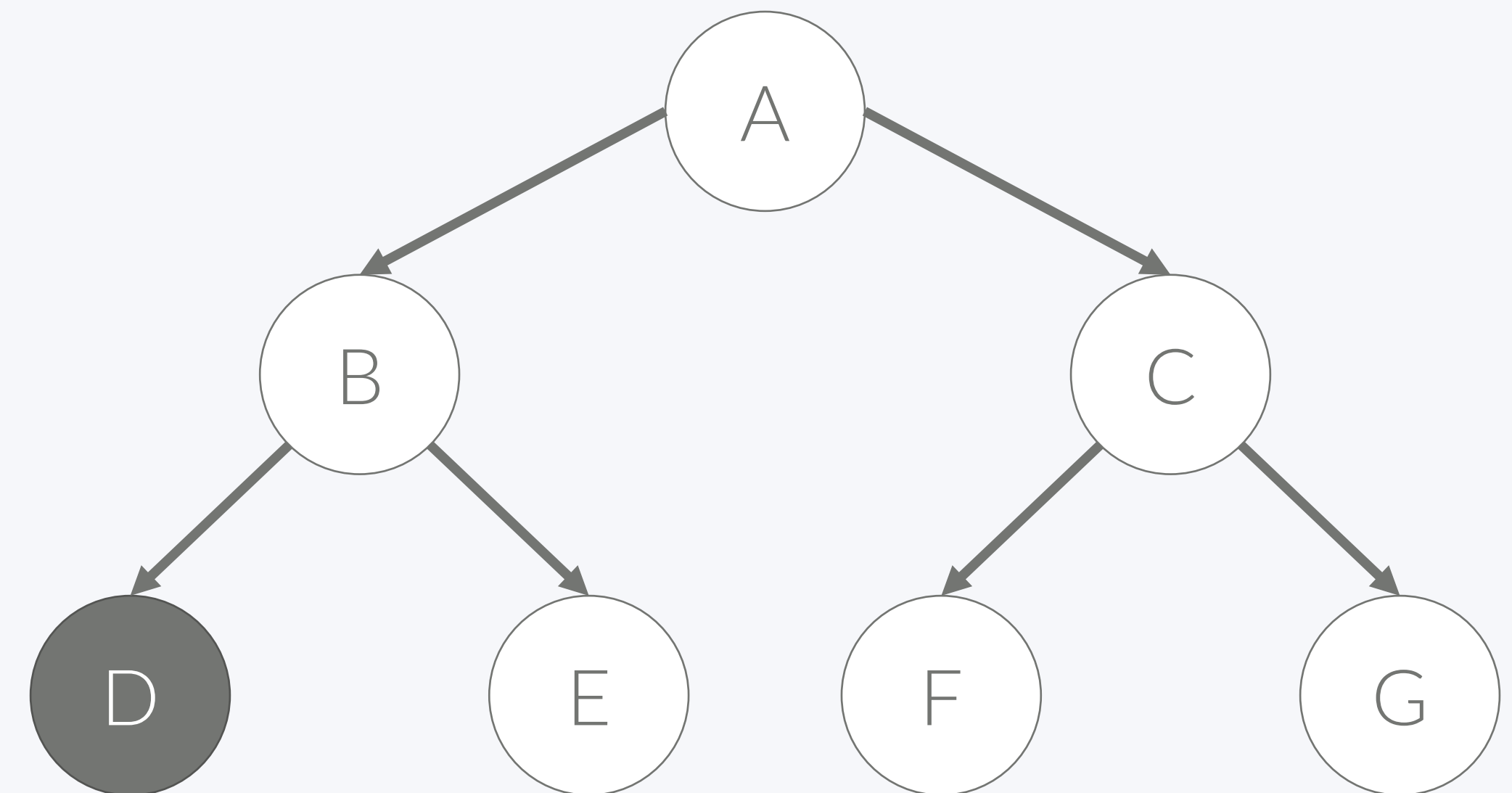


포스트오더

53

Postorder

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - D
 - 오른쪽 자식 포스트오더
 - 노드 방문
- 오른쪽 자식 포스트오더
- 노드 방문

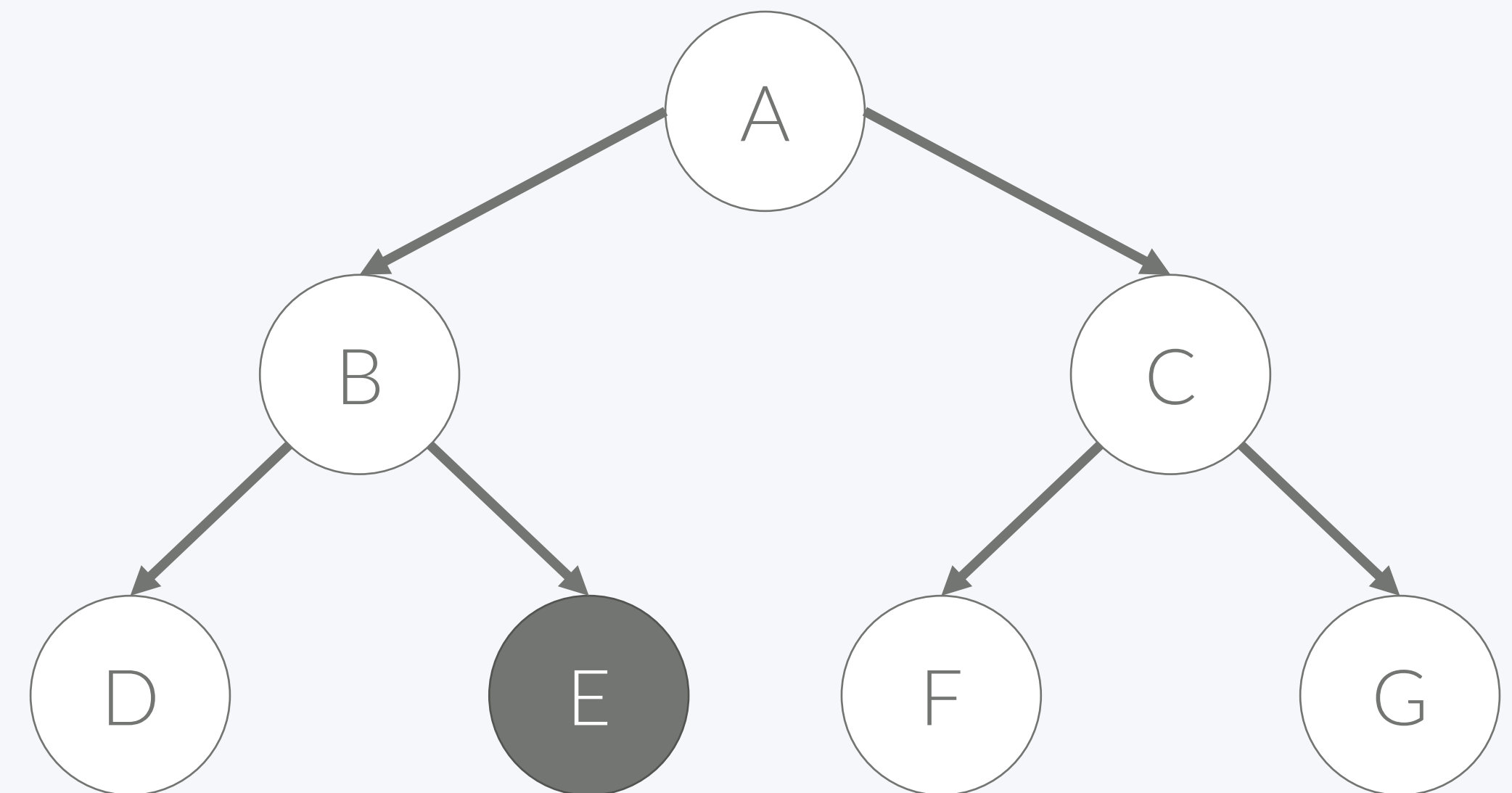


포스트오더

54

Postorder

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - D
 - 오른쪽 자식 포스트오더
 - E
 - 노드 방문
- 오른쪽 자식 포스트오더
- 노드 방문

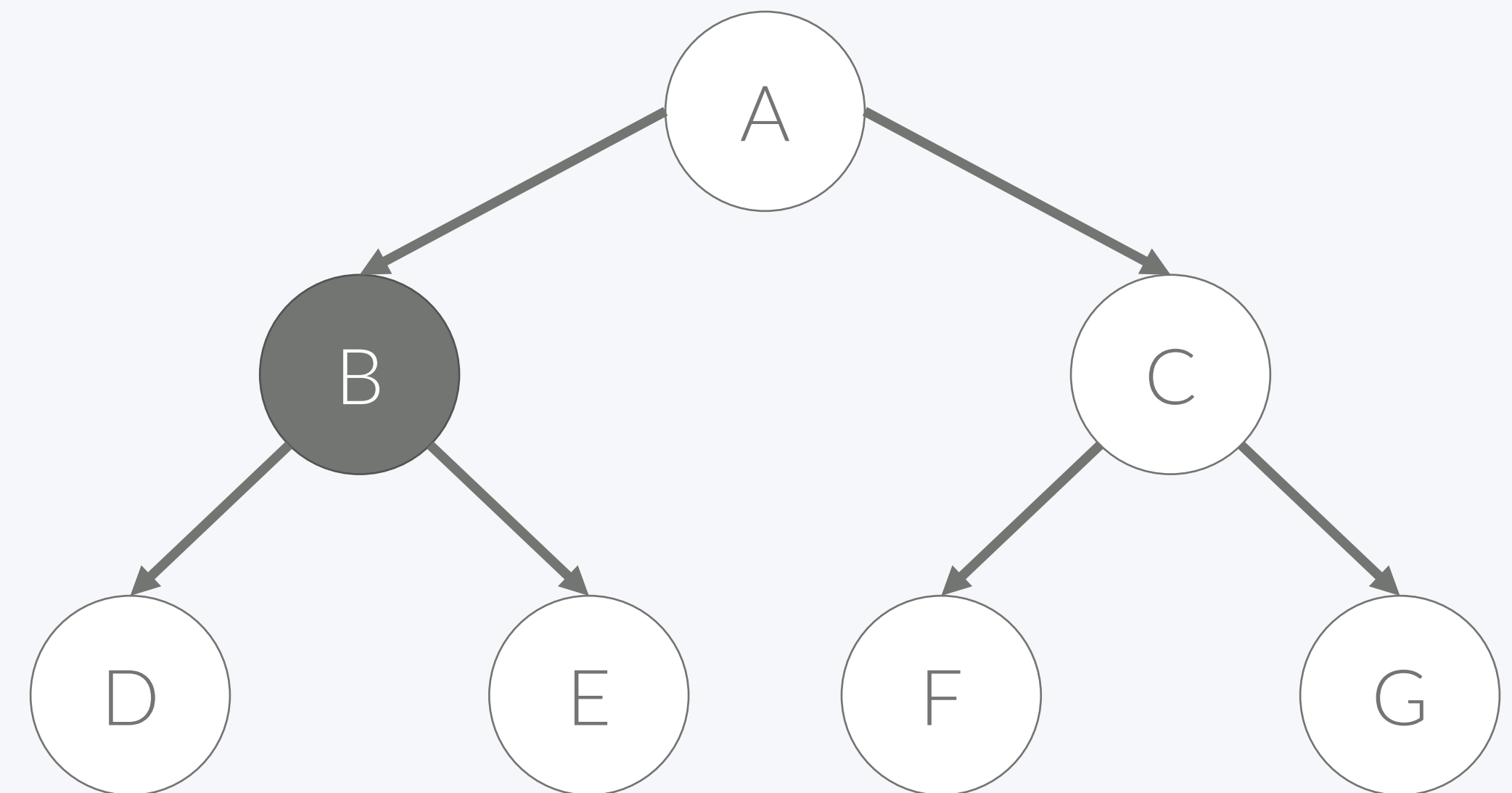


포스트오더

55

Postorder

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - D
 - 오른쪽 자식 포스트오더
 - E
 - 노드 방문
 - B
- 오른쪽 자식 포스트오더
- 노드 방문

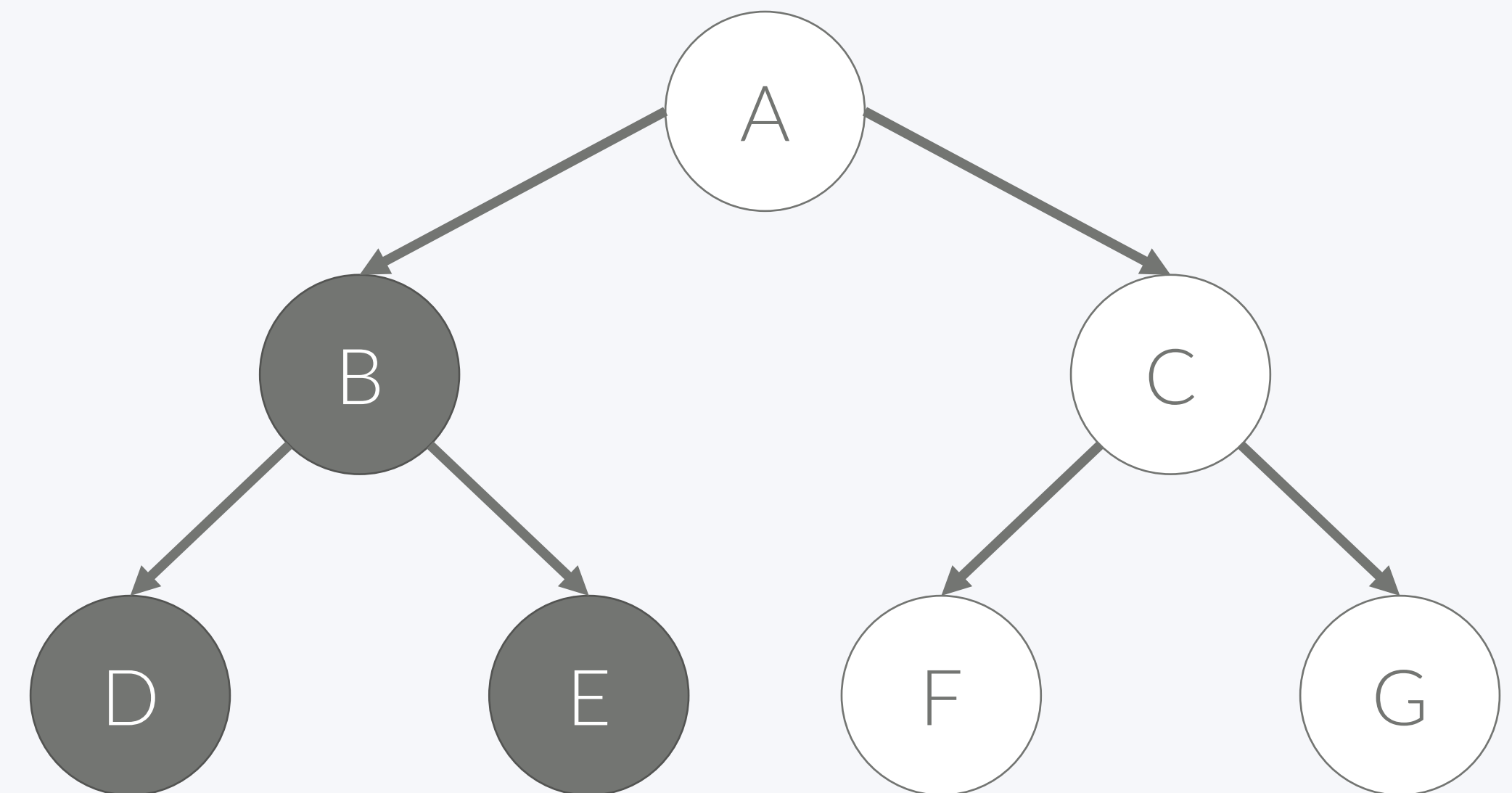


포스트오더

56

Postorder

- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
- 노드 방문

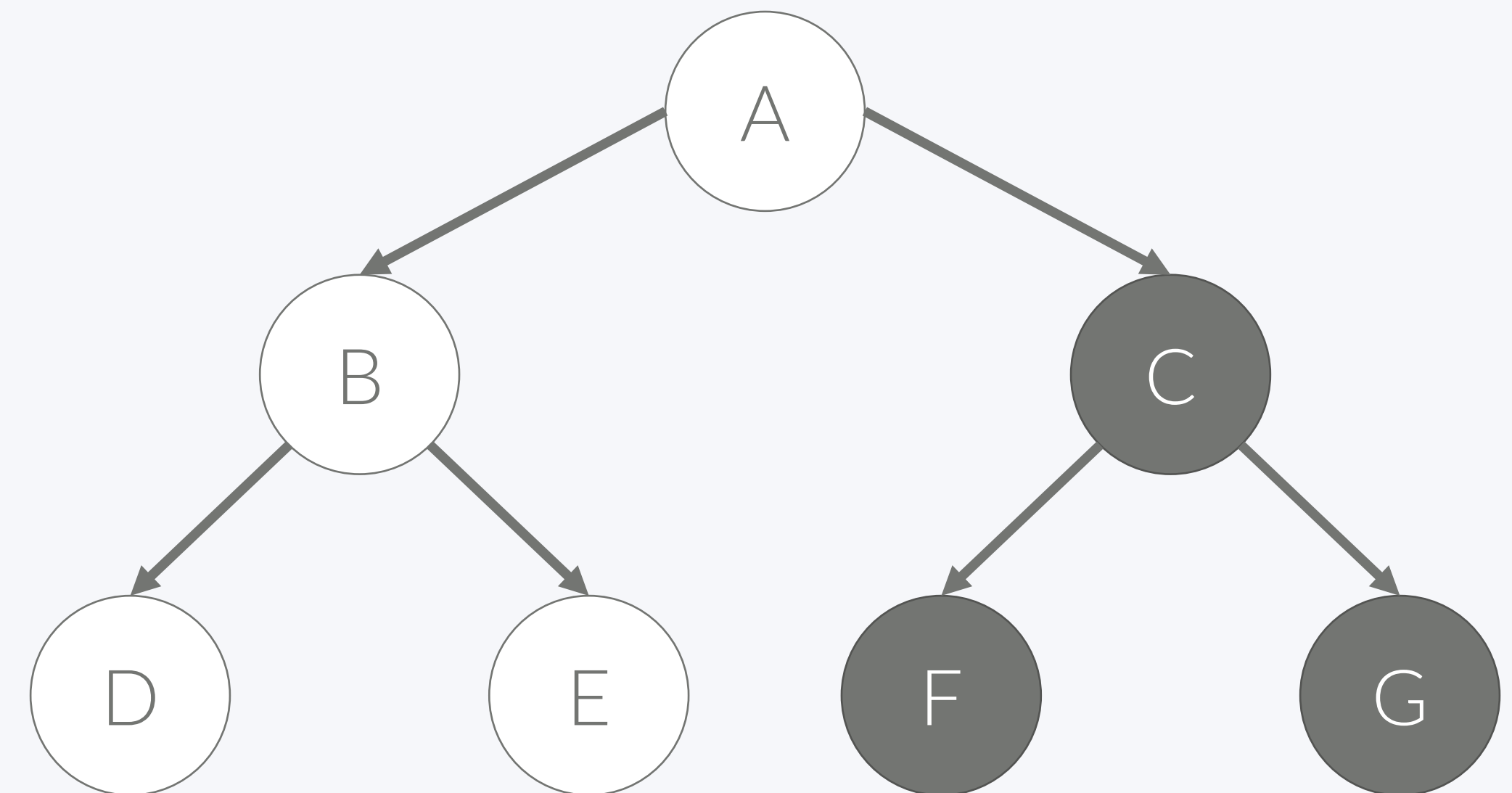


포스트오더

57

Postorder

- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - 오른쪽 자식 포스트오더
 - 노드 방문
- 노드 방문

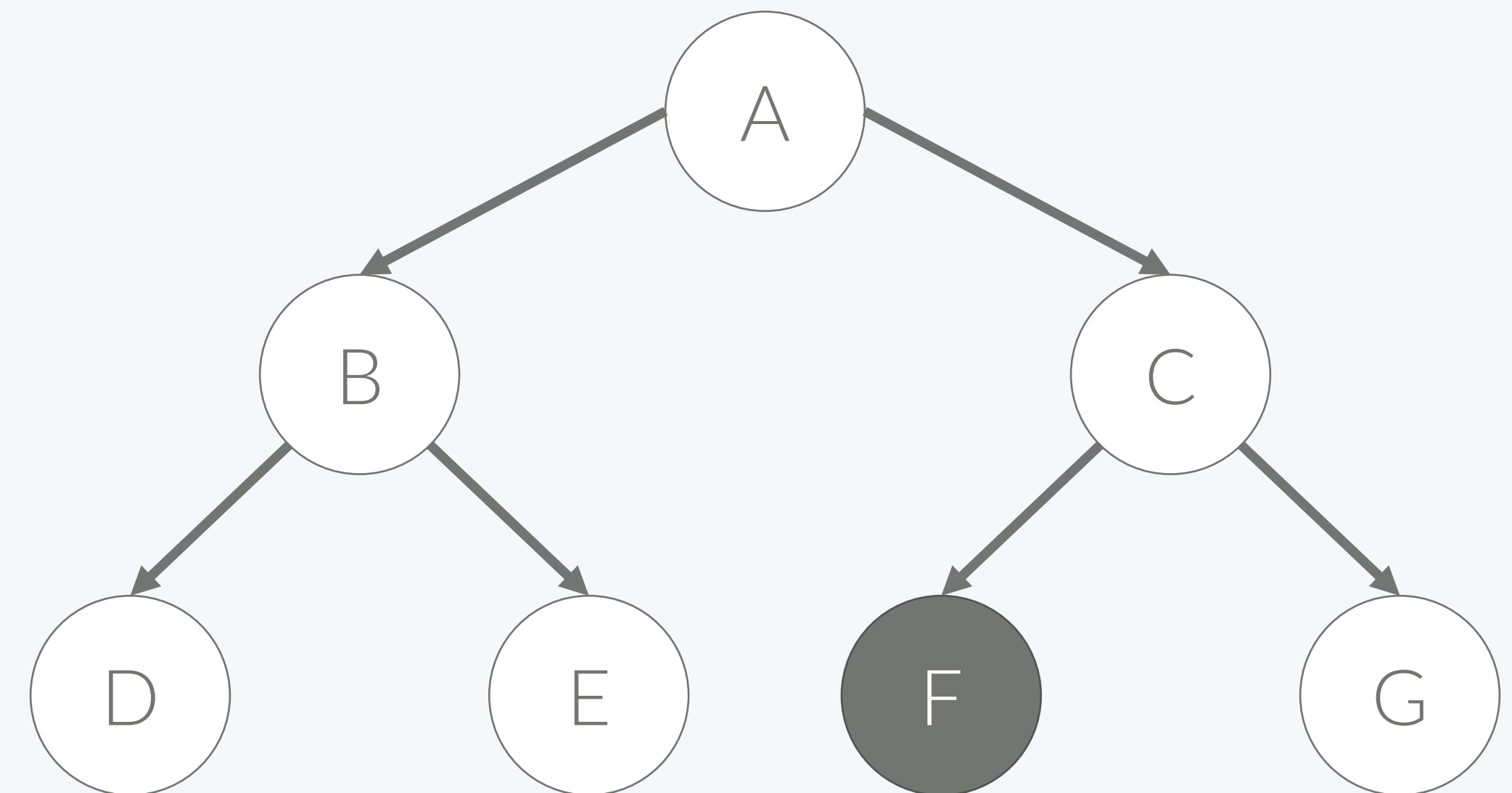


포스트오더

58

Postorder

- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - F
 - 오른쪽 자식 포스트오더
 - 노드 방문
- 노드 방문

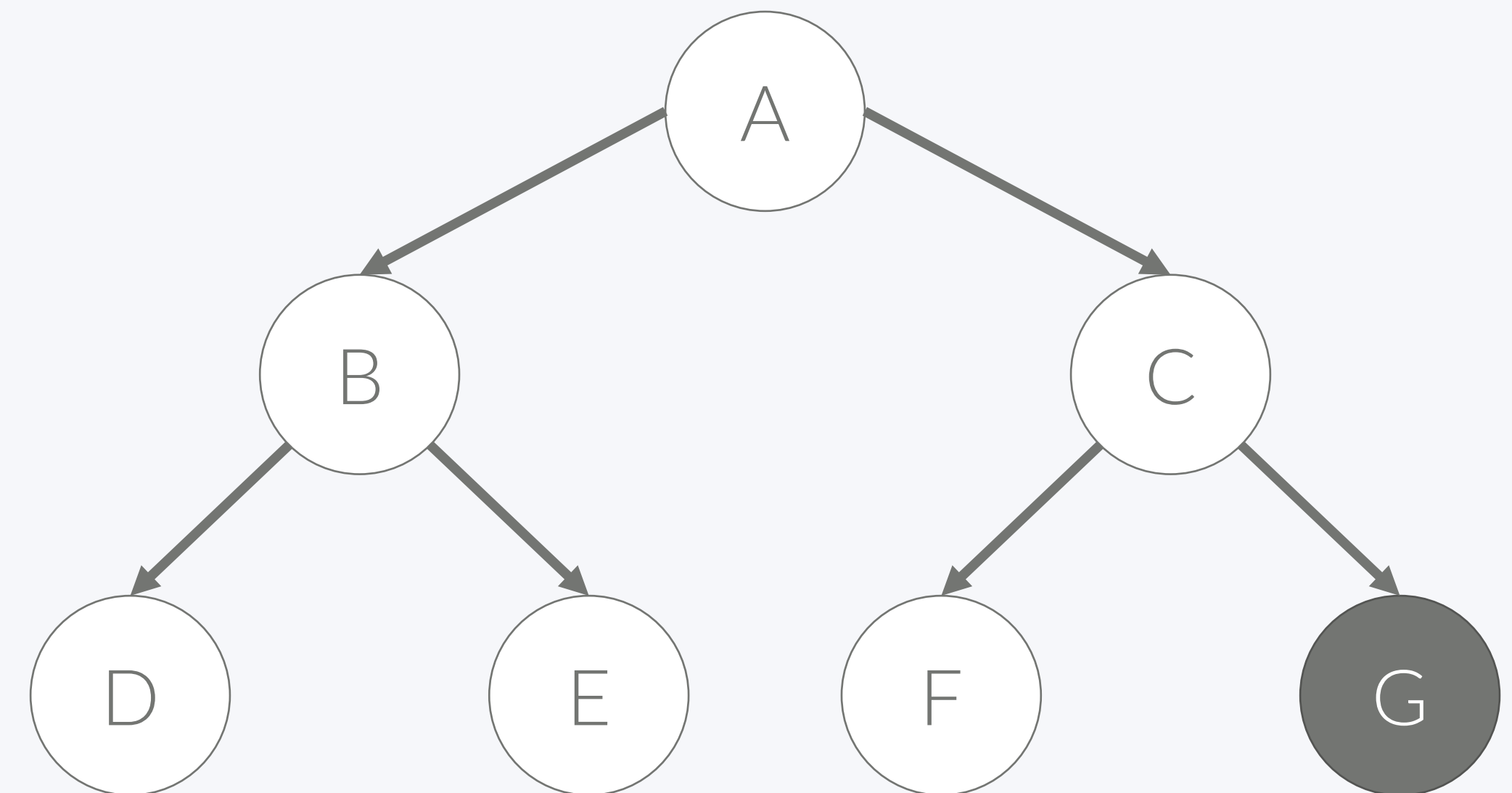


포스트오더

59

Postorder

- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - F
 - 오른쪽 자식 포스트오더
 - G
 - 노드 방문
- 노드 방문

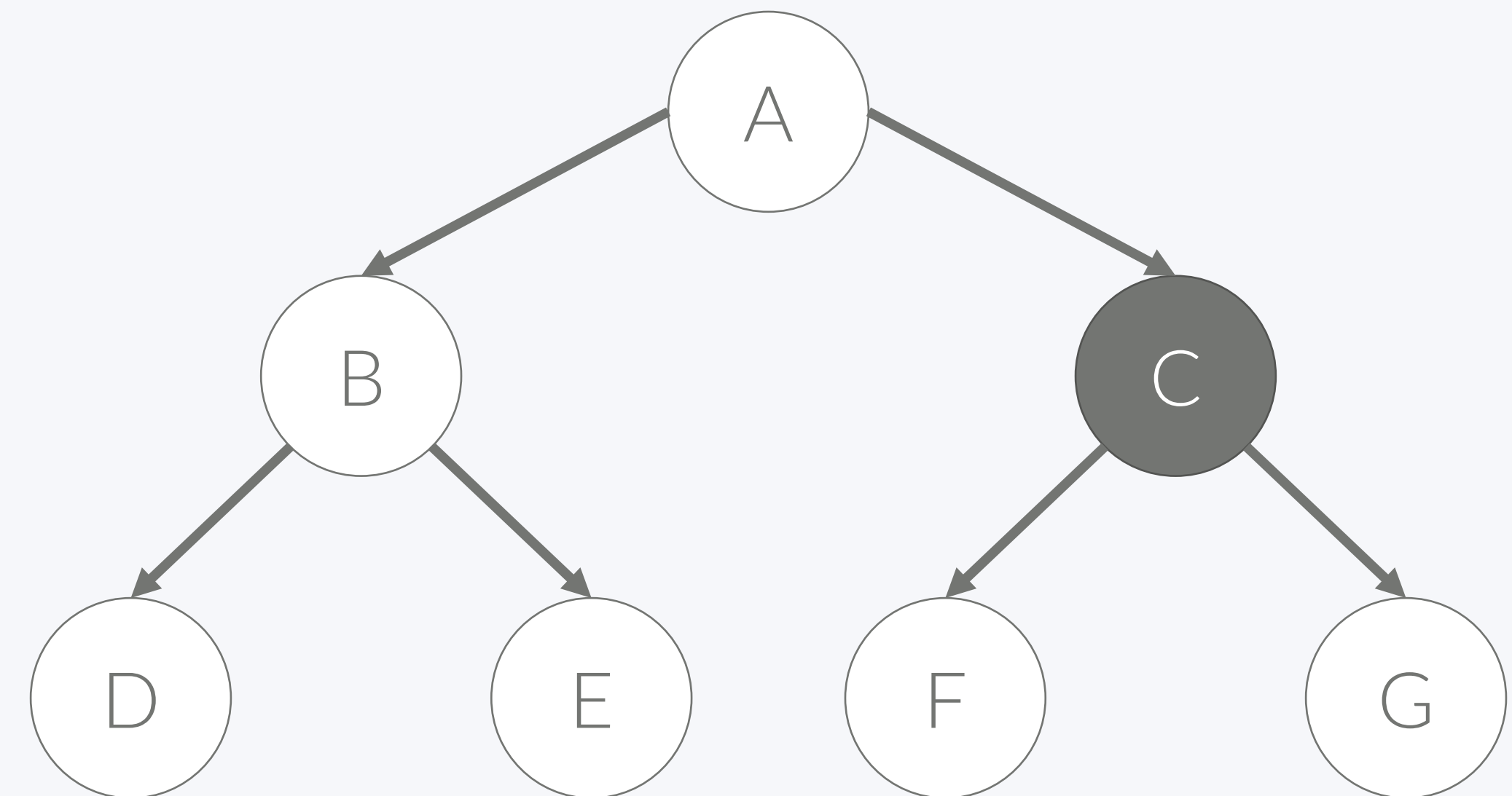


포스트오더

60

Postorder

- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - F
 - 오른쪽 자식 포스트오더
 - G
 - 노드 방문
 - C
- 노드 방문

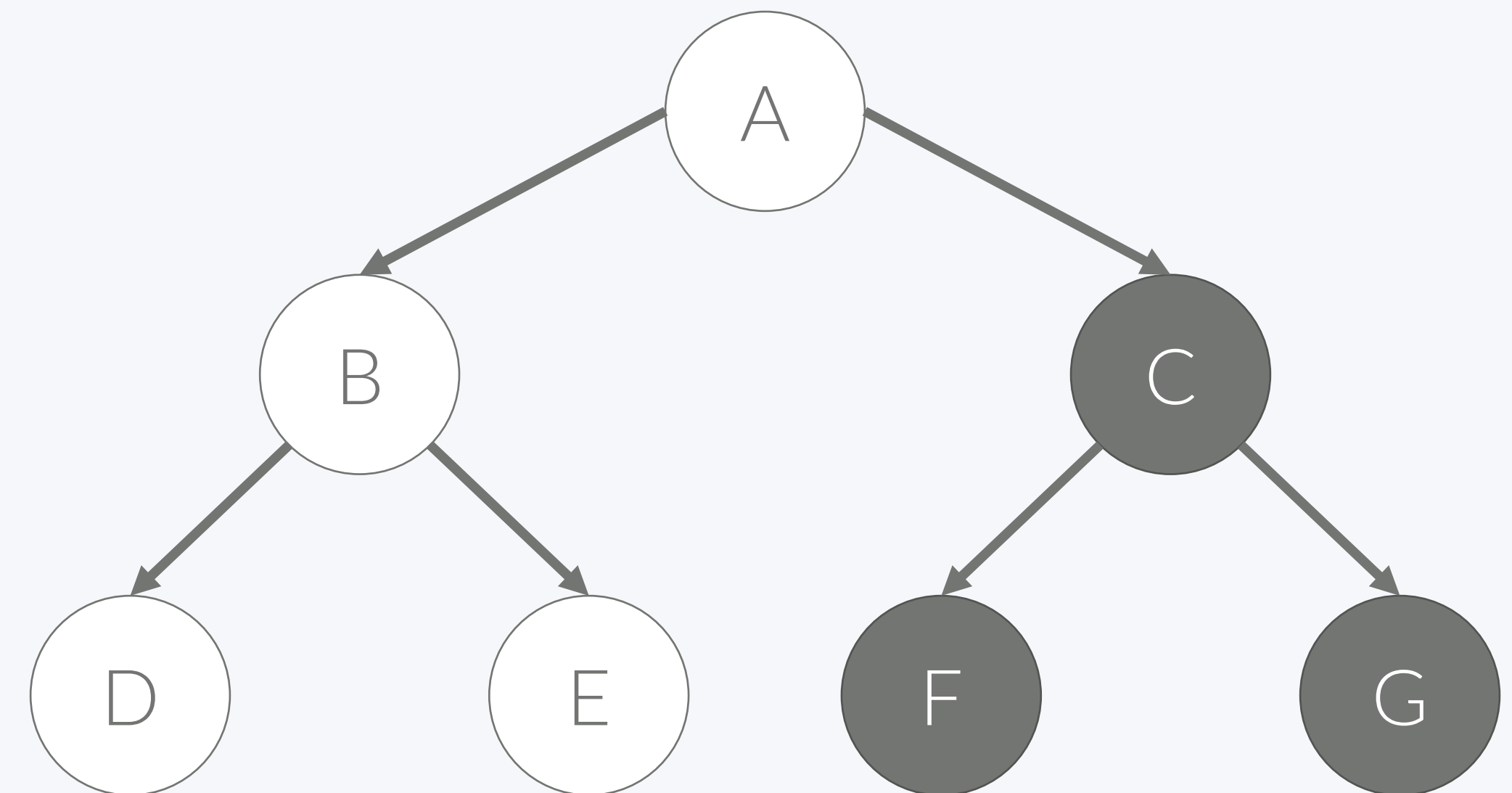


포스트오더

61

Postorder

- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
 - FGC
- 노드 방문

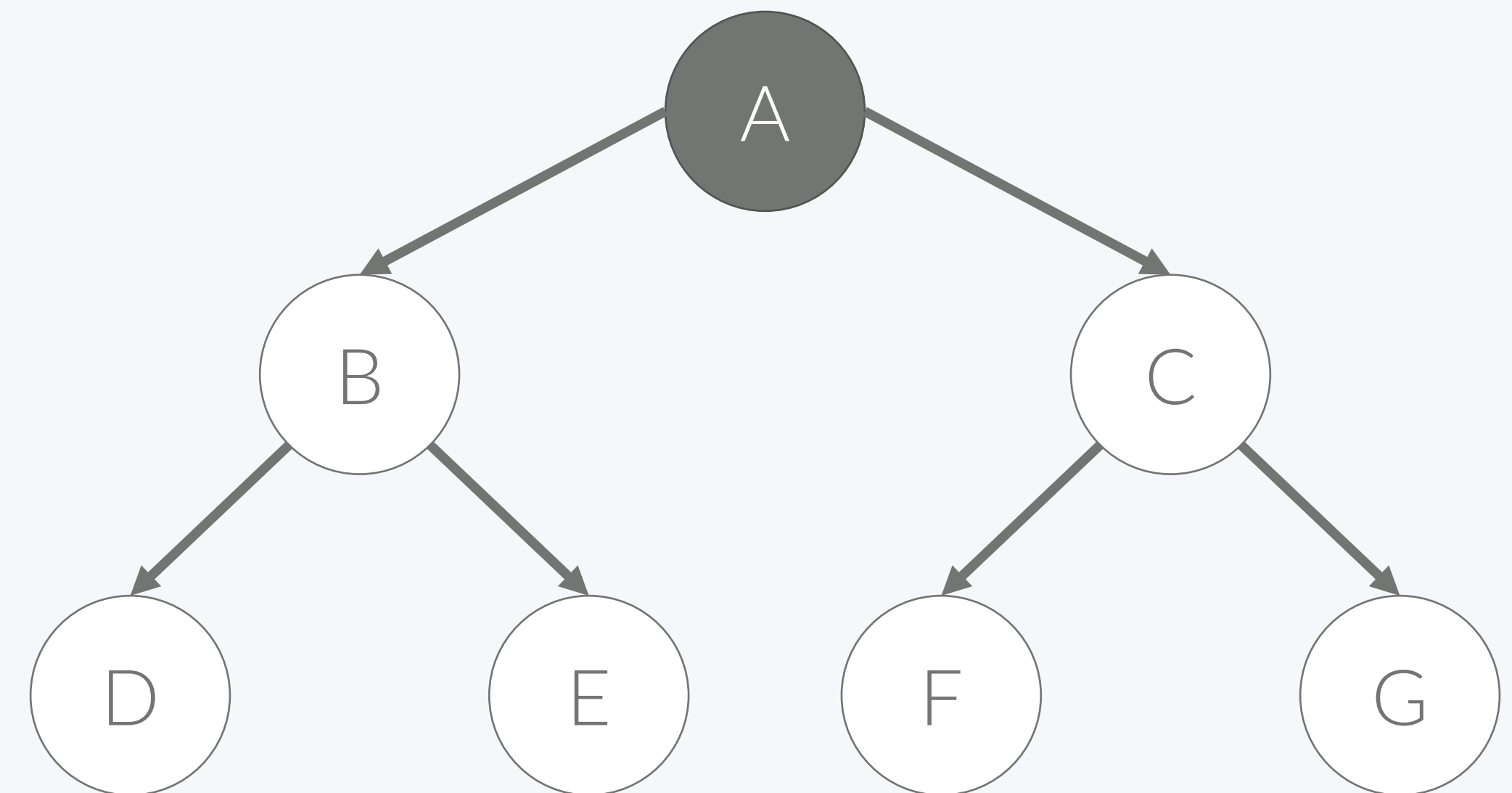


포스트오더

62

Postorder

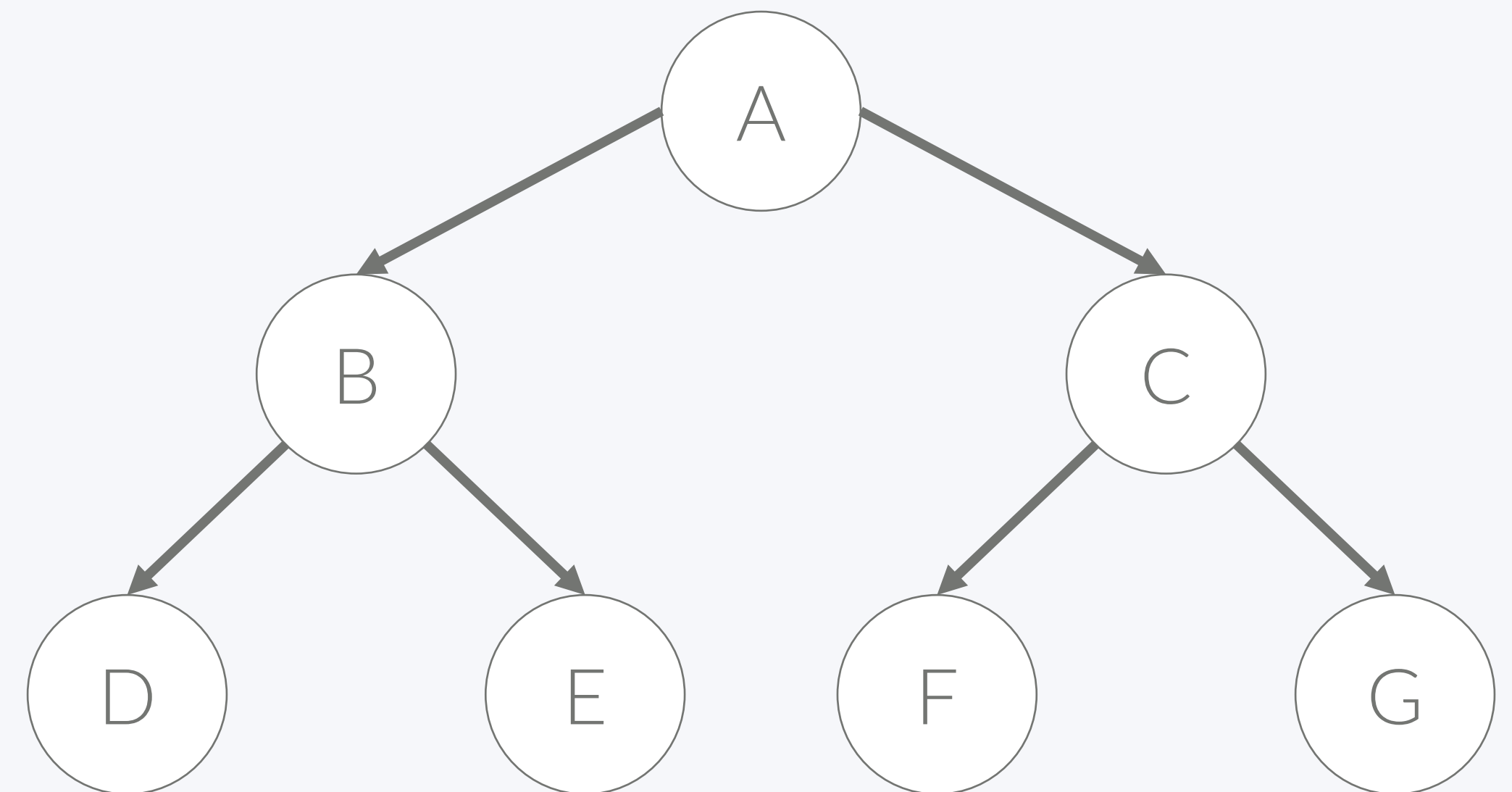
- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
 - FGC
- 노드 방문
 - A



포스트오더

Postorder

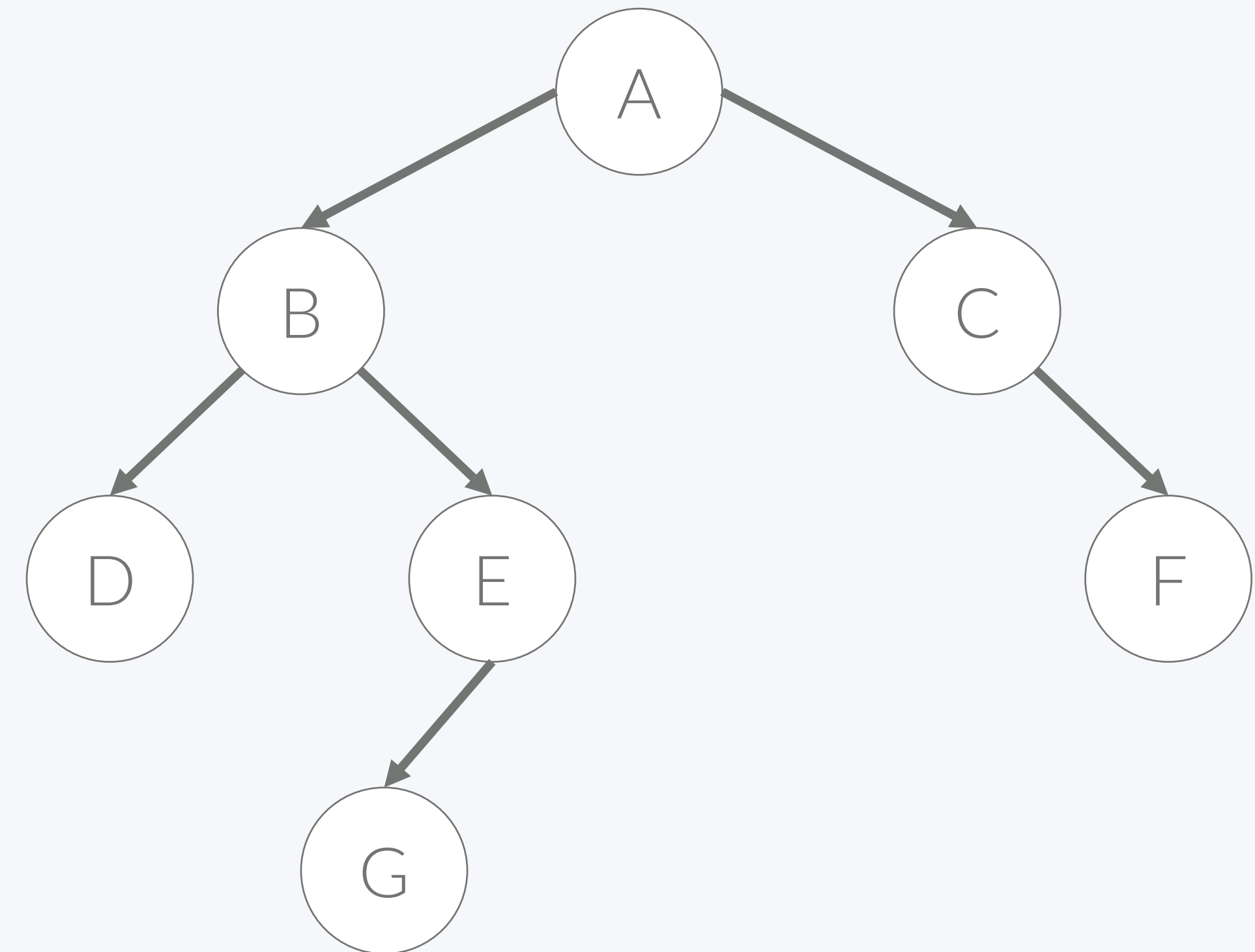
- DEBFGCA



프리오더

Pre-order

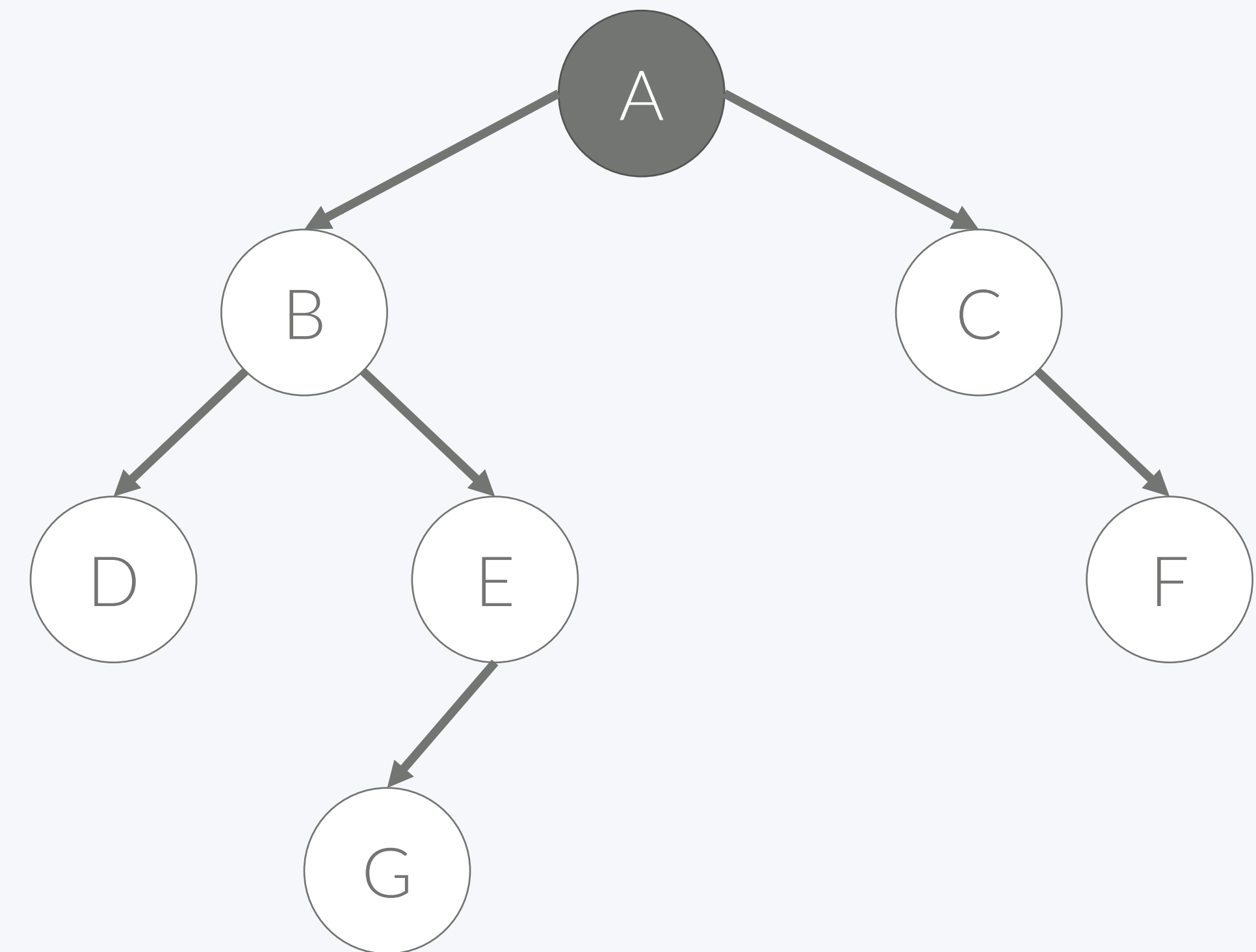
- 노드 방문
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

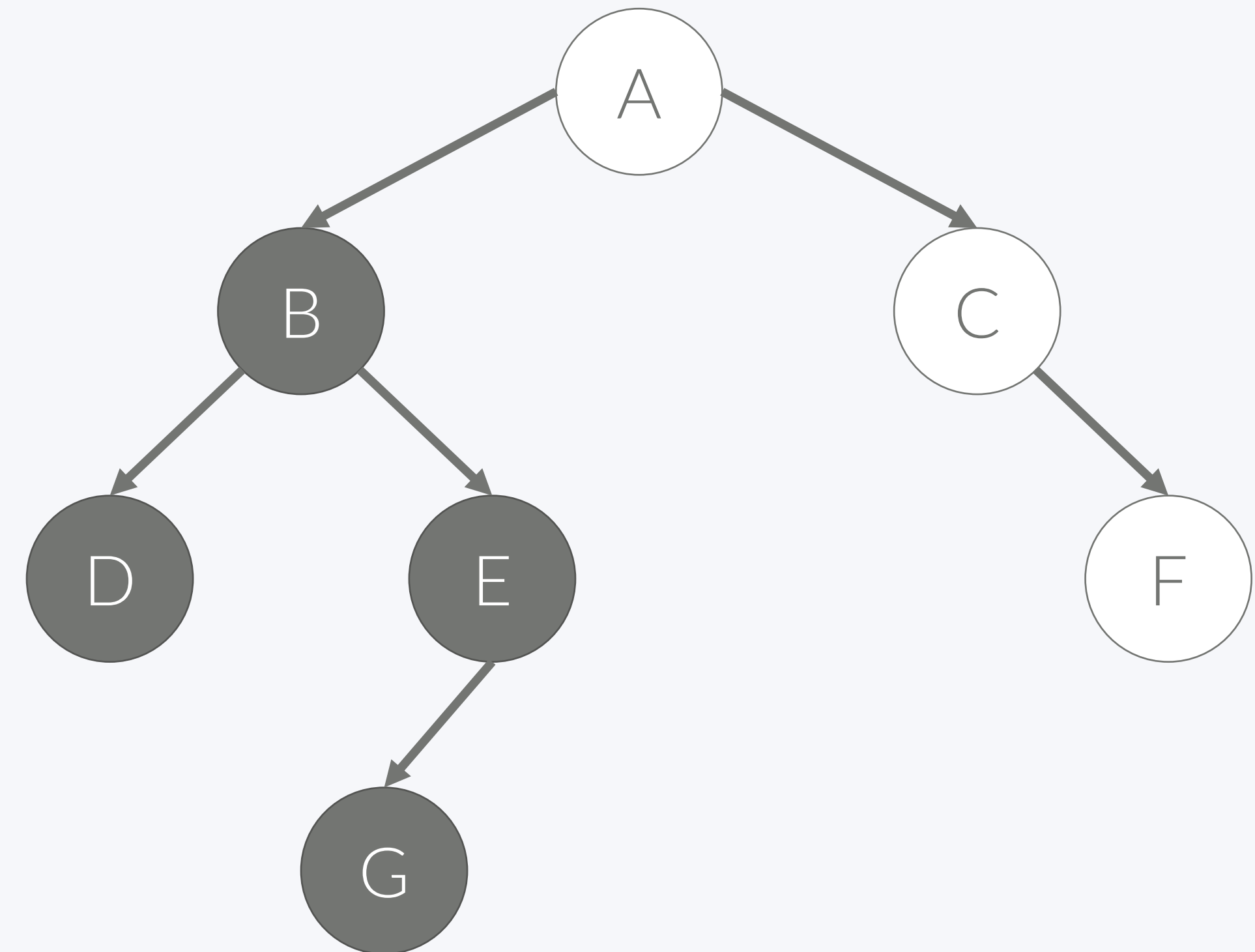
- 노드 방문
 - A
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

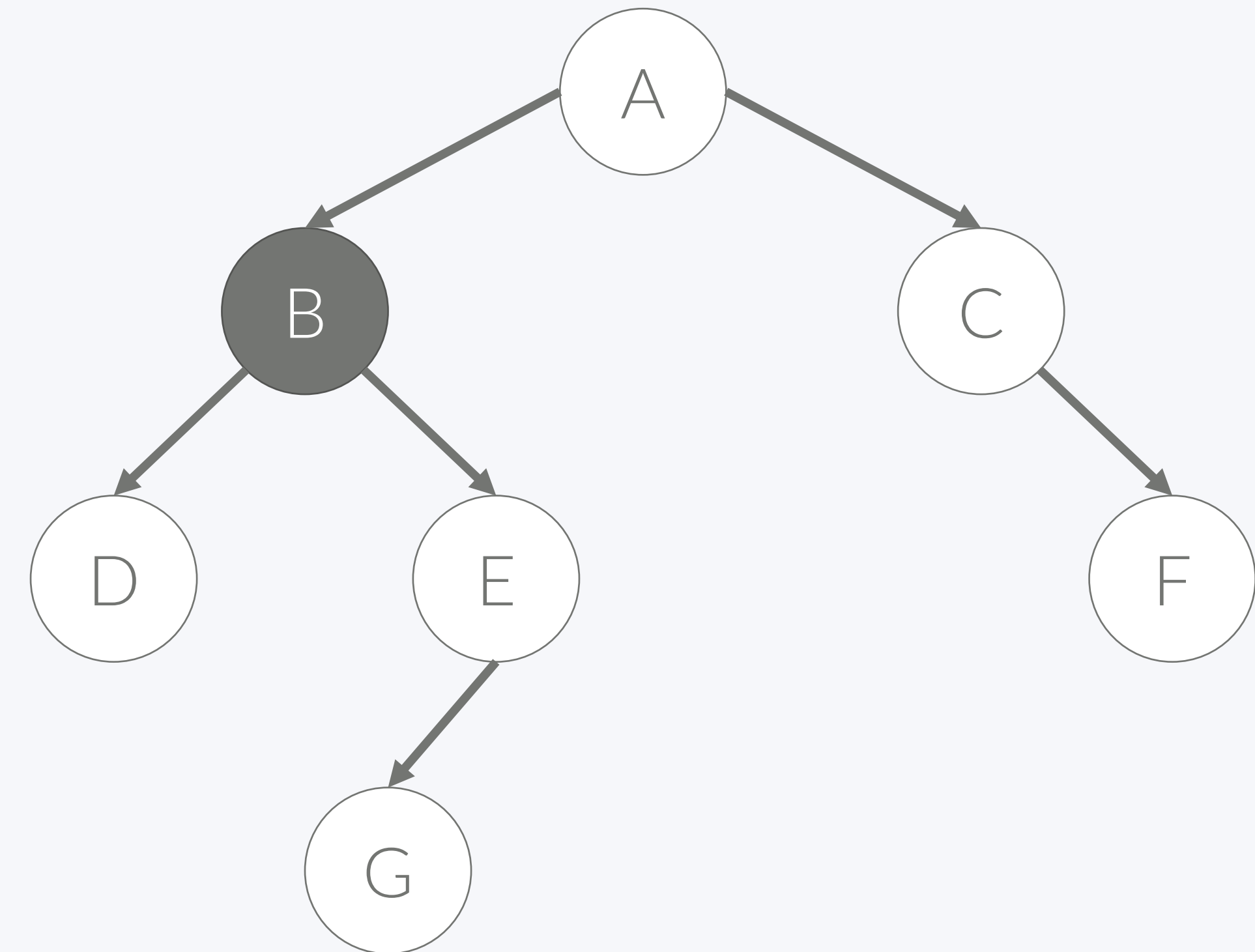
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더

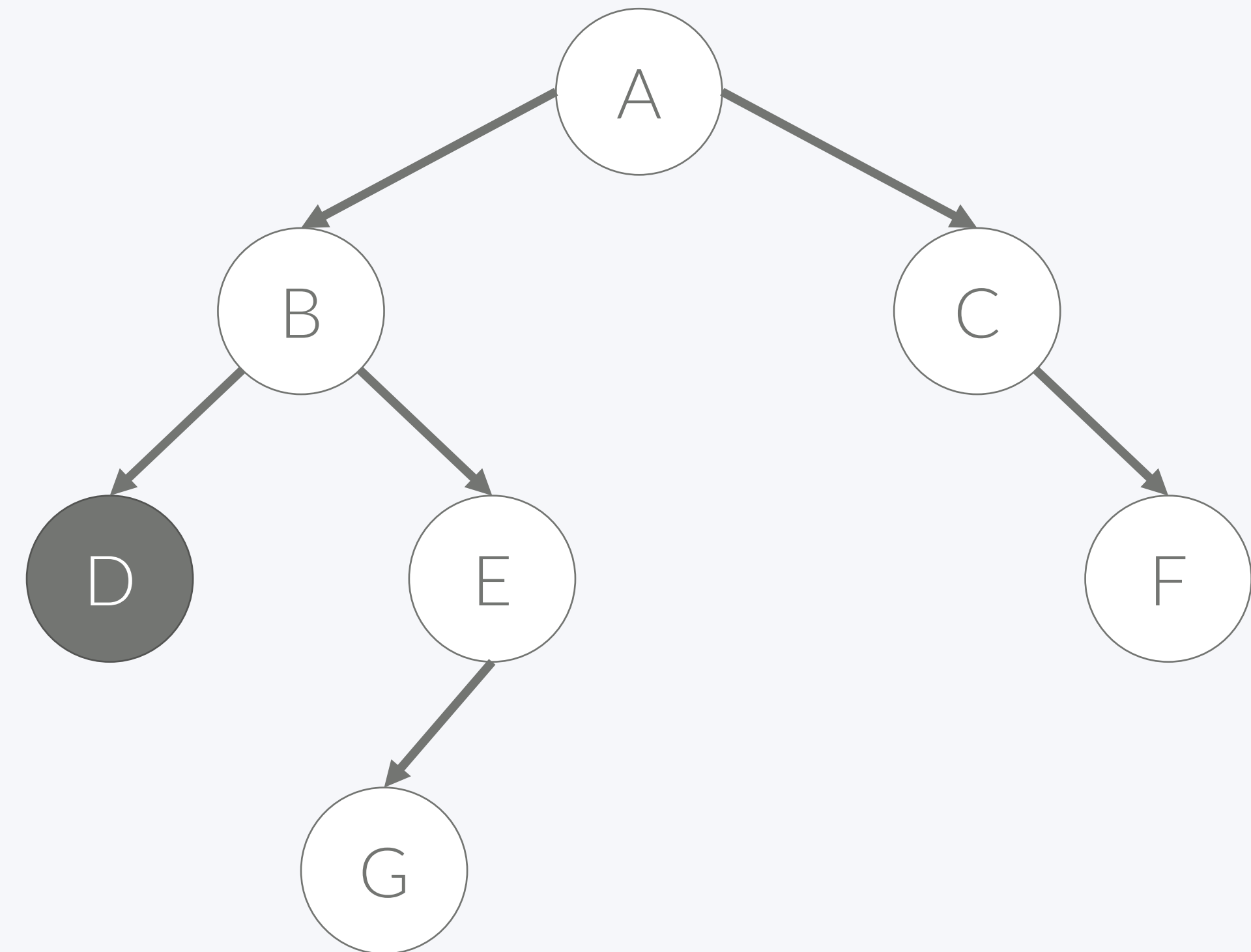


프리오더

68

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더

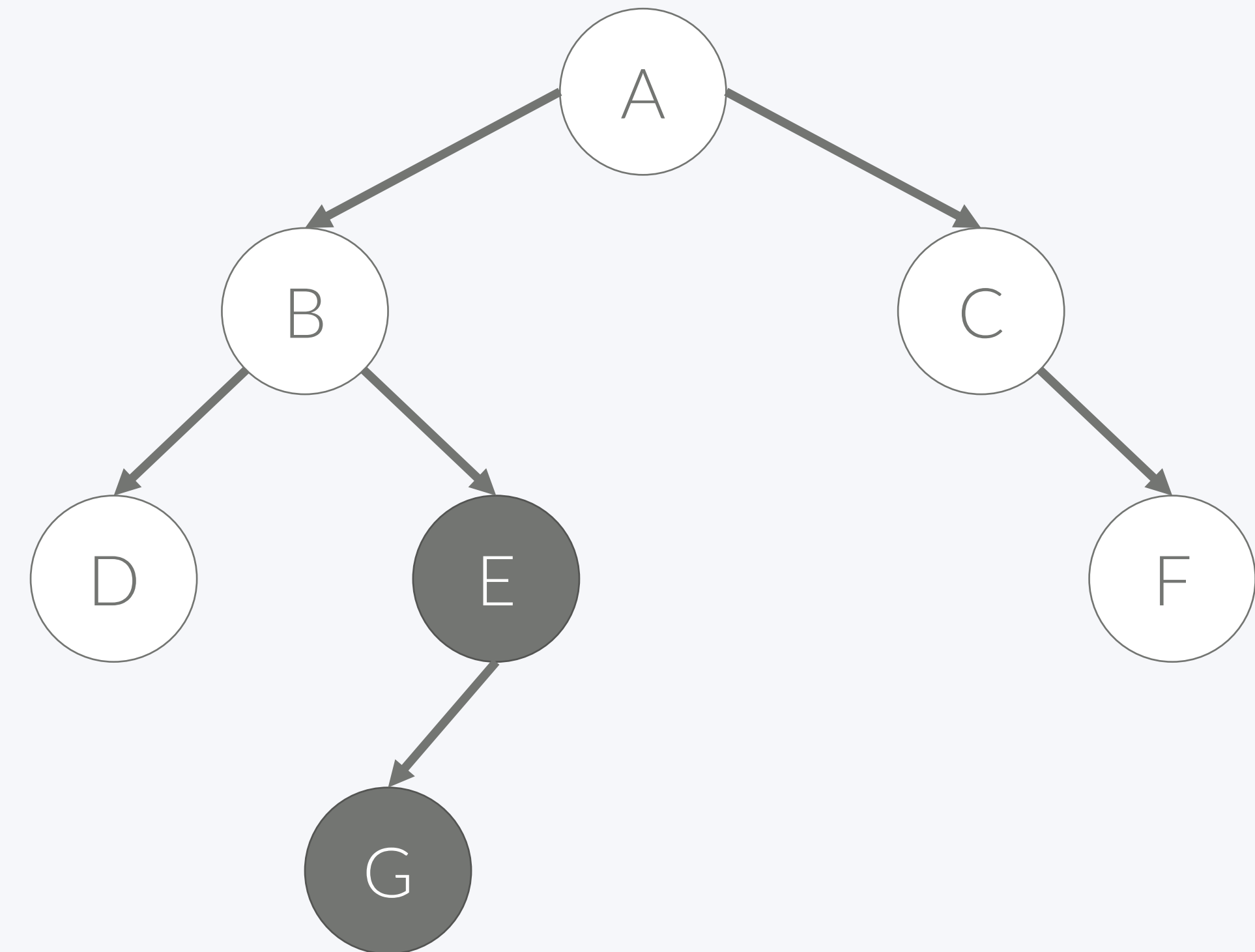


프리오더

69

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
 - 오른쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더

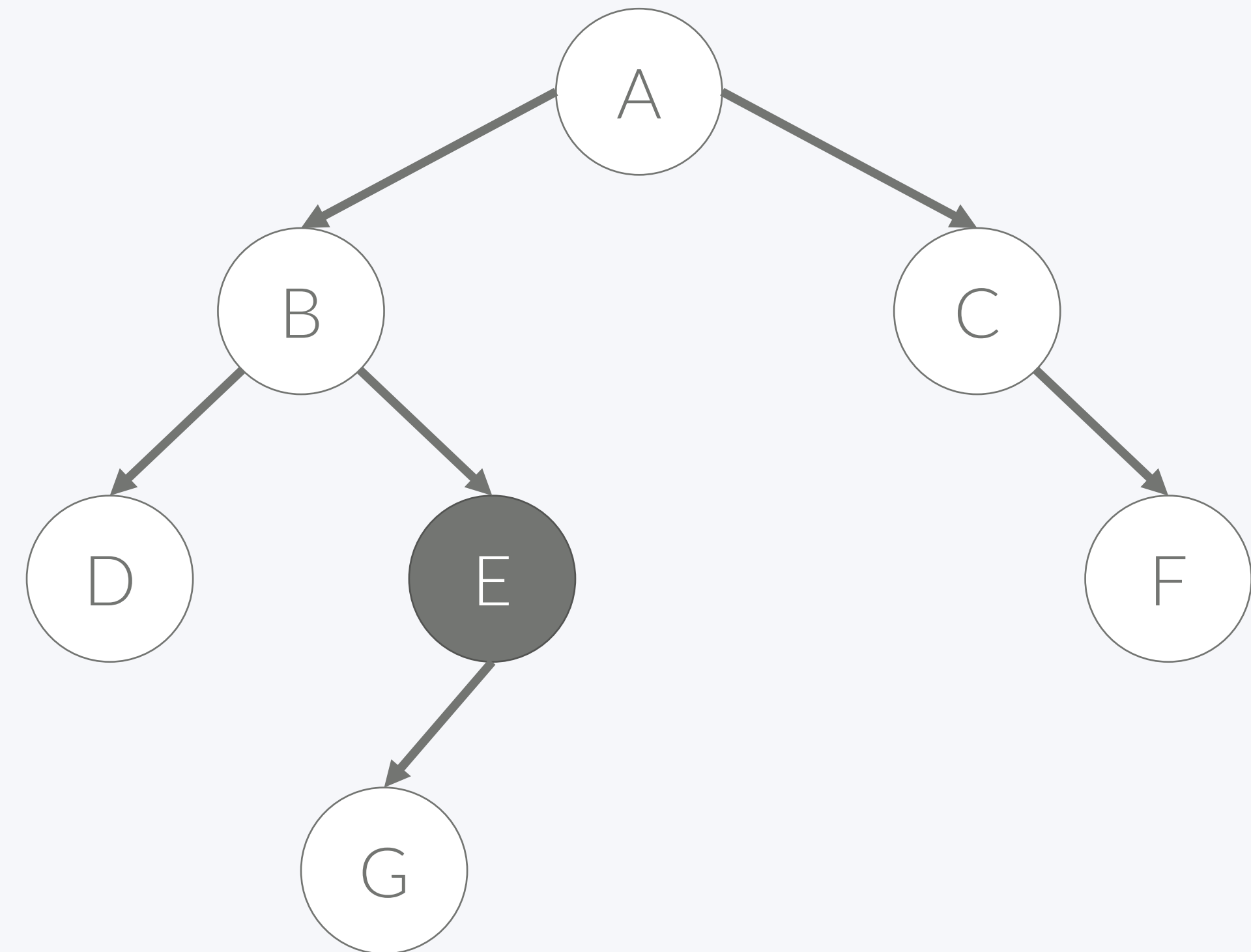


프리오더

70

Pre-order

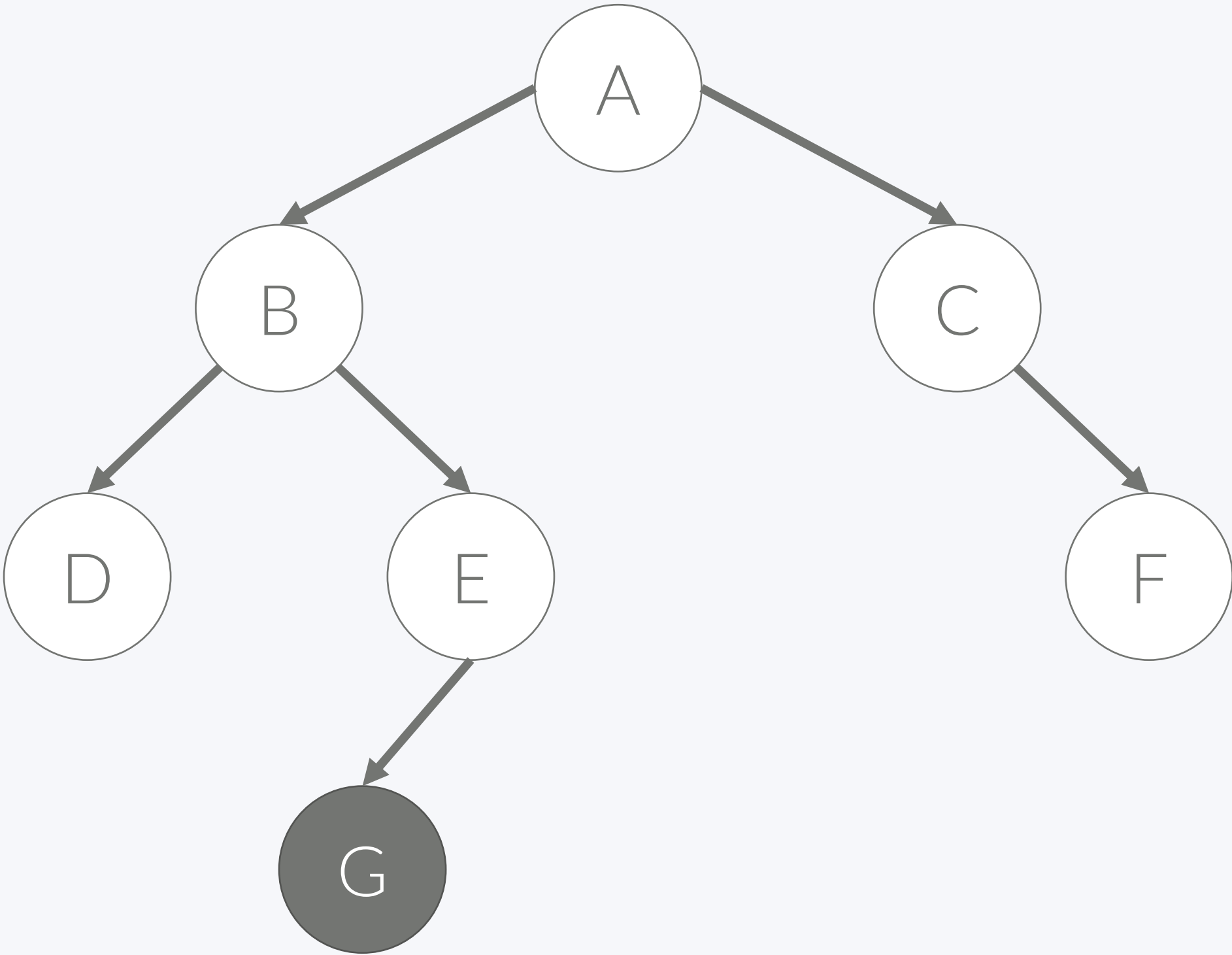
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
- 오른쪽 자식 프리오더
 - 노드 방문
 - E
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더



프리오더

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
- 오른쪽 자식 프리오더
 - 노드 방문
 - E
 - 왼쪽 자식 프리오더
 - G

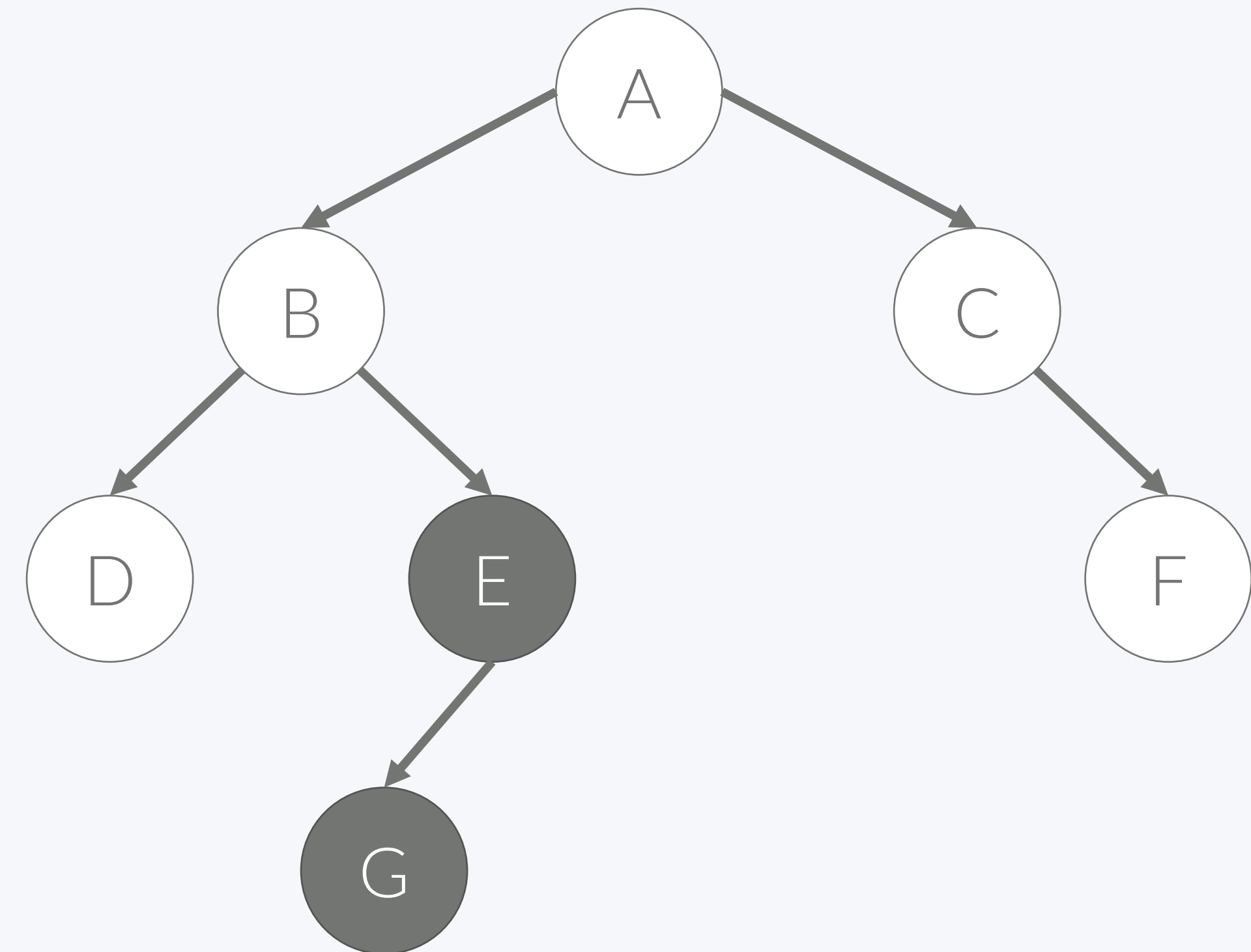


프리오더

72

Pre-order

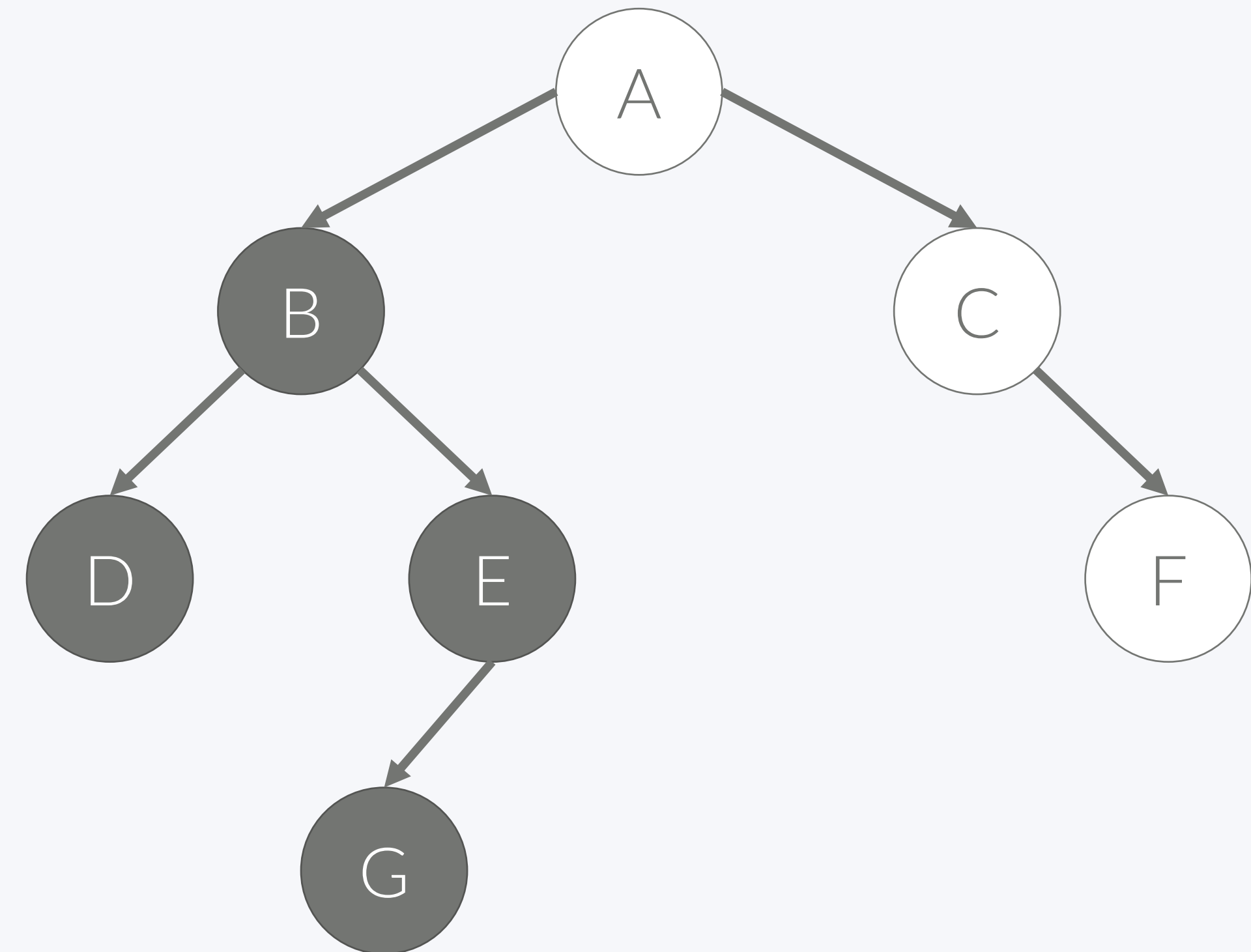
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
 - 오른쪽 자식 프리오더
 - EG
- 오른쪽 자식 프리오더



프리오더

Pre-order

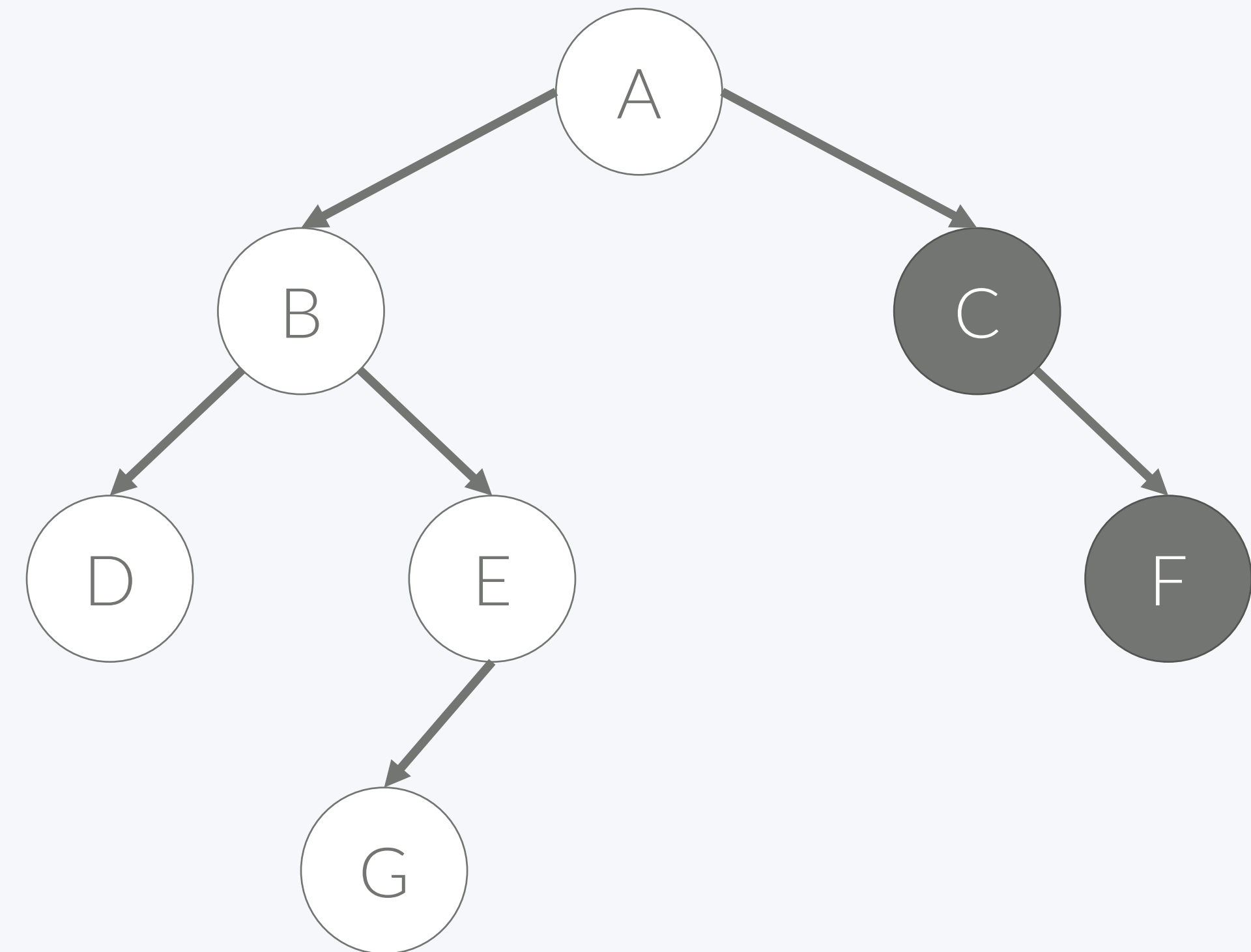
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더



프리오더

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더

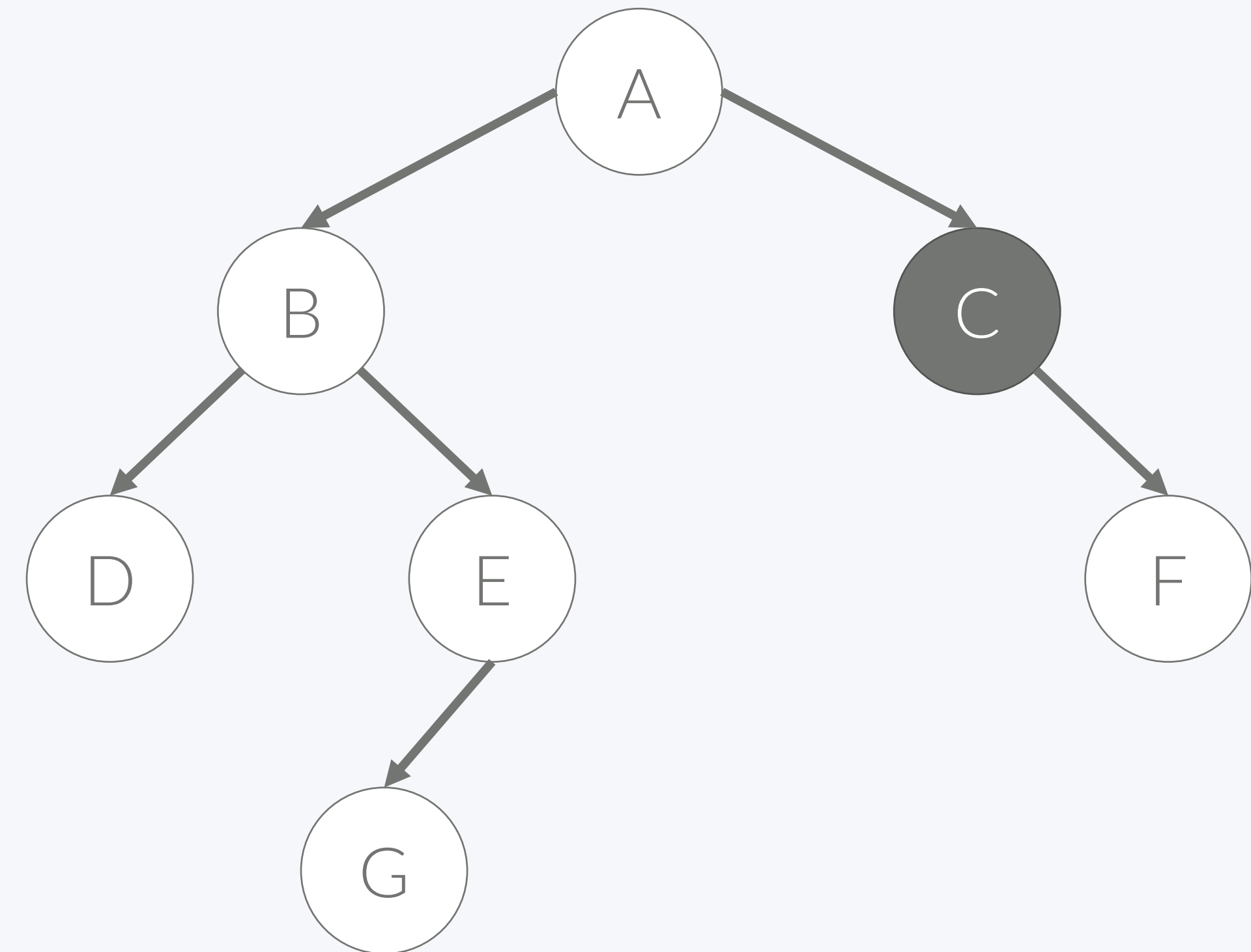


프리오더

75

Pre-order

- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더

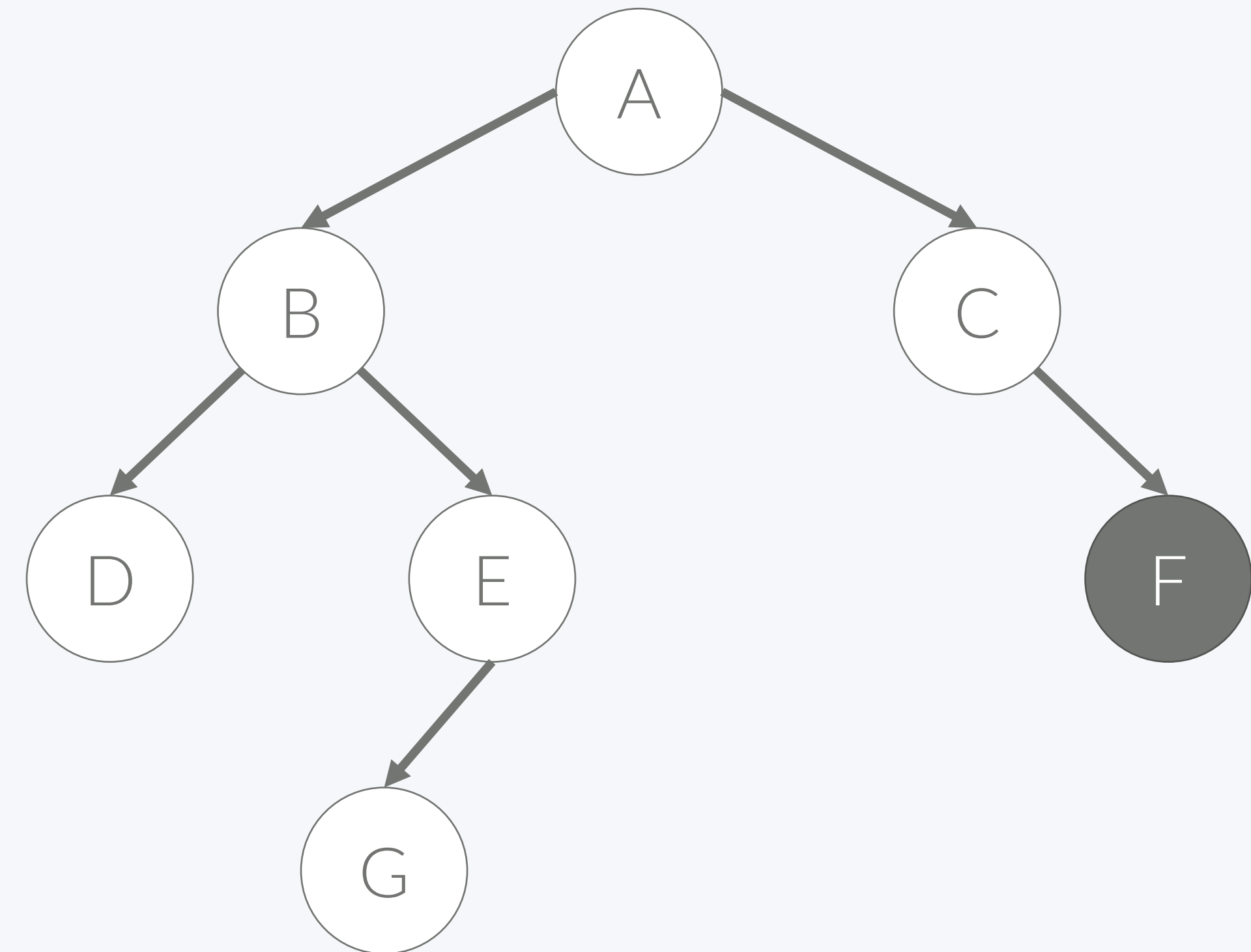


프리오더

76

Pre-order

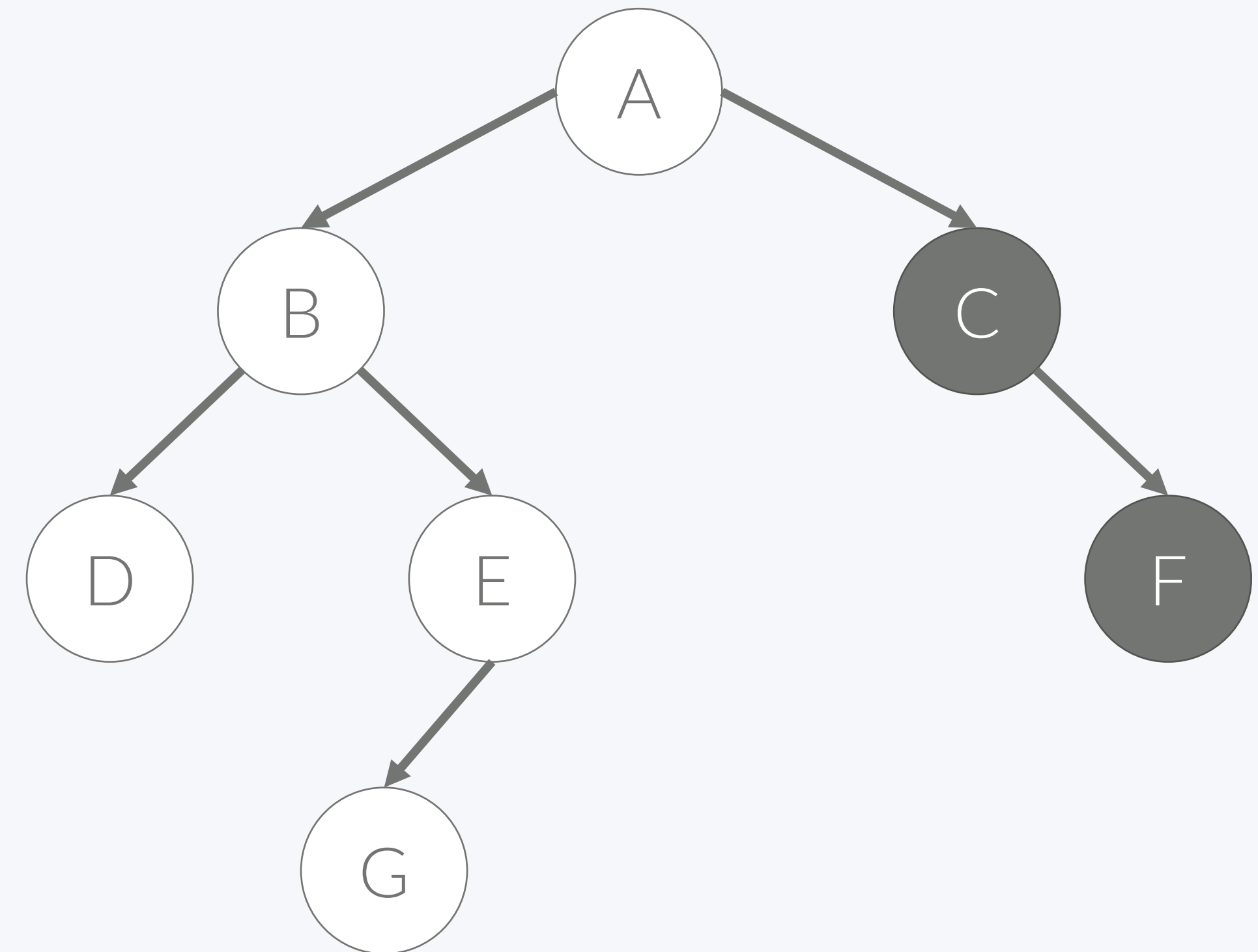
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
 - F



프리오더

Pre-order

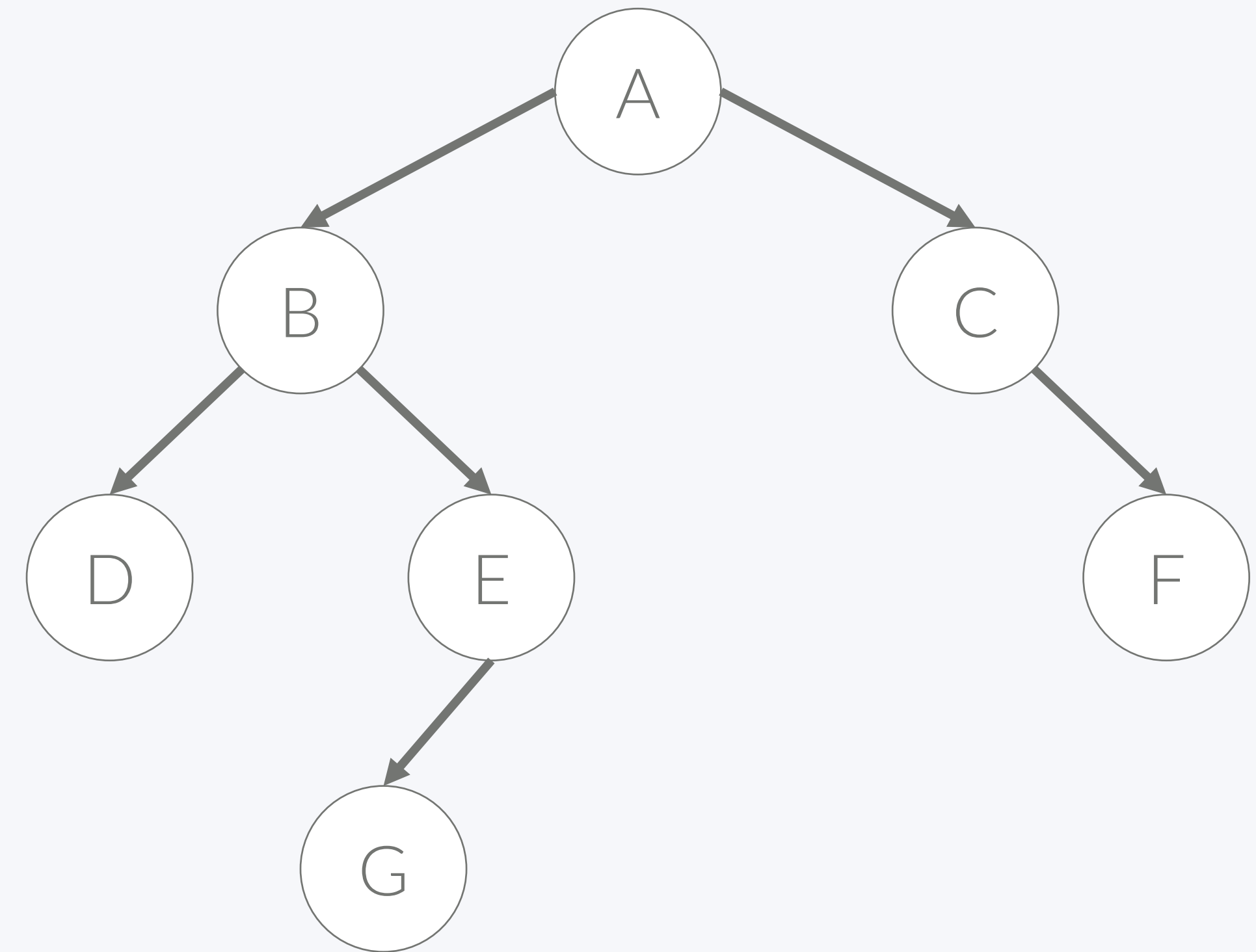
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더
 - CF



프리오더

Pre-order

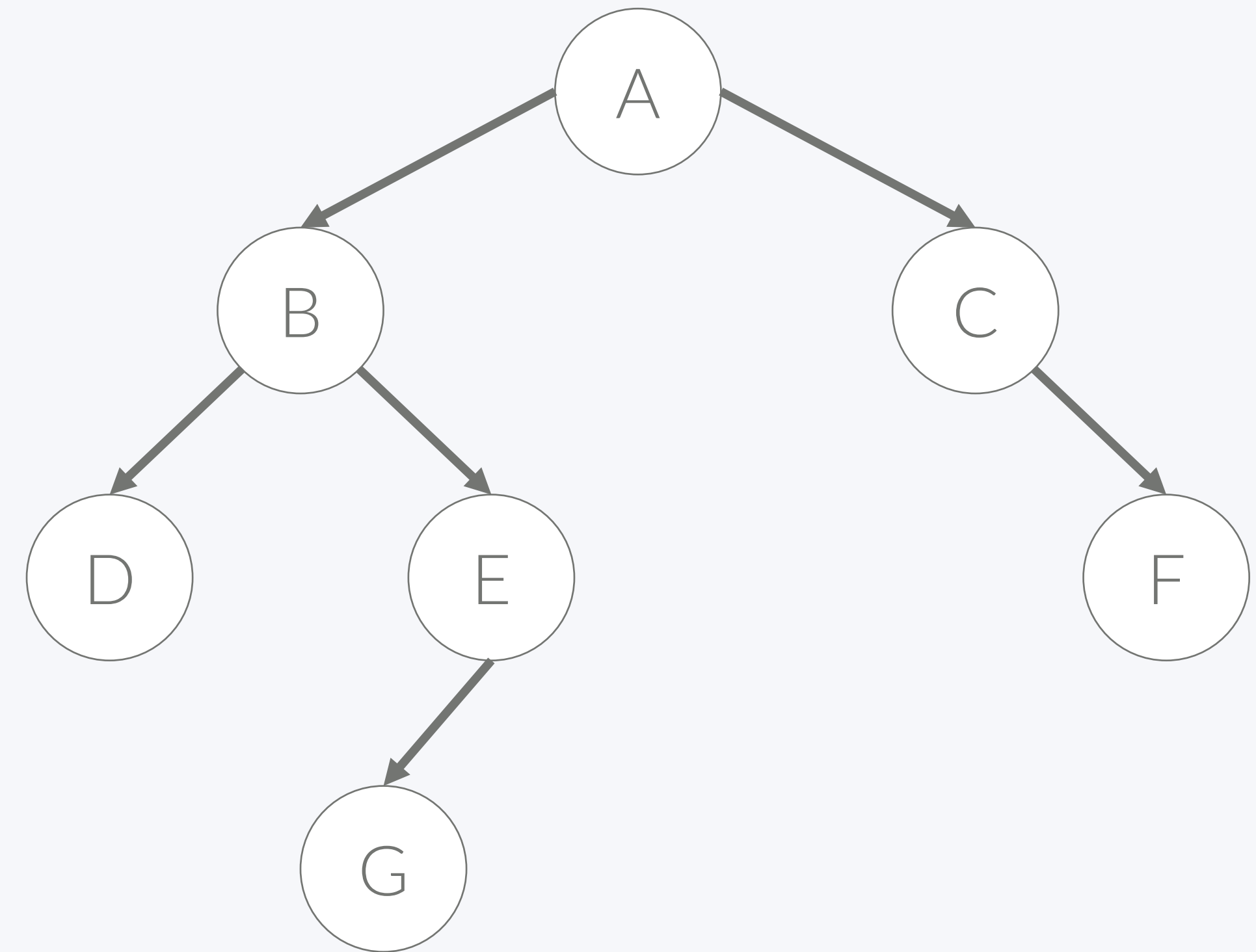
- ABDEGCF



인오더

In-order

- 왼쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

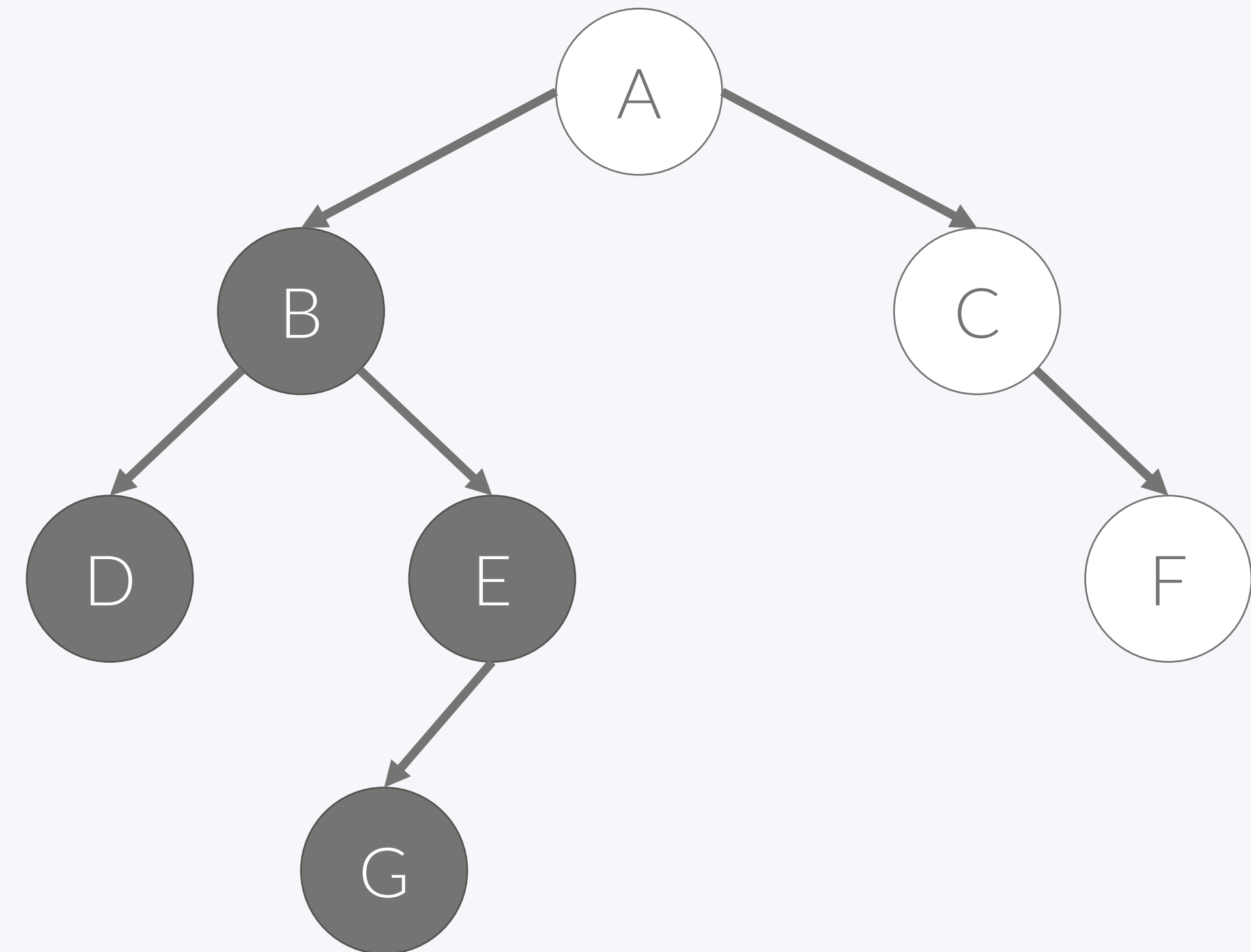


인오더

In-order

80

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - 노드 방문
 - 오른쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

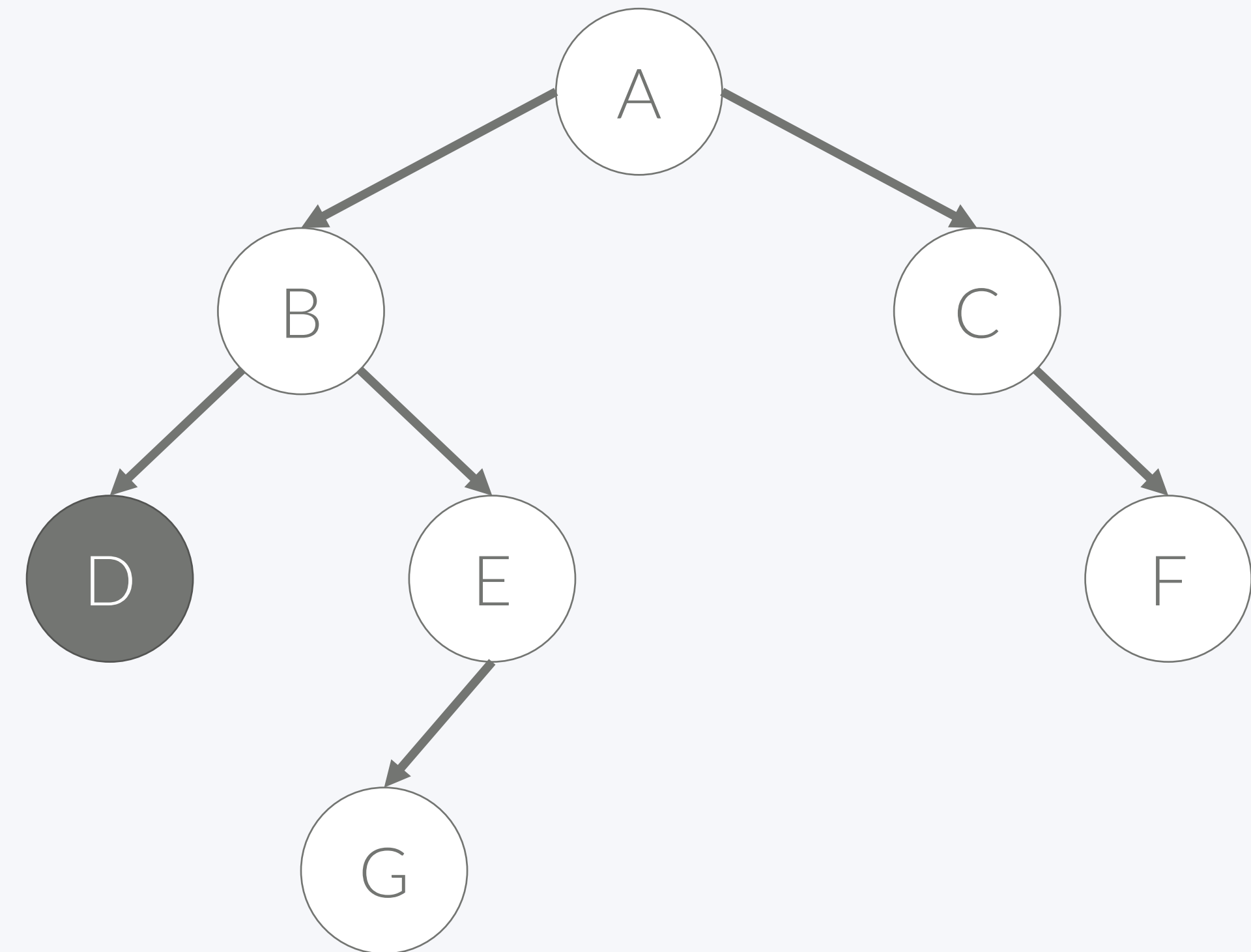


인오더

In-order

81

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
 - 노드 방문
 - 오른쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

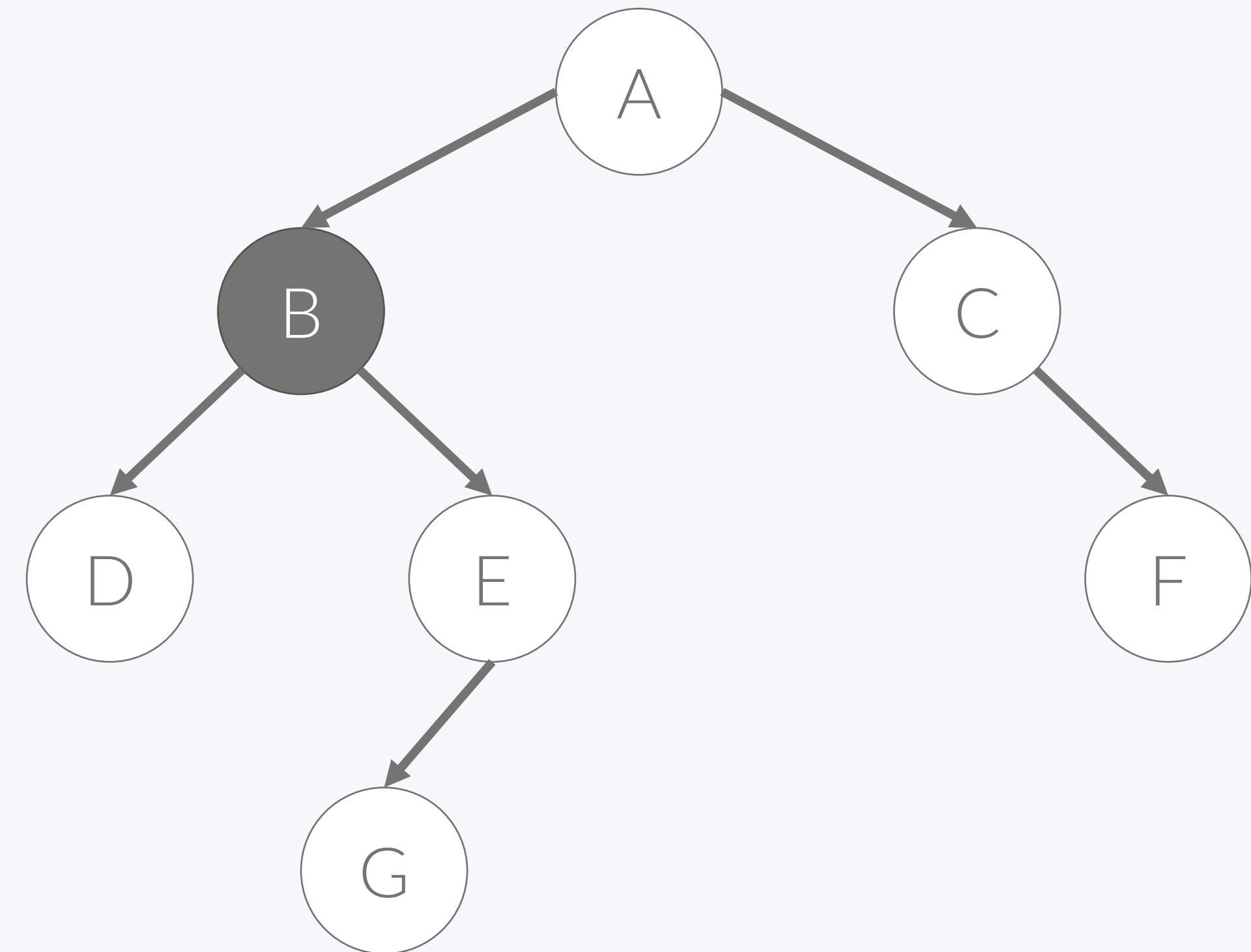


인오더

In-order

82

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
- 노드 방문
 - B
- 오른쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

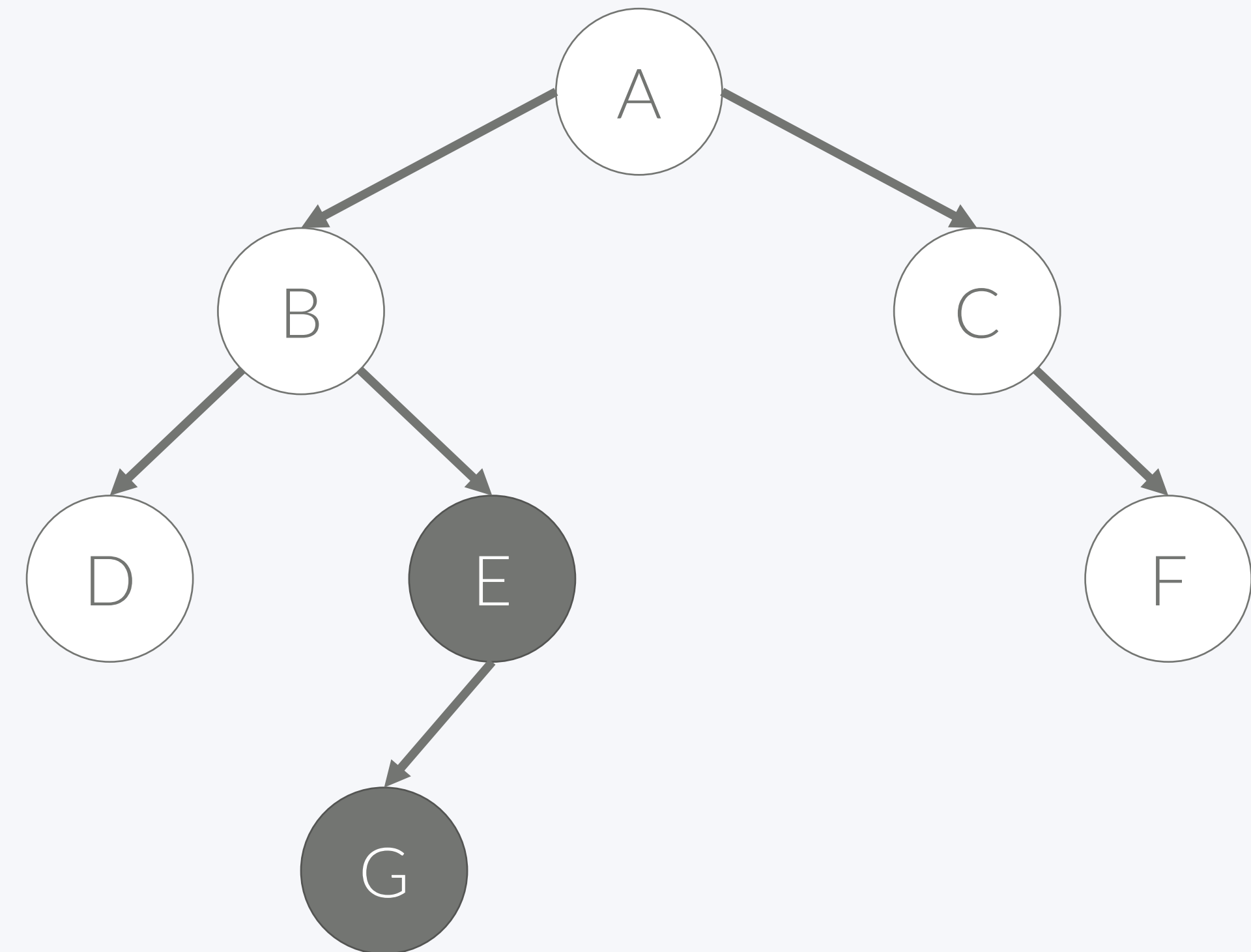


인오더

In-order

83

- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
 - 노드 방문
 - B
 - 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - 노드 방문
 - 오른쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더



인오더

In-order

84

- 왼쪽 자식 인오더

- 왼쪽 자식 인오더

- D

- 노드 방문

- B

- 오른쪽 자식 인오더

- 왼쪽 자식 인오더

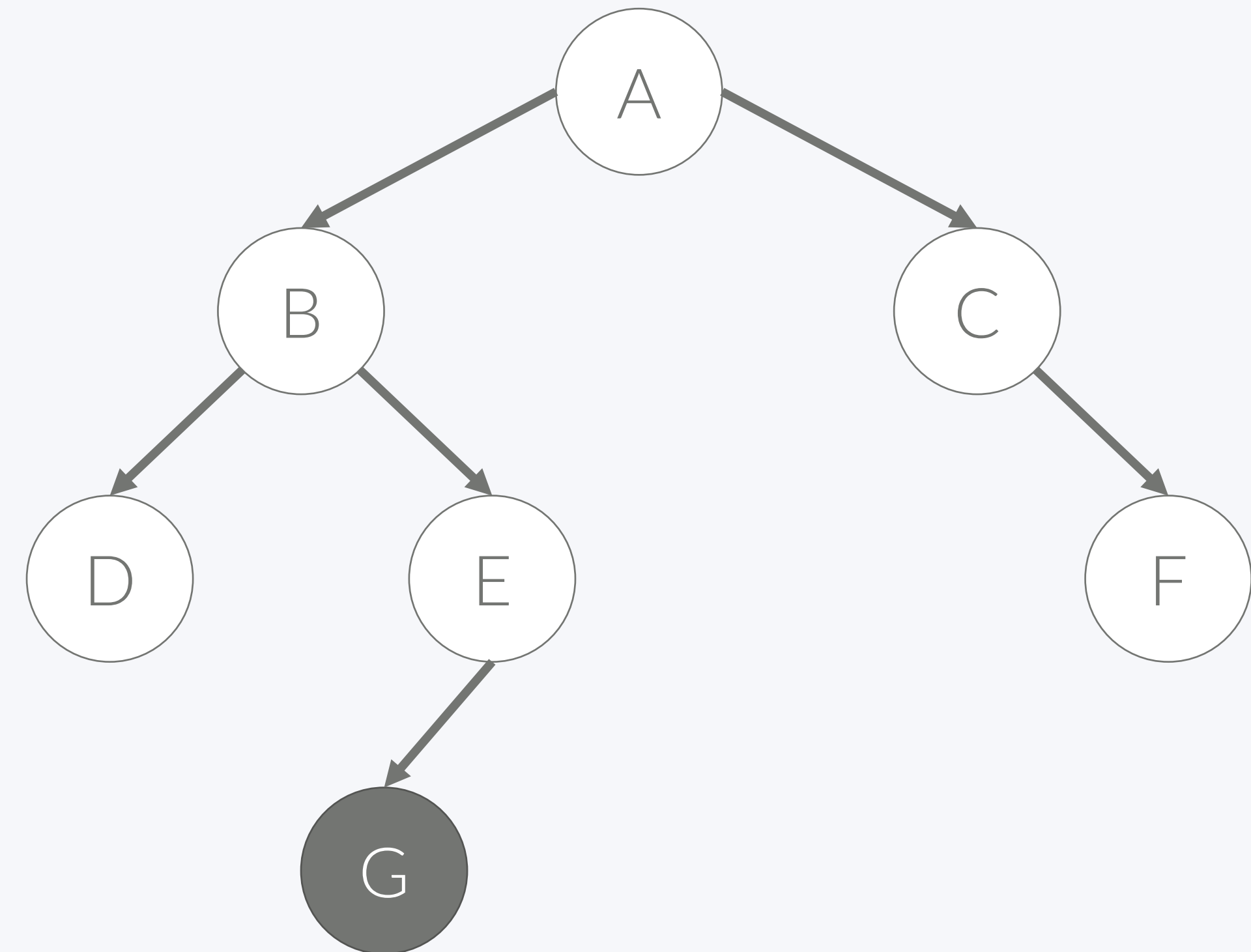
- G

- 노드 방문

- 오른쪽 자식 인오더

- 노드 방문

- 오른쪽 자식 인오더



인오더

In-order

85

- 왼쪽 자식 인오더

- 왼쪽 자식 인오더

- D

- 노드 방문

- B

- 오른쪽 자식 인오더

- 왼쪽 자식 인오더

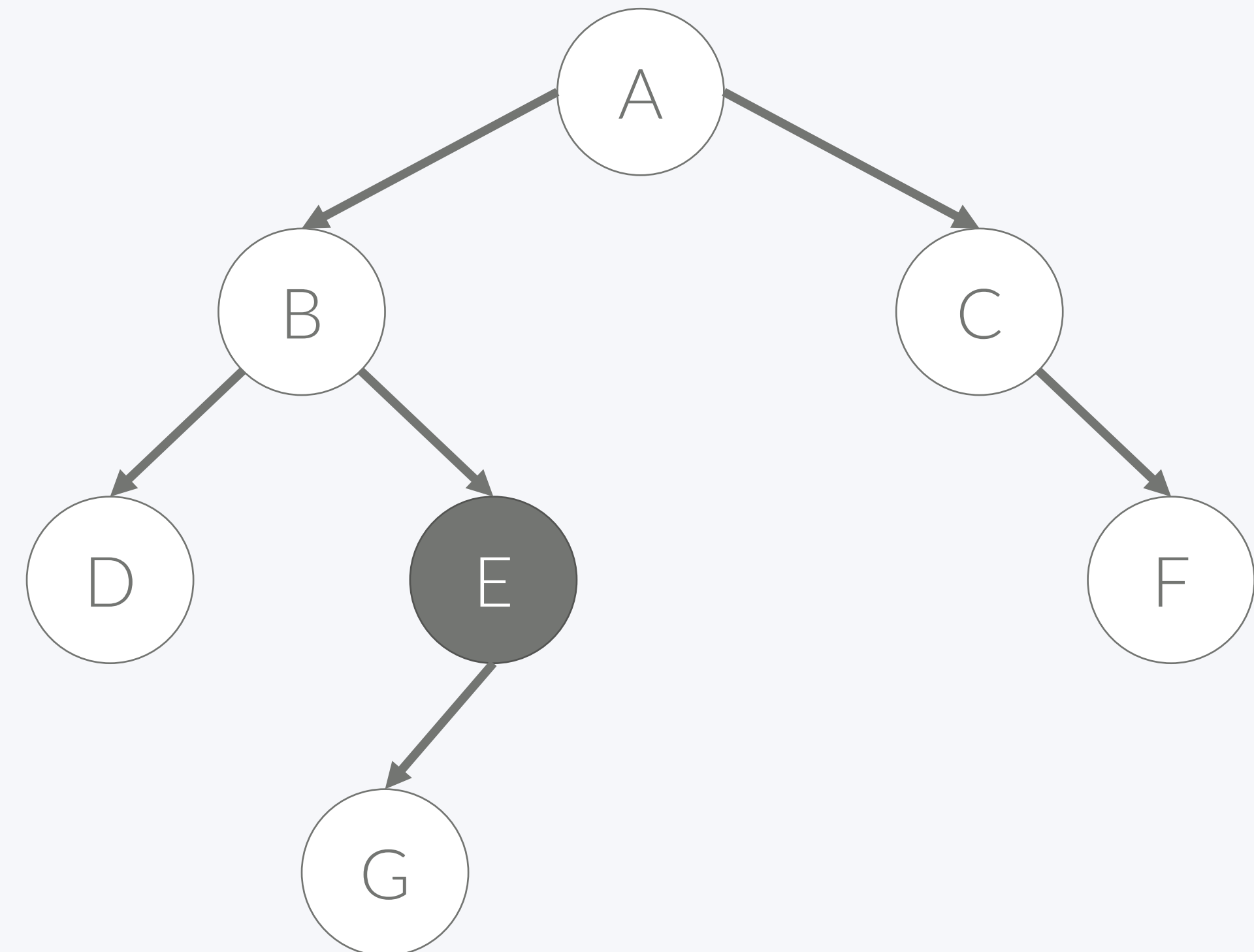
- G

- 노드 방문

- E

- 오른쪽 자식 인오더

- 노드 방문

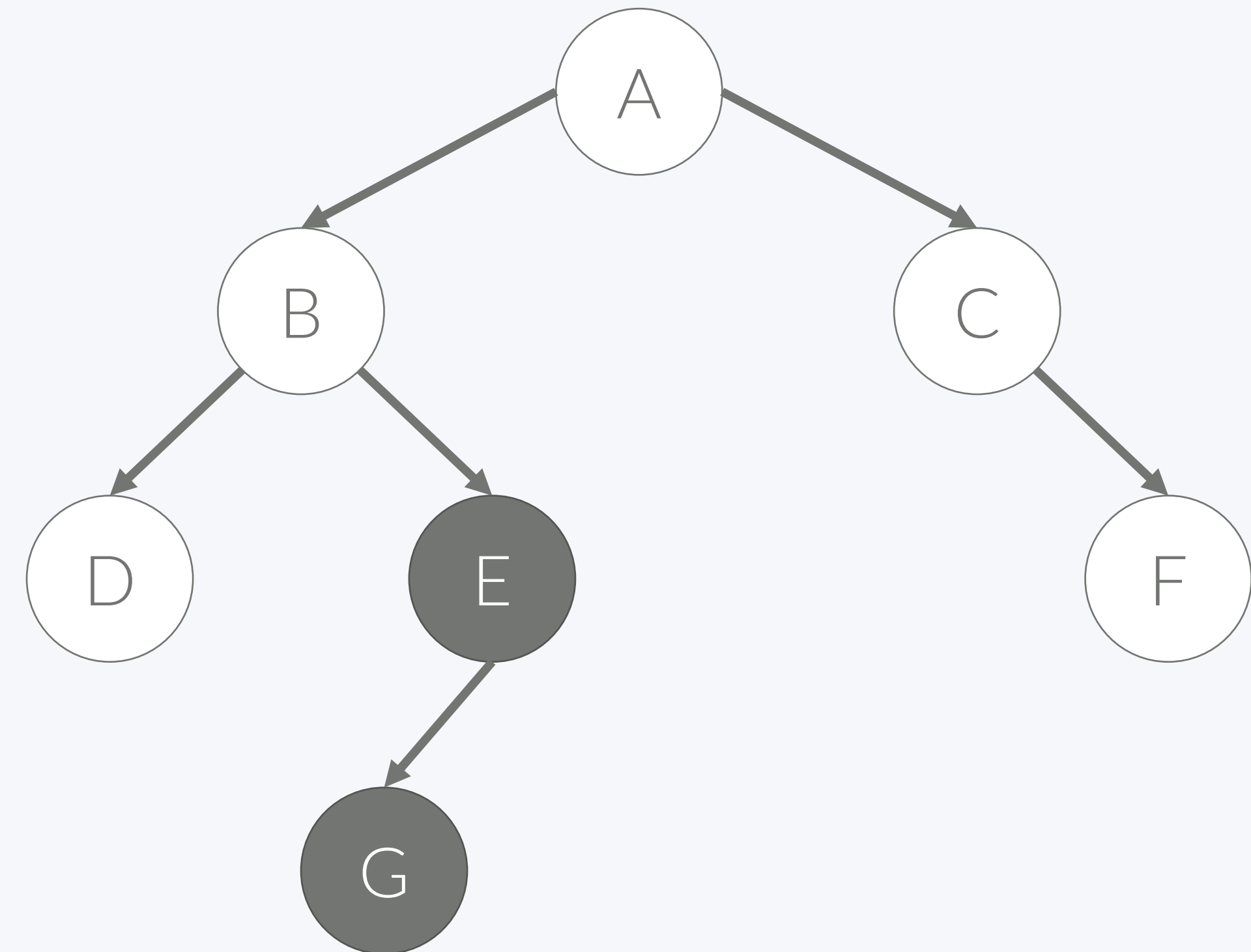


인오더

In-order

86

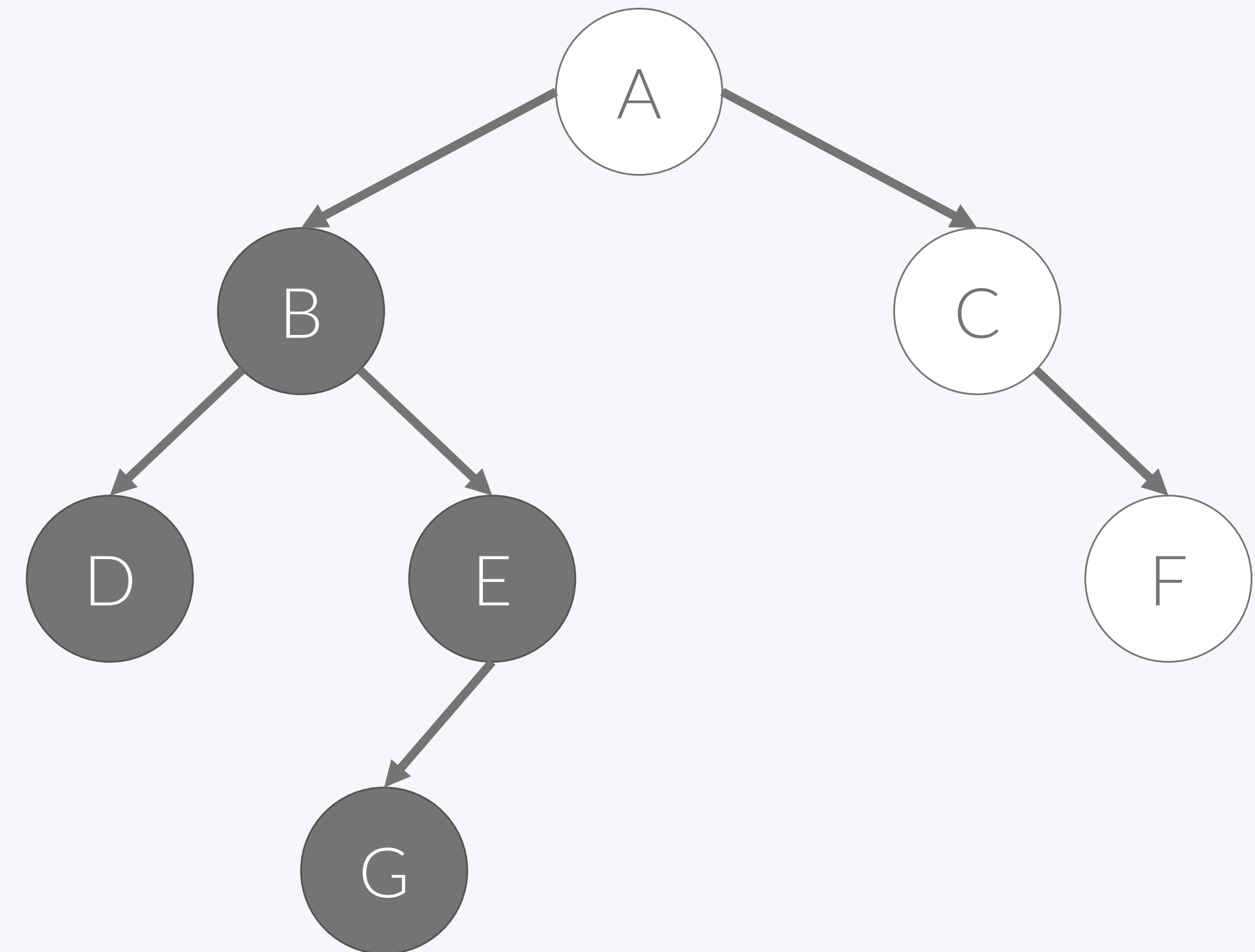
- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
 - 노드 방문
 - B
 - 오른쪽 자식 인오더
 - GE
- 노드 방문
- 오른쪽 자식 인오더



인오더

In-order

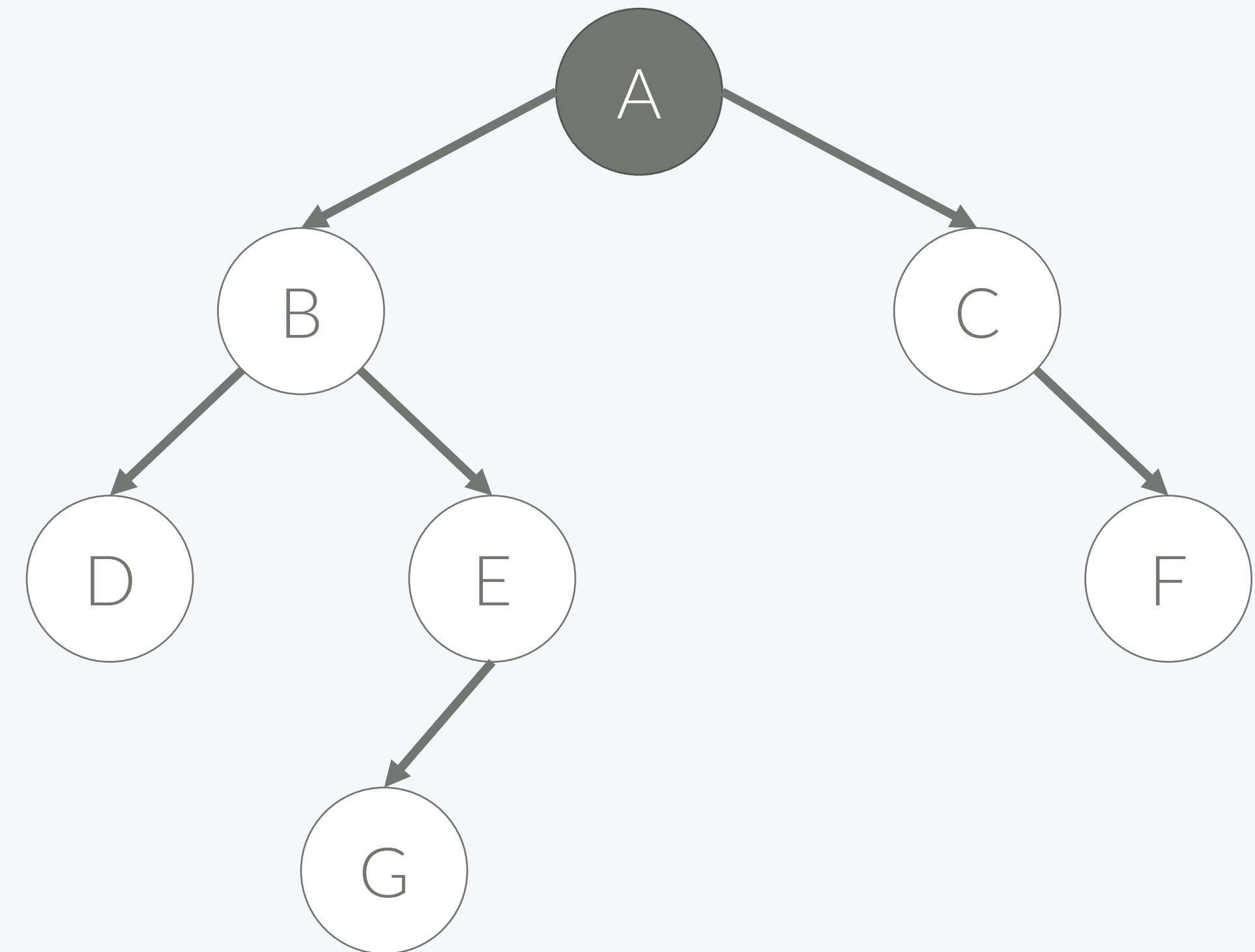
- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
- 오른쪽 자식 인오더



인오더

In-order

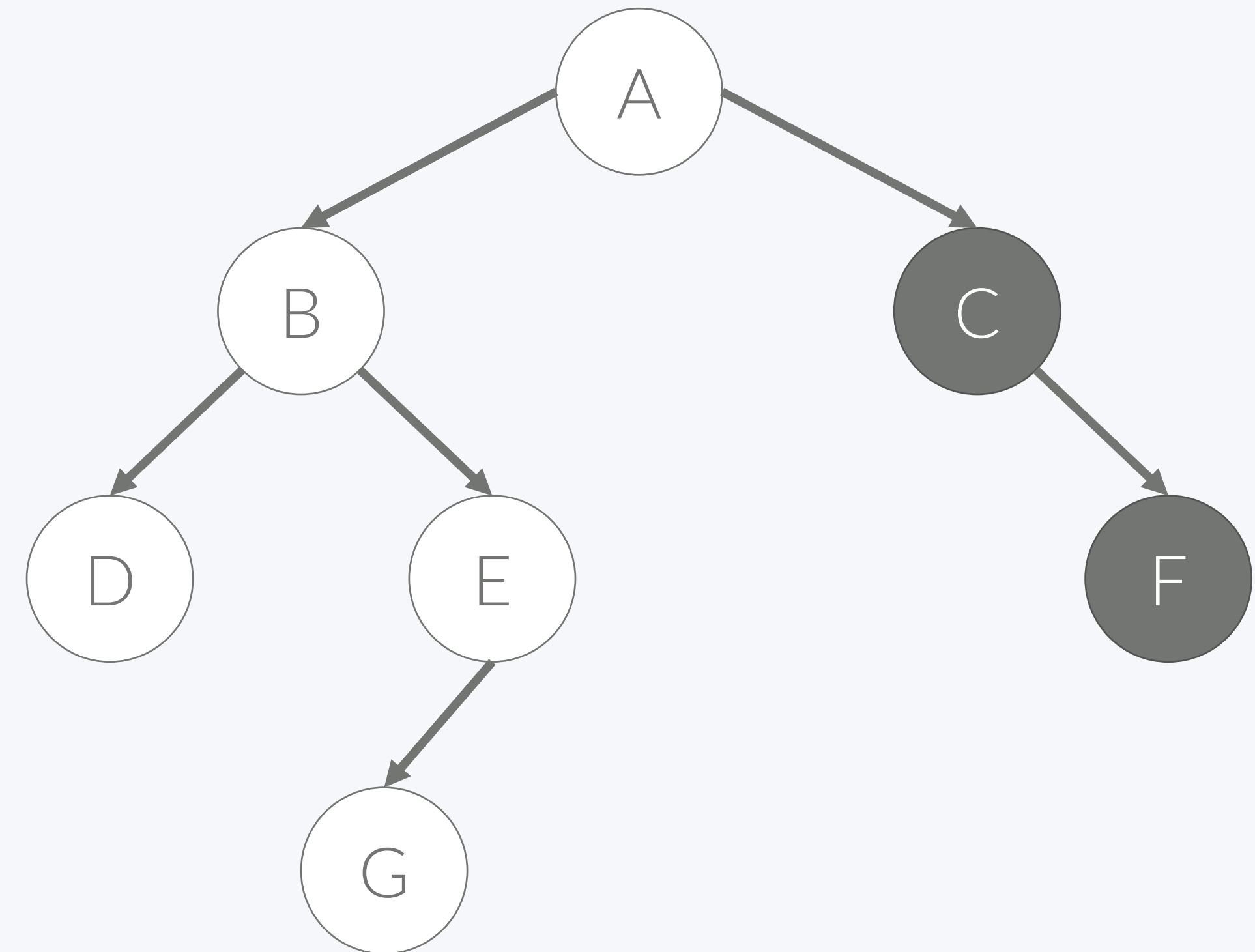
- 왼쪽 자식 인오더
 - DBGE
- **노드 방문**
 - A
- 오른쪽 자식 인오더



인오더

In-order

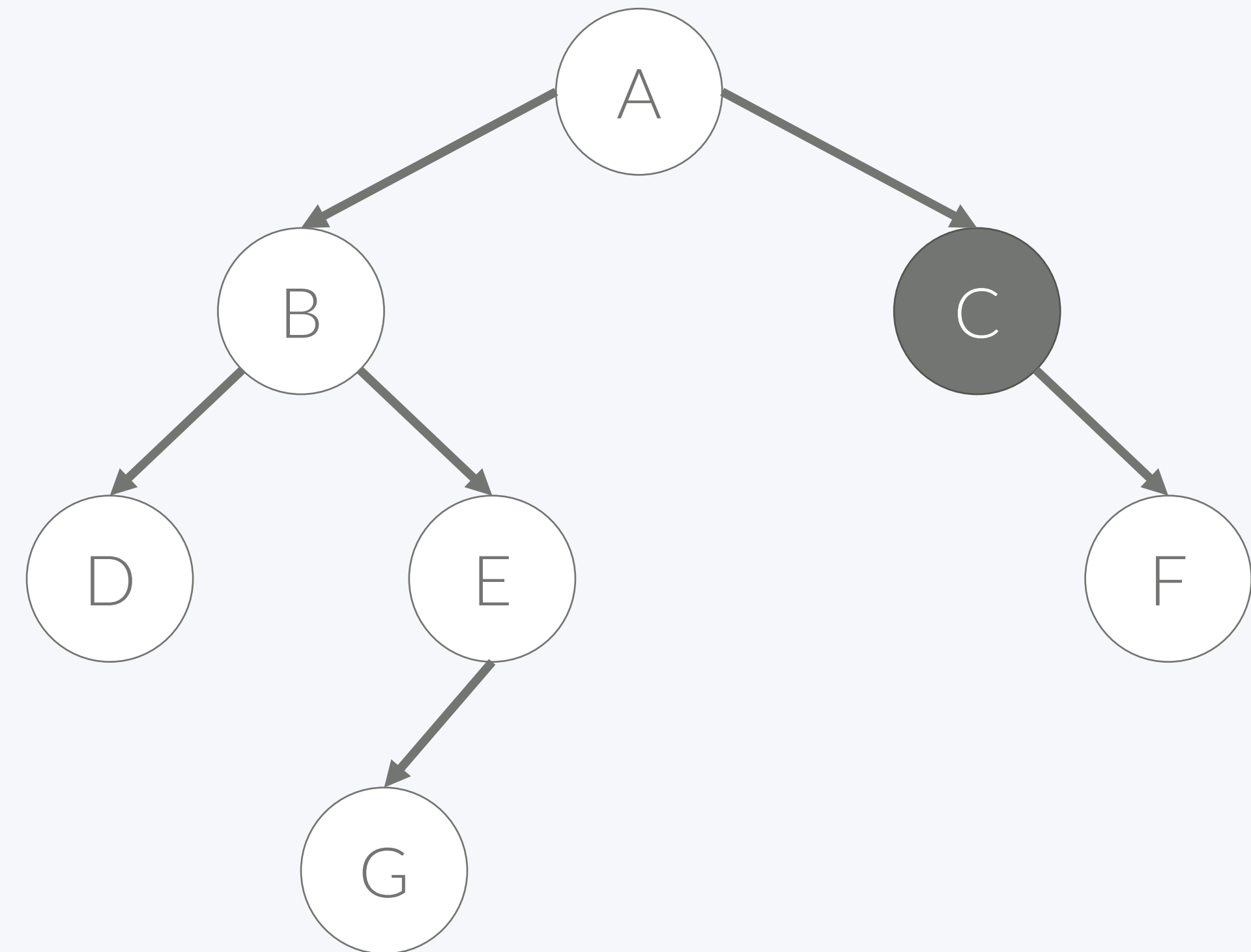
- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - 노드 방문
 - 오른쪽 자식 인오더



인오더

In-order

- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - **노드 방문**
 - C
 - 오른쪽 자식 인오더

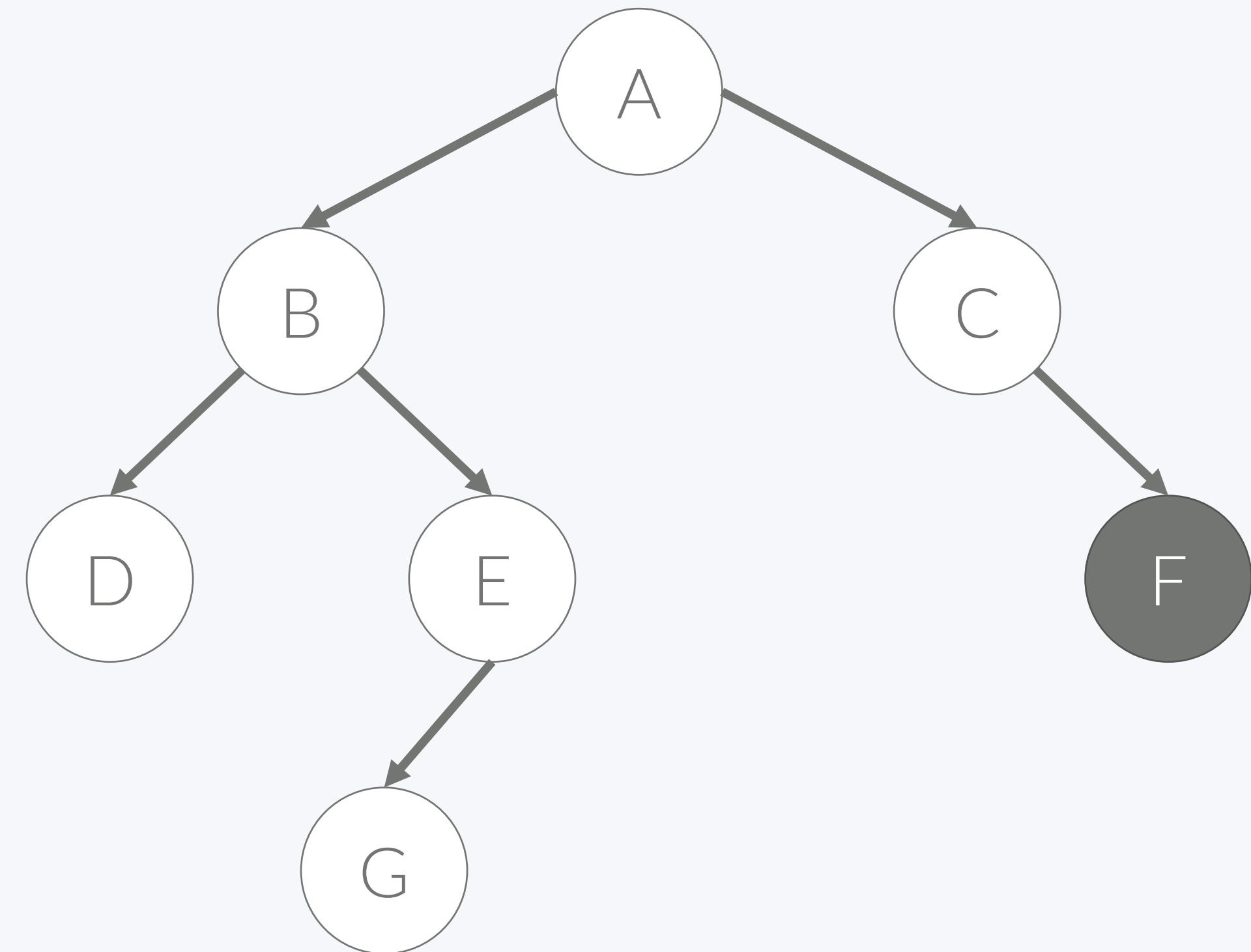


인오더

91

In-order

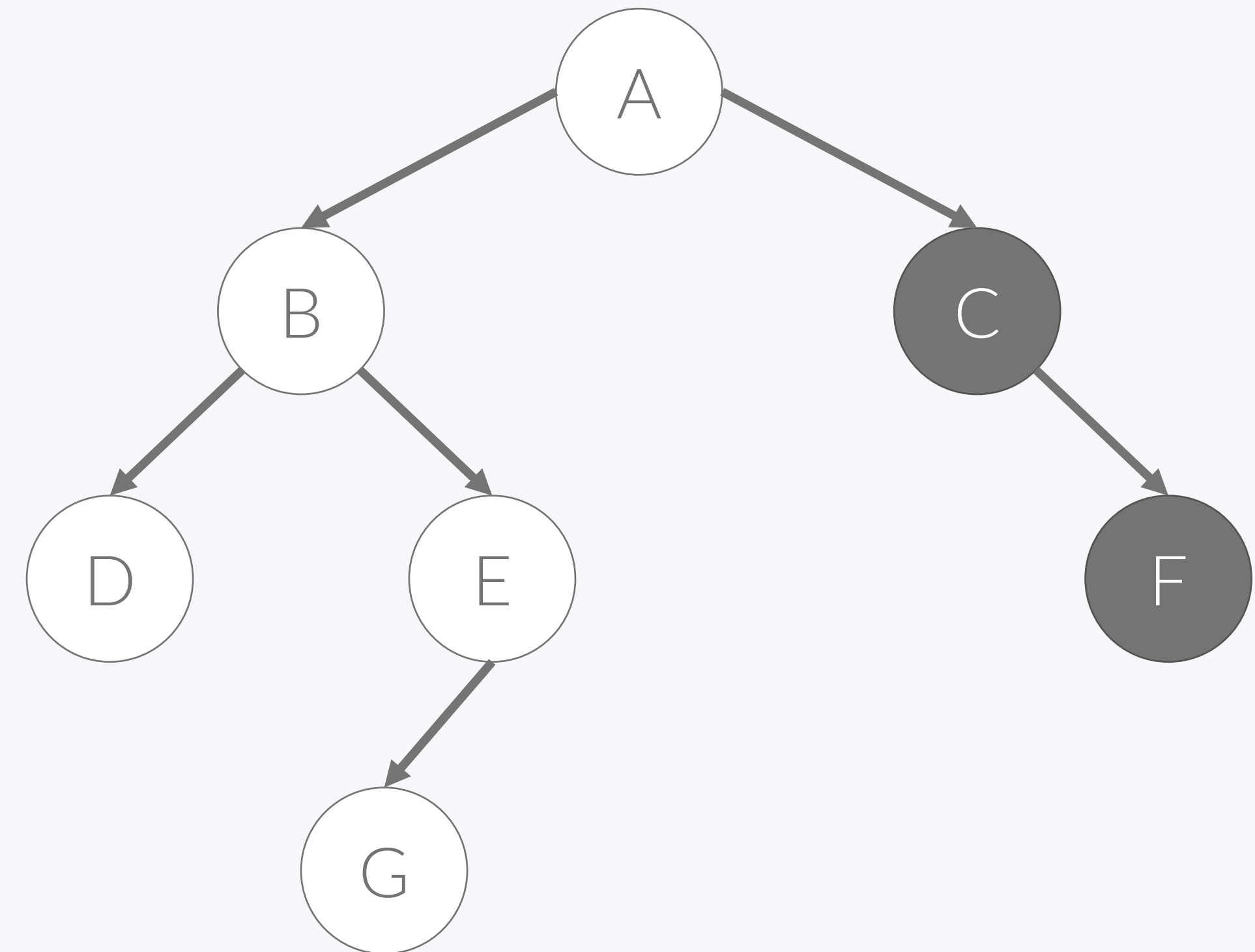
- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - 노드 방문
 - C
 - 오른쪽 자식 인오더
 - F



인오더

In-order

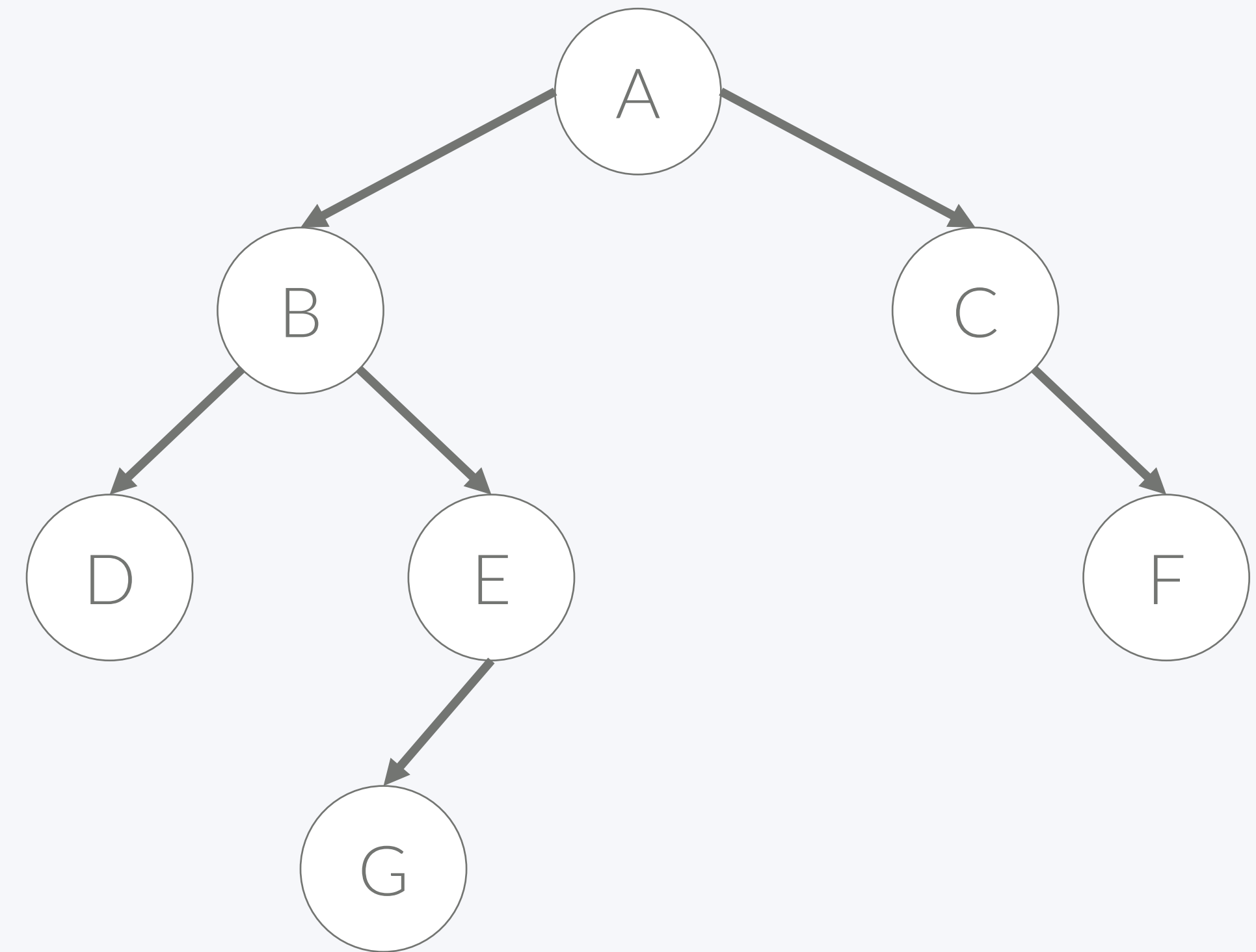
- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - CF



인오더

In-order

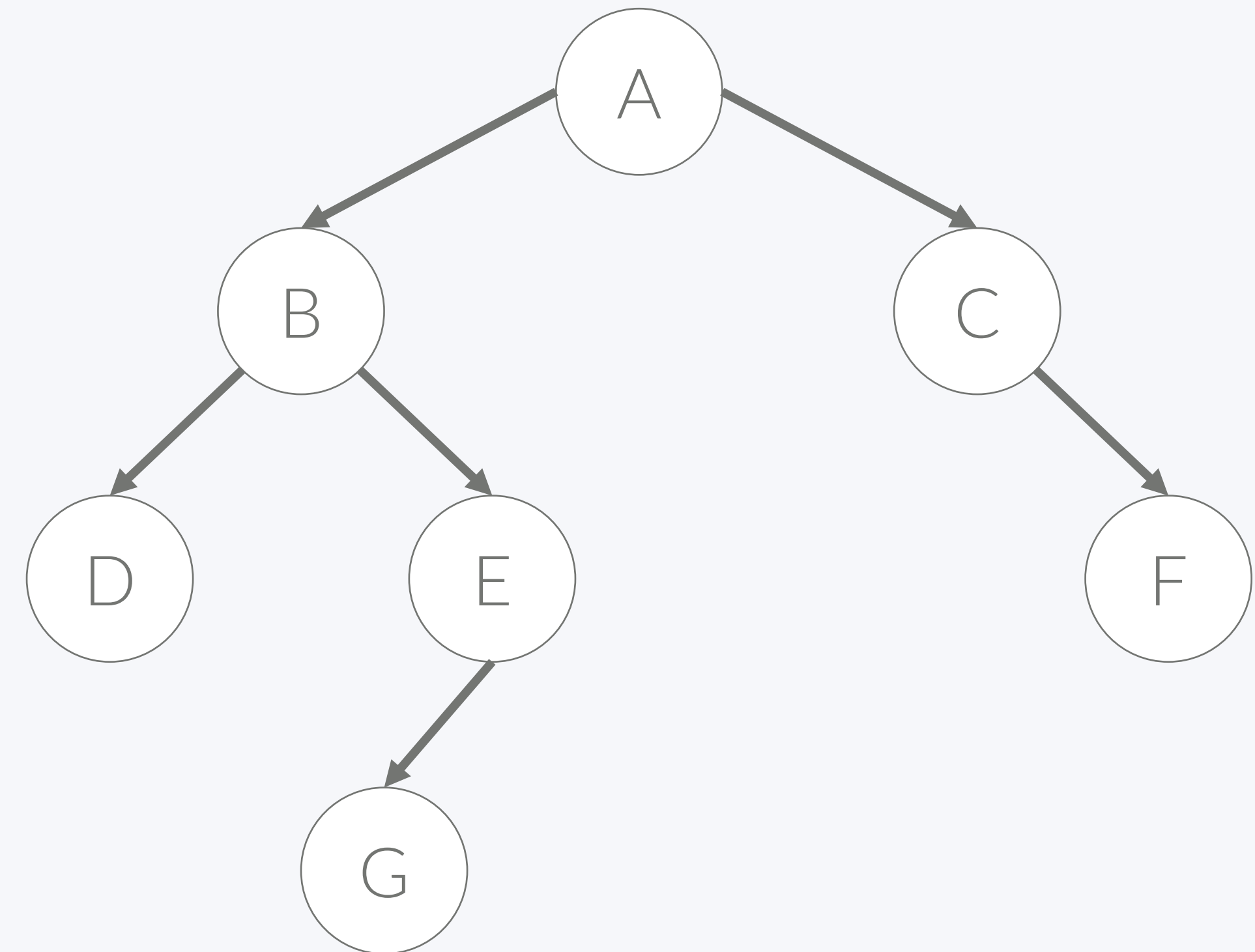
- DBGEACF



포스트오더

Post-order

- 왼쪽 자식 포스트오더
- 오른쪽 자식 포스트오더
- 노드 방문

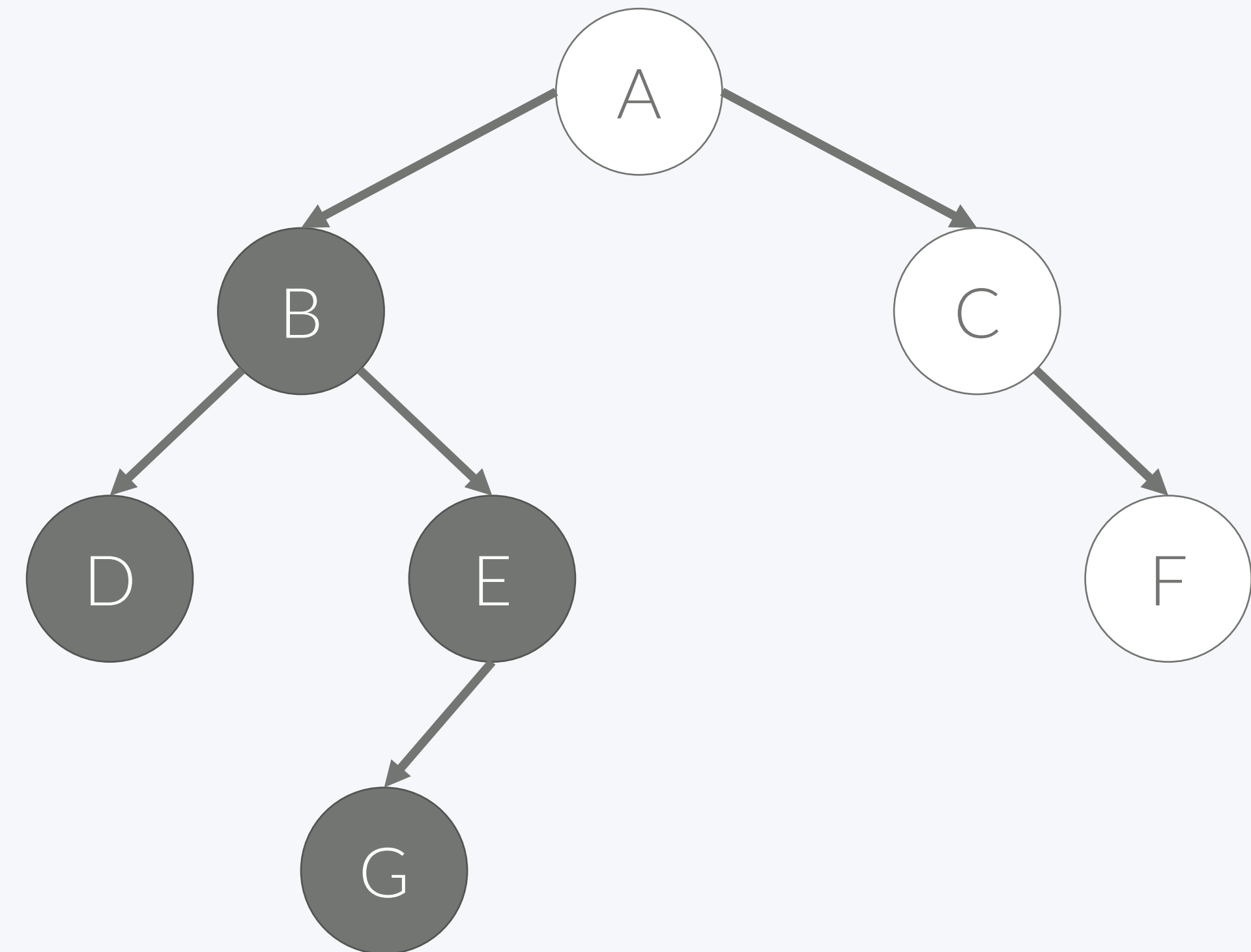


포스트오더

95

Post-order

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - 오른쪽 자식 포스트오더
 - 노드 방문
- 오른쪽 자식 포스트오더
- 노드 방문

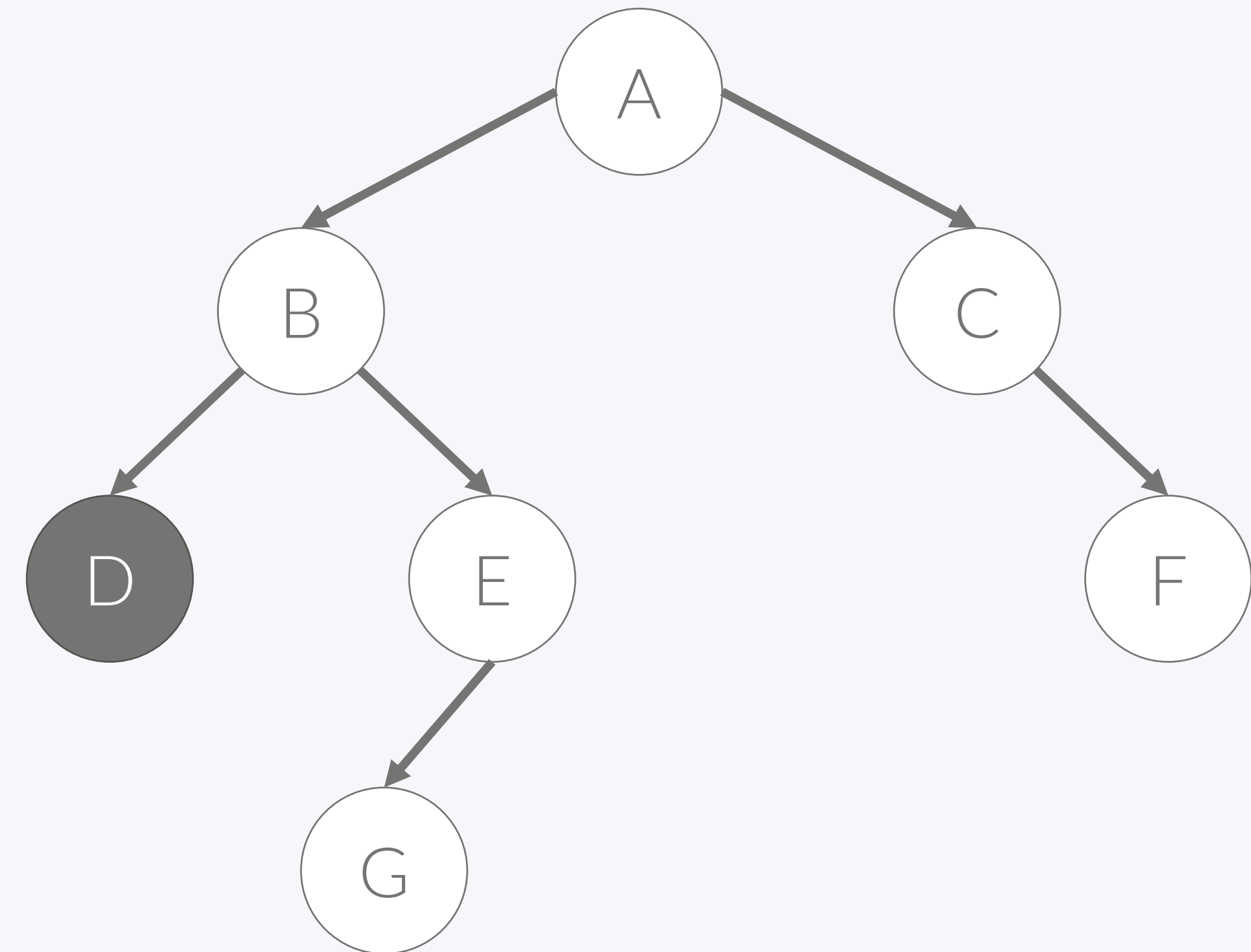


포스트오더

96

Post-order

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - D
 - 오른쪽 자식 포스트오더
 - 노드 방문
- 오른쪽 자식 포스트오더
- 노드 방문

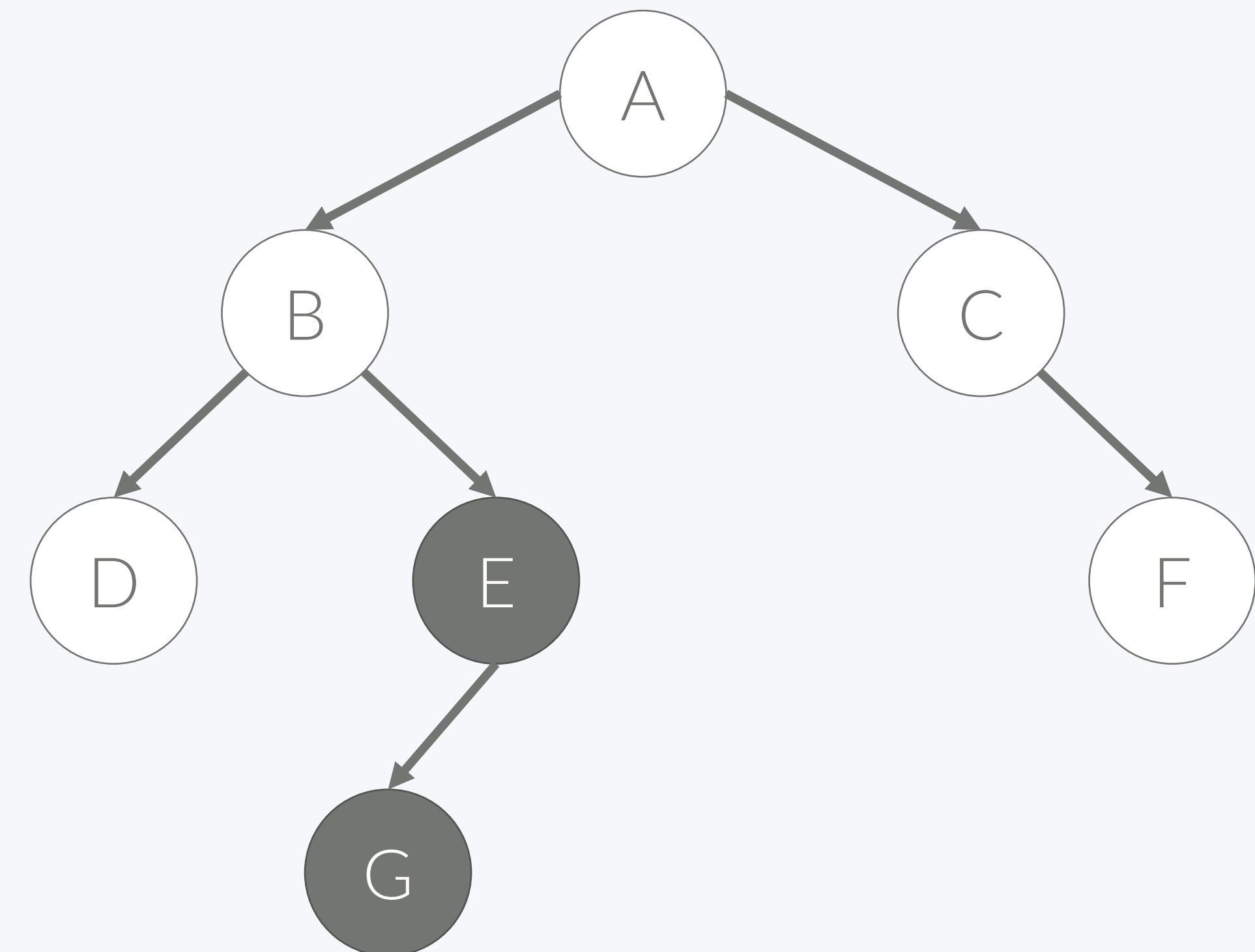


포스트오더

97

Post-order

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - D
 - 오른쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - 오른쪽 자식 포스트오더
 - 노등 방문
 - 노드 방문
-
- 오른쪽 자식 포스트오더
 - 노드 방문



포스트오더

98

Post-order

- 왼쪽 자식 포스트오더

- 왼쪽 자식 포스트오더
 - D

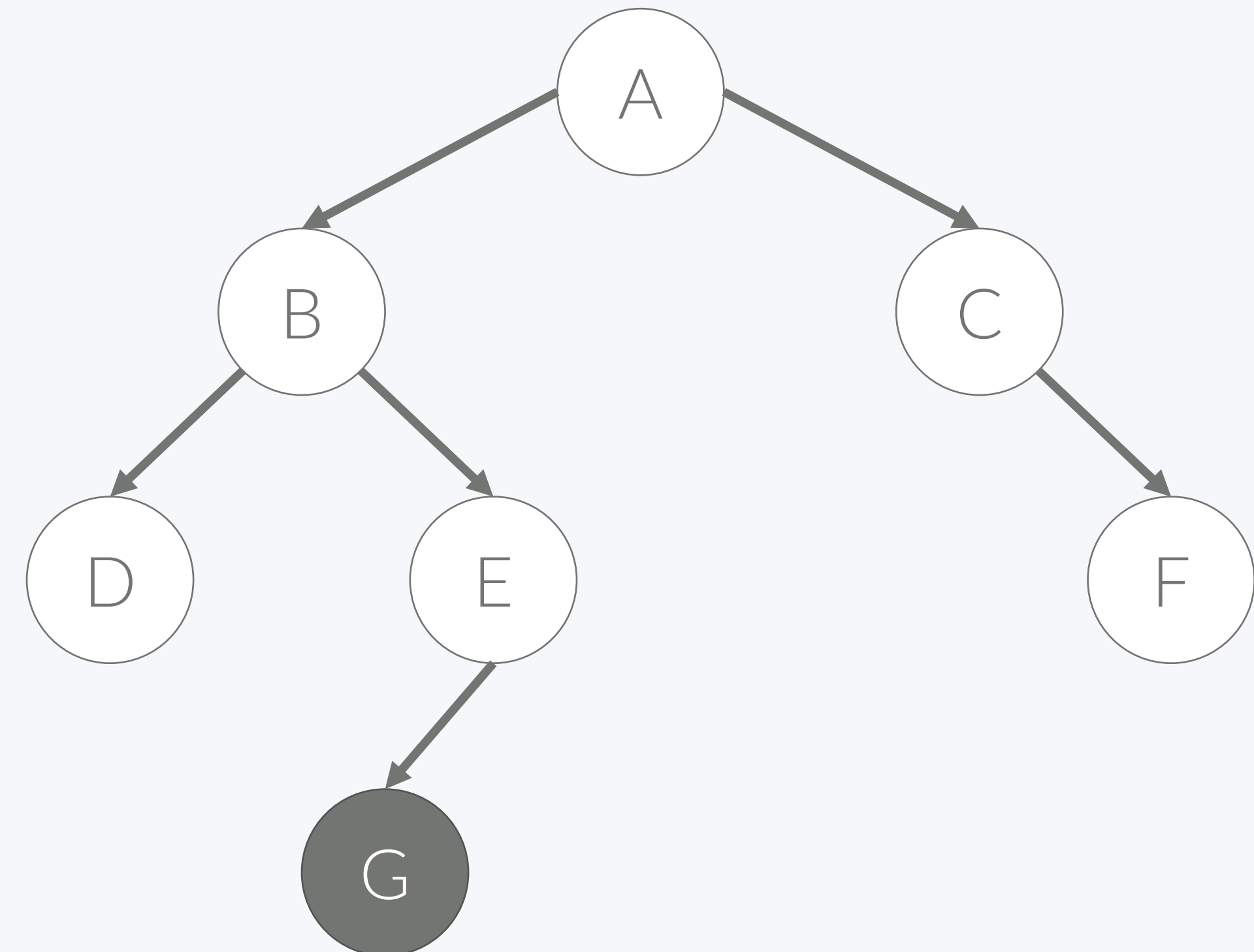
- 오른쪽 자식 포스트오더

- 왼쪽 자식 포스트오더
 - G
- 오른쪽 자식 포스트오더
- 노등 방문

- 노드 방문

- 오른쪽 자식 포스트오더

- 노드 방문



포스트오더

99

Post-order

- 왼쪽 자식 포스트오더

- 왼쪽 자식 포스트오더
 - D

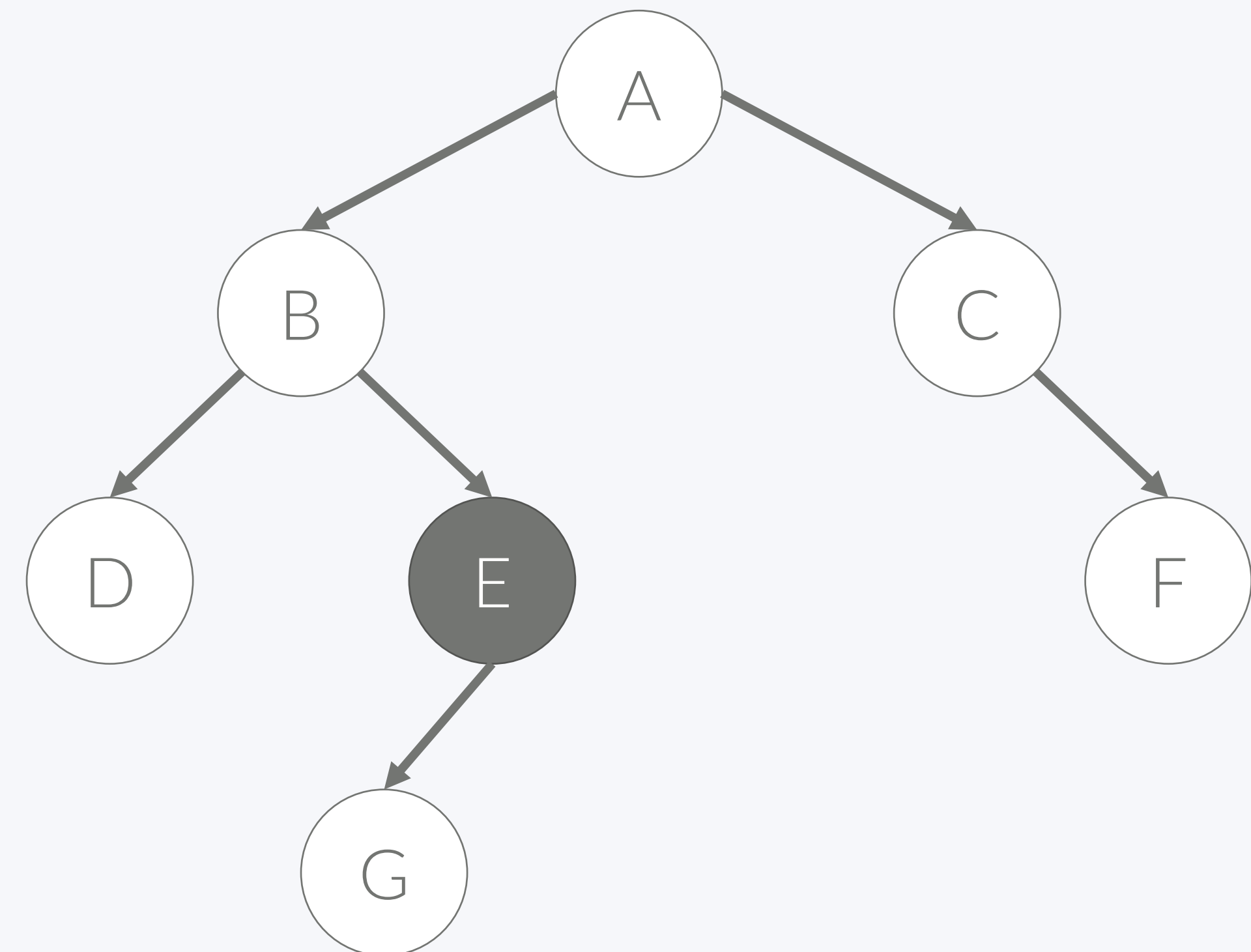
- 오른쪽 자식 포스트오더

- 왼쪽 자식 포스트오더
 - G
- 오른쪽 자식 포스트오더
- 노등 방문
 - E

- 노드 방문

- 오른쪽 자식 포스트오더

- 노드 방문

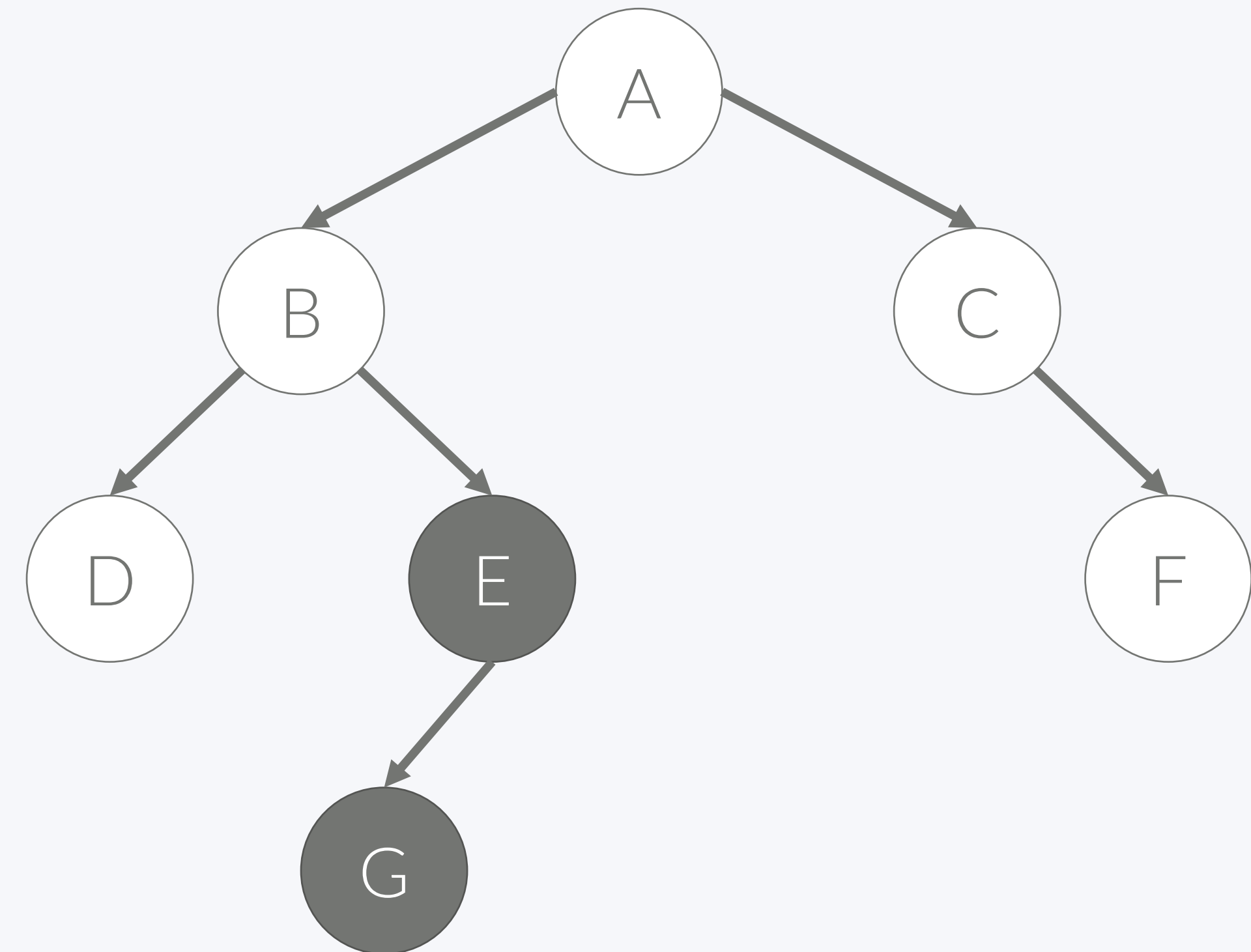


포스트오더

100

Post-order

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - D
 - 오른쪽 자식 포스트오더
 - GE
 - 노드 방문
- 오른쪽 자식 포스트오더
- 노드 방문

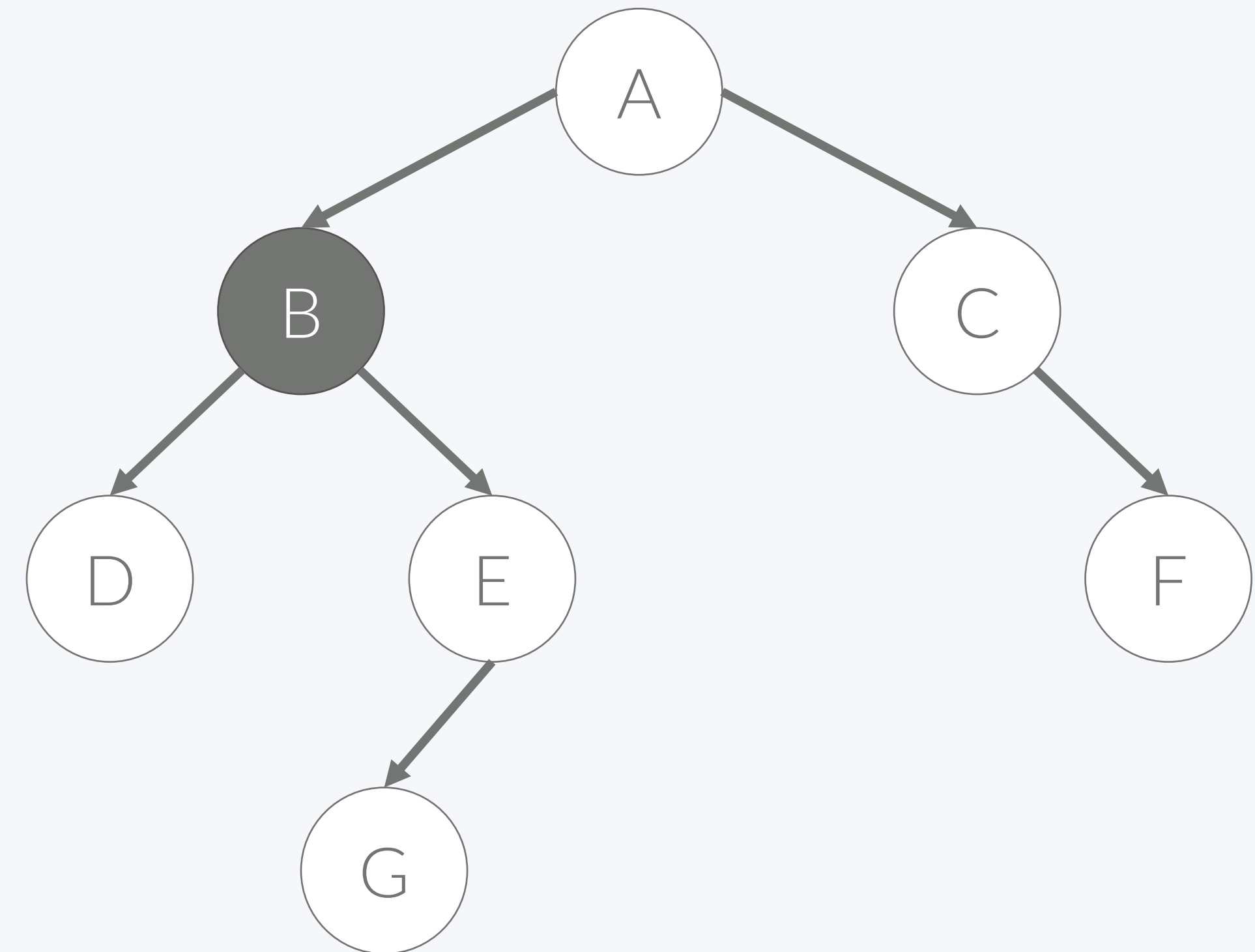


포스트오더

101

Post-order

- 왼쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - D
 - 오른쪽 자식 포스트오더
 - GE
 - 노드 방문
 - B
- 오른쪽 자식 포스트오더
- 노드 방문

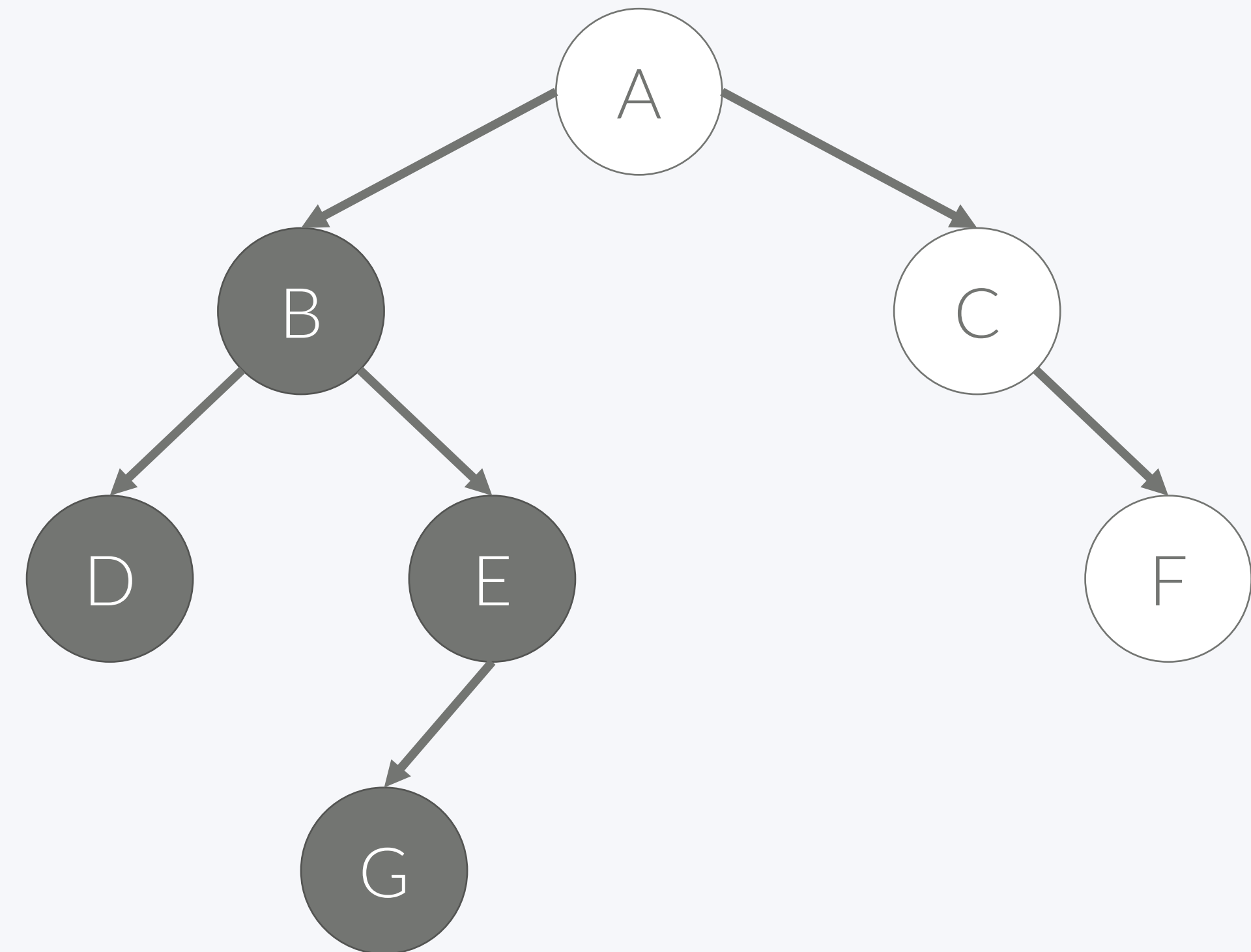


포스트오더

102

Post-order

- 왼쪽 자식 포스트오더
 - DGEB
- 오른쪽 자식 포스트오더
- 노드 방문

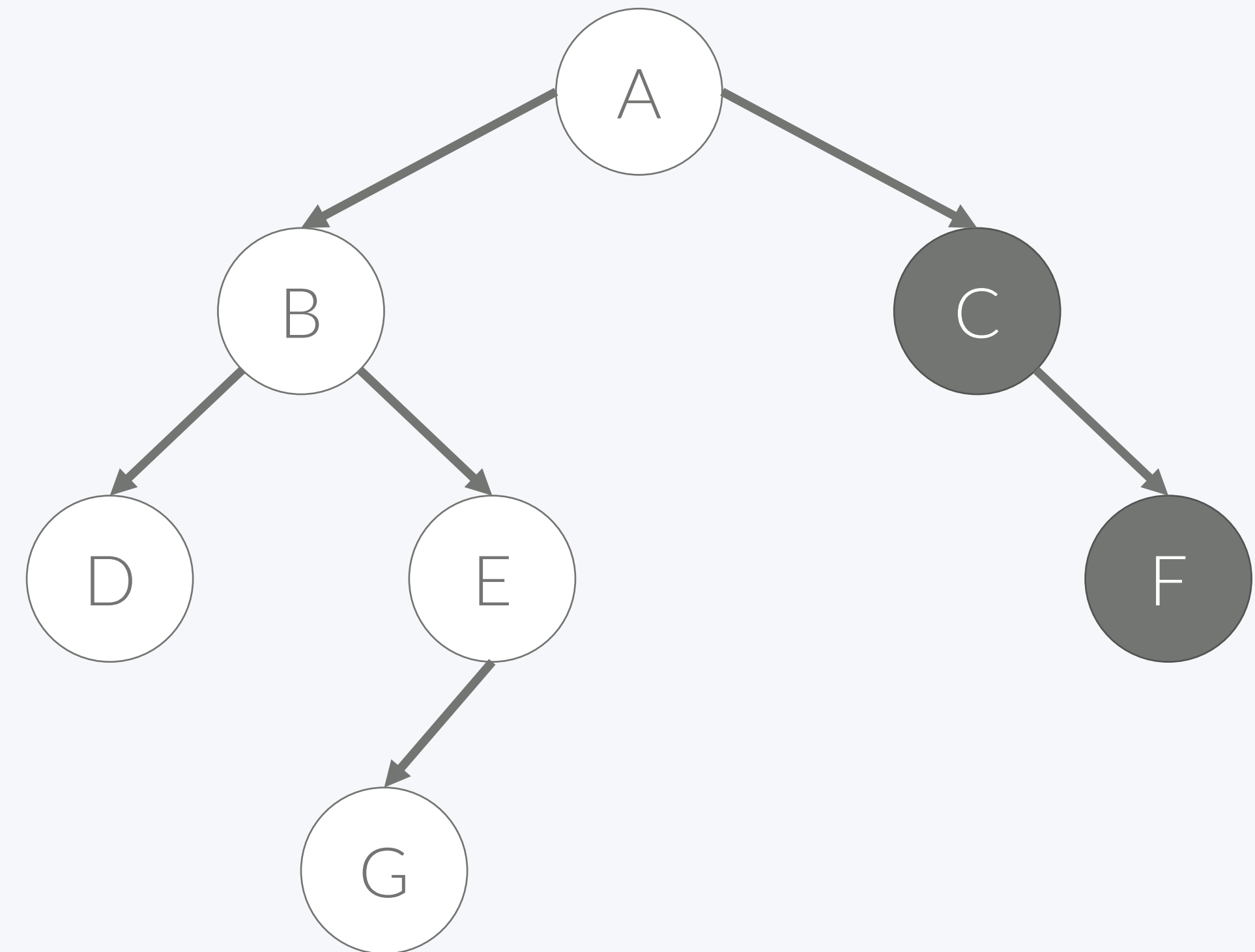


포스트오더

103

Post-order

- 왼쪽 자식 포스트오더
 - DGEB
- 오른쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - 오른쪽 자식 포스트오더
 - 노드 방문
- 노드 방문

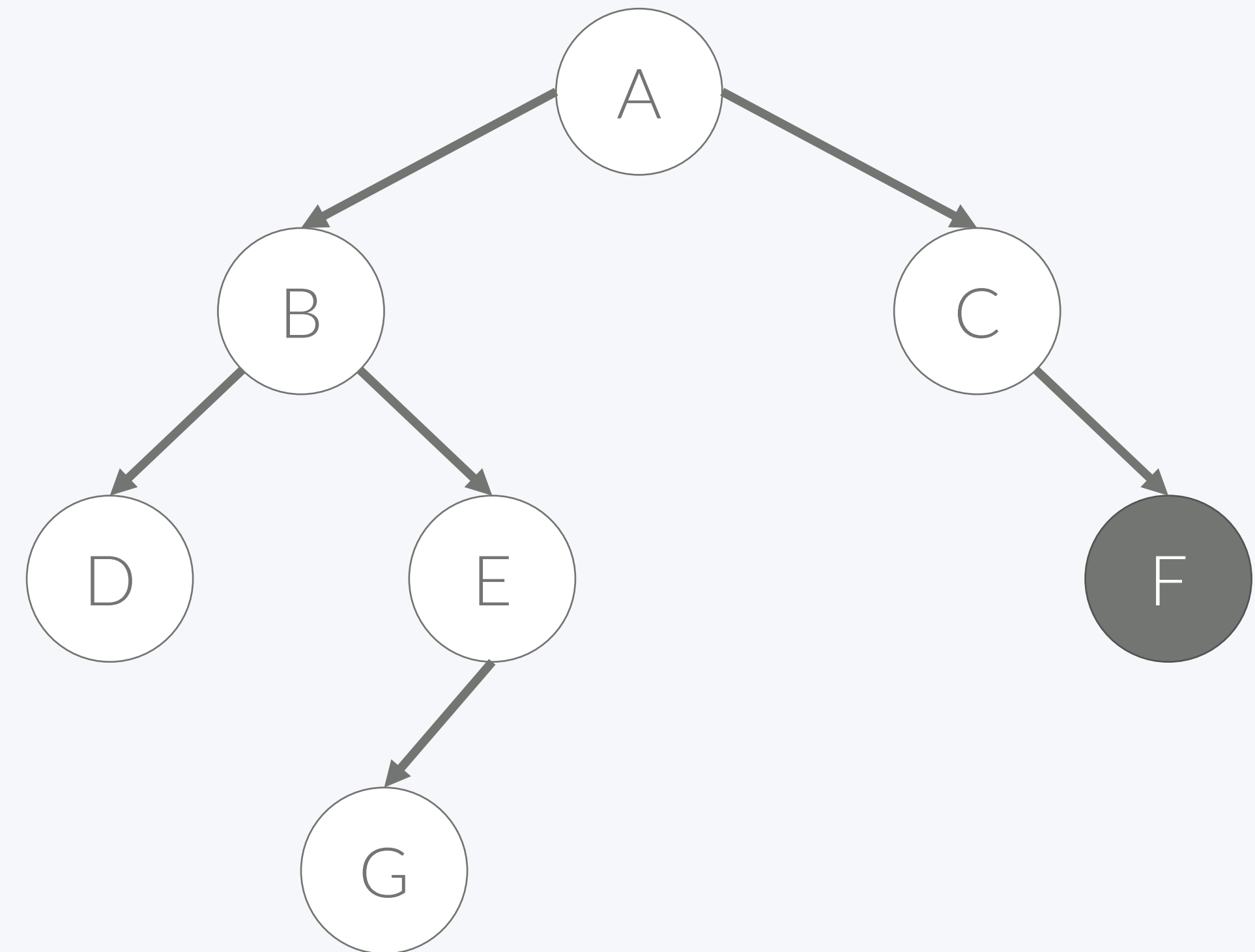


포스트오더

104

Post-order

- 왼쪽 자식 포스트오더
 - DGEB
- **오른쪽 자식 포스트오더**
 - 왼쪽 자식 포스트오더
 - **오른쪽 자식 포스트오더**
 - F
 - 노드 방문
- 노드 방문

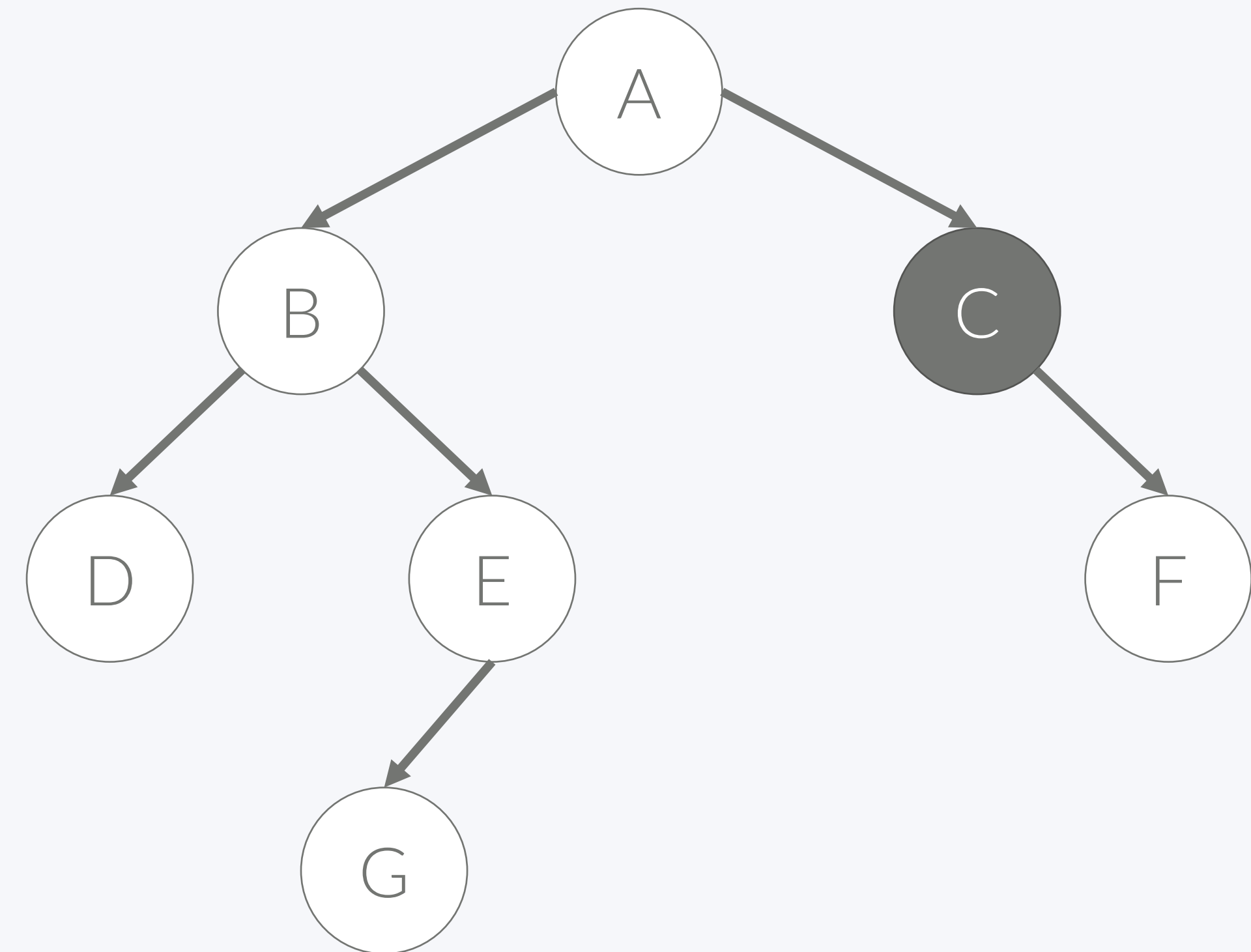


포스트오더

105

Post-order

- 왼쪽 자식 포스트오더
 - DGEB
- 오른쪽 자식 포스트오더
 - 왼쪽 자식 포스트오더
 - 오른쪽 자식 포스트오더
 - F
 - 노드 방문
 - C
- 노드 방문

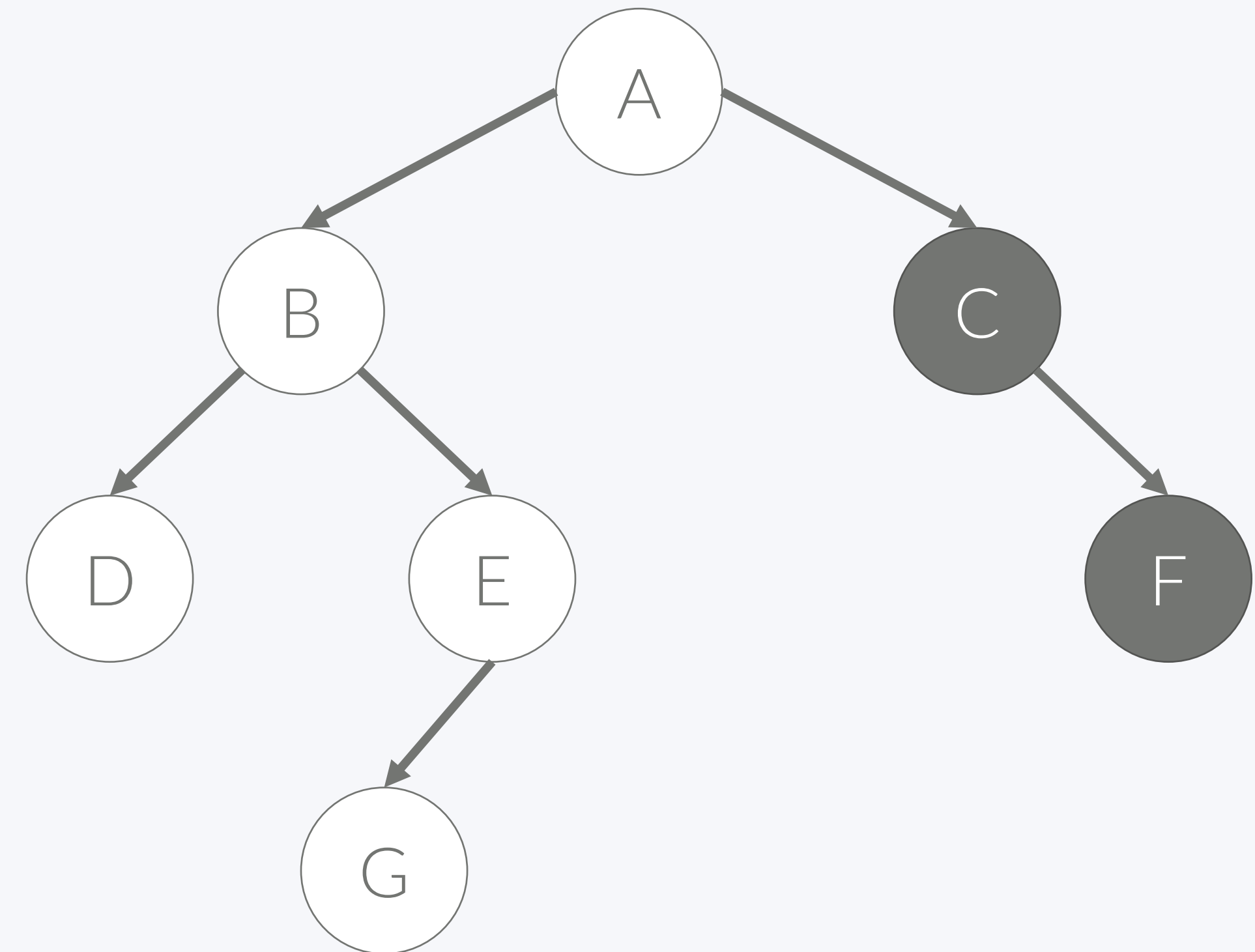


포스트오더

106

Post-order

- 왼쪽 자식 포스트오더
 - DGEB
- 오른쪽 자식 포스트오더
 - FC
- 노드 방문

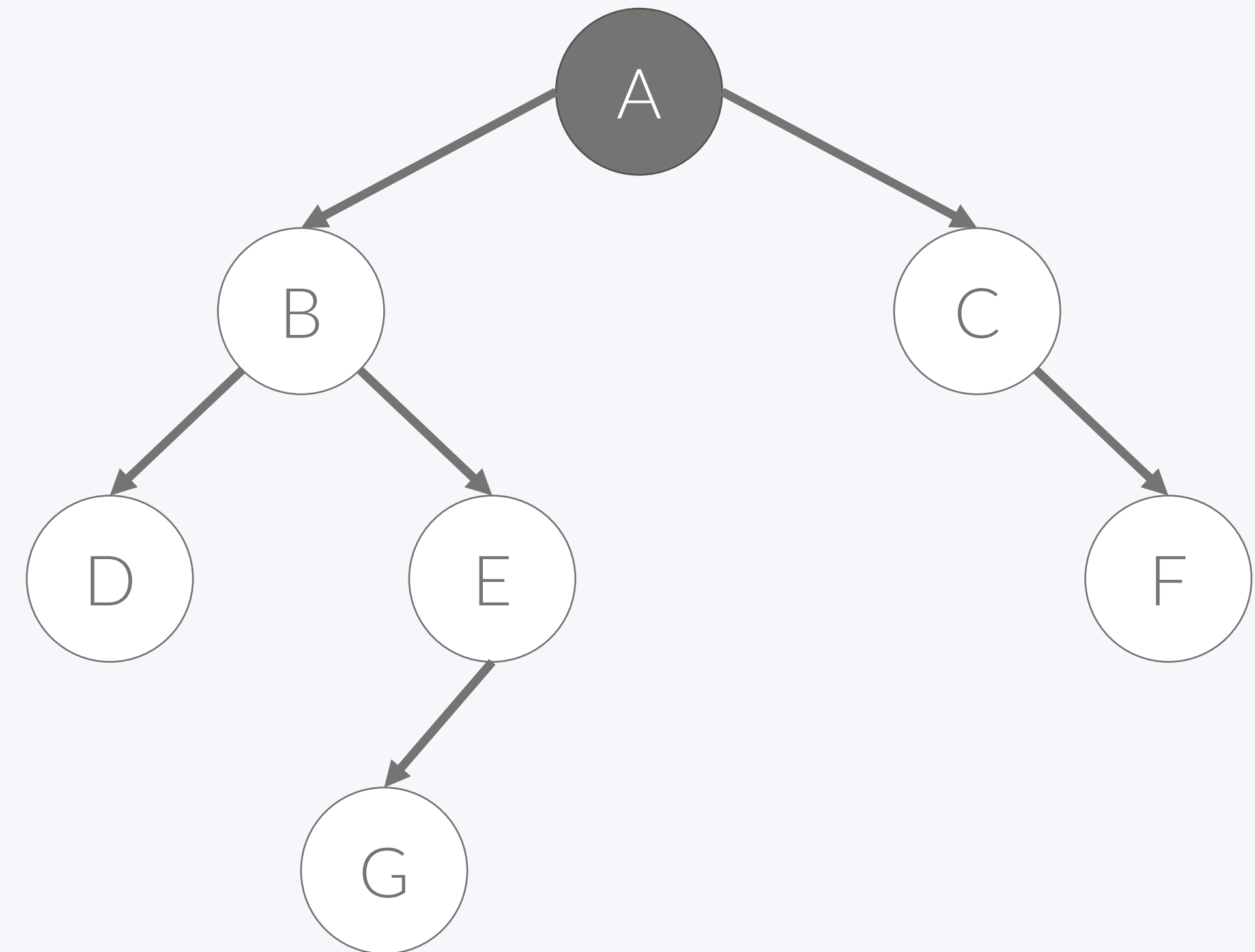


포스트오더

107

Post-order

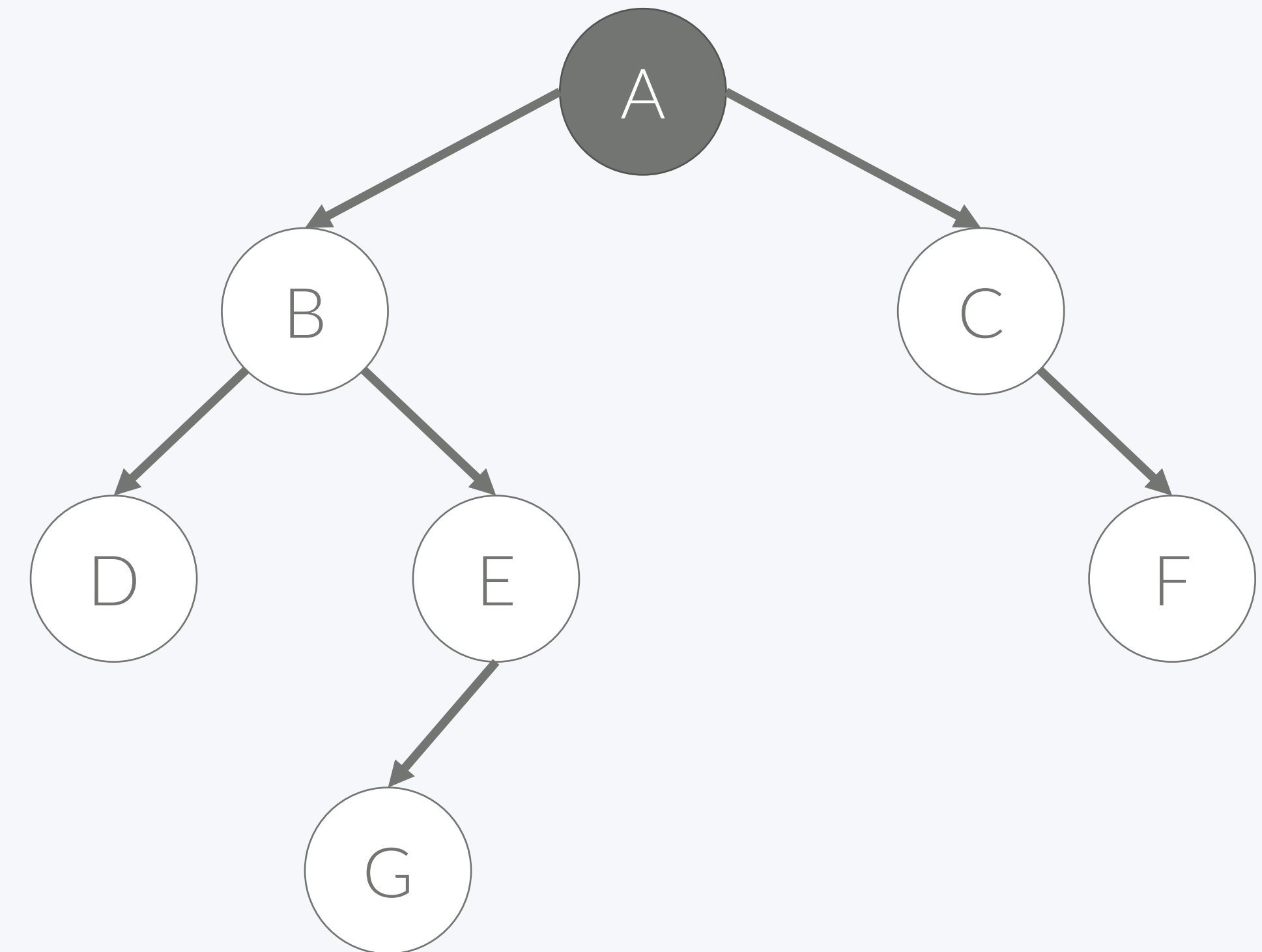
- 왼쪽 자식 포스트오더
 - DGEB
- 오른쪽 자식 포스트오더
 - FC
- 노드 방문
 - A



포스트오더

Post-order

- DGEBFCA



트리 순회

109

<https://www.acmicpc.net/problem/1991>

- 이진 트리의 프리오더, 인오더, 포스트오더 순서를 출력하는 문제

트리 순회

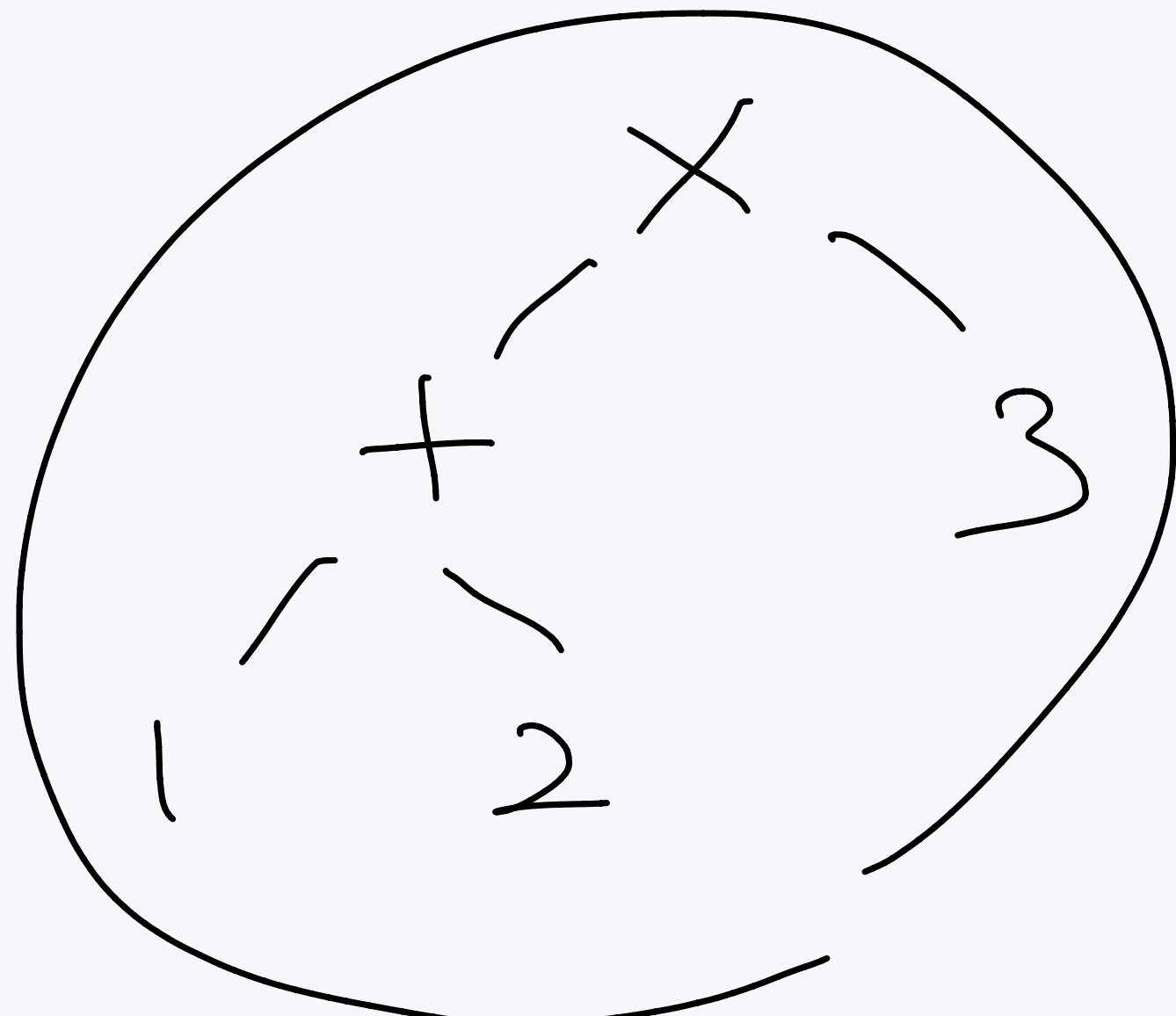
110

<https://www.acmicpc.net/problem/1991>

- 소스: <http://boj.kr/df63db6a94c3482b93a098af39e7affa>

$$(1+2) \times 3$$

$$(12+3) \times$$



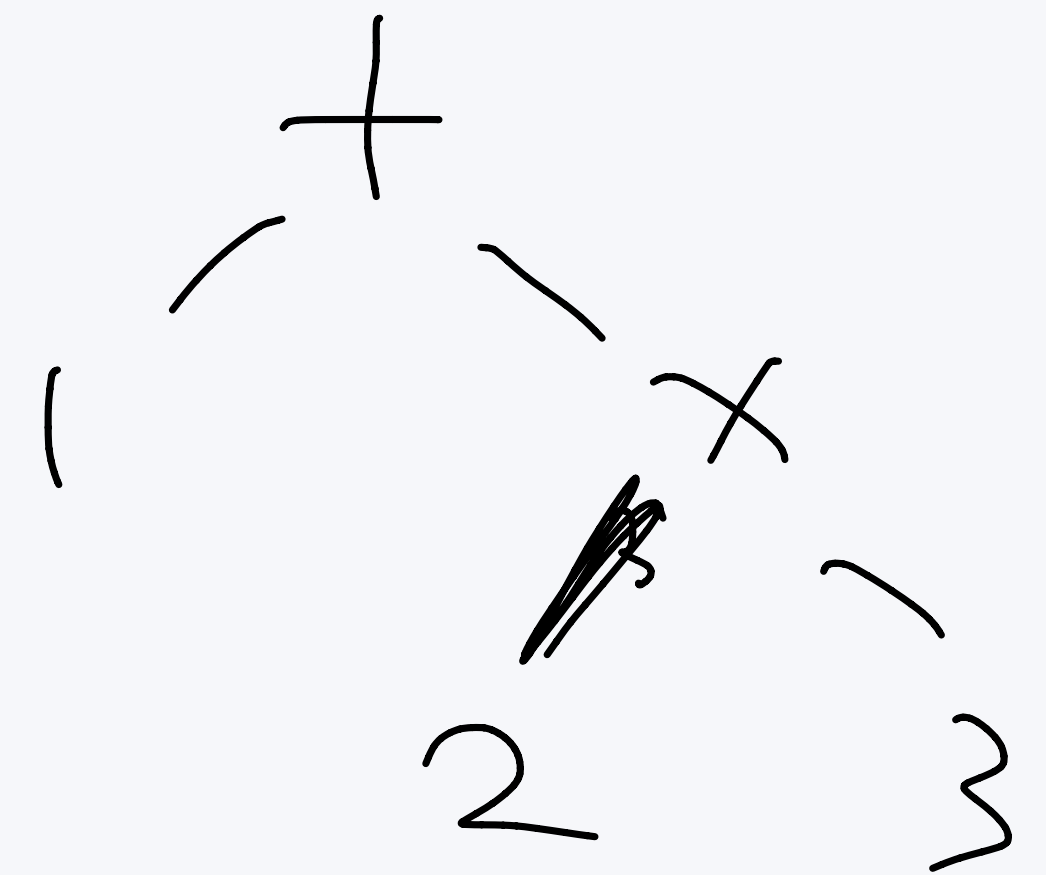
$$\text{Infix} = 1+2$$

$$\text{Postfix} = 12+$$

$$\text{Prefix} = +12$$

$$1+2 \times 3$$

$$123 \times +$$



트리의 높이와 너비

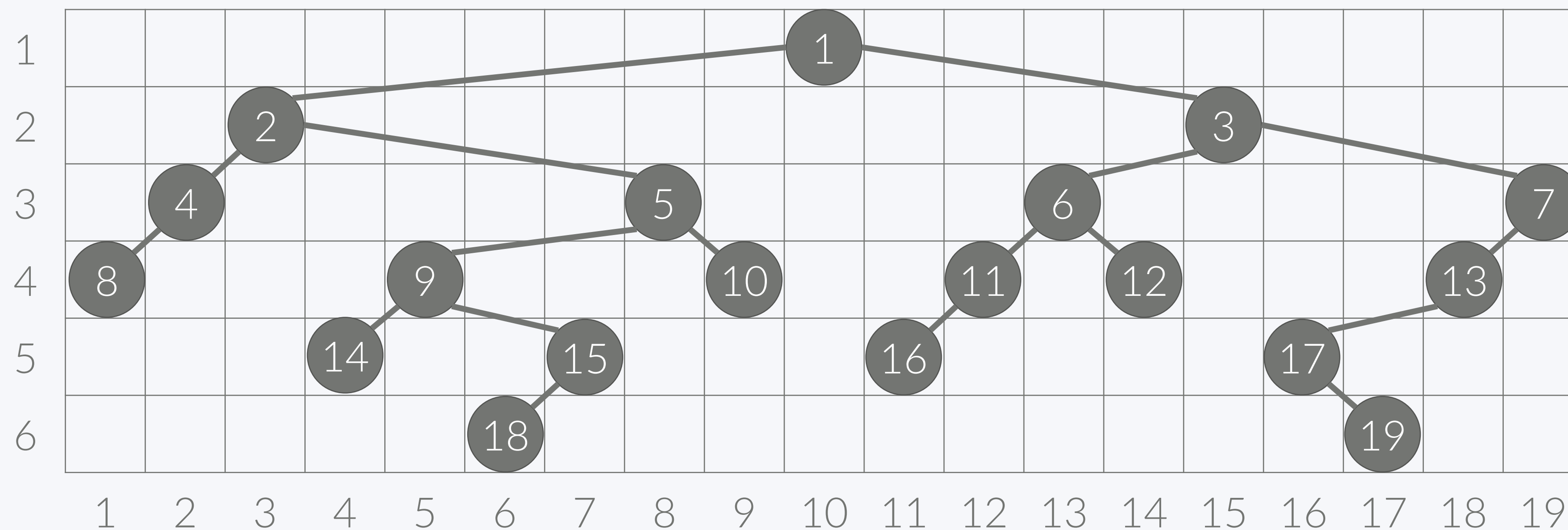
<https://www.acmicpc.net/problem/2250>

- 이진 트리를 다음과 같은 규칙에 따라 격자에 그리려고 한다
 1. 이진트리에서 같은 레벨(level)에 있는 노드는 같은 행에 위치한다.
 2. 한 열에는 한 노드만 존재한다.
 3. 임의의 노드의 왼쪽 부트리(left subtree)에 있는 노드들은 해당 노드보다 왼쪽의 열에 위치하고, 오른쪽 부트리(right subtree)에 있는 노드들은 해당 노드보다 오른쪽의 열에 위치한다.
 4. 노드가 배치된 가장 왼쪽 열과 오른쪽 열 사이엔 아무 노드도 없이 비어있는 열은 없다.

트리의 높이와 너비

<https://www.acmicpc.net/problem/2250>

- 이진 트리를 다음과 같은 규칙에 따라 격자에 그리려고 한다



트리의 높이와 너비

113

<https://www.acmicpc.net/problem/2250>

- 각 노드가 배치되는 순서는 인오더와 같다
- 인오더를 수행하면서, 몇 번인지, 몇 번째 레벨인지 기록한다

트리의 높이와 너비

114

<https://www.acmicpc.net/problem/2250>

- 소스: <http://boj.kr/dae0019b770d4b20a25cdf67a8e74163>

트리에서 순회하기

DFS, BFS

트리의 탐색

이진 트리의 경우

u, v
124

트리의 탐색

BFS

- 트리의 탐색은 DFS/BFS 알고리즘을 이용해서 할 수 있다.
- 트리는 사이클이 없는 그래프이기 때문에
- 임의의 두 정점 사이의 경로는 1개이다.
- 따라서, BFS 알고리즘을 이용해서 최단 거리를 구할 수 있다.
- 이유: 경로가 1개라 찾은 그 경로가 최단 경로

트리의 부모 찾기

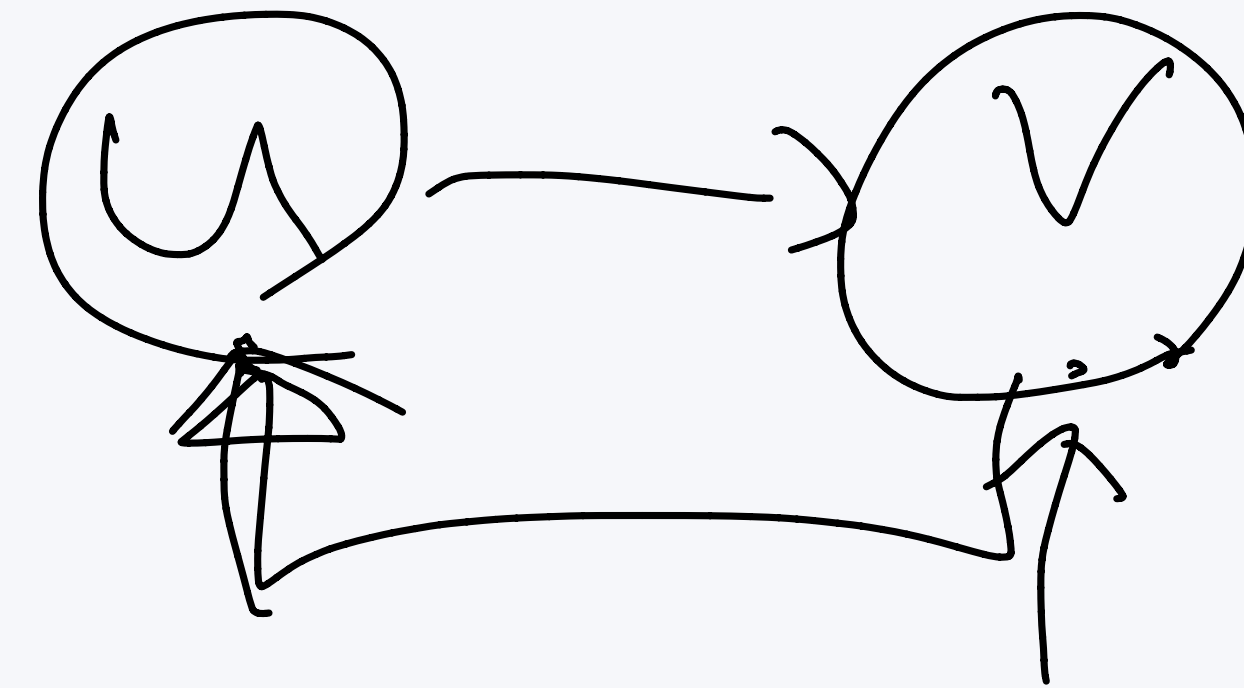
<https://www.acmicpc.net/problem/11725>

- 그래프로 트리를 입력받고
 - 루트를 1이라고 정했을 때
 - 각 노드의 부모를 찾는 문제
-
- BFS 탐색으로 해결할 수 있다.

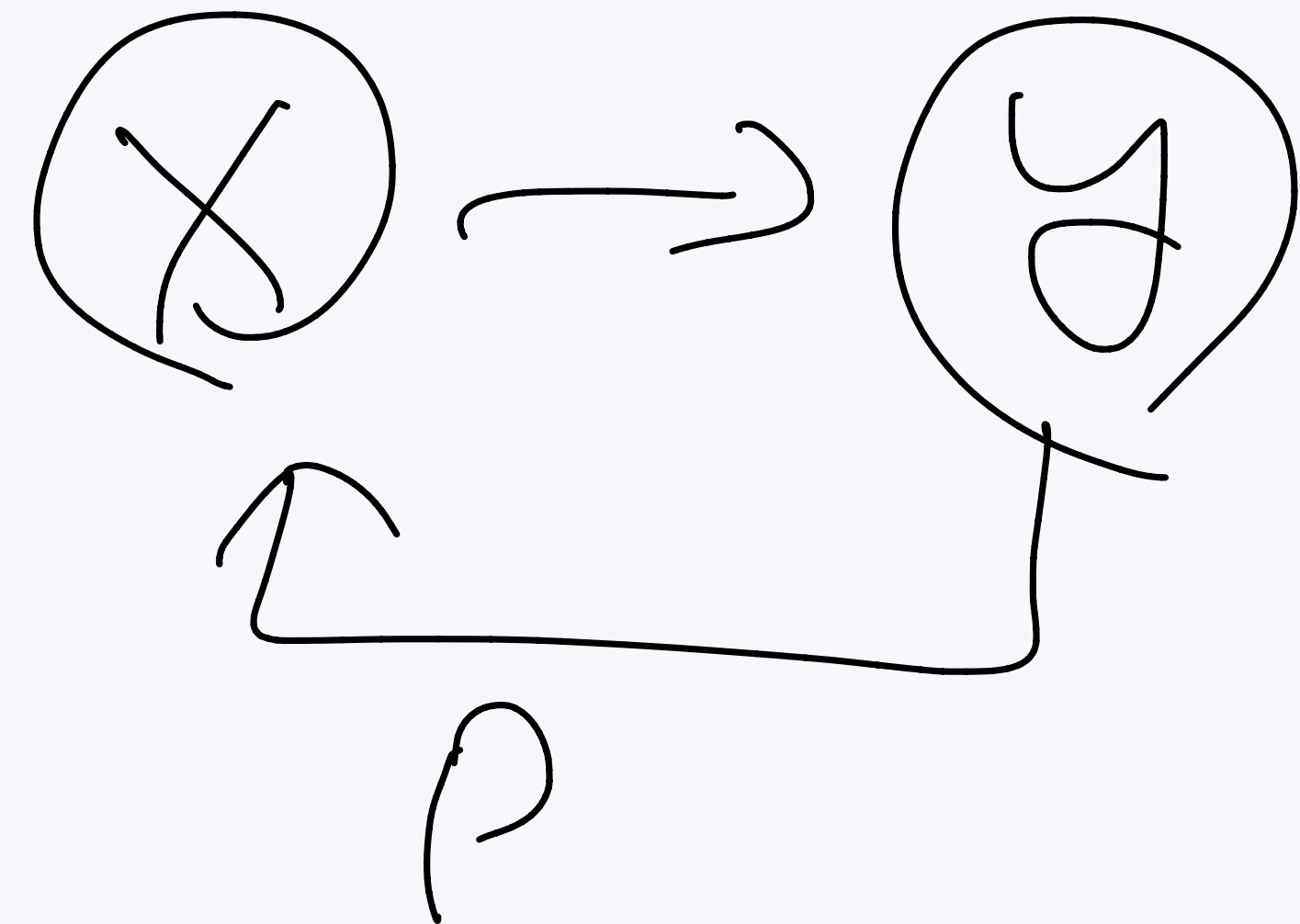
트리의 부모 찾기

<https://www.acmicpc.net/problem/11725>

118



```
queue<int> q;
depth[1] = 0; check[1] = true; parent[1] = 0; q.push(1);
while (!q.empty()) {
    int x = q.front(); q.pop();
    for (int y : a[x]) {
        if (!check[y]) {
            depth[y] = depth[x] + 1;
            check[y] = true;
            parent[y] = x;
            q.push(y);
        }
    }
}
```



트리의 부모 찾기

119

<https://www.acmicpc.net/problem/11725>

- 소스: <http://boj.kr/02fac9bbb2754e2e926e607247cf1018>

트리의 지름

가장 짧은 길이를 트리의 지름이라고 함

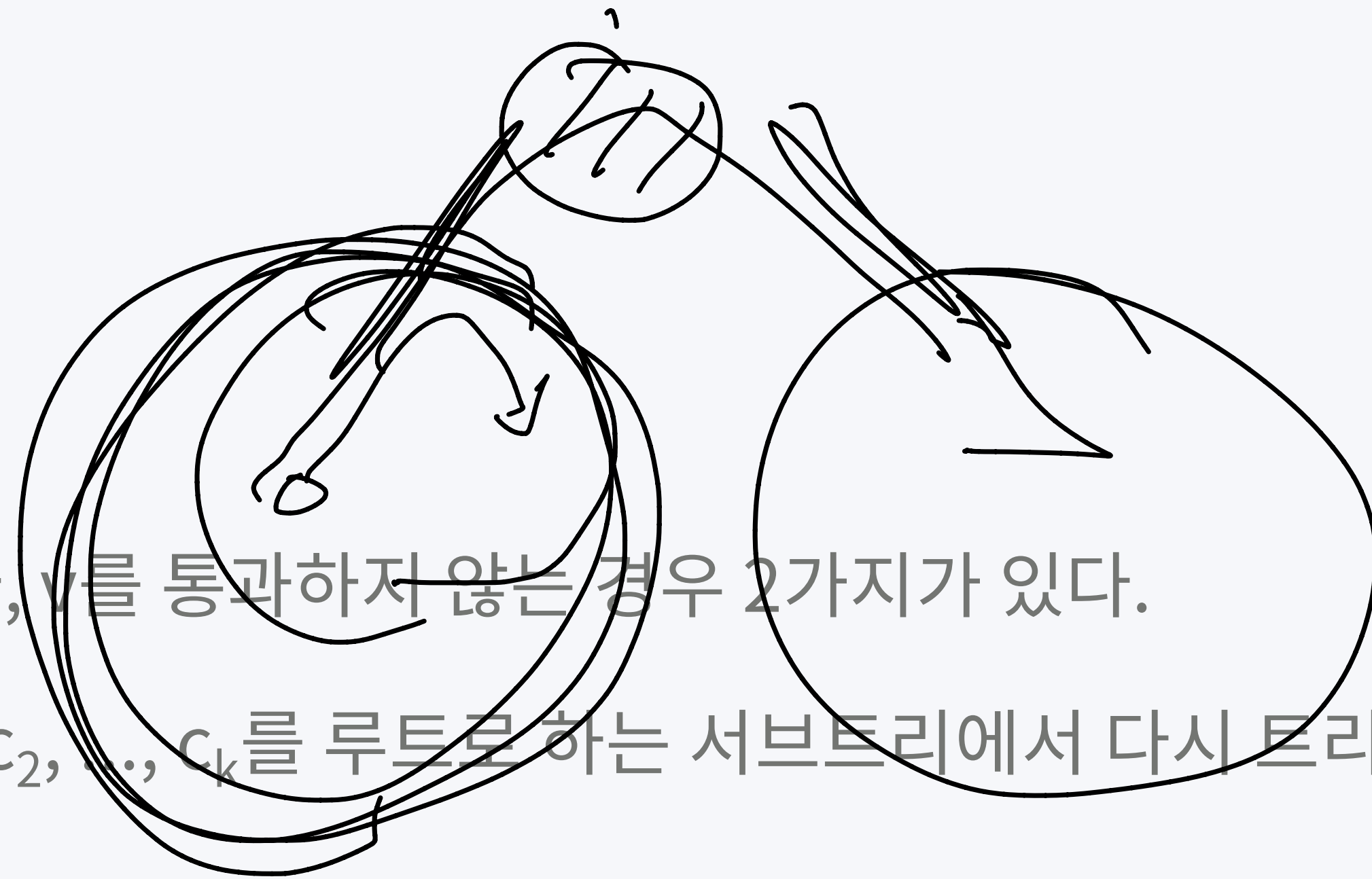
- 트리에 존재하는 모든 경로 중에서 가장 긴 것의 길이를 트리의 지름이라고 한다
- 트리의 지름은 탐색 2번으로 구할 수 있다.
 1. 한 정점에서 모든 정점까지의 거리를 구한다. 이 때, 가장 먼 거리인 정점을 u 라고 한다.
 2. u 에서 모든 정점까지의 거리를 구한다. 이 때, 가장 먼 거리인 정점 v 를 구한다.
- $d(u, v)$ 를 u 와 v 사이의 거리라고 했을 때, $d(u, v)$ 가 트리의 지름이다

트리의 지름

122

Diameter

- 포스트 오더를 이용해서도 구할 수 있다
- 루트가 v 일 때, 트리의 지름은 v 를 통과하거나, v 를 통과하지 않는 경우 2가지가 있다.
- v 를 통과하지 않는 경우에는 각각의 자식 c_1, c_2, \dots, c_k 를 루트로 하는 서브트리에서 다시 트리의 지름을 구한다
- v 를 통과하는 경우에는 트리의 지름은 각각의 자식 c_1, c_2, \dots, c_k 에서 리프노드까지의 거리 중에 가장 큰 값 2개를 이용해서 만들 수 있다



트리의 지름

123

<https://www.acmicpc.net/problem/1167>

- 트리의 지름을 구하는 문제

트리의 지름

124

<https://www.acmicpc.net/problem/1167>

- 탐색 2번 이용한 소스: <http://boj.kr/ae22c0131578408b99c08c1064643e9d>
- 포스트 오더 이용한 소스: <http://boj.kr/2c1ff6d873f647b496b6baba6e2ad74b>

트리의 지름

125

<https://www.acmicpc.net/problem/1967>

- 트리의 지름을 구하는 문제

트리의 지름

126

<https://www.acmicpc.net/problem/1967>

- 탐색 2번 이용한 소스: <http://boj.kr/8c473026f8b74265bb65d68aa62c6d40>
- 포스트 오더 이용한 소스: <http://boj.kr/2697de4b9b1c439897d464a33421be98>