

트리 1

최백준 choi@startlink.io

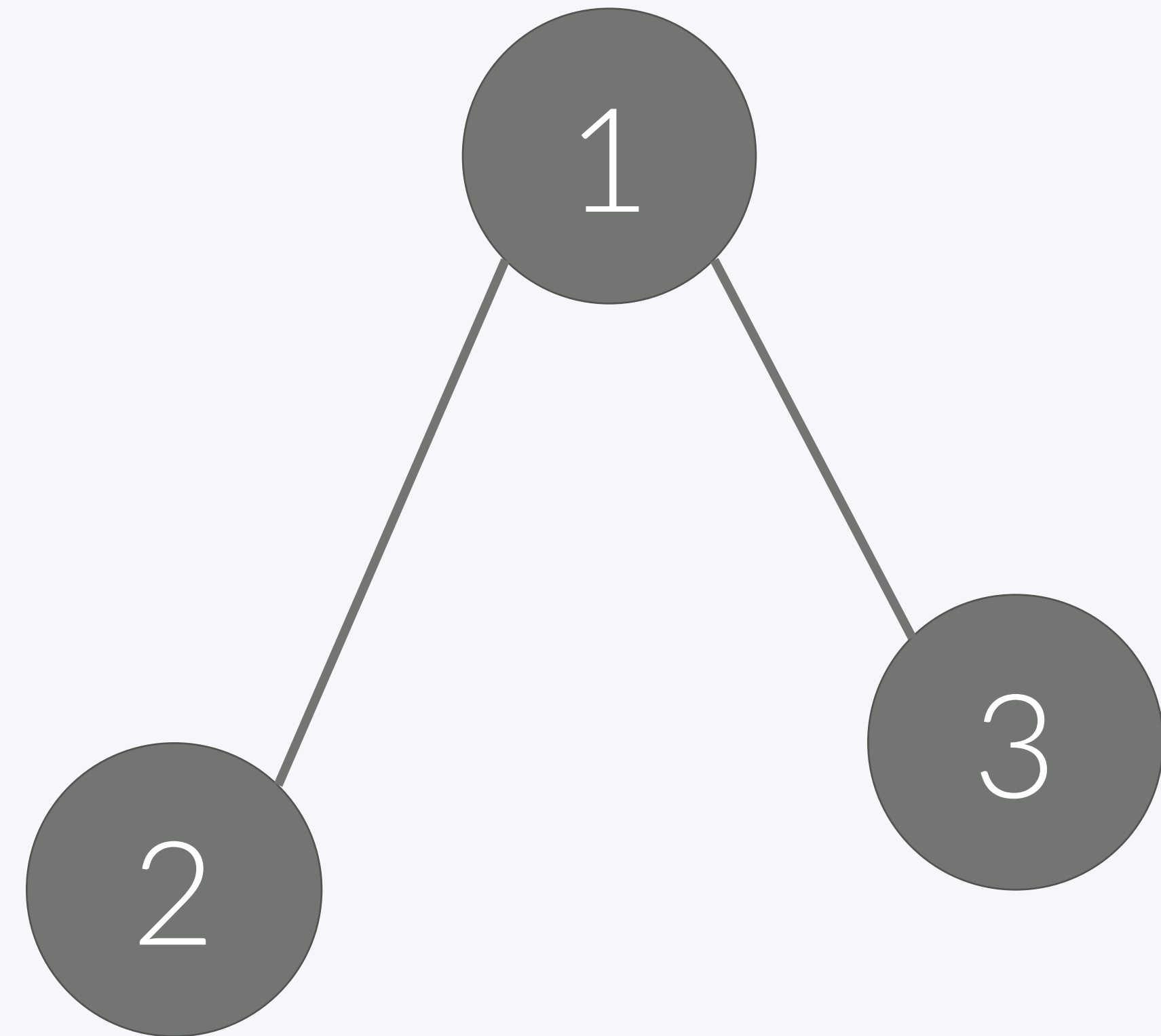
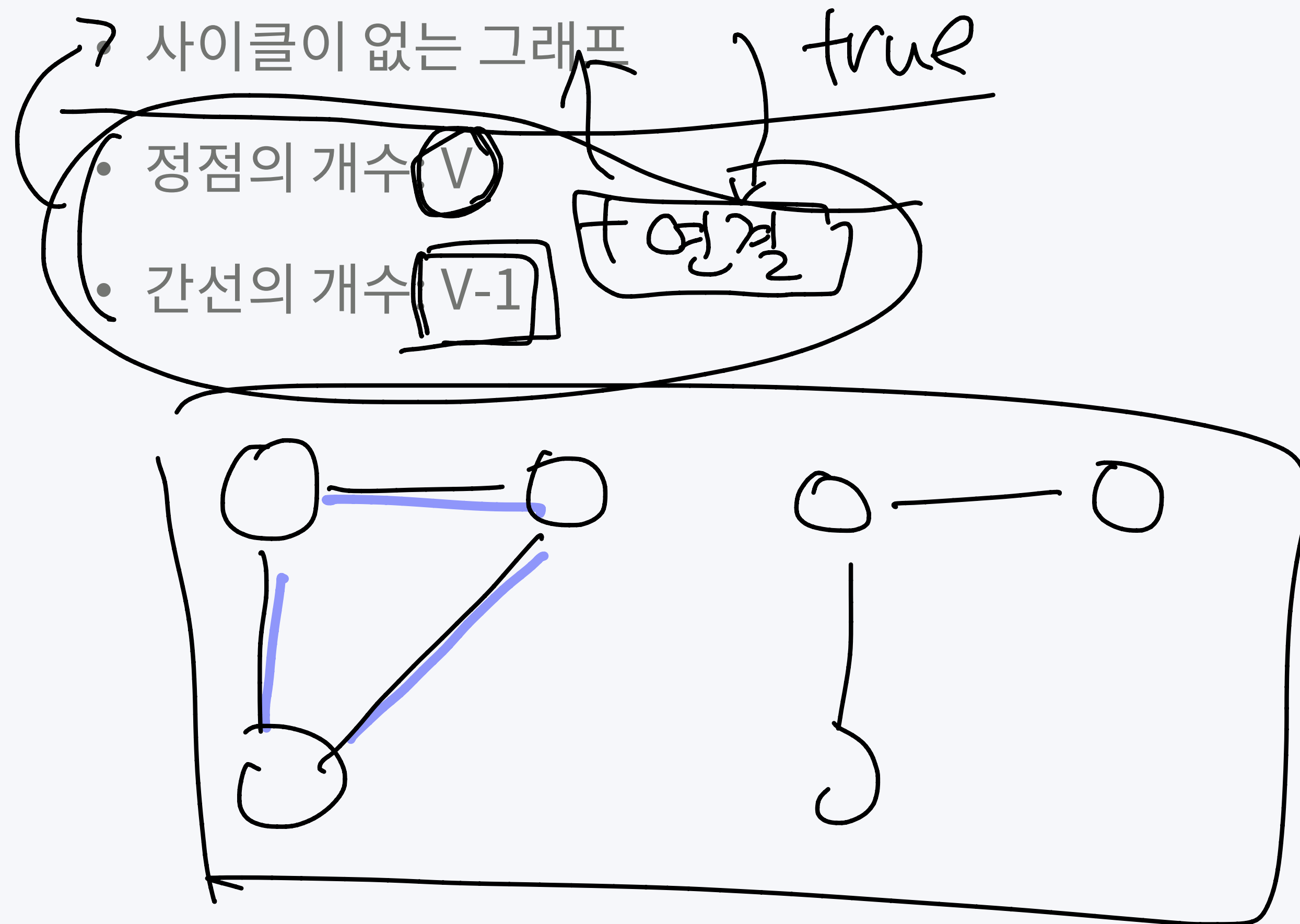
트리

트리

Tree

3

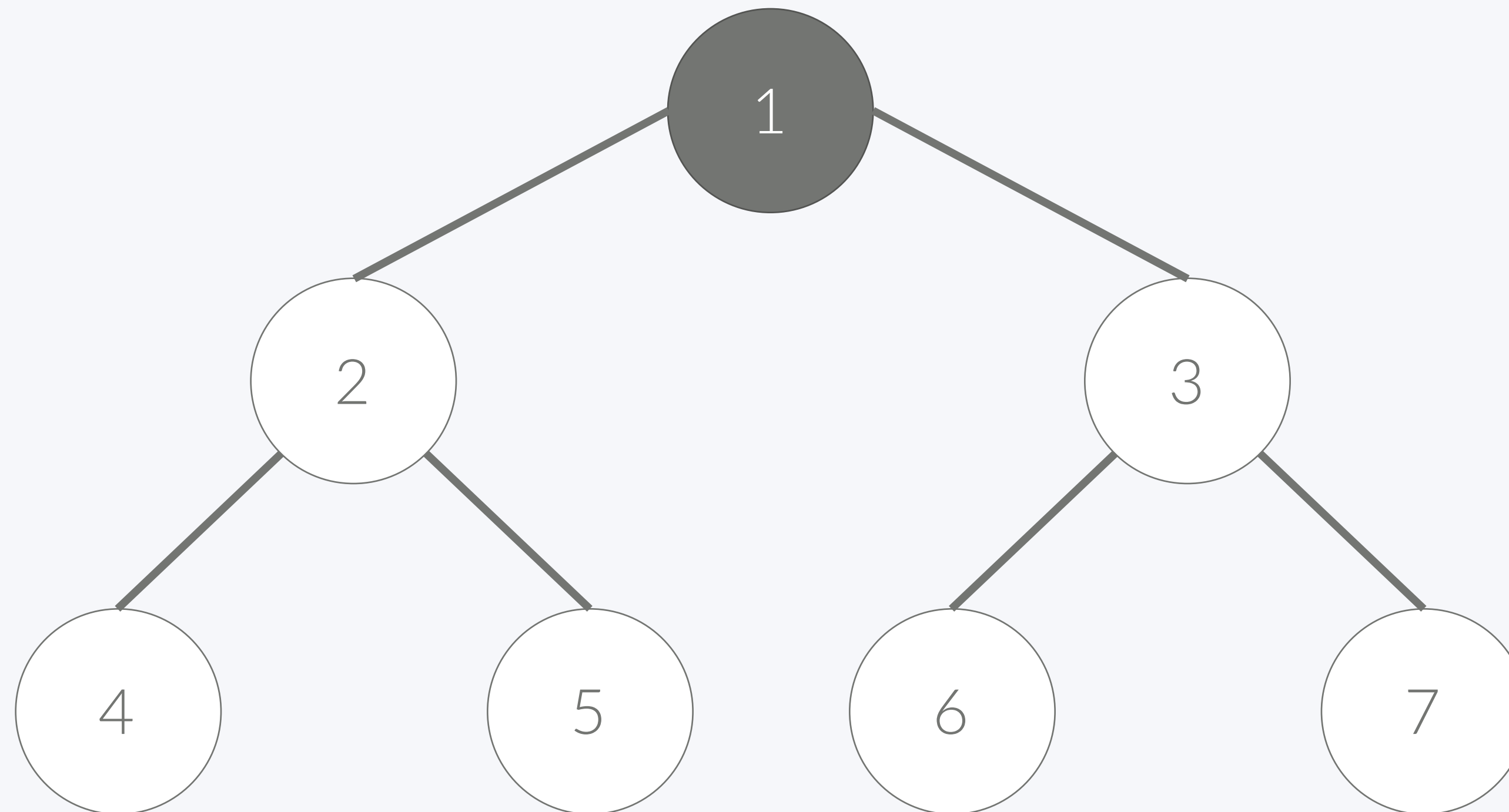
- 자료구조의 일종



루트 있는 트리

Rooted Tree

- 루트가 있는 트리
- 1번이 루트이다

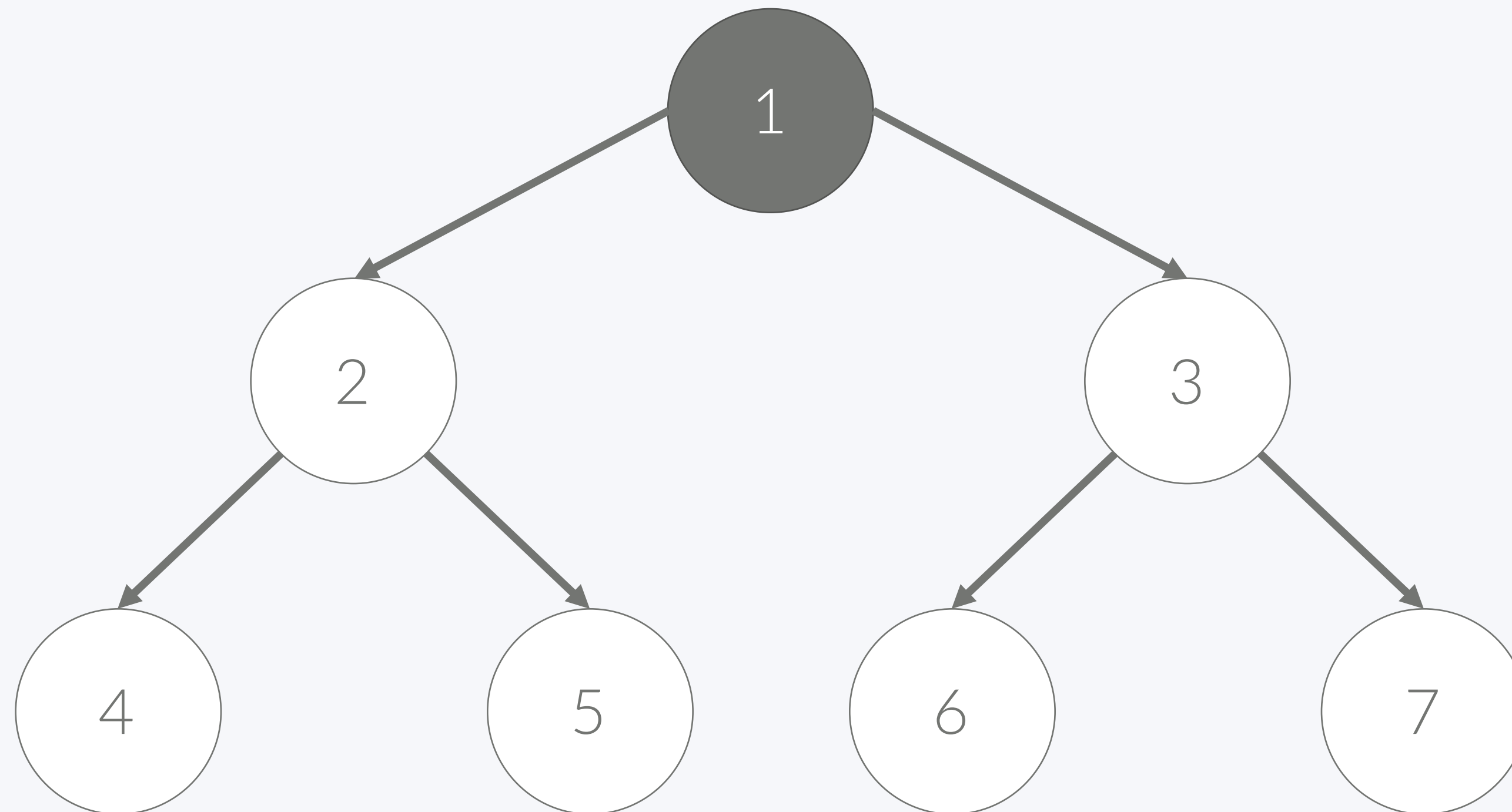


루트 있는 트리

Rooted Tree

5

- 루트 부터 아래로 방향을 정할 수 있다

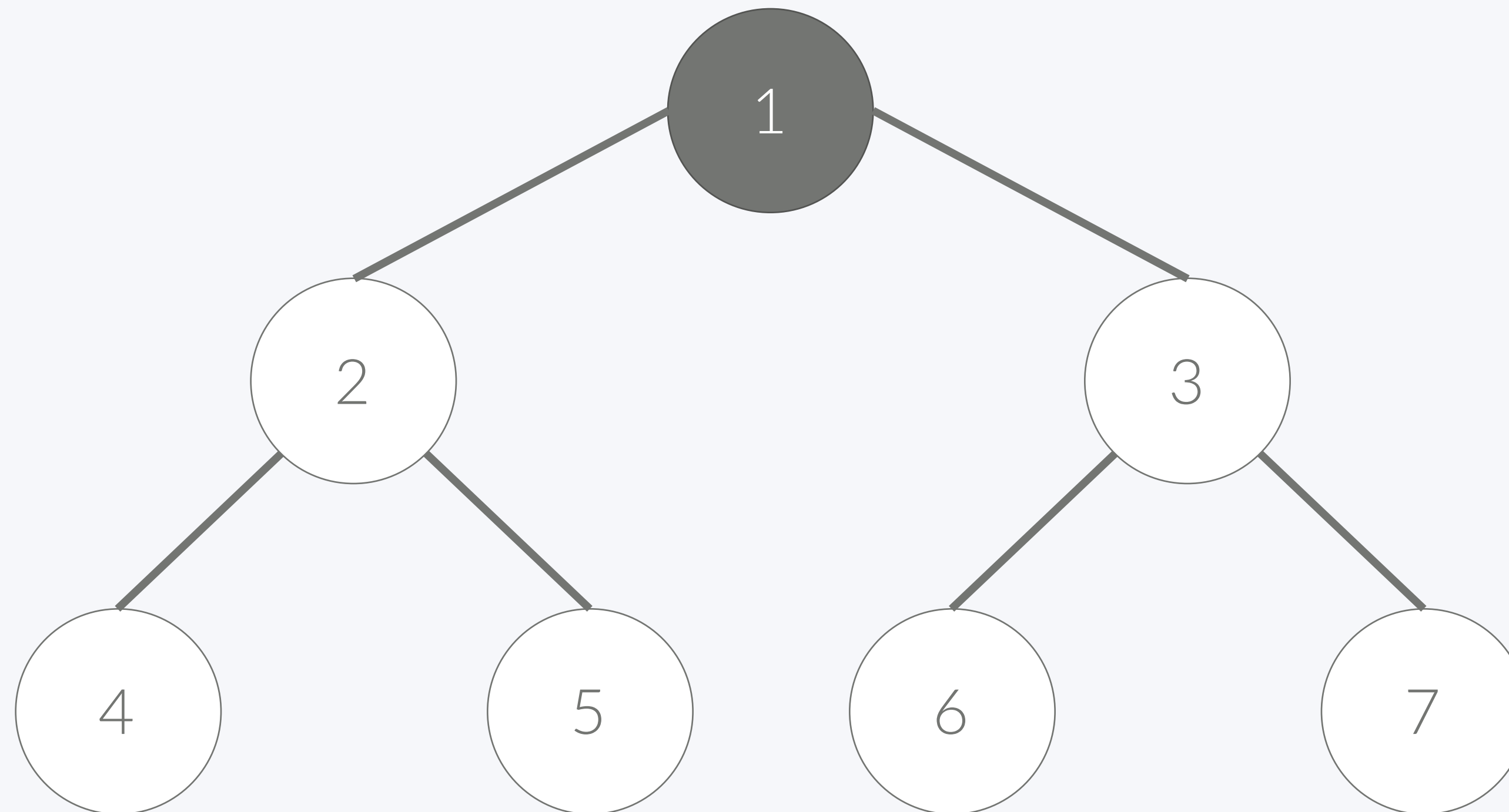


부모

Parent

- 1은 2의 부모
- 2는 4의 부모

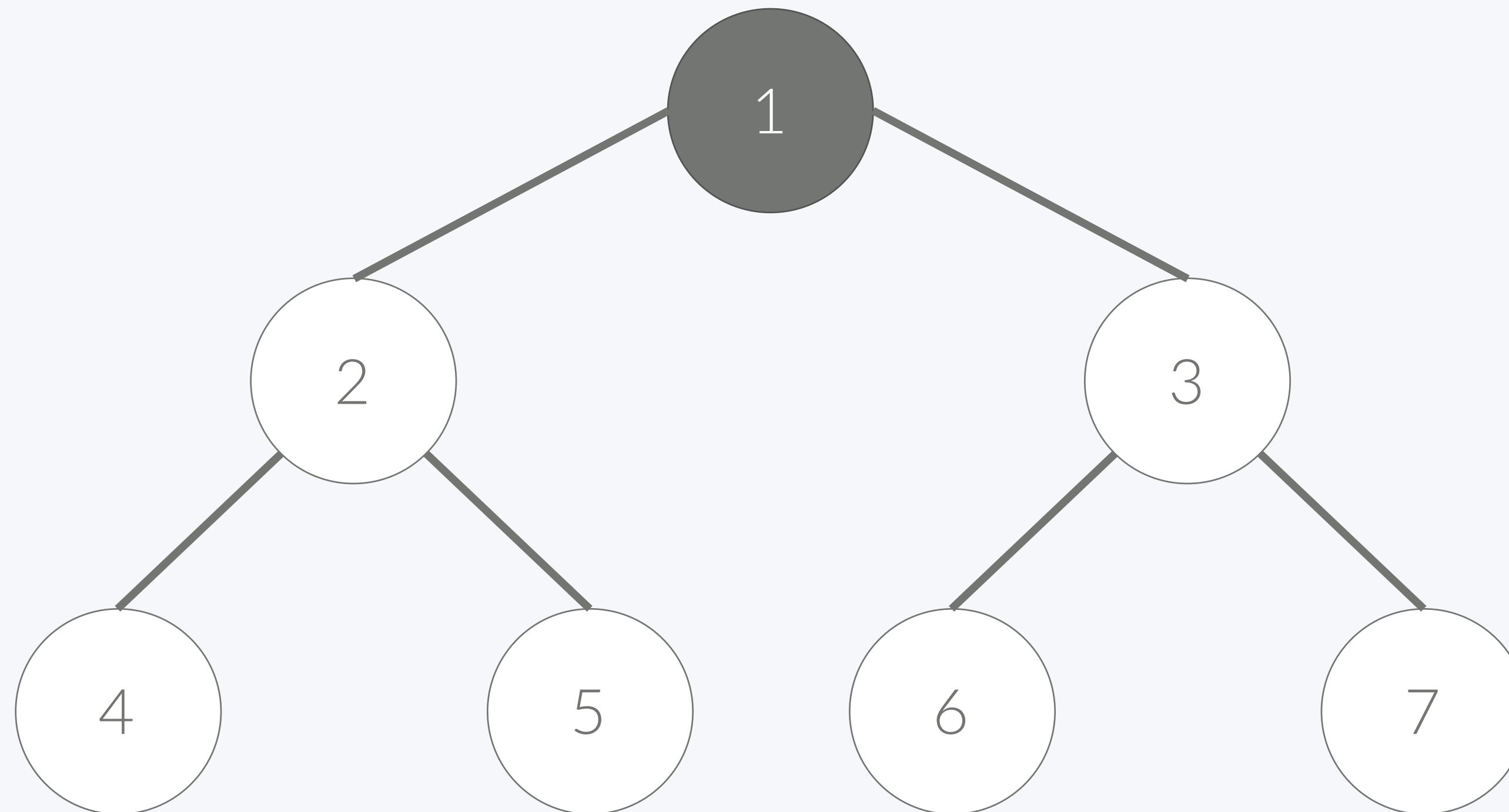
Parent $\times \Rightarrow \frac{2}{1} \bar{E}$



자식

Children

- 2는 1의 자식
- 4는 2의 자식
- 3의 자식: 6, 7

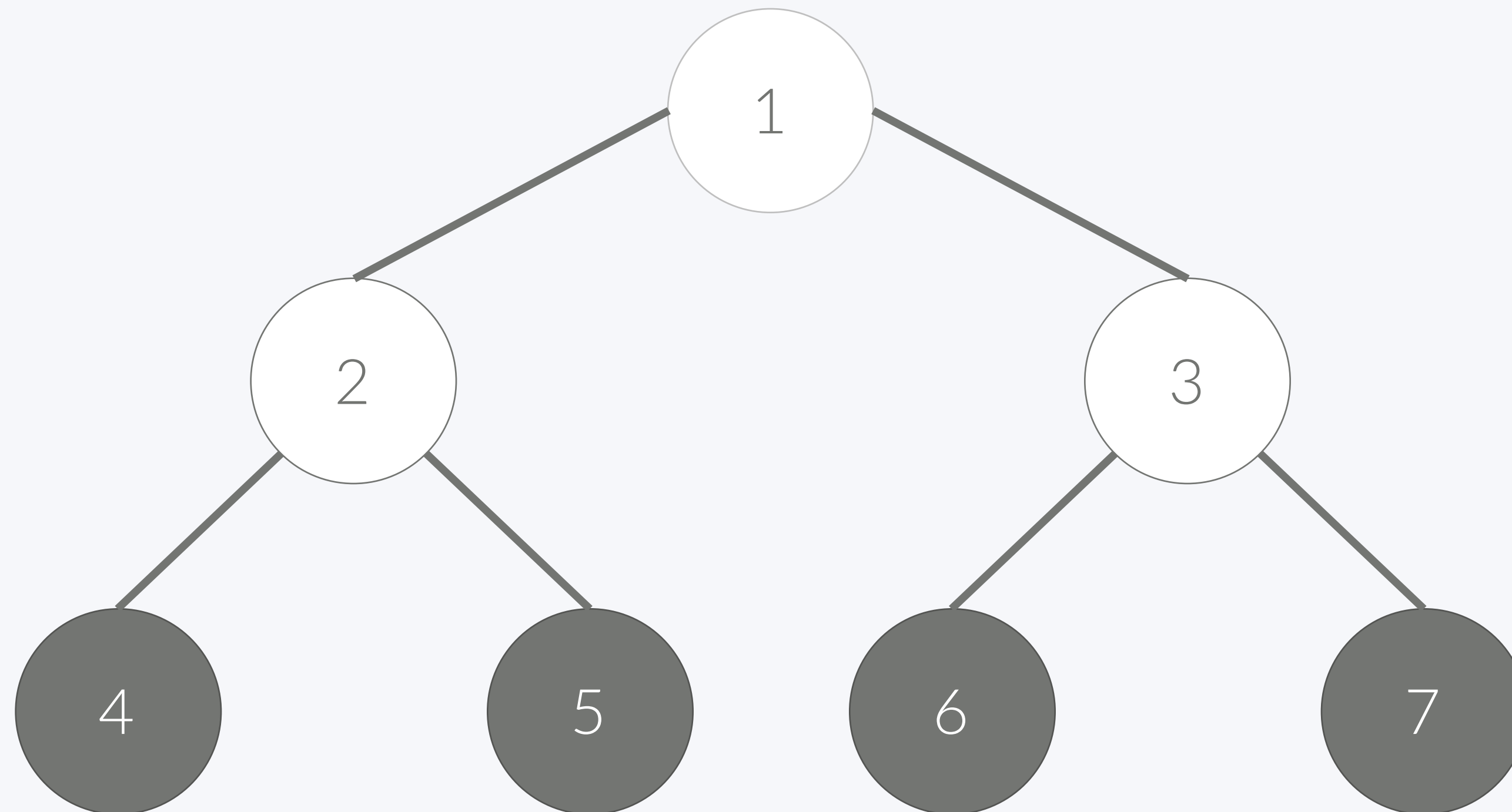


단말 정점

Leaf Node

- 4, 5, 6, 7

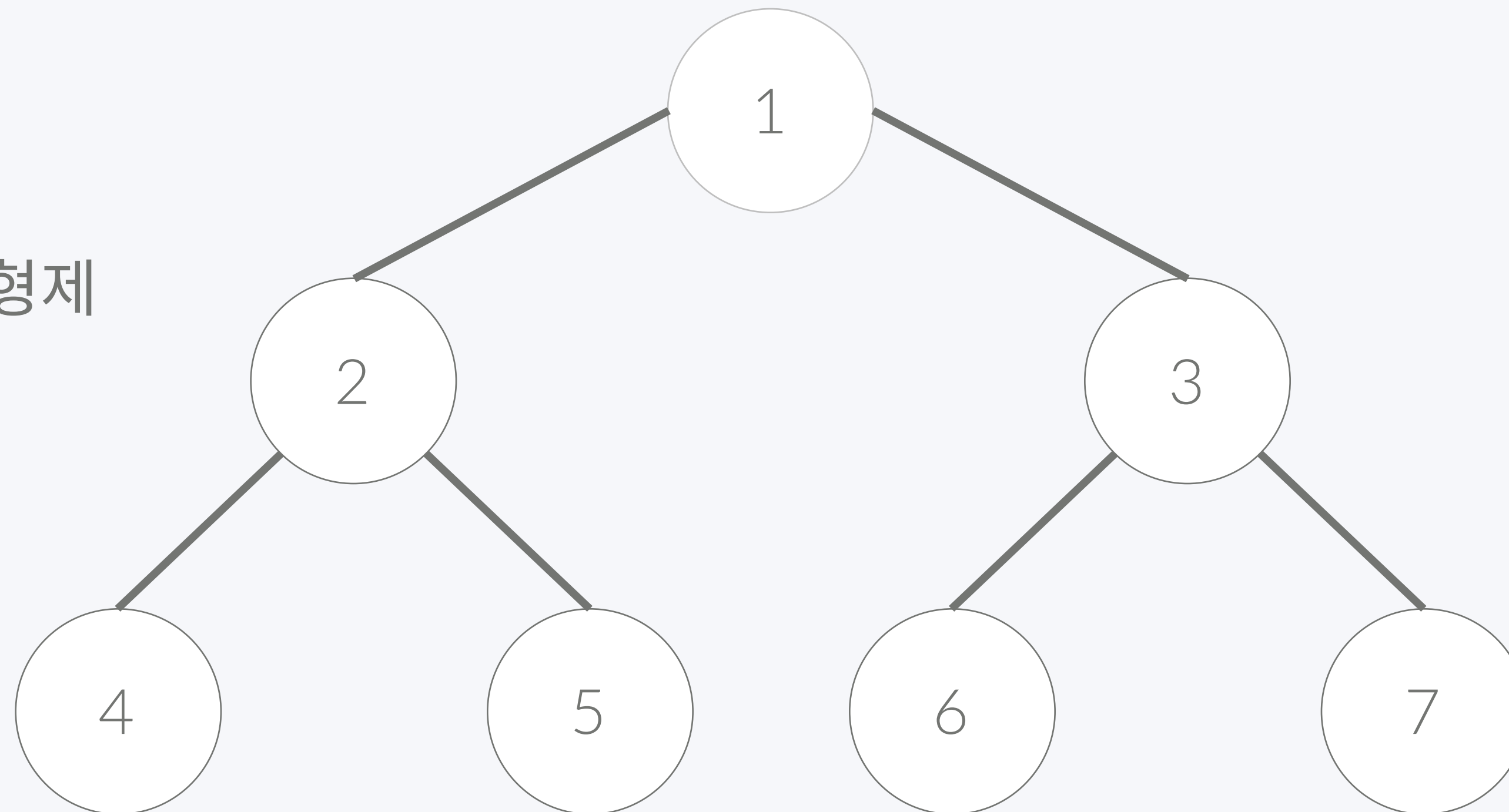
Terminal Node



형제

Sibling

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



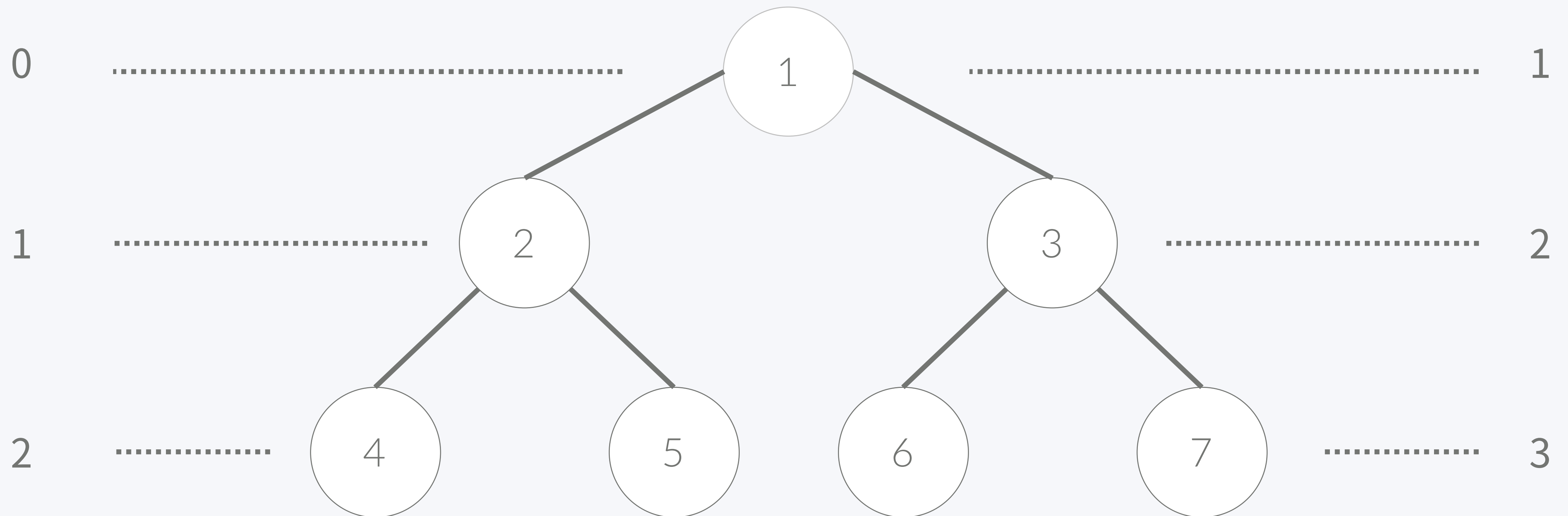
깊이

Depth

level

10

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

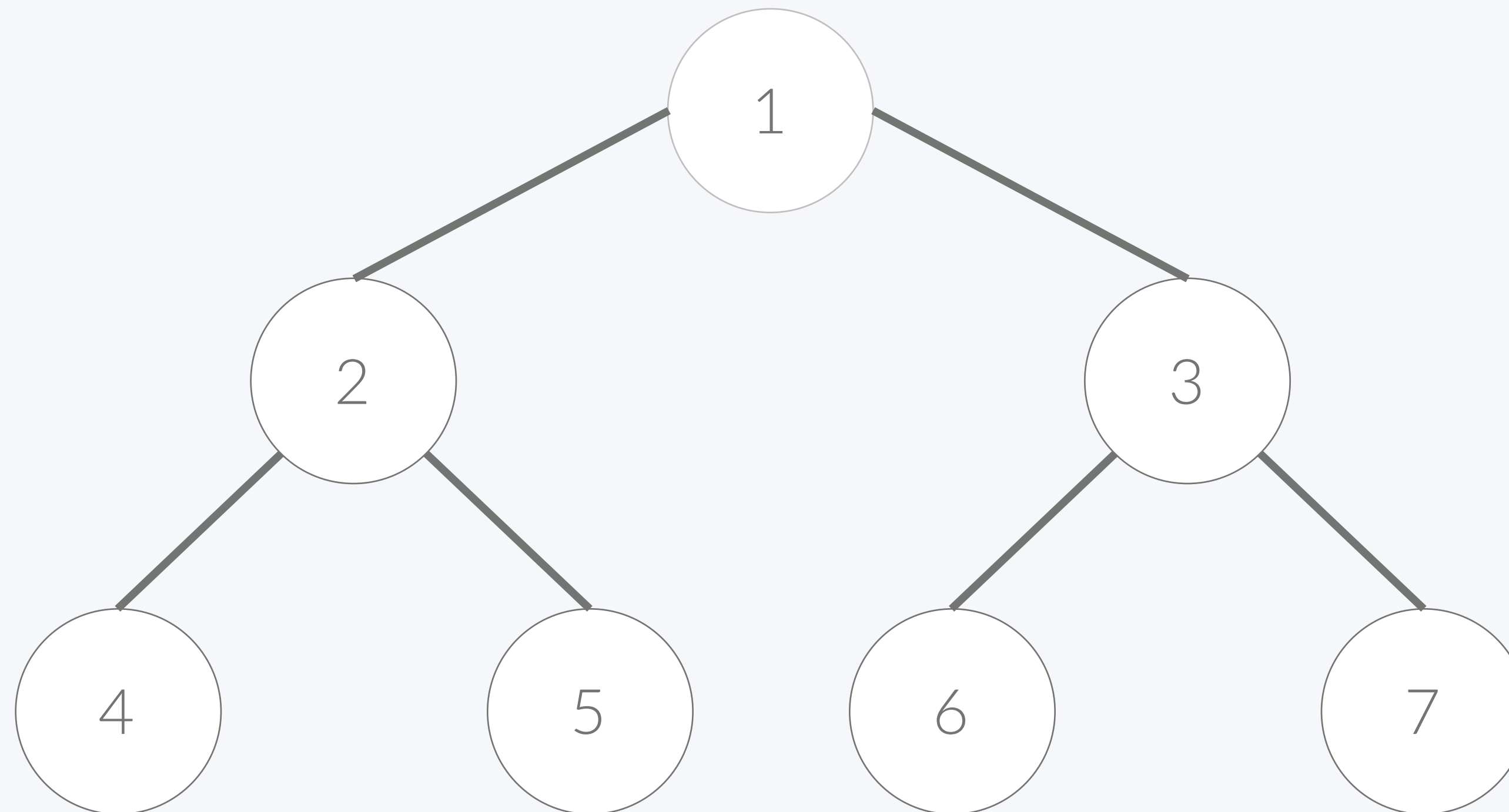


높이

Height

11

- 깊이 중 가장 큰 값 2 또는 3

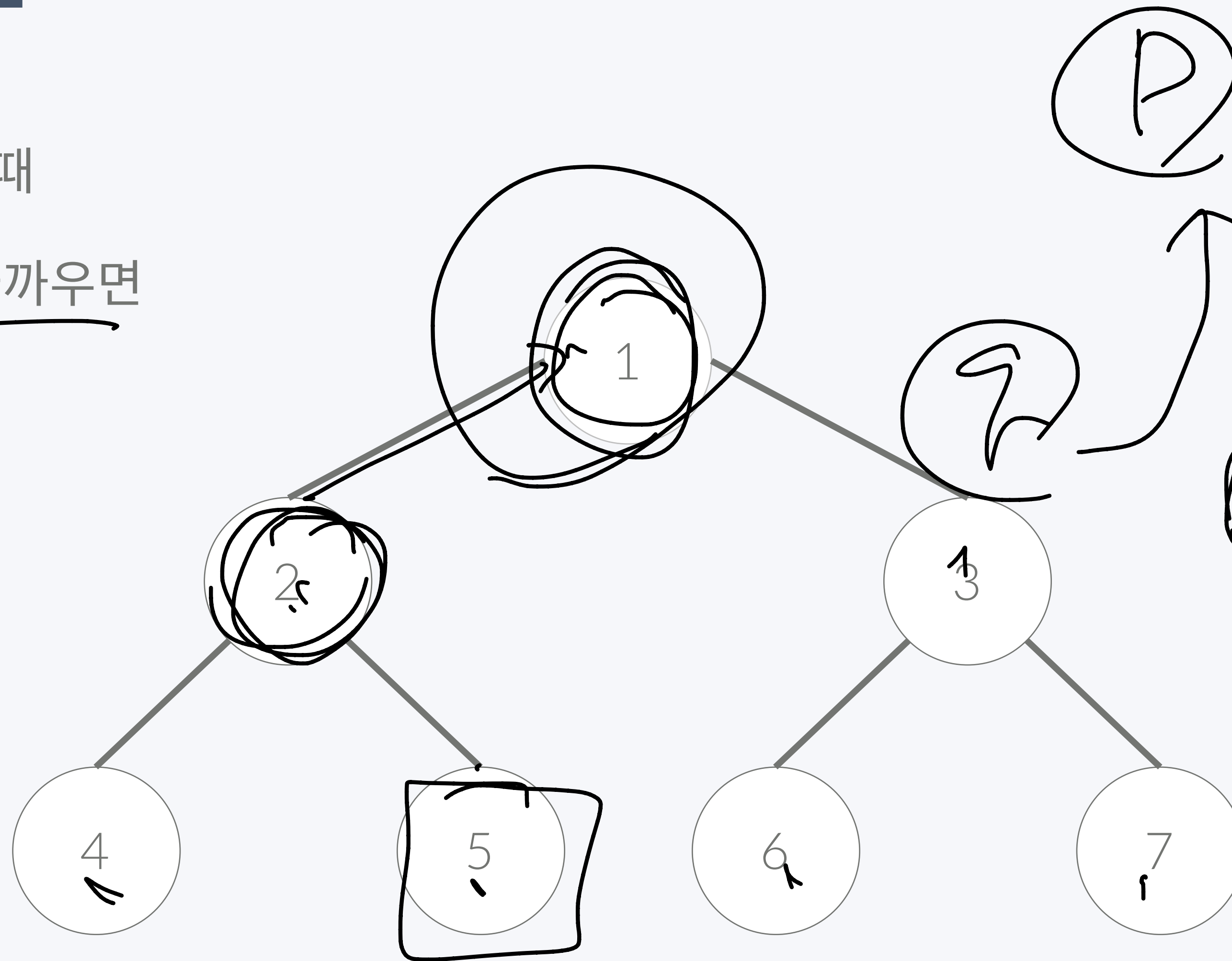


조상, 자손

Ancestor, Descendent

12

- $p \rightarrow q$ 로 갈 수 있을 때
- p가 q보다 루트에 가까우면
- p는 q의 조상
- q는 p의 자손

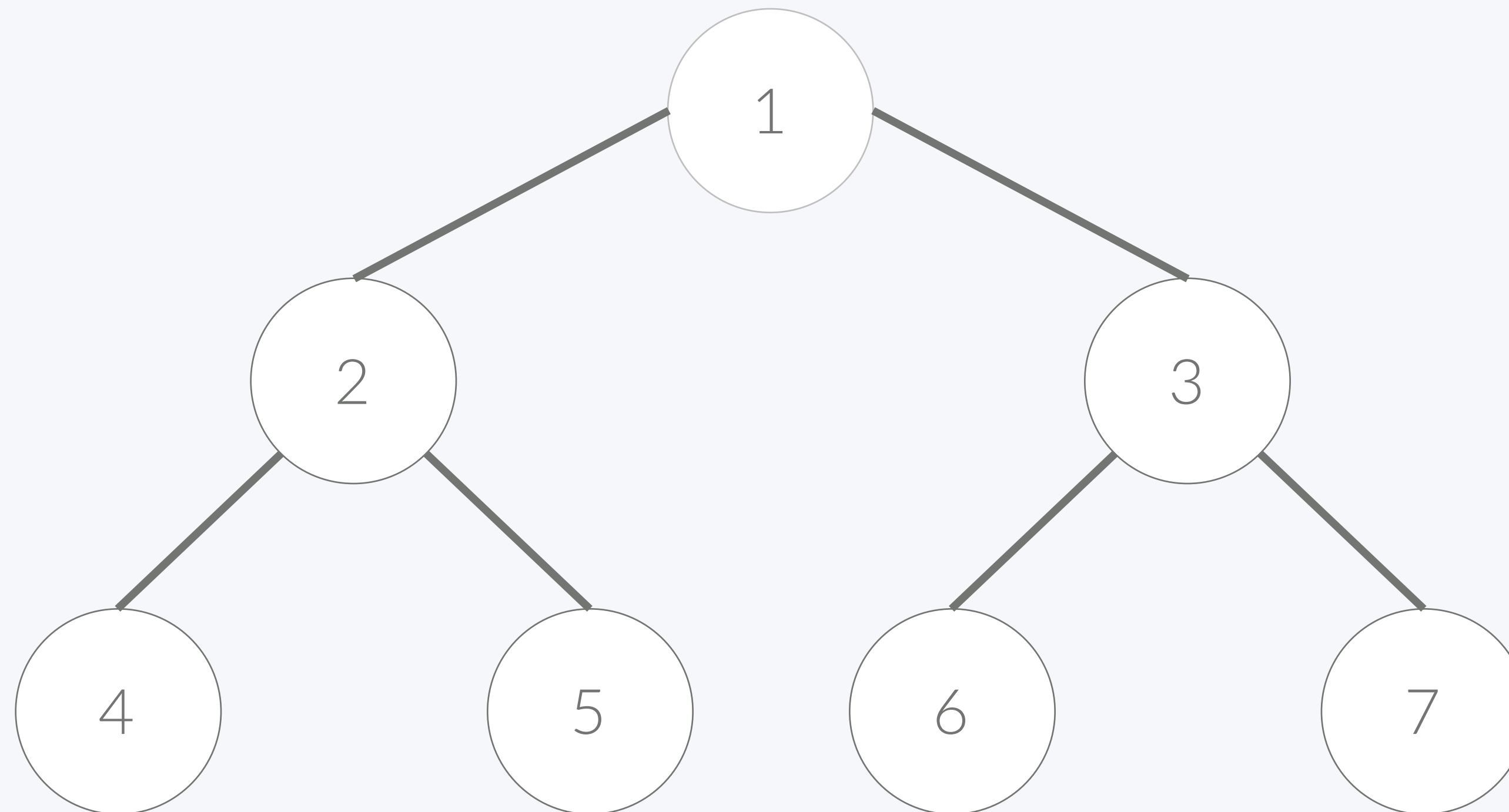


이진 트리

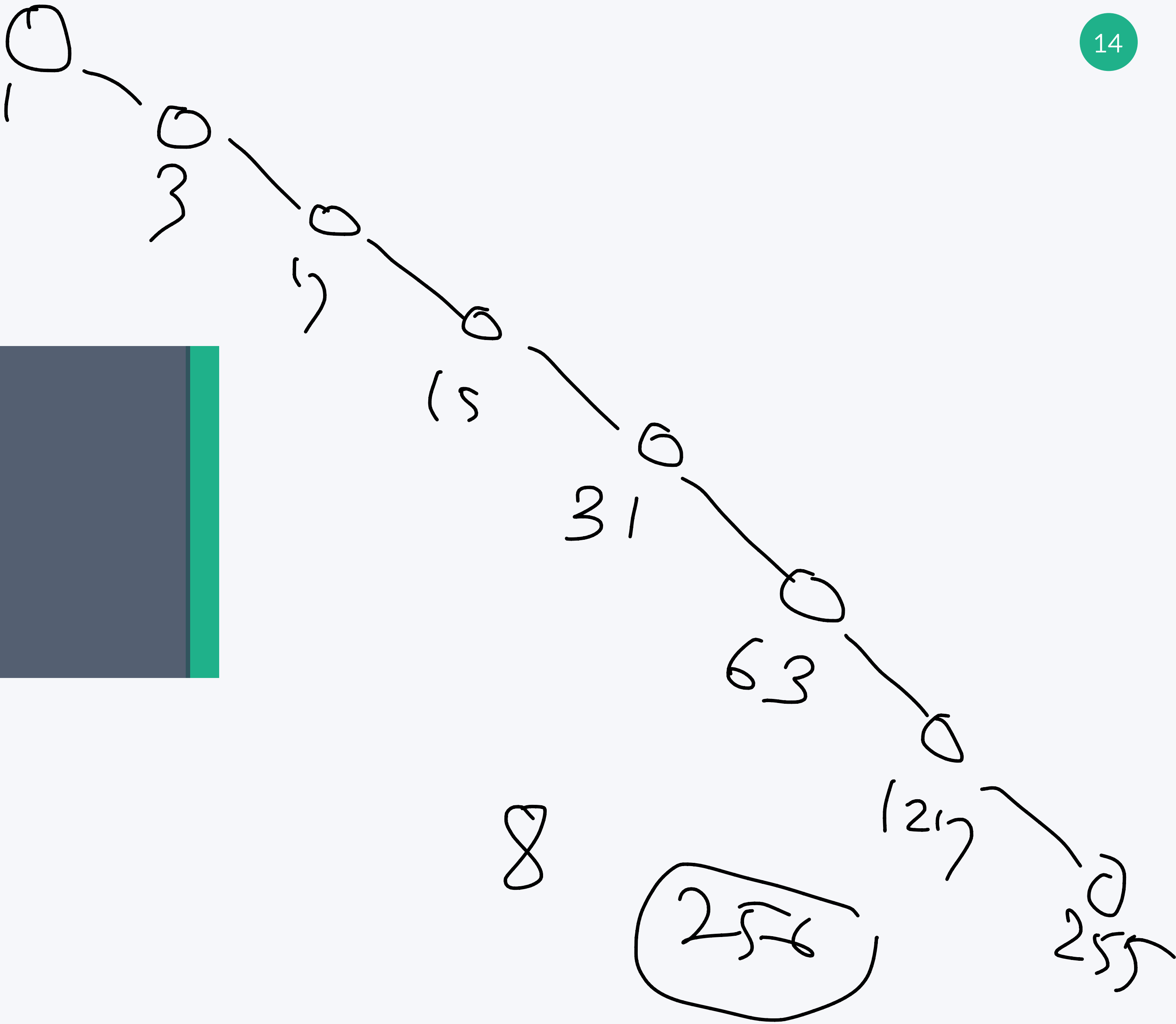
Binary Tree

13

- 자식을 최대 2개만 가지고 있는 트리

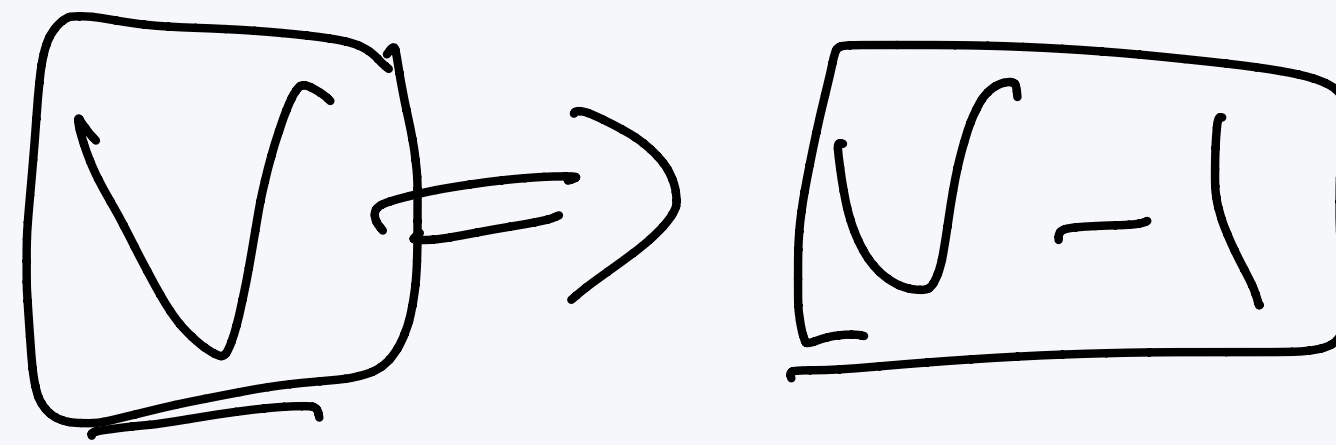


트리의 표현



트리의 표현

Representation of Tree



인접행렬: 비효율
인접리스트

- 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
(1개) \Rightarrow 2개
- 부모가 0개인 경우는 트리의 루트인데, 이 경우 부모를 -1이나 0으로 처리하는 방식을 사용한다

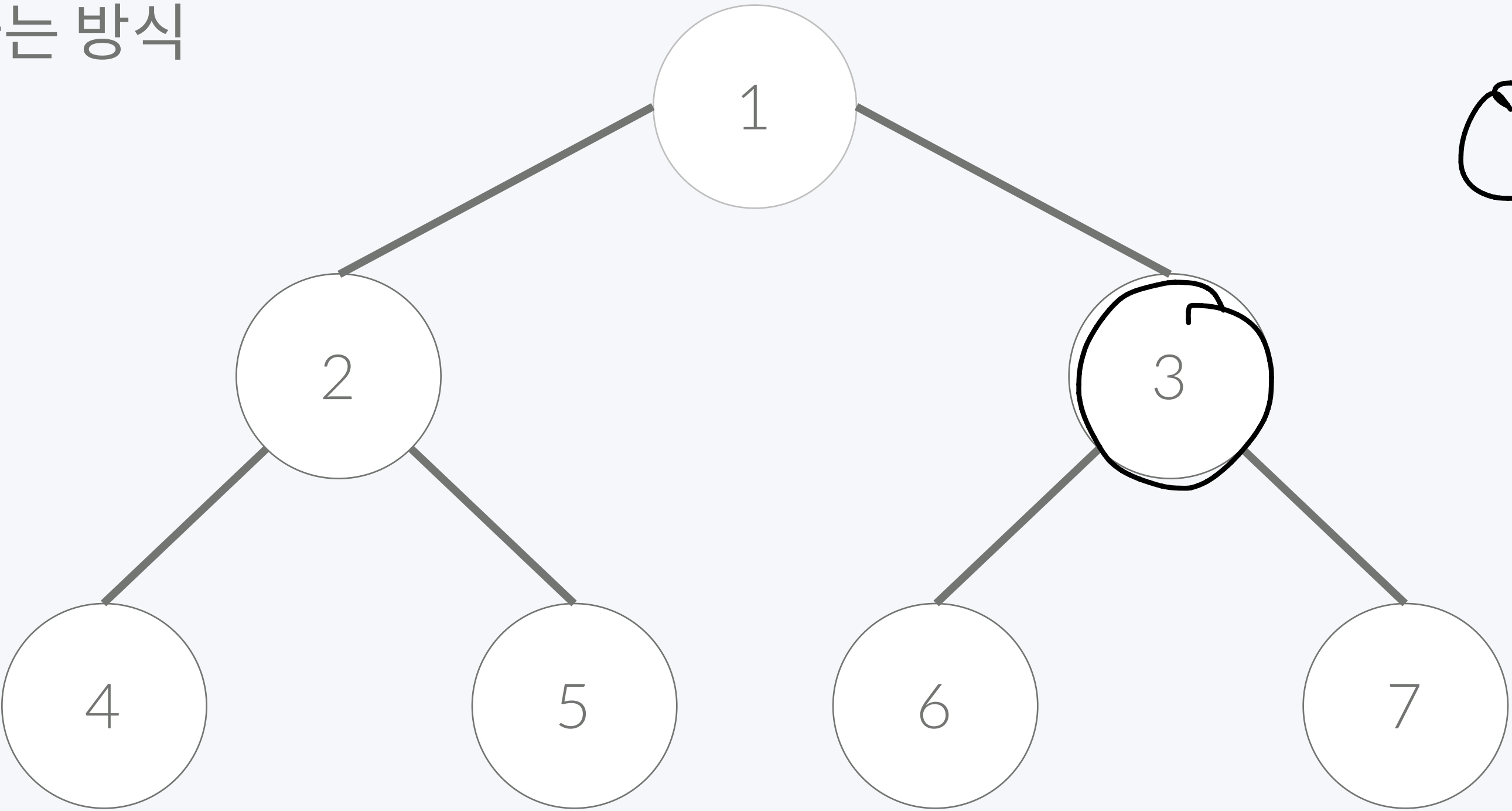
트리의 표현

Representation of Tree

- 트리의 부모만 저장하는 방식

$O(1)$

$O(V)$



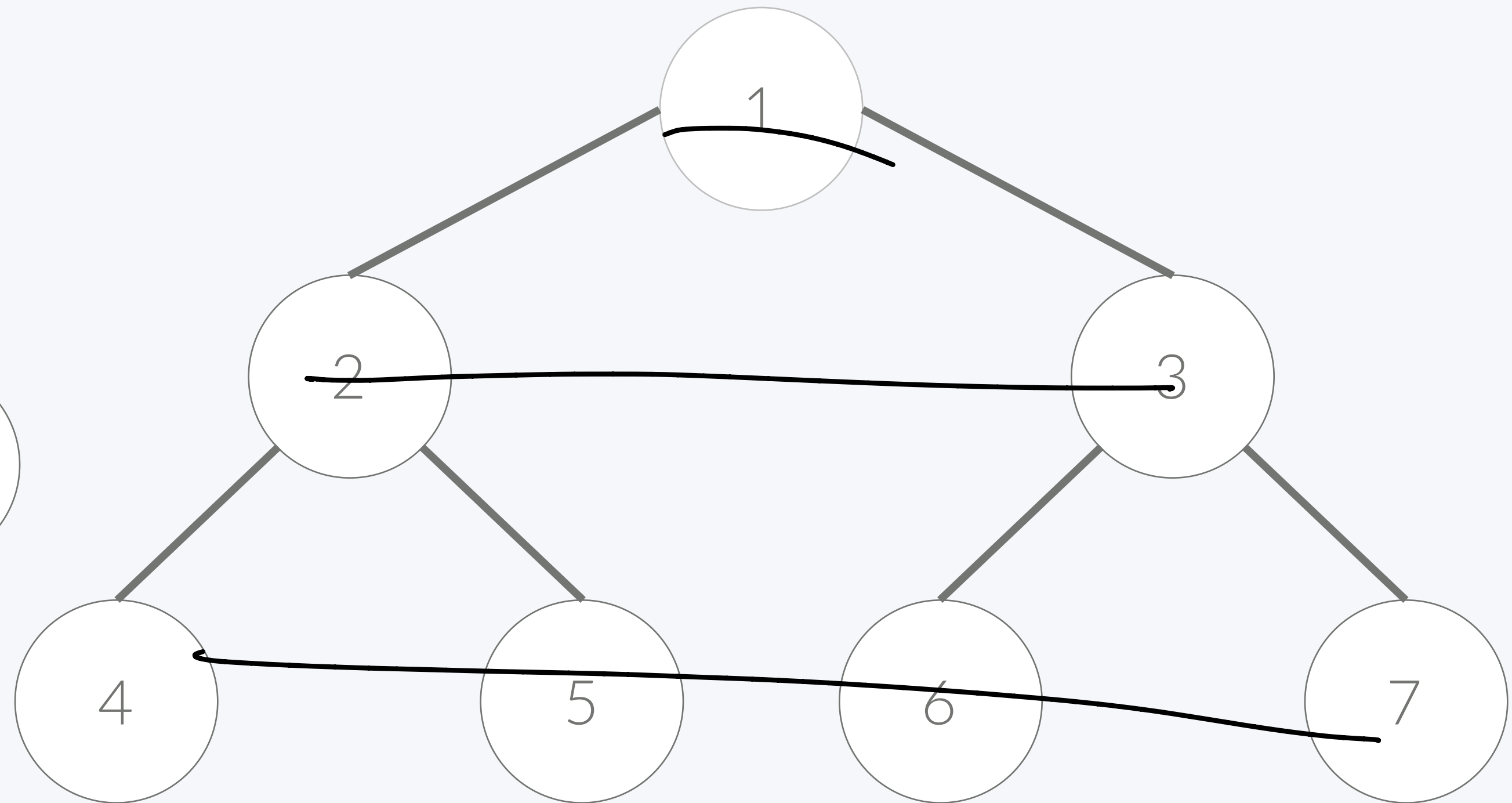
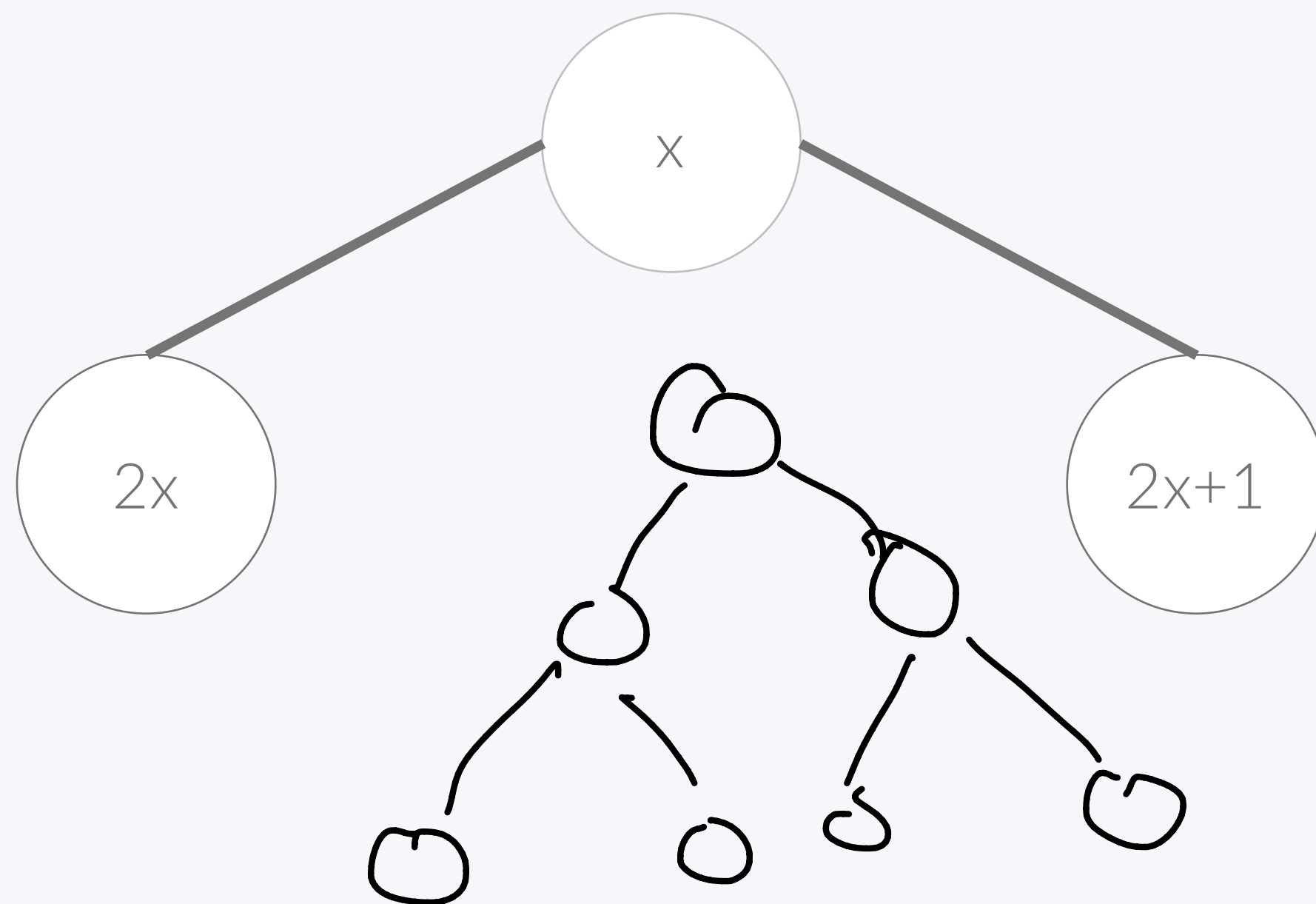
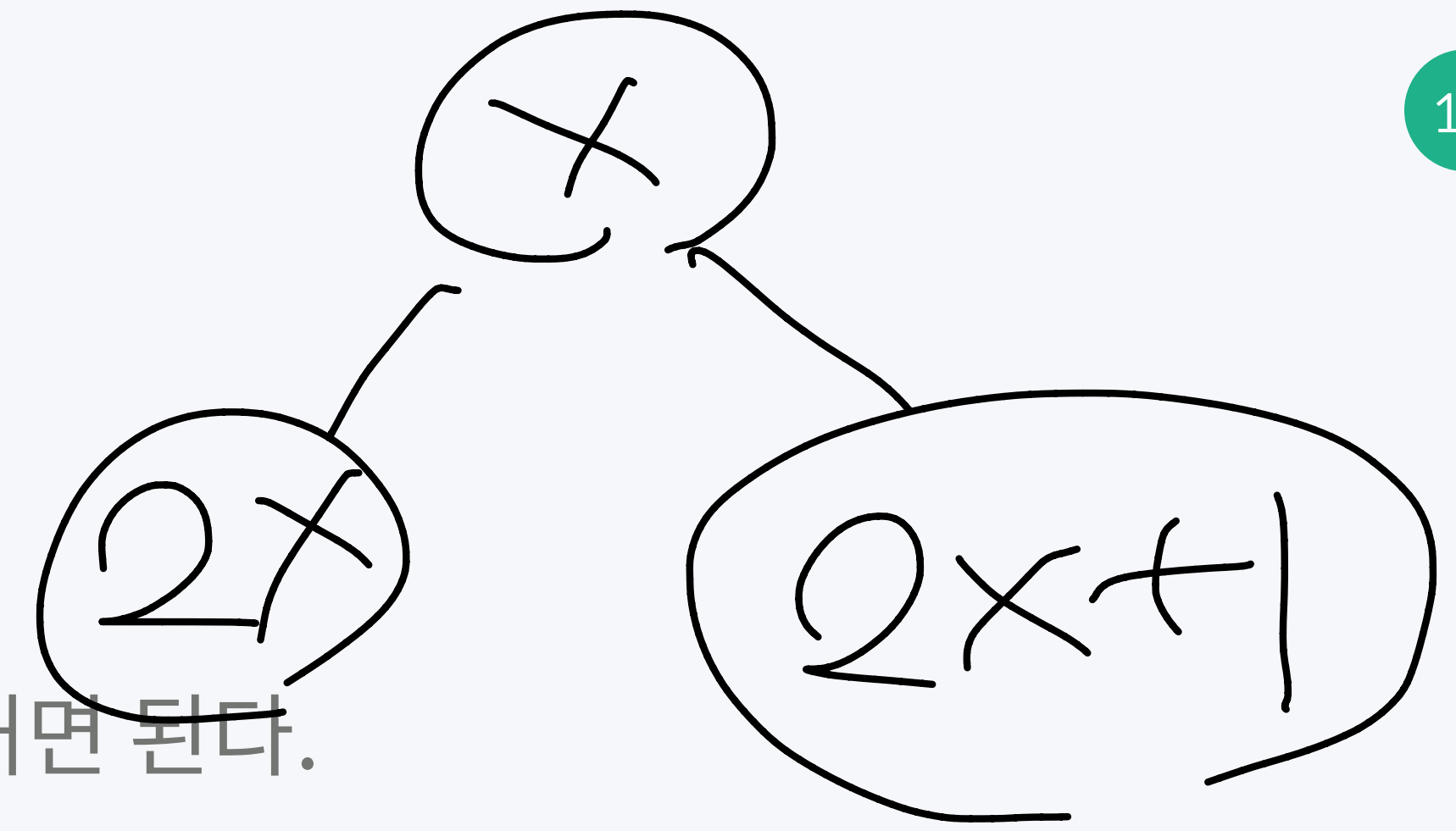
i	1	2	3	4	5	6	7
parent[i]	0	1	1	2	2	3	3

트리의 표현

17

Representation of Tree

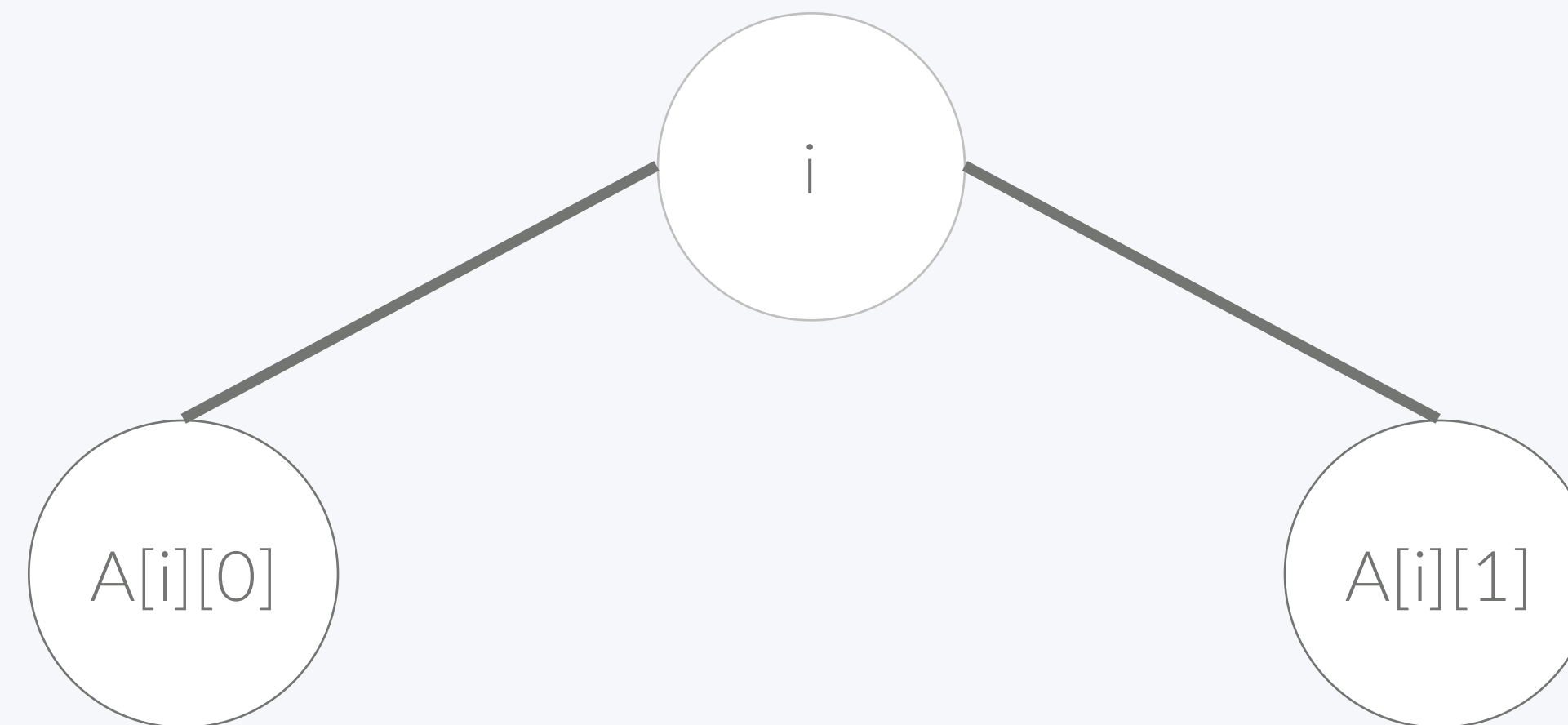
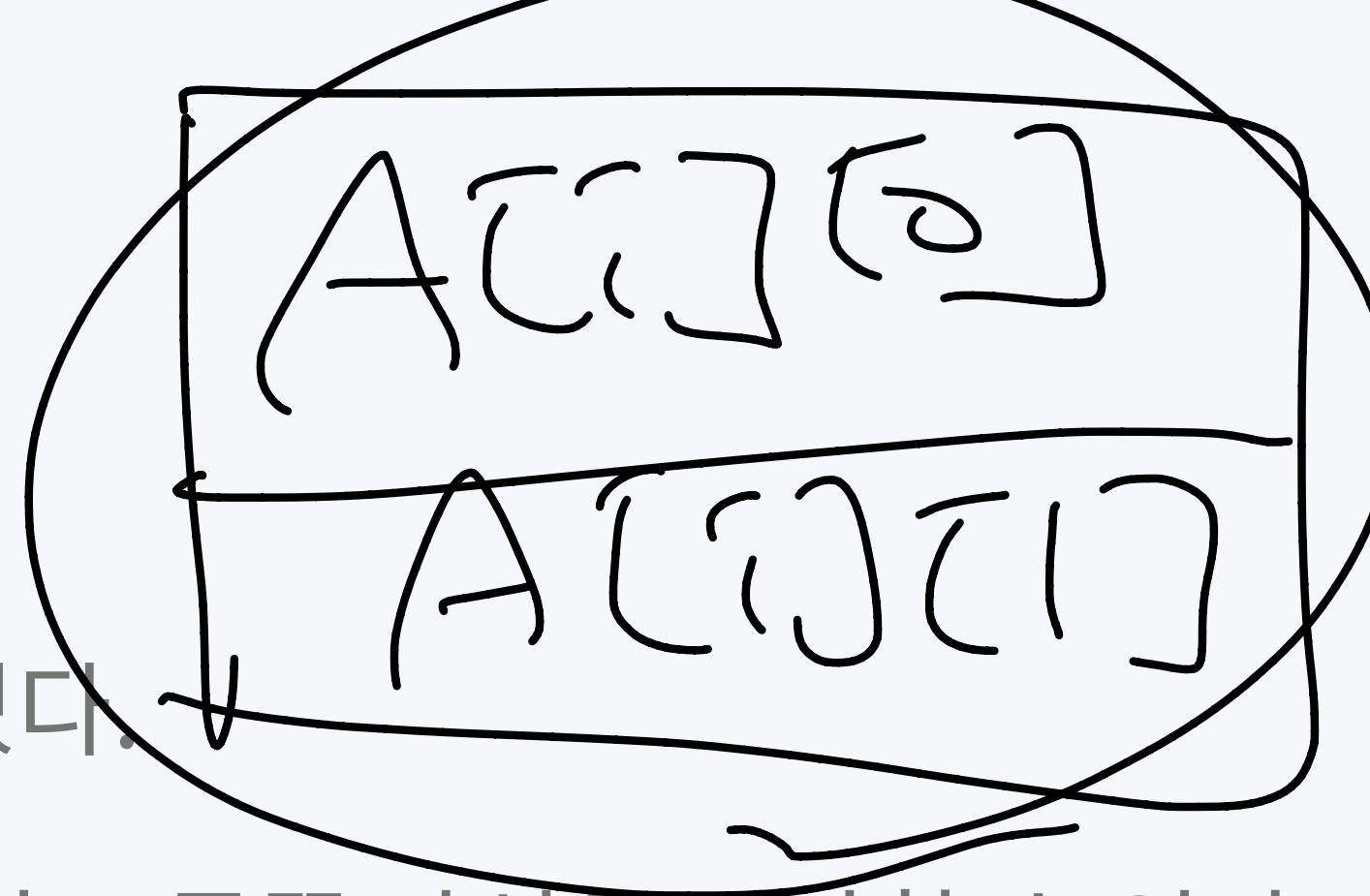
- 이진 트리의 경우에는 배열로 표현할 수 있다.
- 부모의 노드가 x 인 경우에 자식의 노드는 $2x$, $2x+1$ 로 나타내면 된다.



트리의 표현

Representation of Tree

- 이진 트리의 경우에는 배열로 표현할 수 있다.
- 또는, $A[i][0]$ 에 i 의 왼쪽 자식, $A[i][1]$ 에 i 의 오른쪽 자식을 저장할 수 있다.



이진 트리

DFS

BFS

트리의 순회

트리의 순회

Tree Traversal

- 트리의 모든 노드를 방문하는 순서이다.
- 그래프의 경우에는 DFS와 BFS가 있었다
- 트리에서도 위의 두 방법을 사용할 수 있지만, 트리에서만 사용할 수 있는 세 방법이 있다.
- 프리오더 전위 순회
- 인오더 중위 순회
- 포스트오더 후위 순회
- 세 방법의 차이는 노드 방문을 언제 하냐의 차이이다

트리의 순회

Tree Traversal

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

트리의 순회

Tree Traversal

- 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

Binary Tree

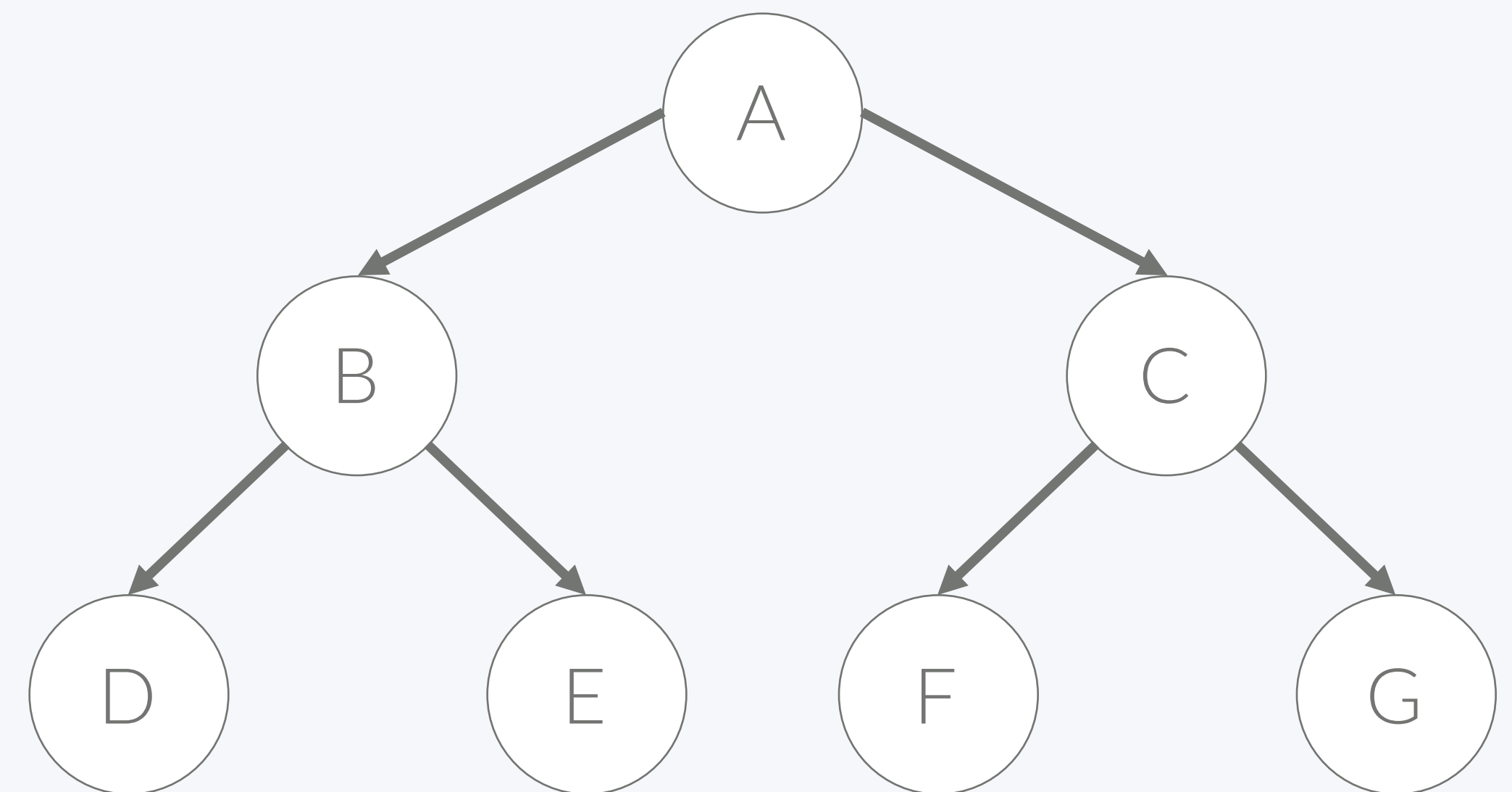
23

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

프리오더

Pre-order

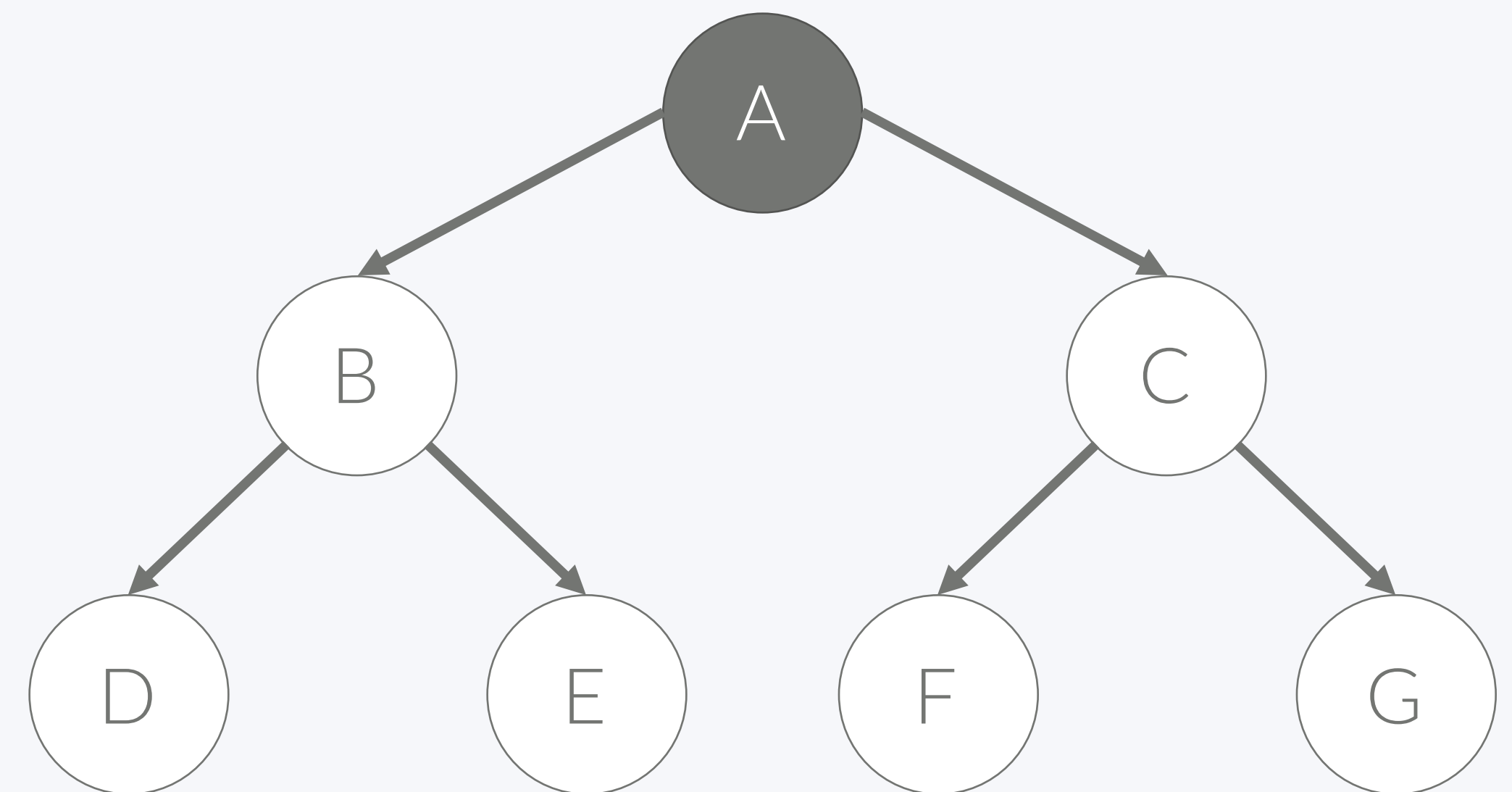
- 노드 방문
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

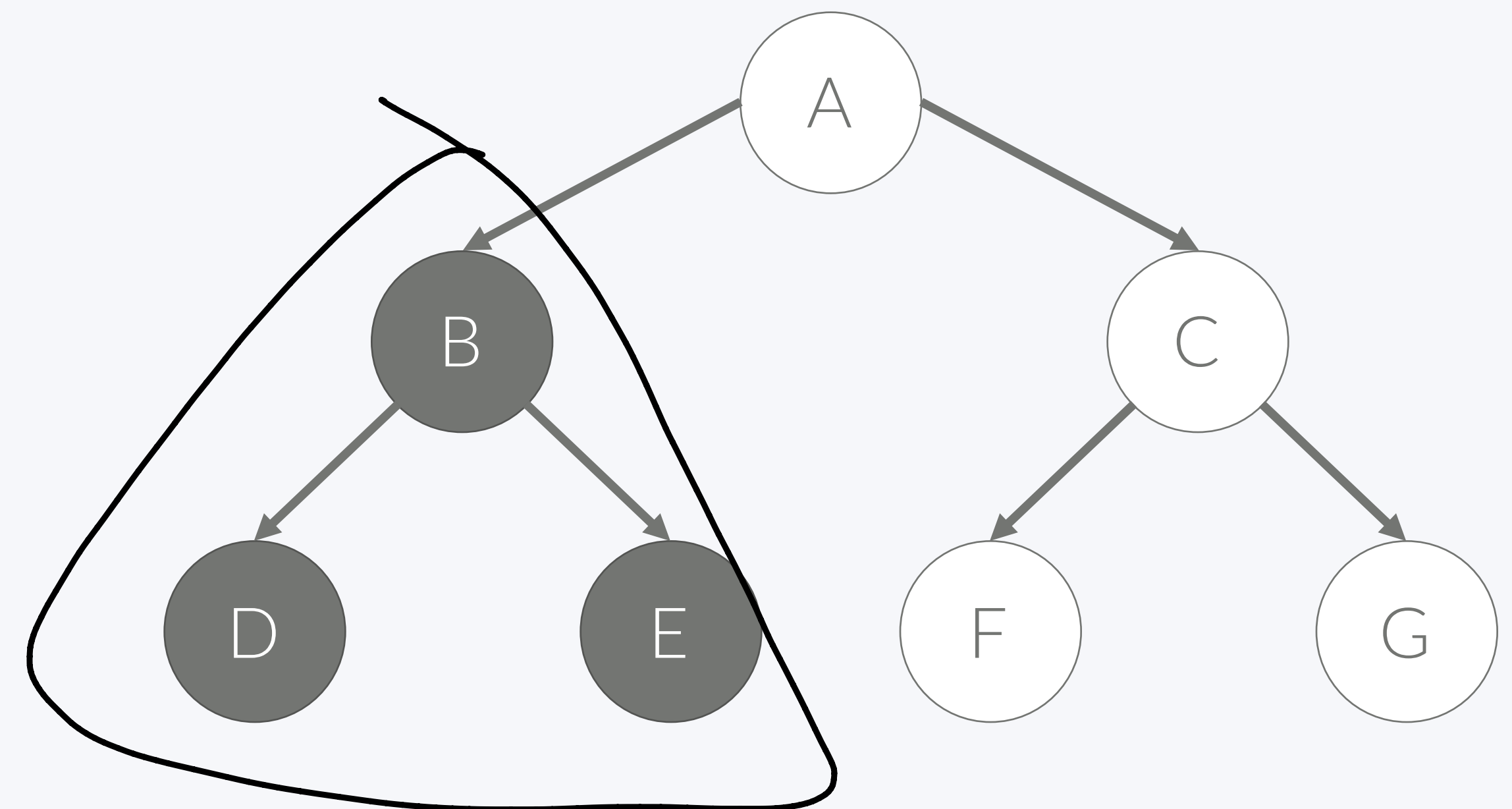
- 노드 방문
 - A
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

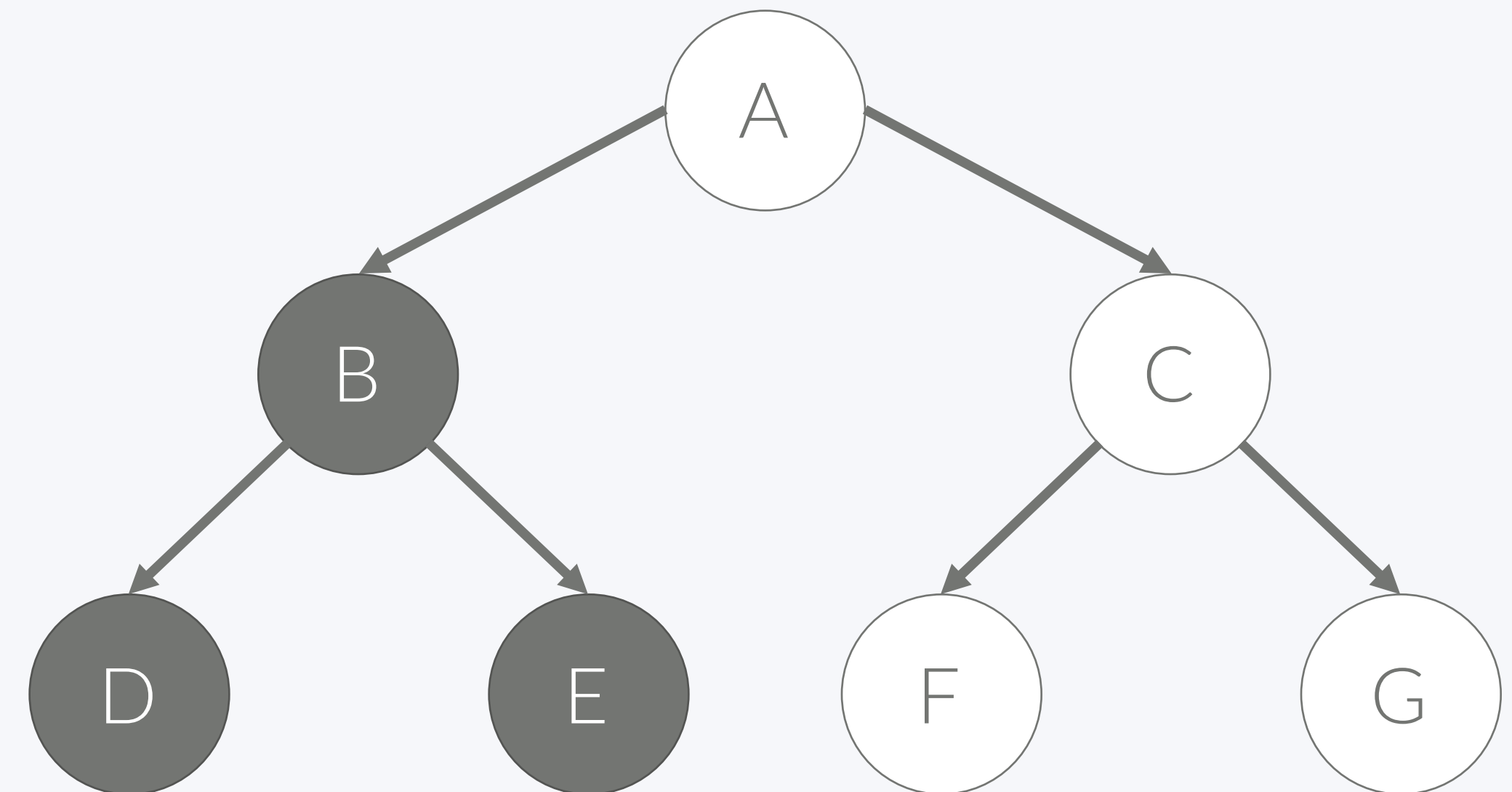
- 노드 방문
 - A
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

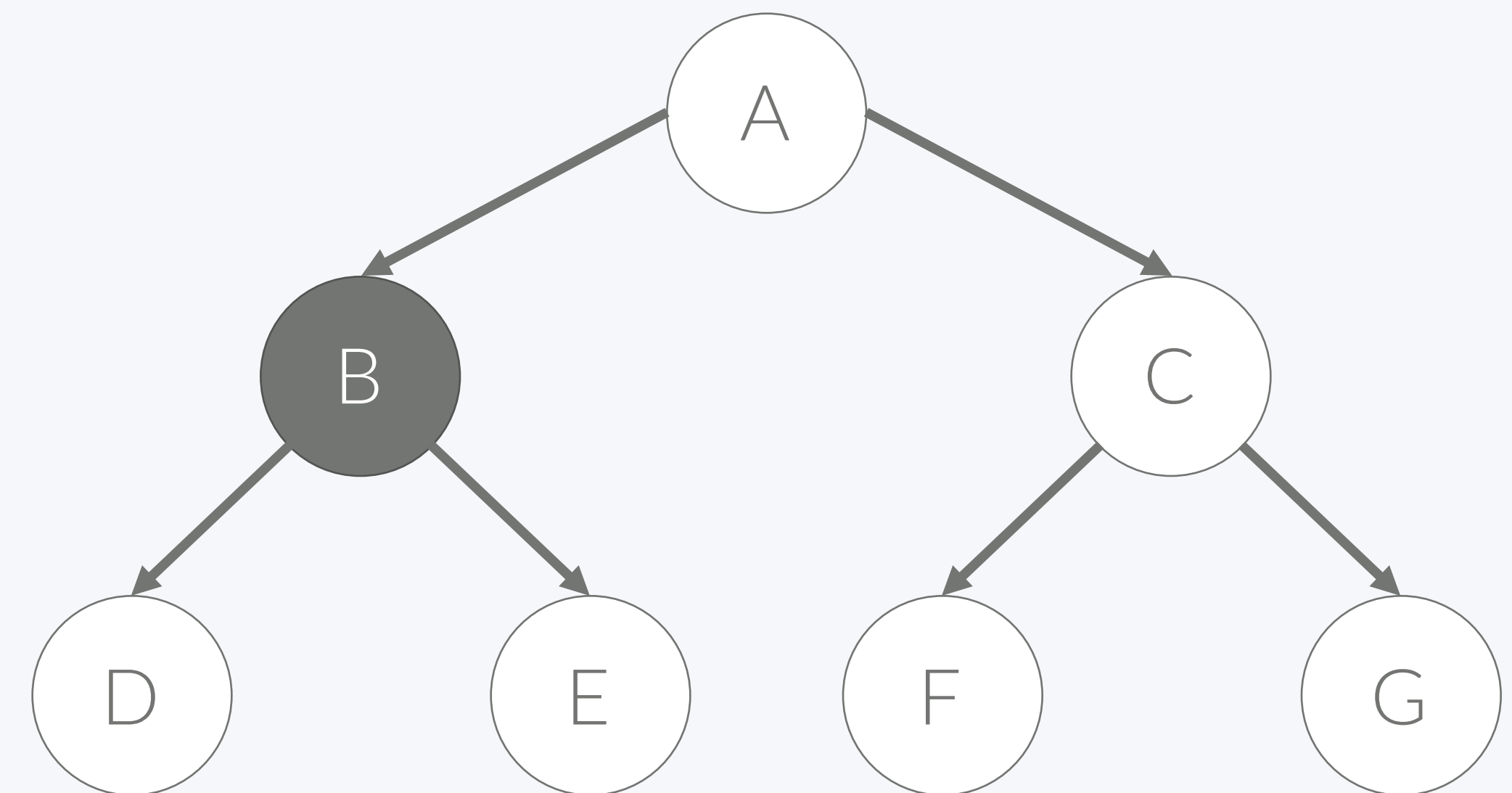
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

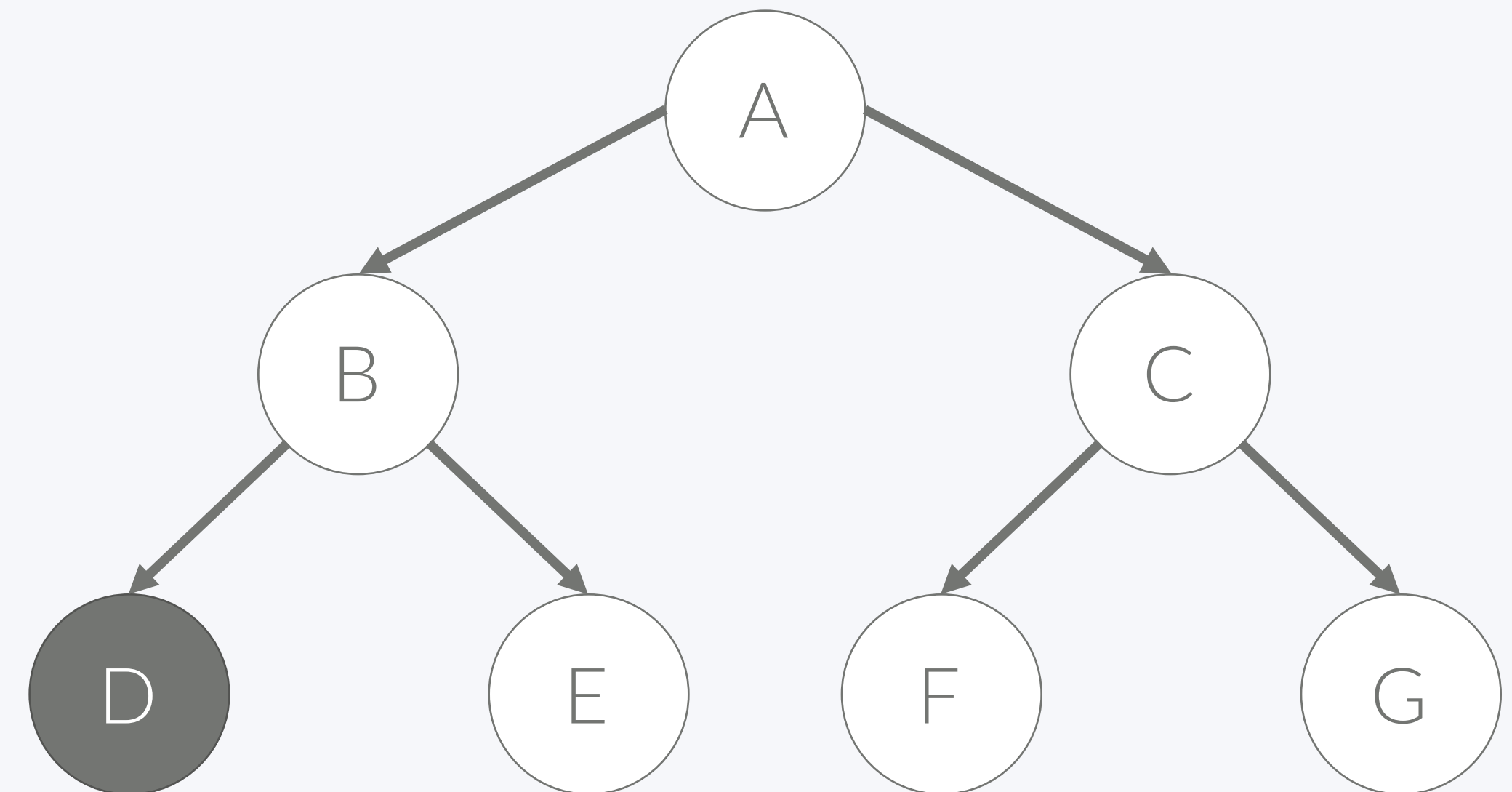
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

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

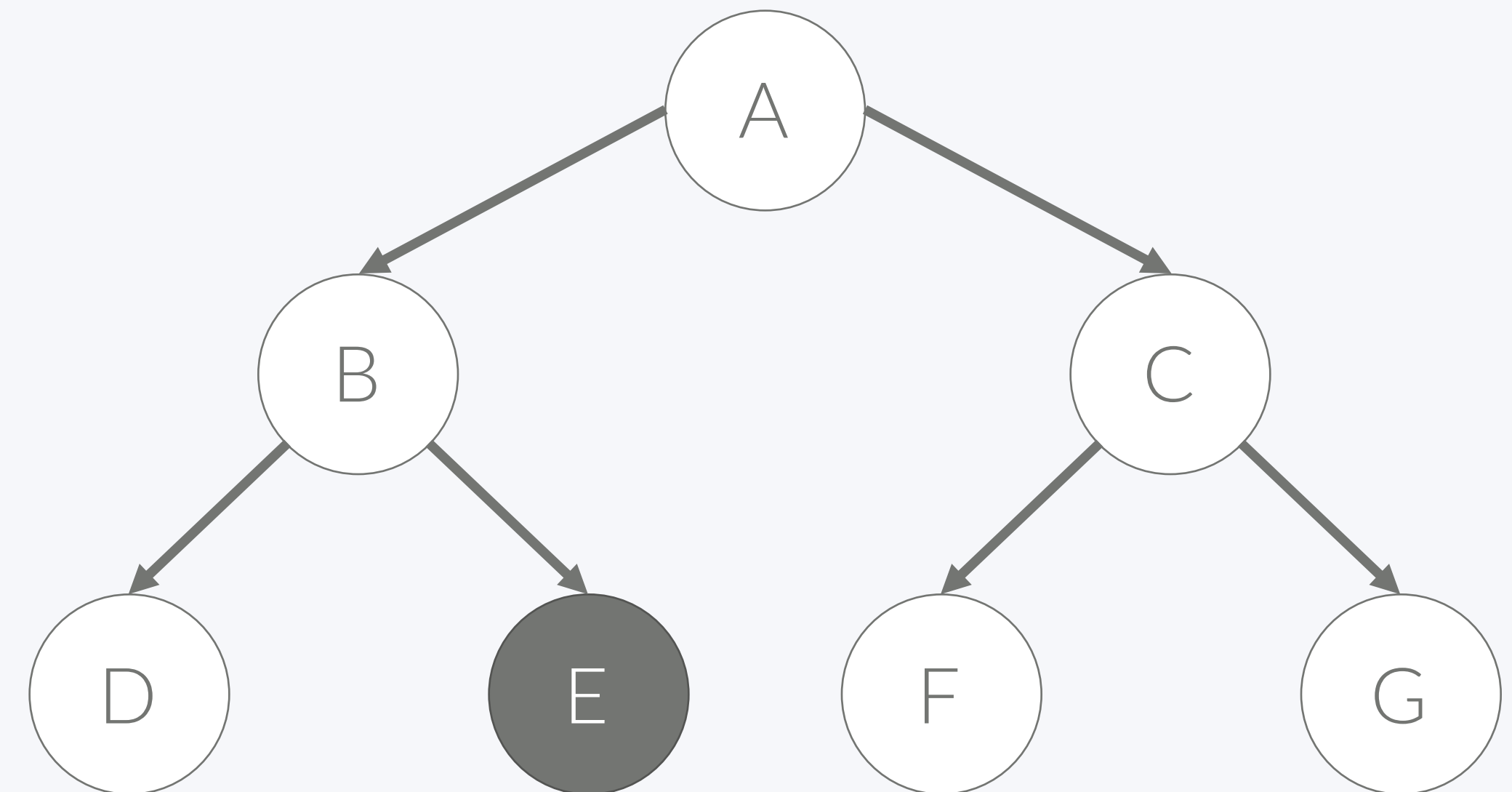


프리오더

30

Pre-order

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

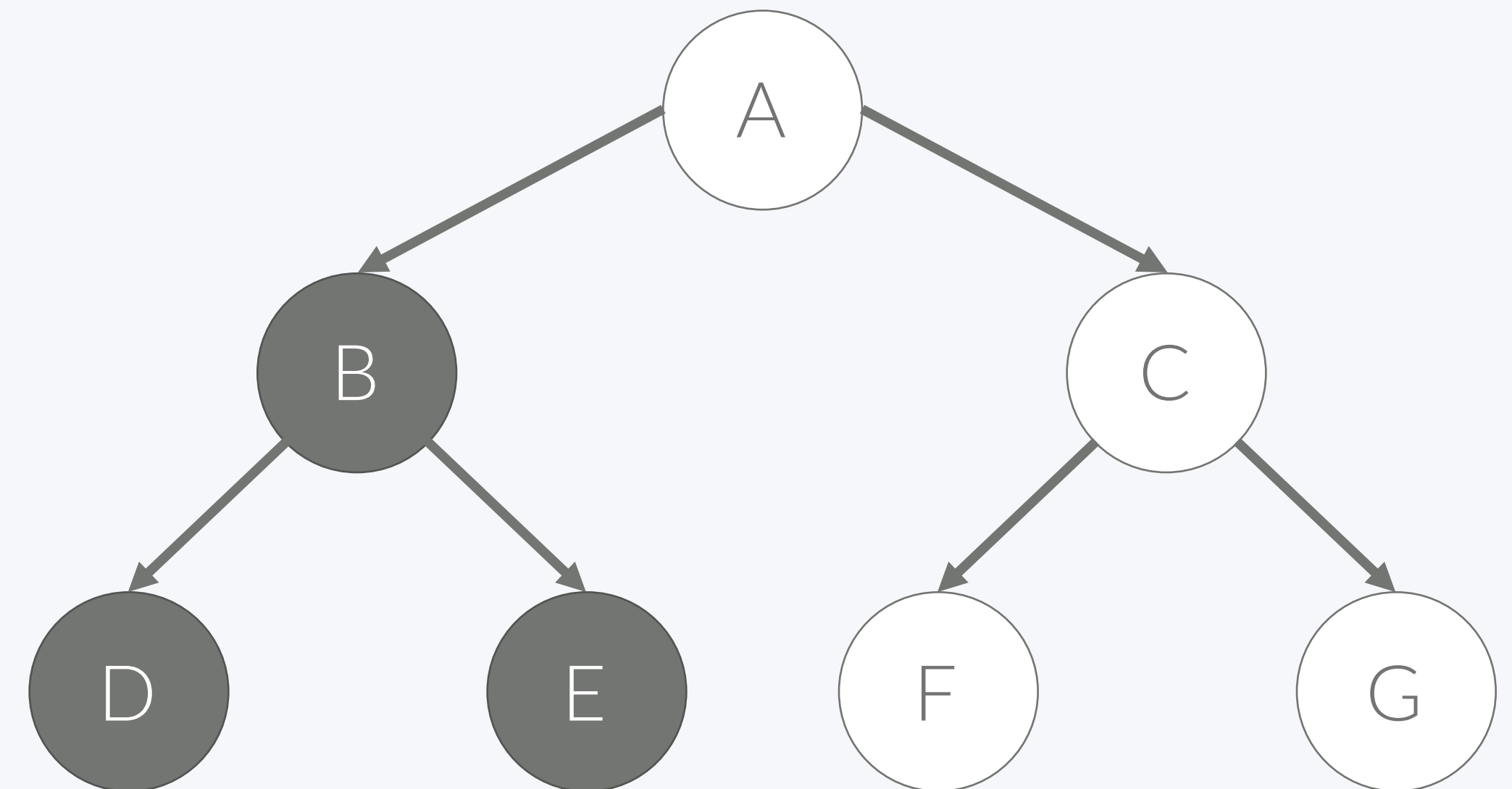


프리오더

31

Pre-order

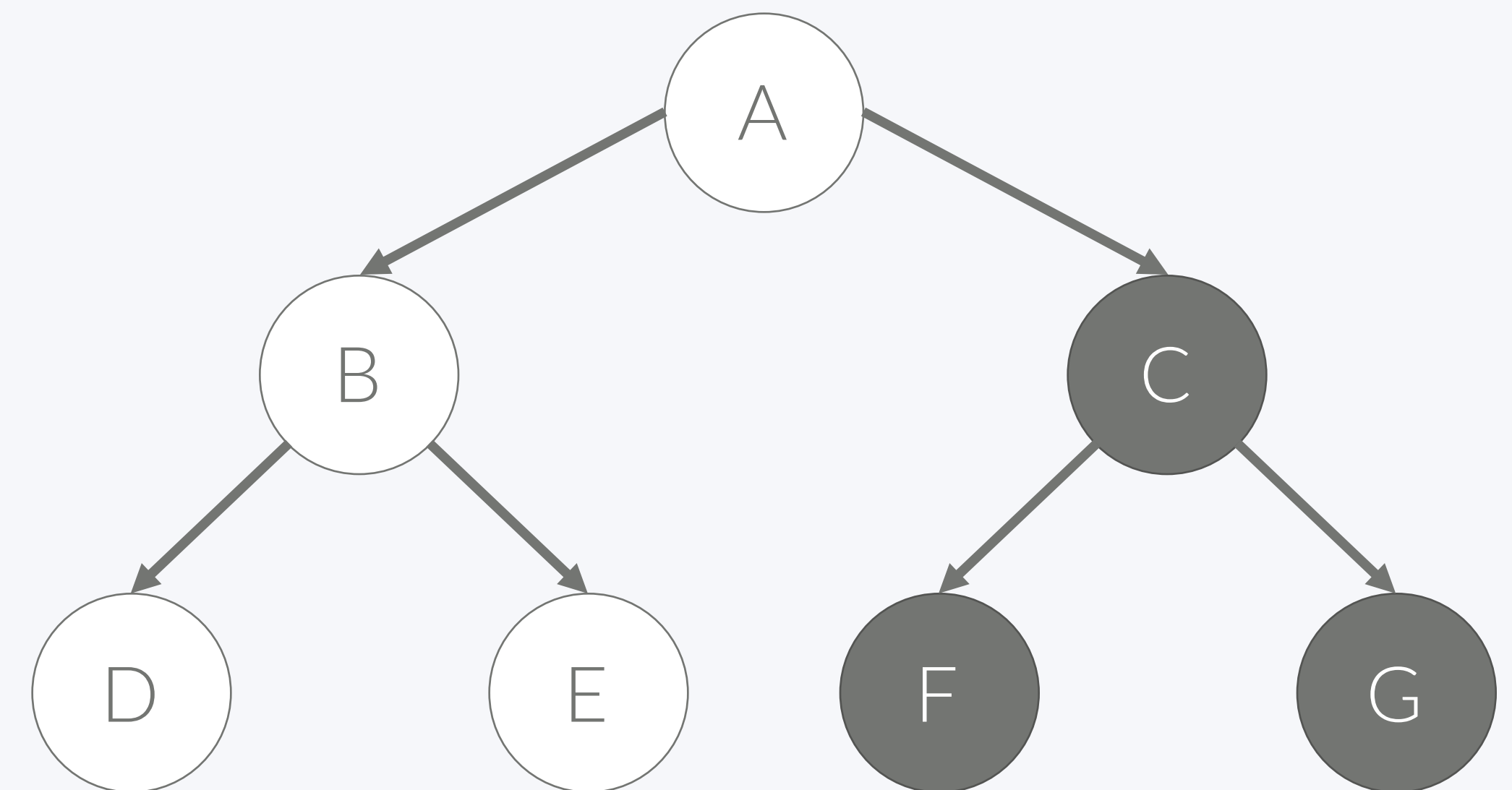
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더



프리오더

Pre-order

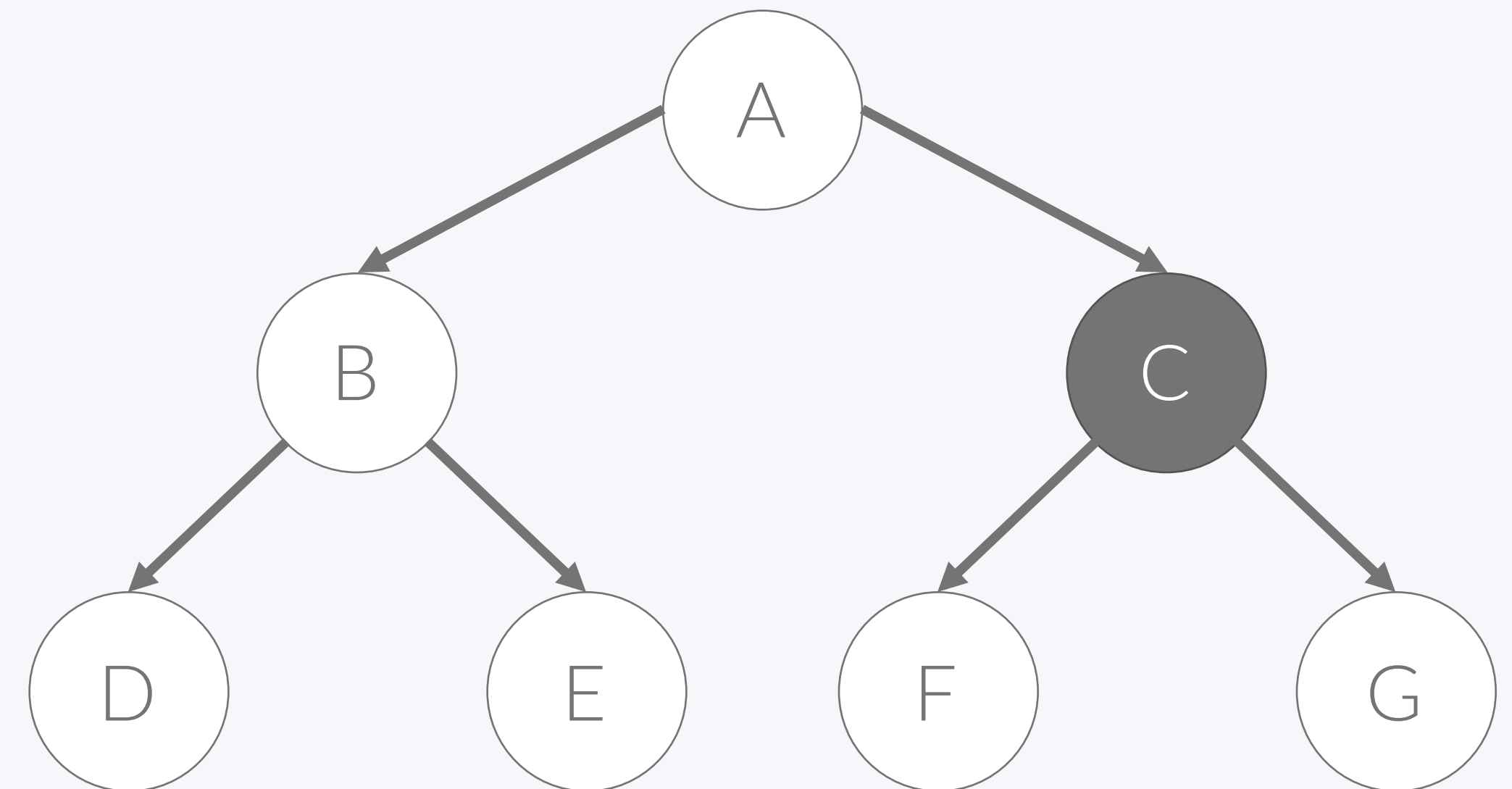
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더



프리오더

Pre-order

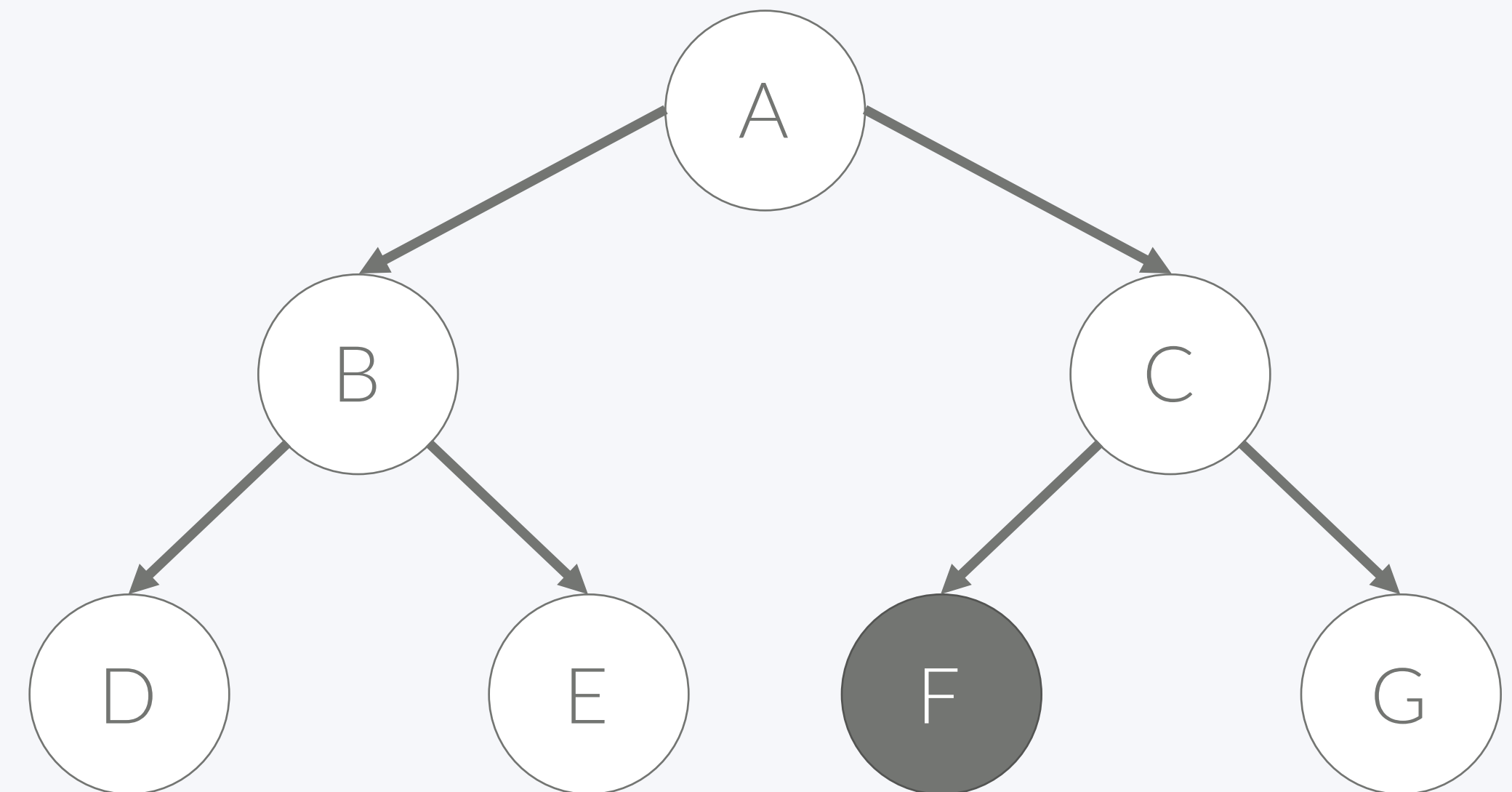
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더



프리오더

Pre-order

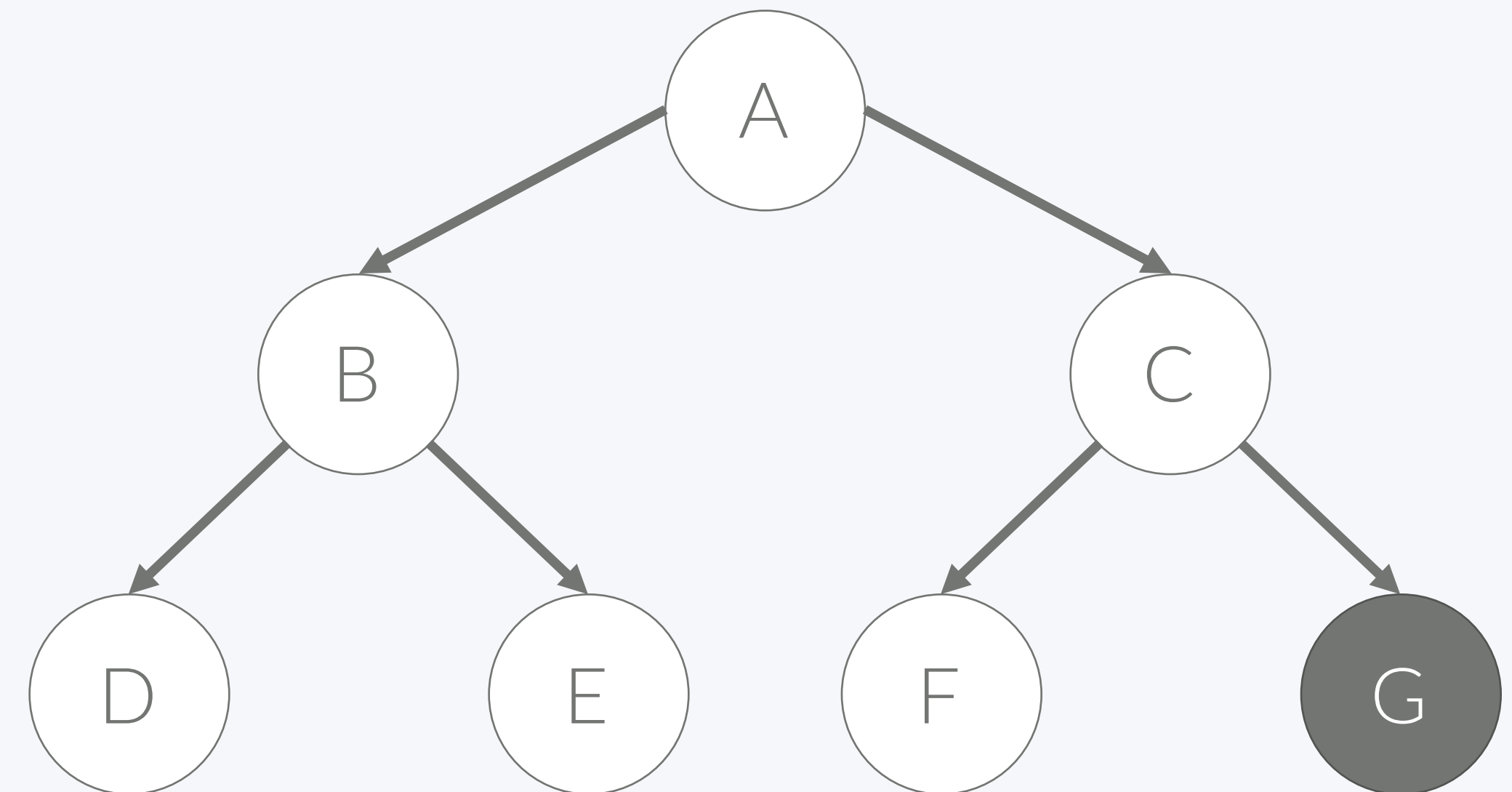
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - F
 - 오른쪽 자식 프리오더



프리오더

Pre-order

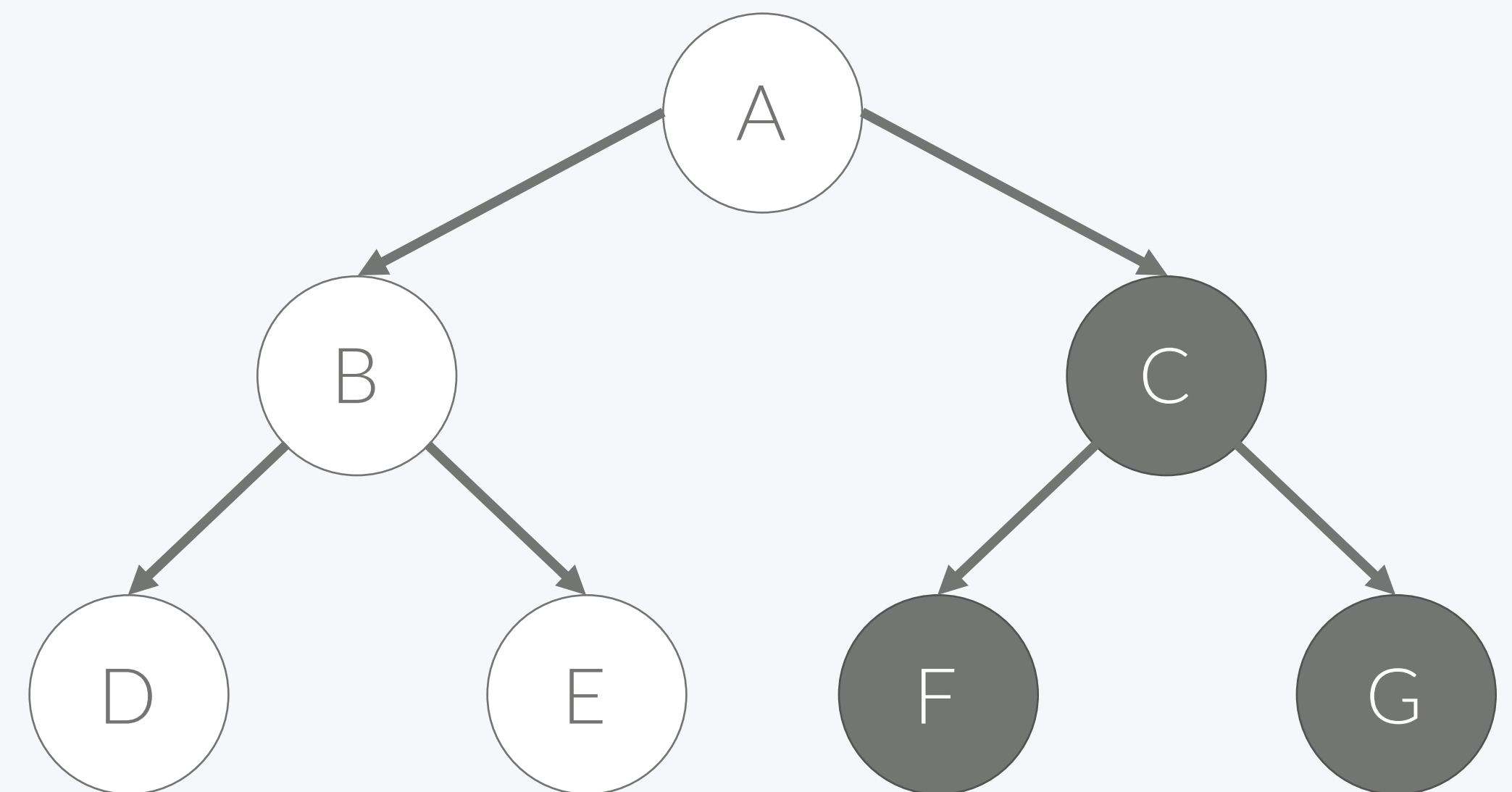
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - F
 - 오른쪽 자식 프리오더
 - G



프리오더

Pre-order

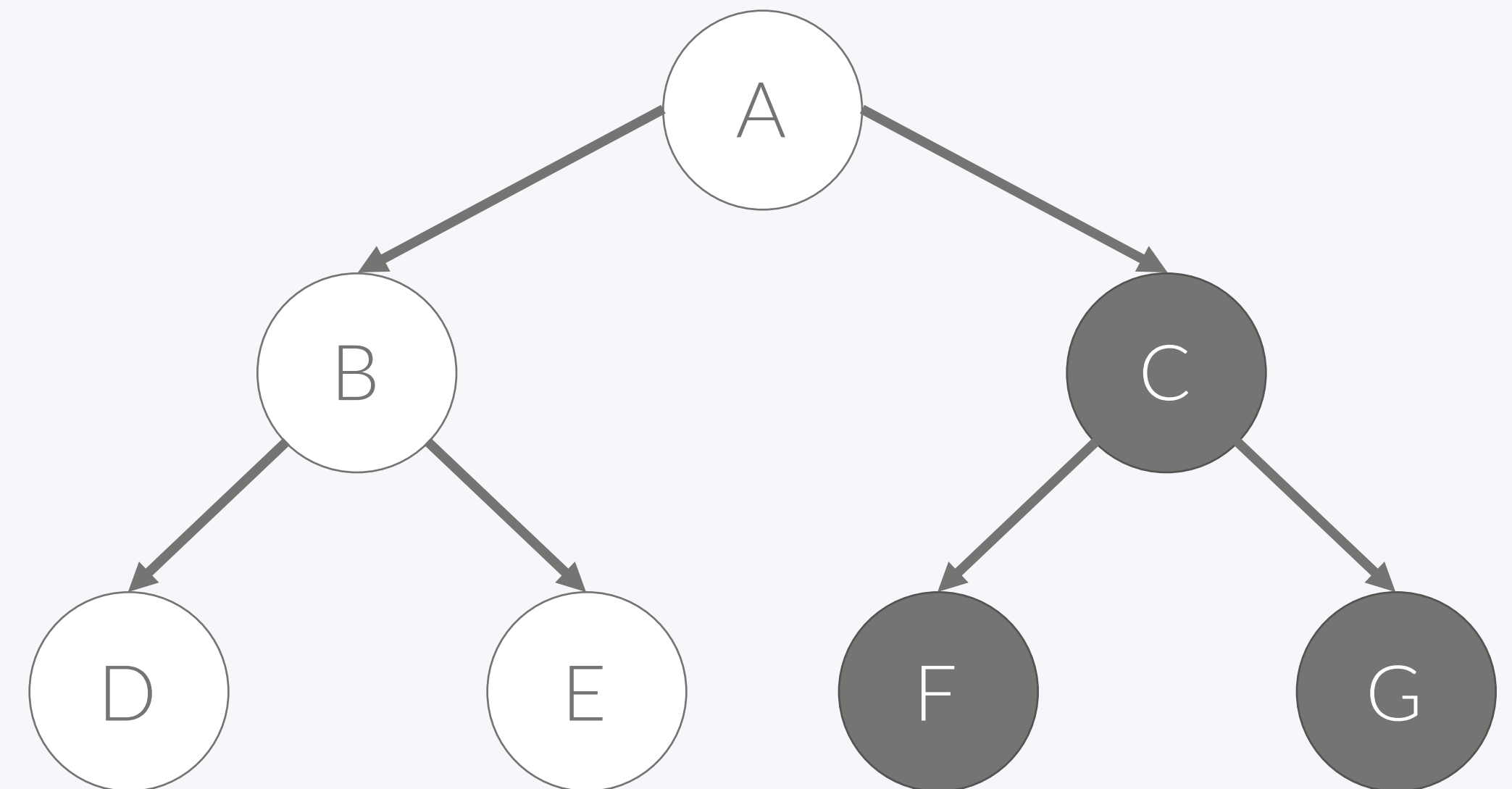
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDE
- 오른쪽 자식 프리오더
 - CFG



프리오더

Pre-order

- ABDECFG
- 그래프의 DFS이 순서와 같다

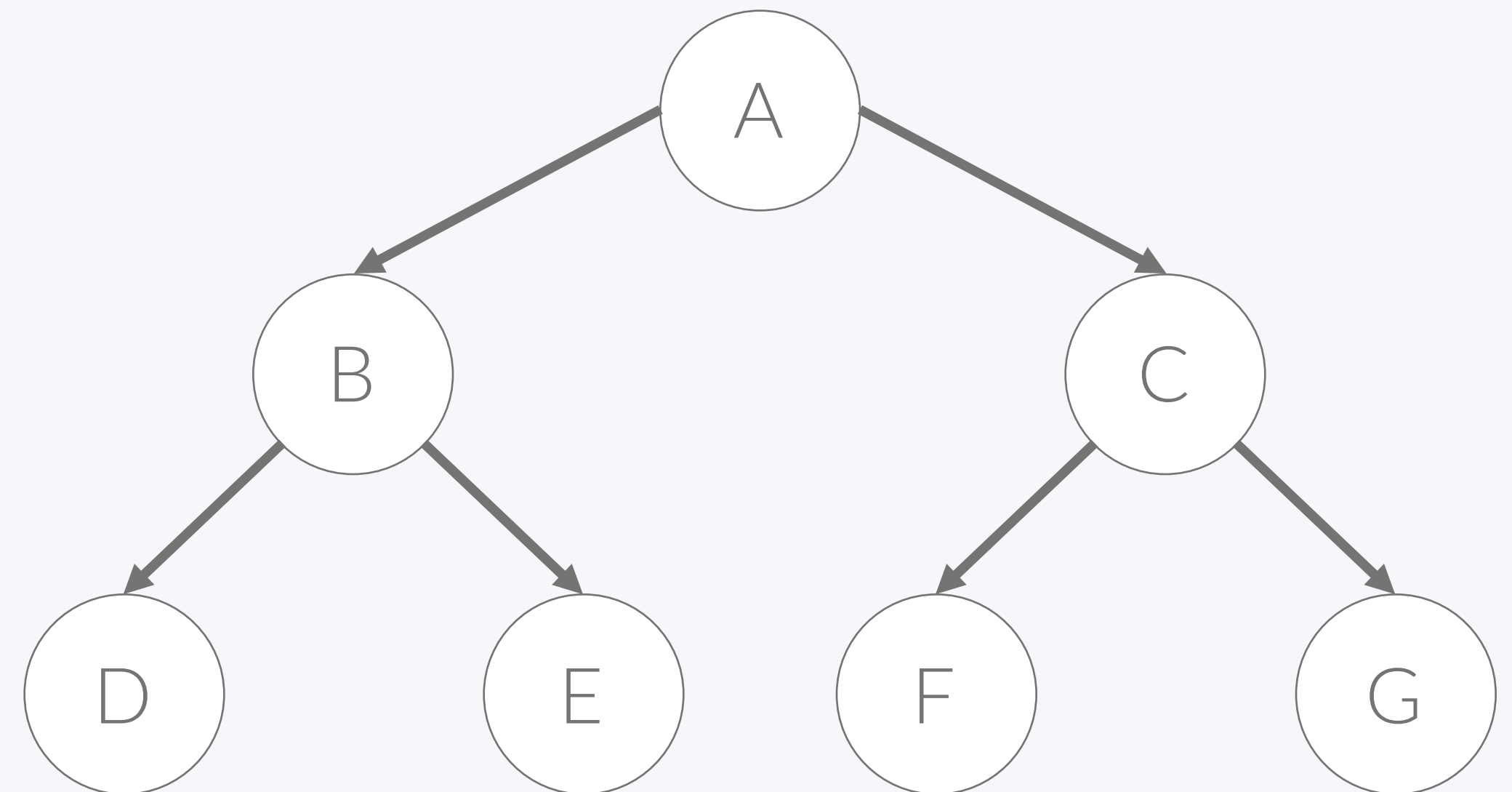


인오더

In-order

38

- 왼쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

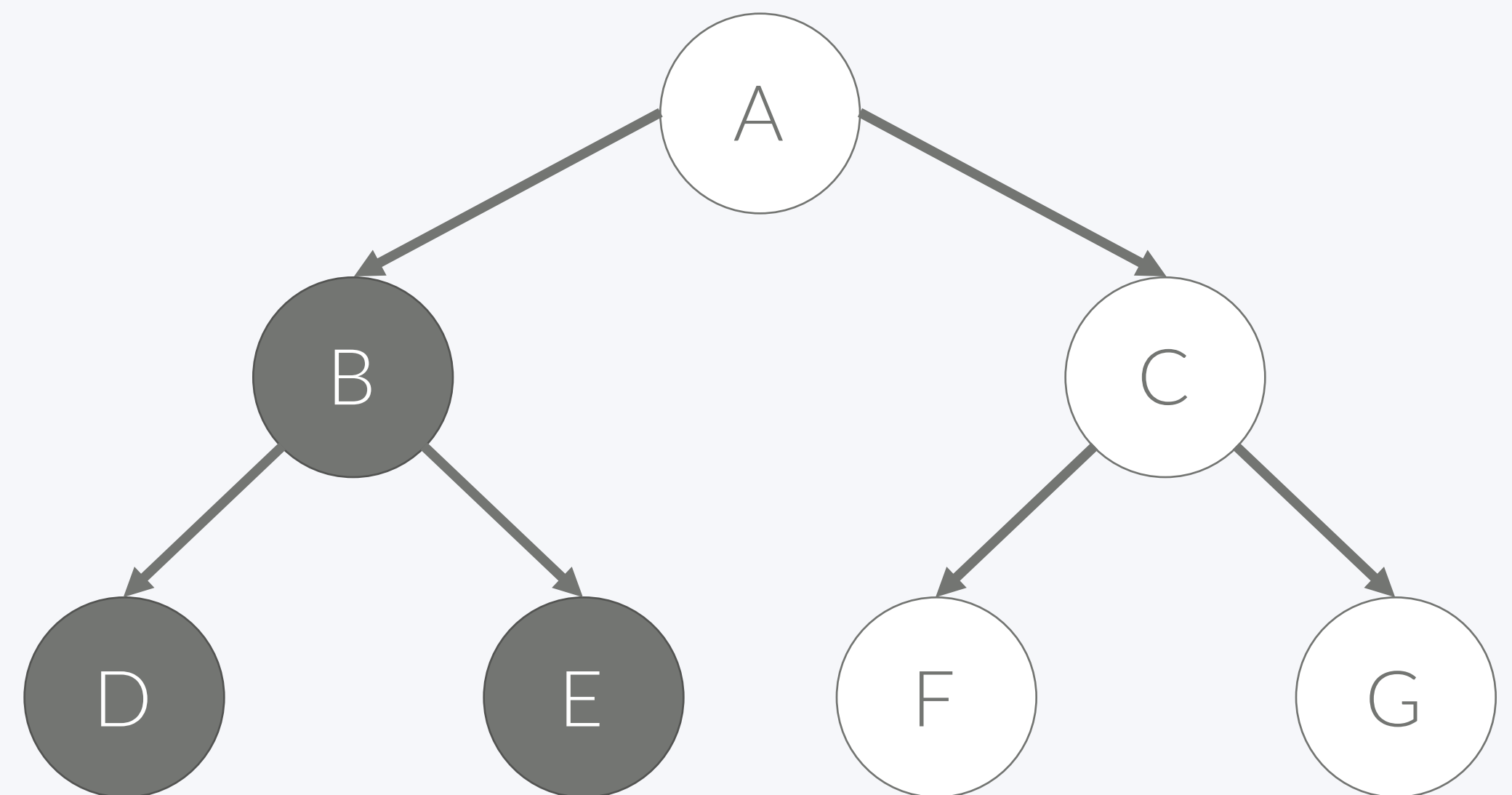


인오더

In-order

39

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

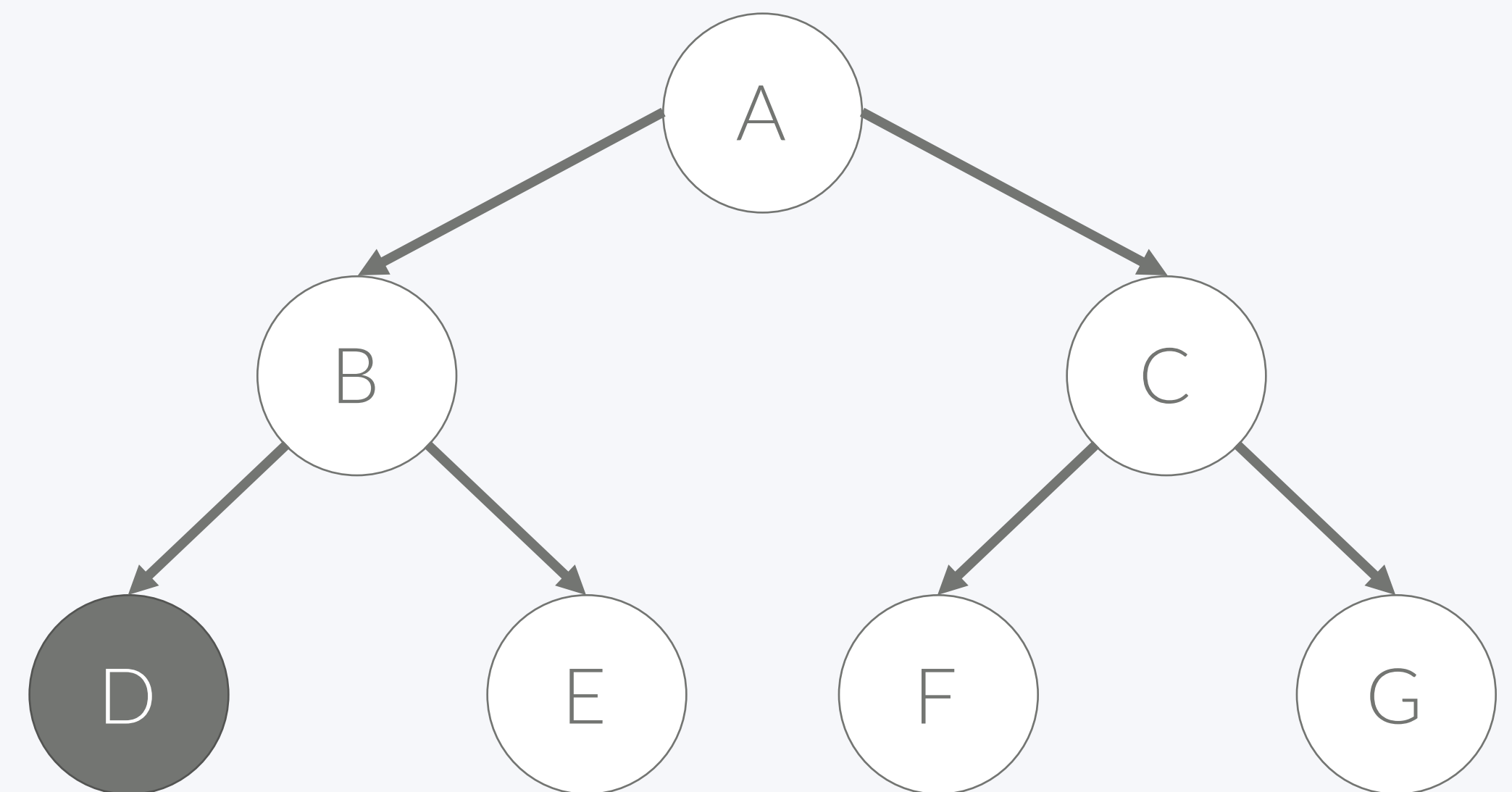


인오더

In-order

40

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

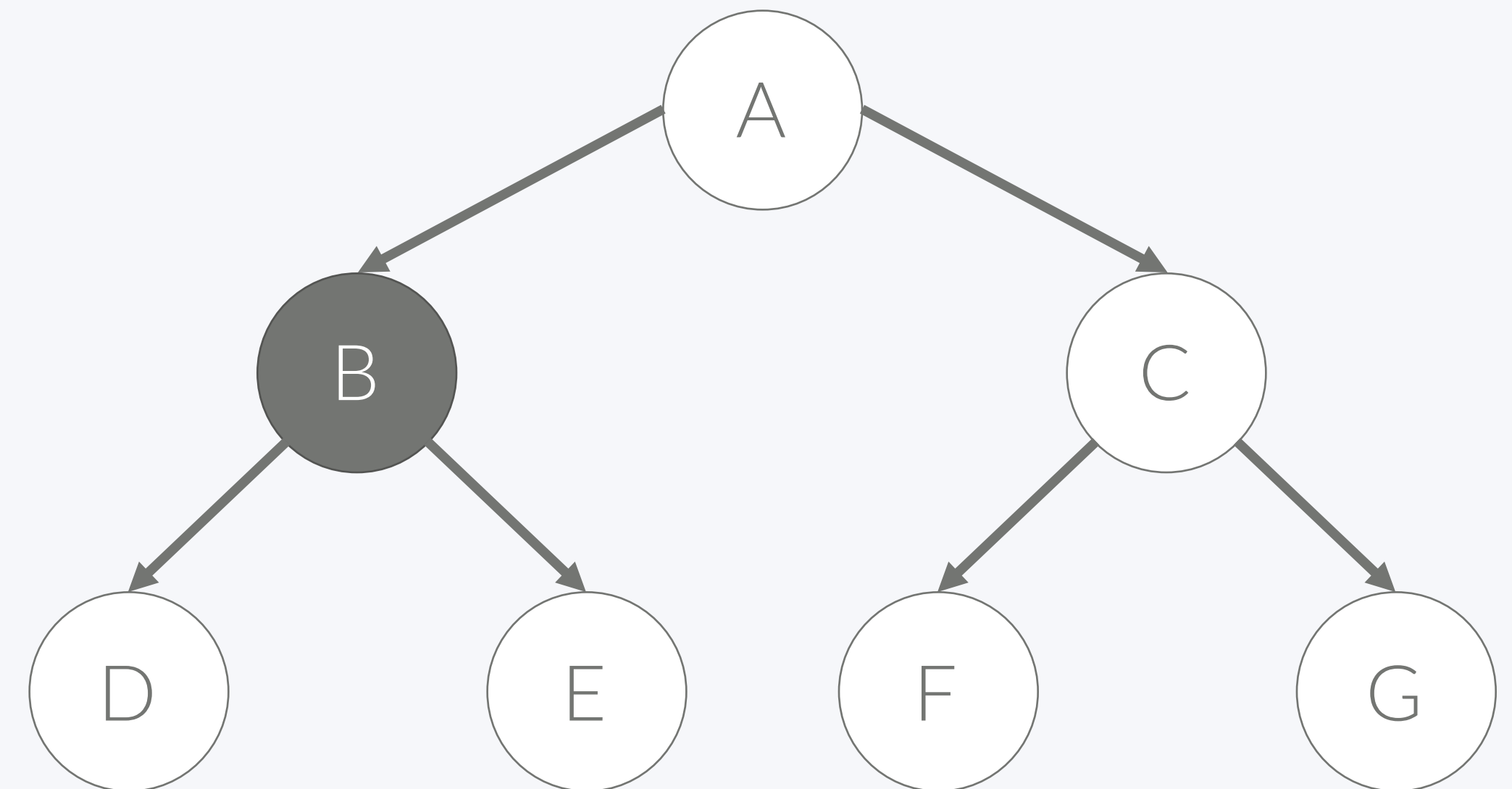


인오더

In-order

41

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

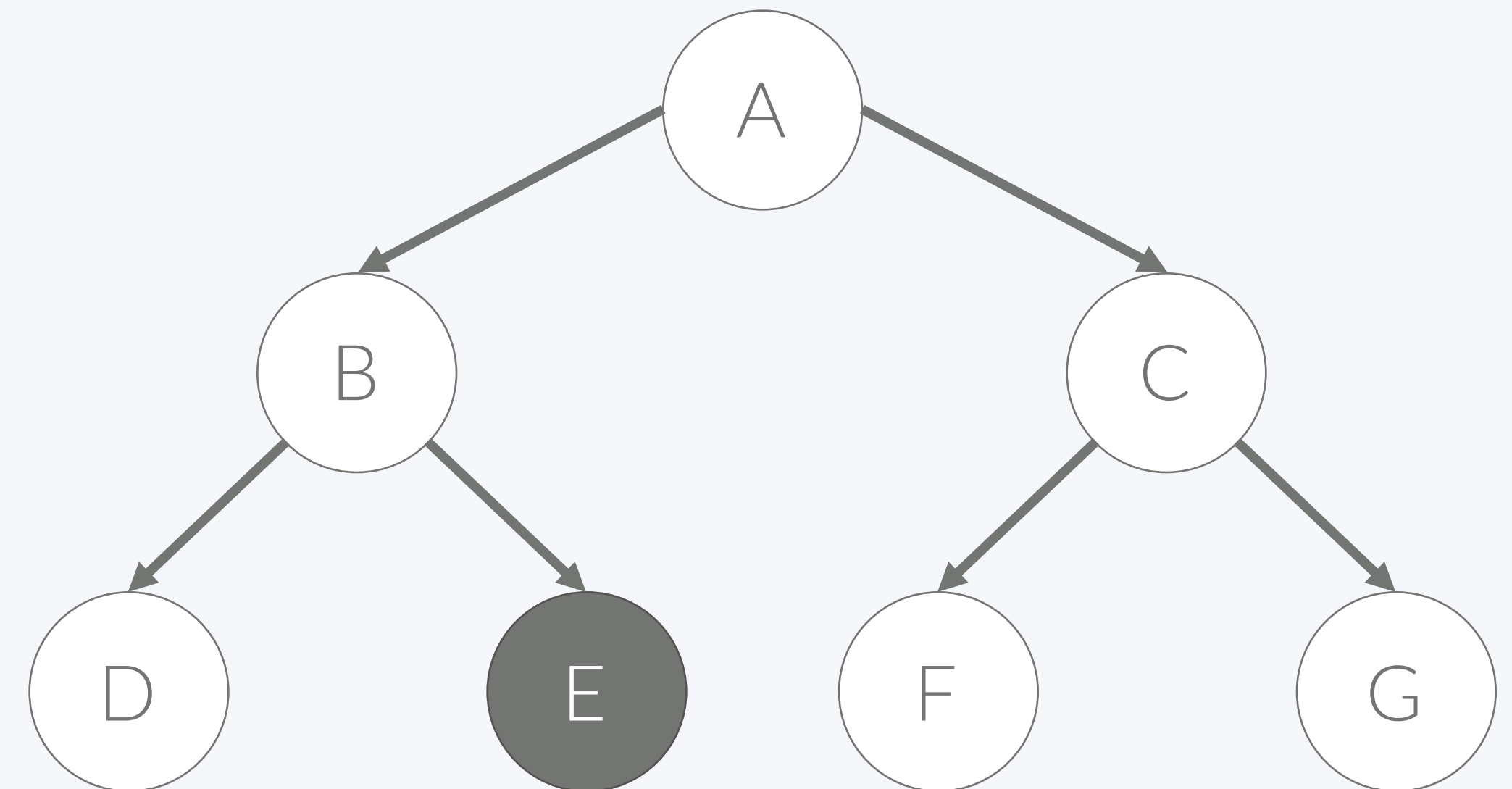


인오더

In-order

42

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

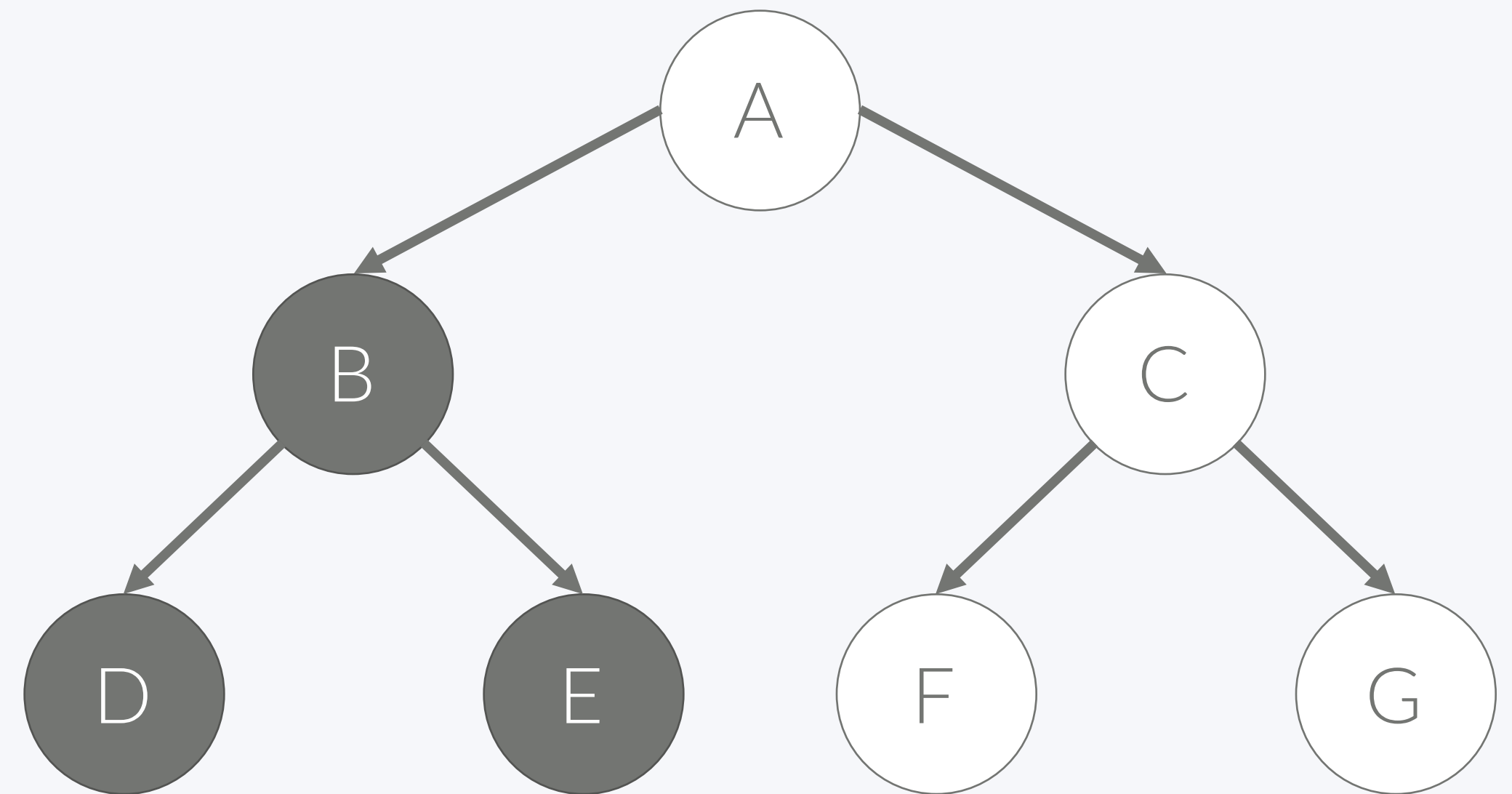


인오더

In-order

43

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

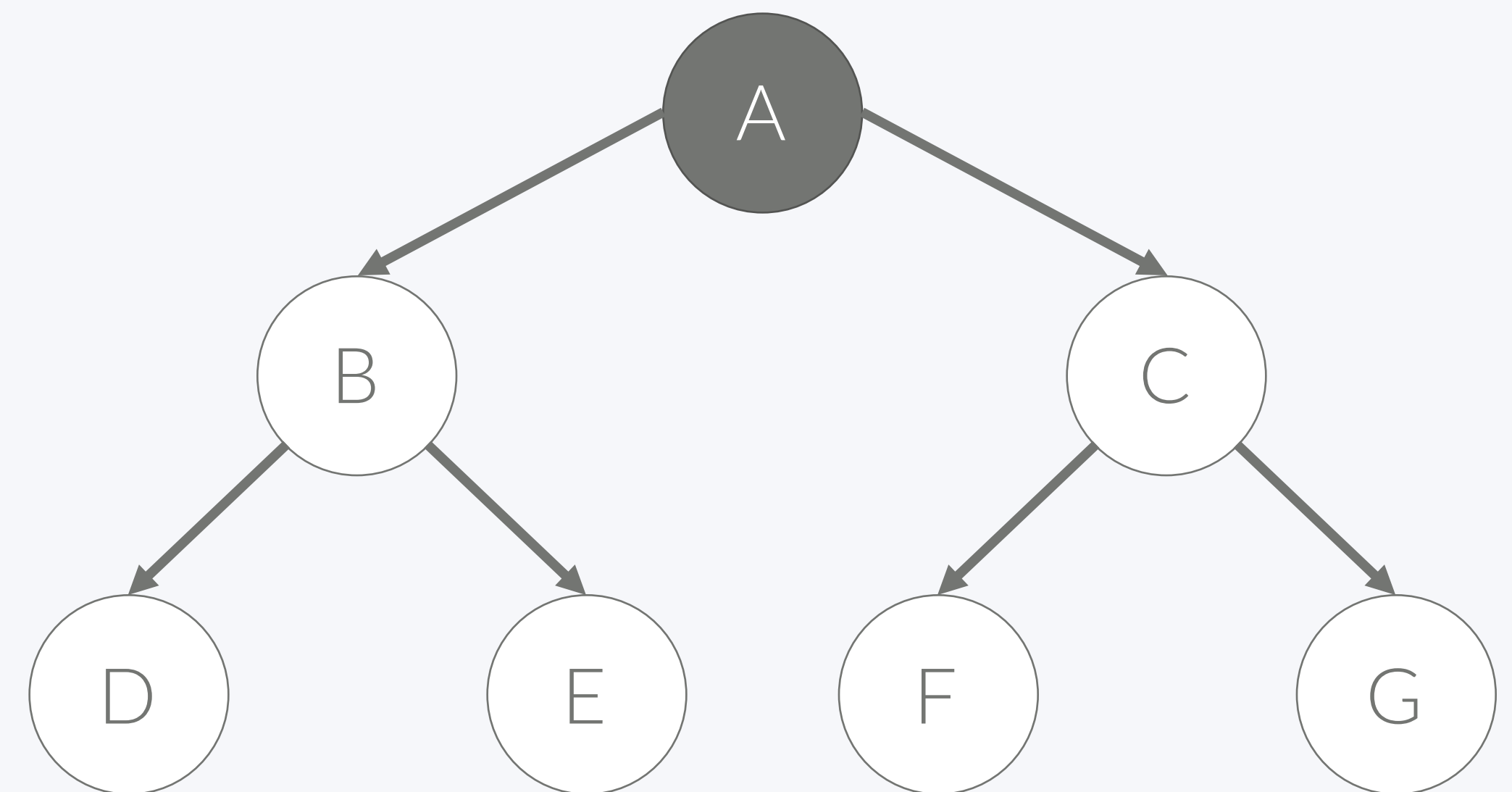


인오더

In-order

44

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

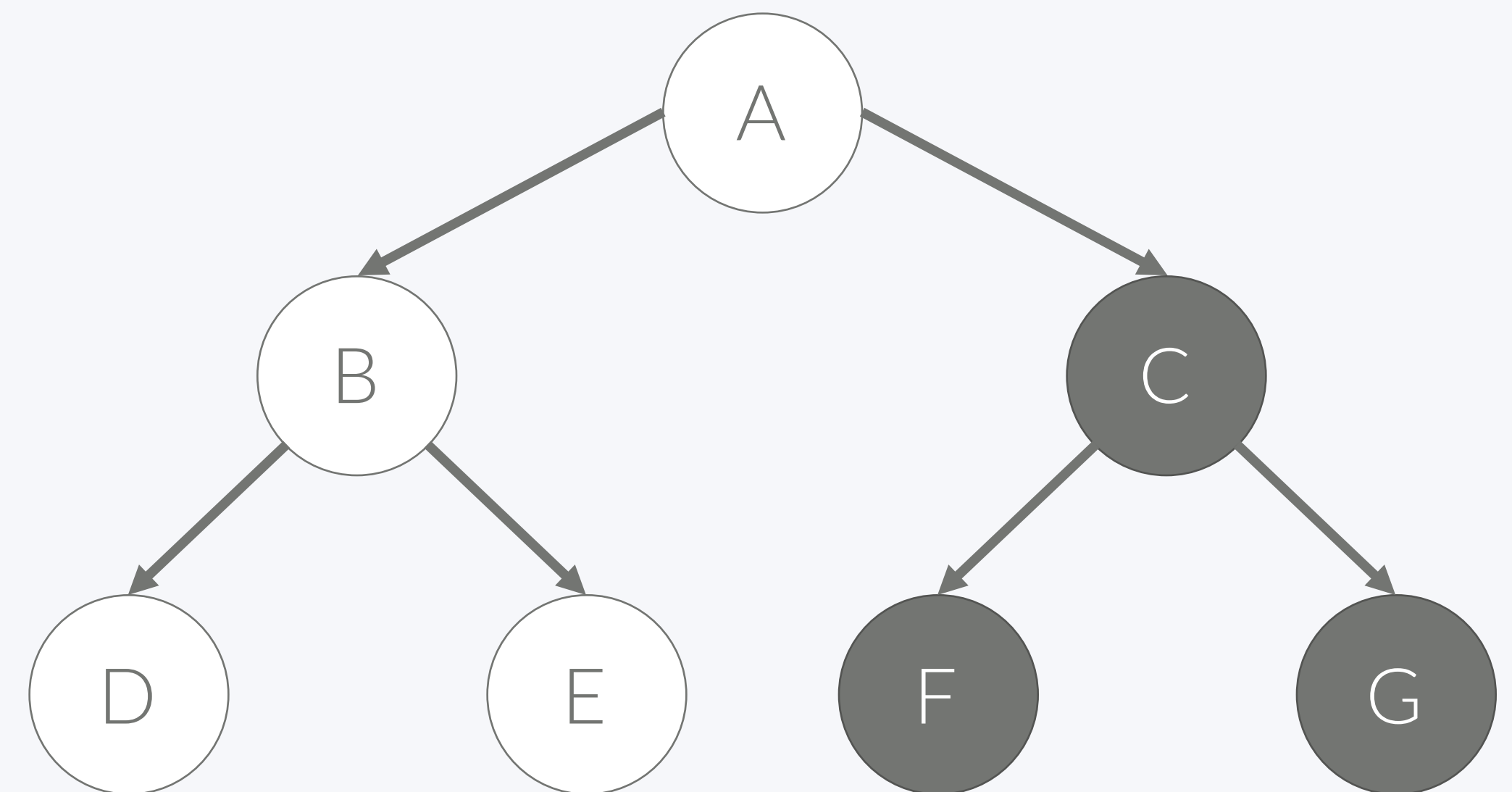


인오더

In-order

45

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

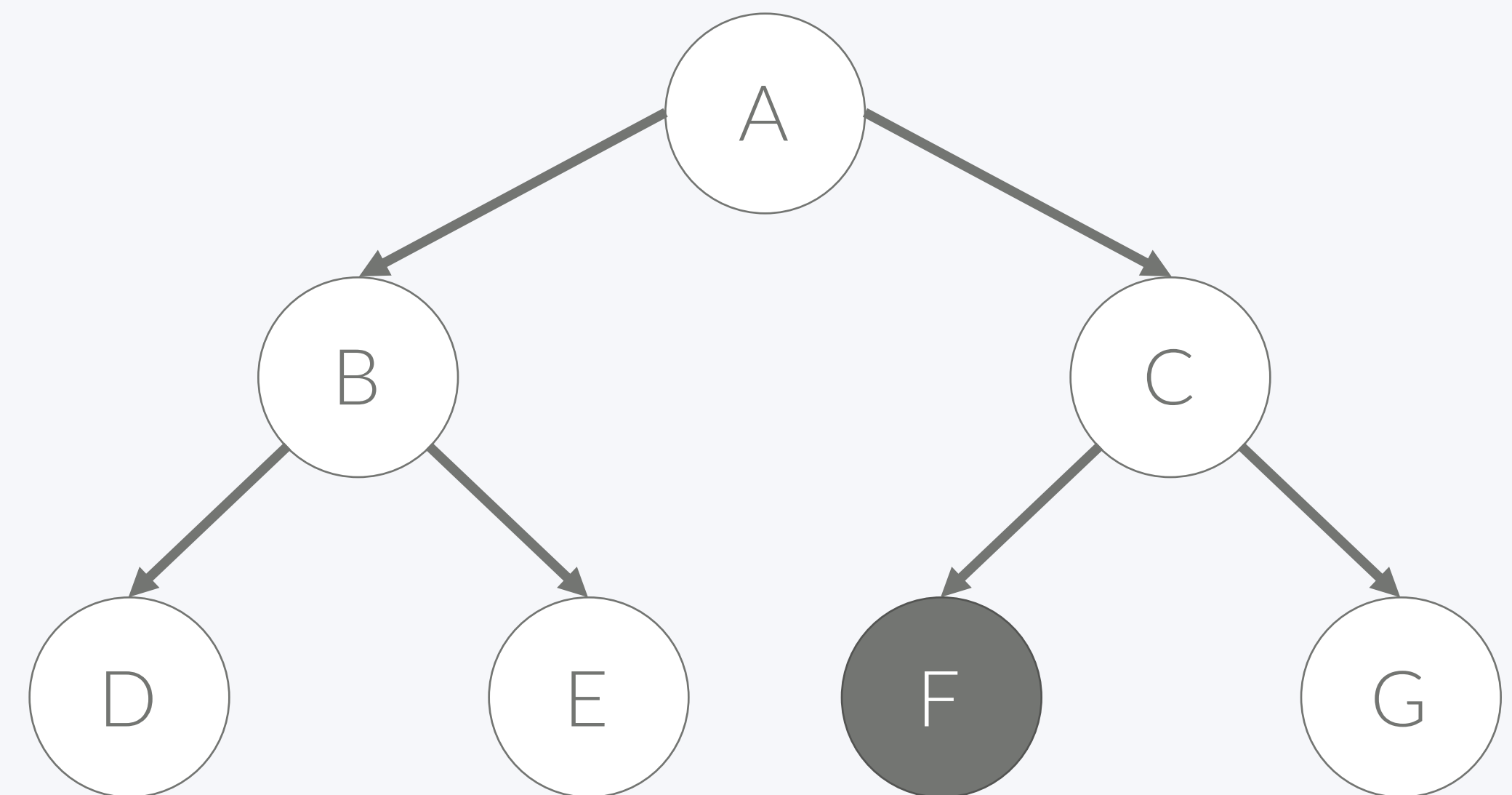


인오더

In-order

46

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

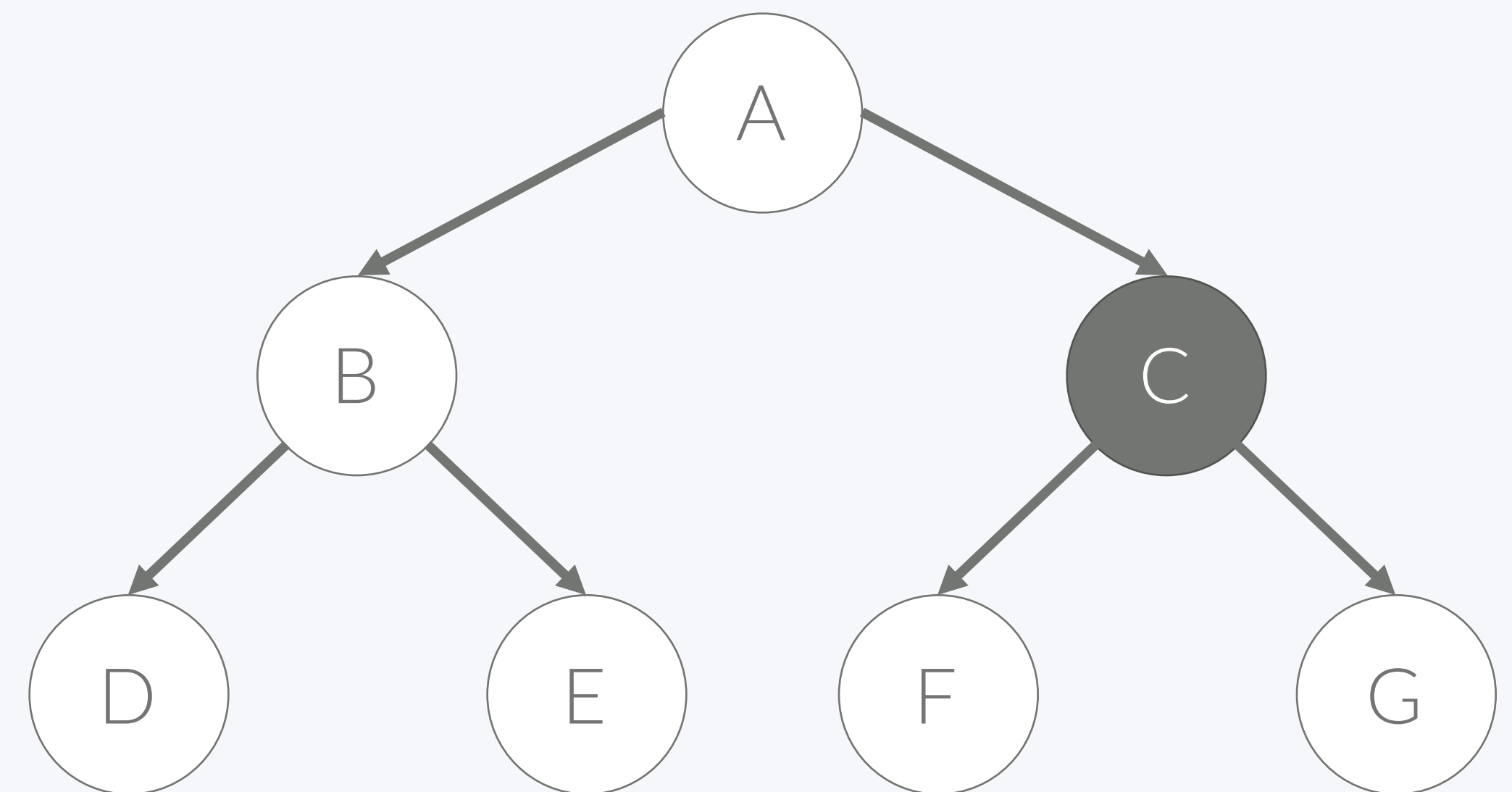


인오더

47

In-order

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

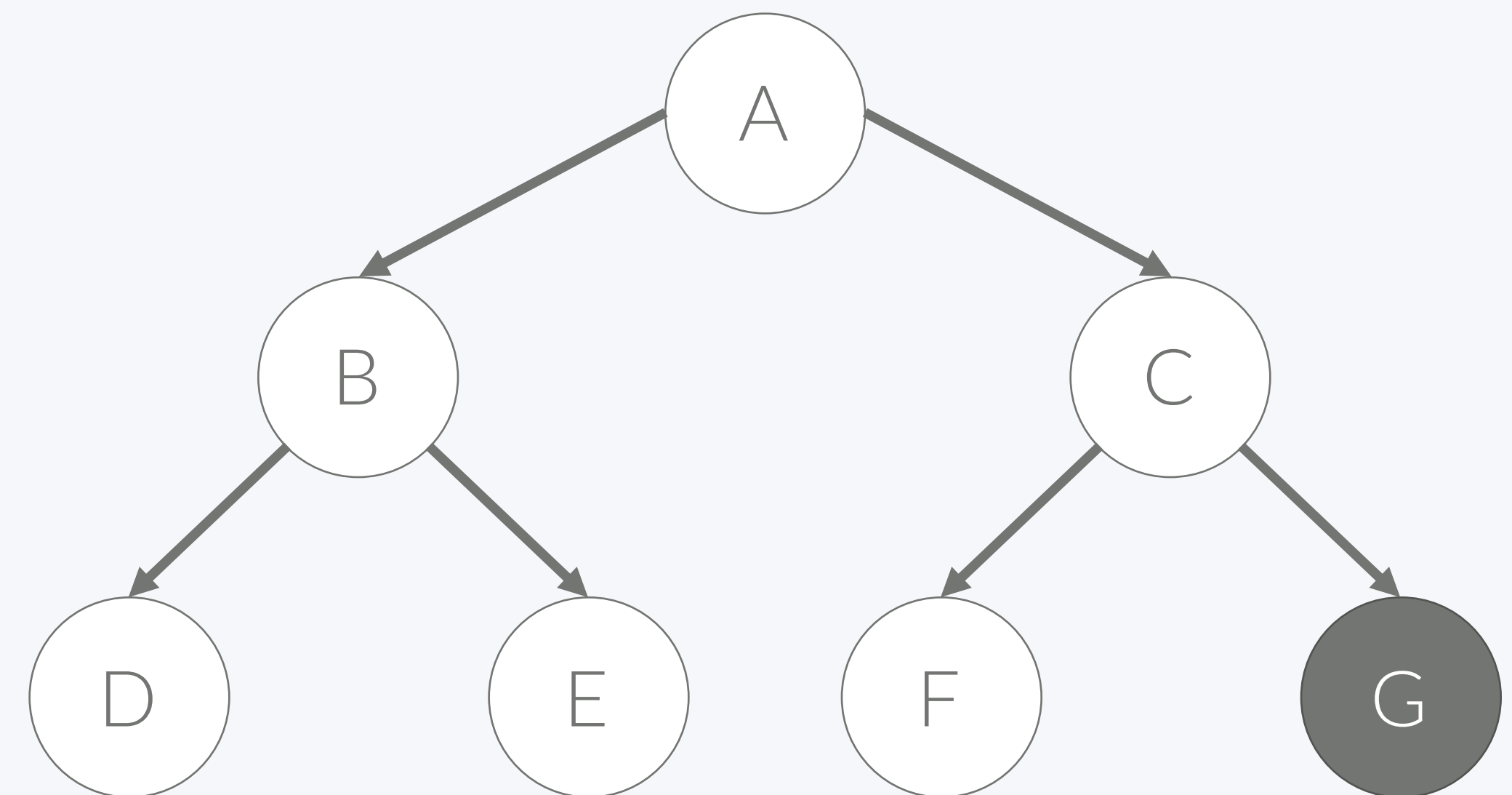


인오더

48

In-order

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

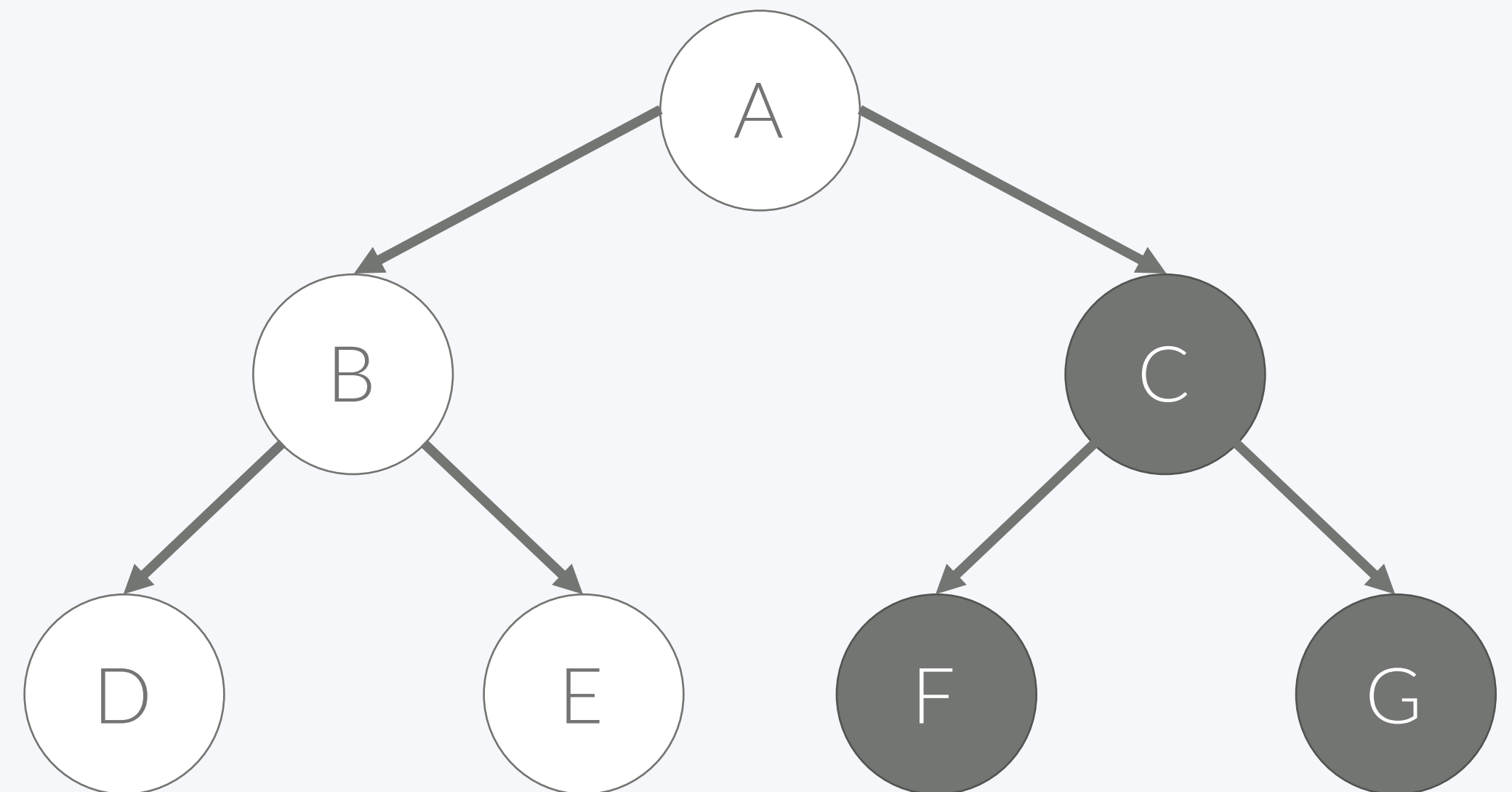


인오더

In-order

49

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



인오더

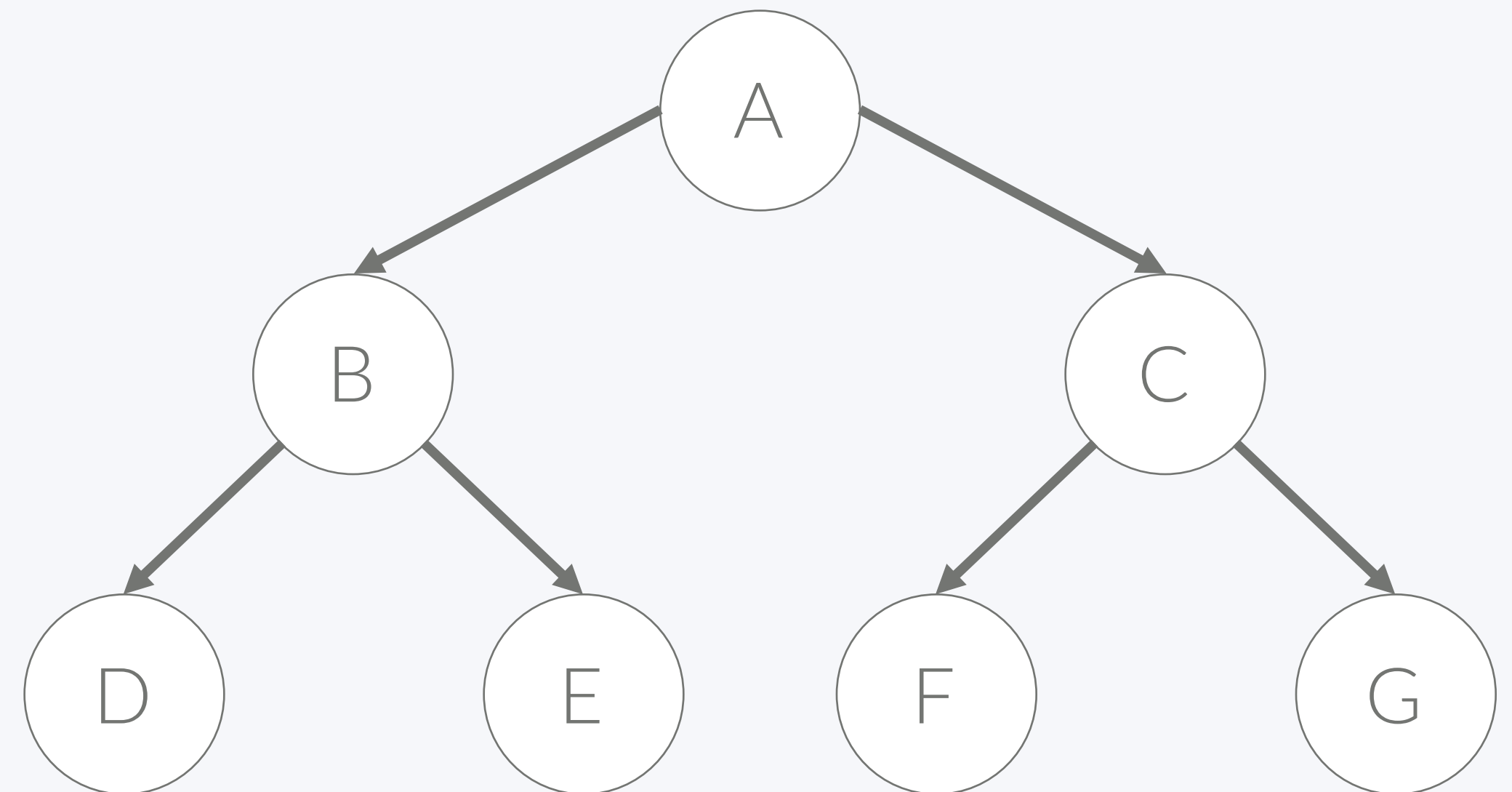
In-order

- DBEAFCG

BST

Delete

50

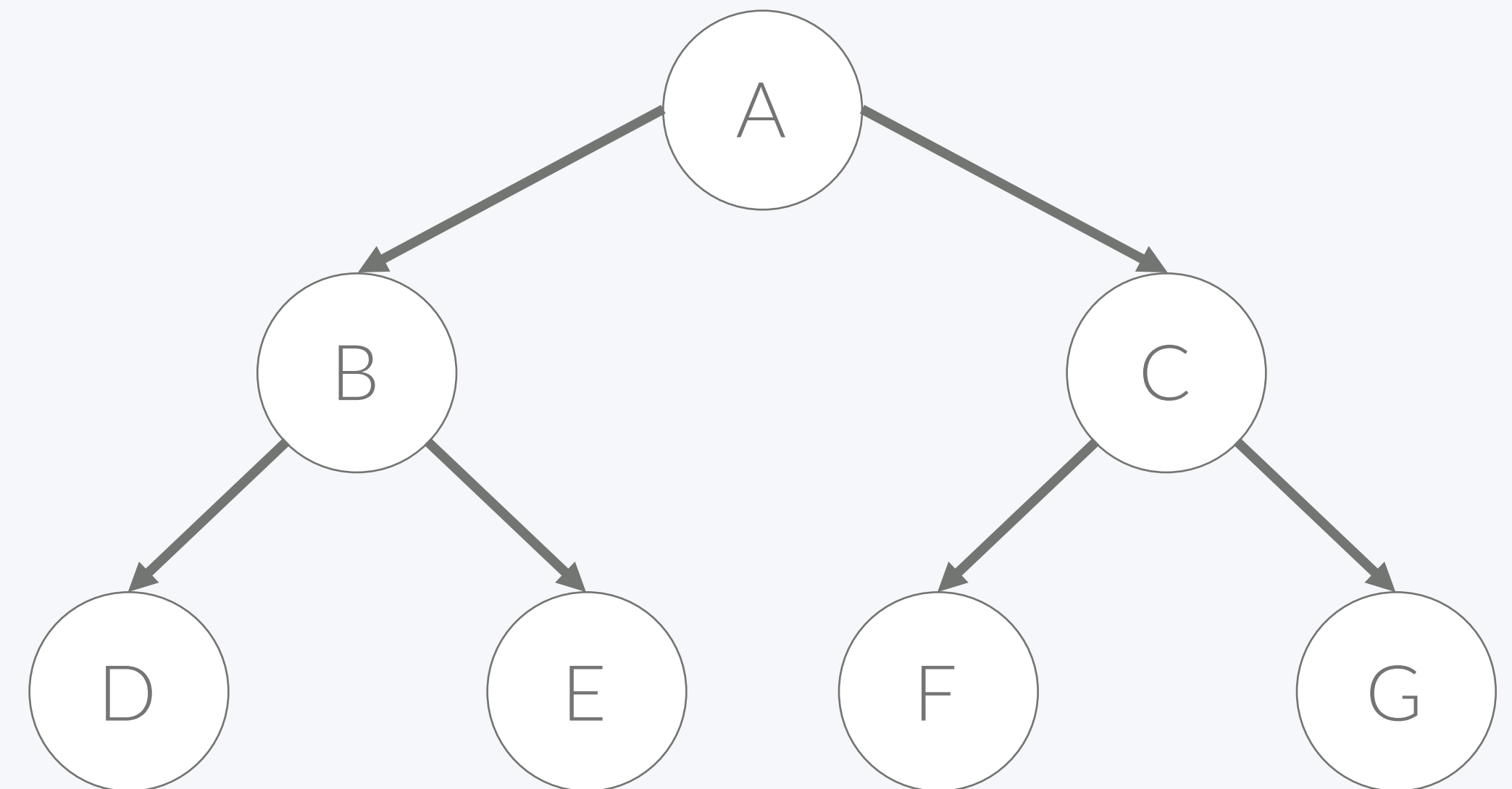


포스트오더

51

Postorder

- 왼쪽 자식 포스트오더
- 오른쪽 자식 포스트오더
- 노드 방문

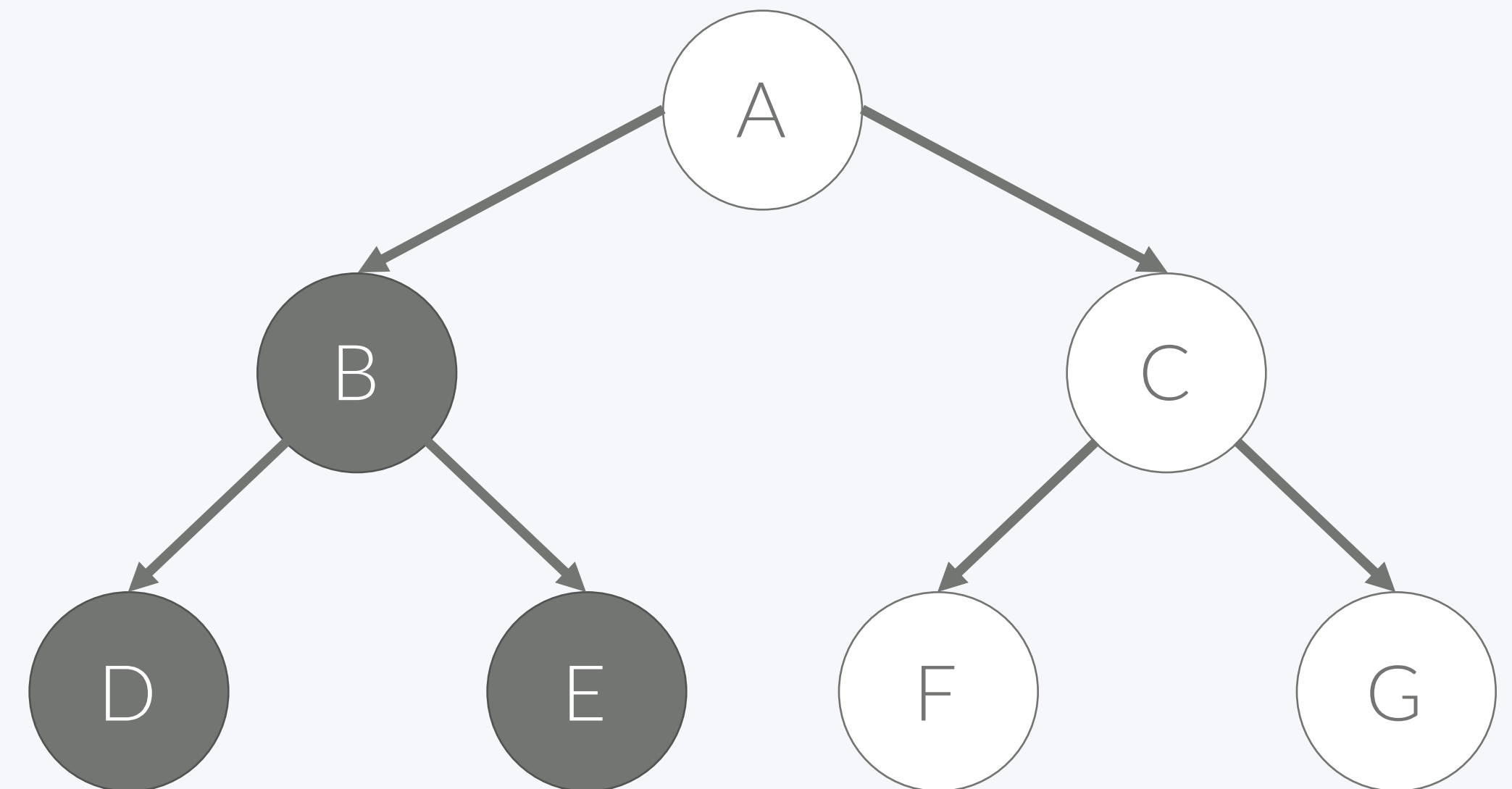


포스트오더

52

Postorder

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

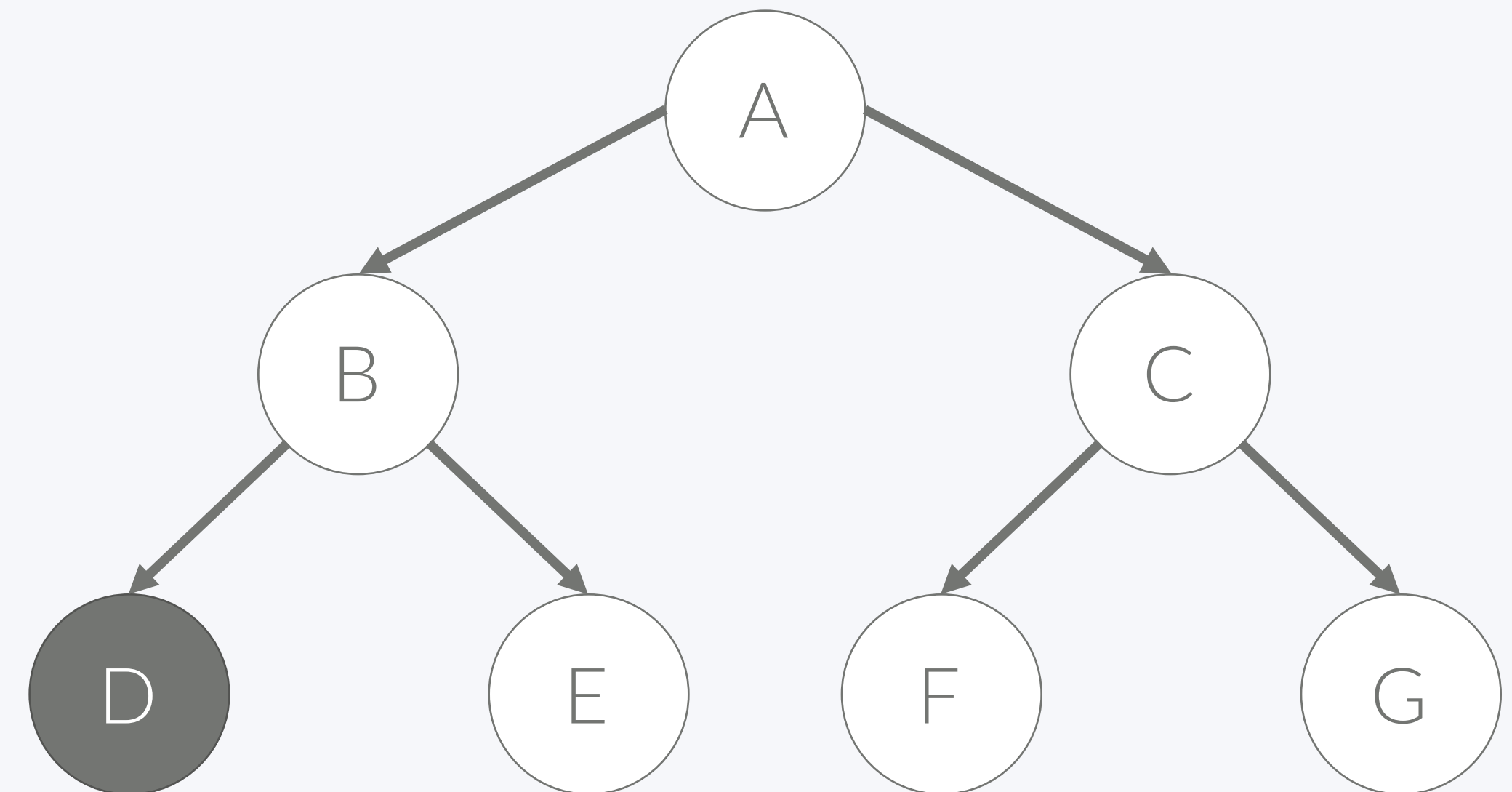


포스트오더

53

Postorder

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

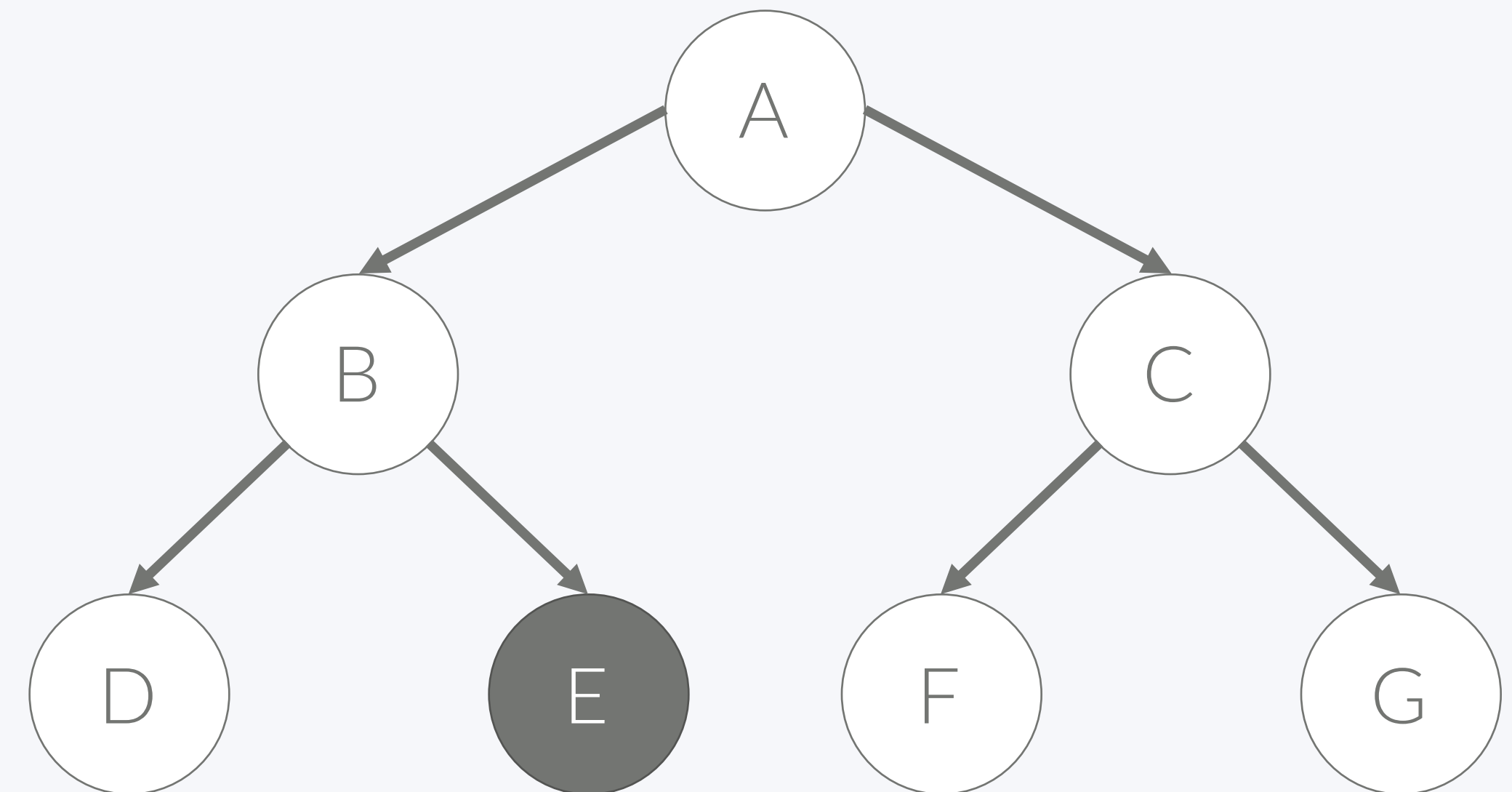


포스트오더

54

Postorder

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

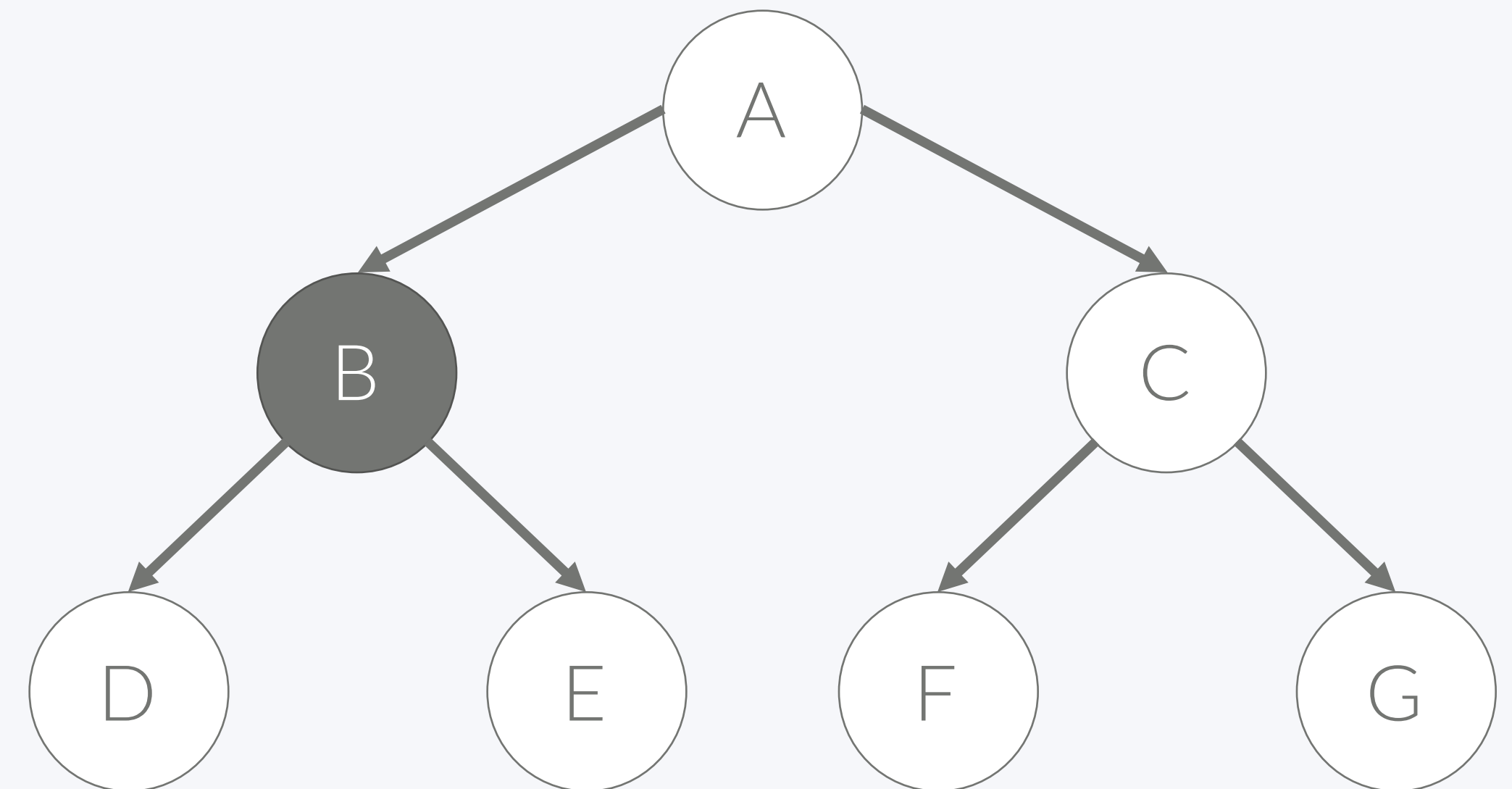


포스트오더

55

Postorder

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

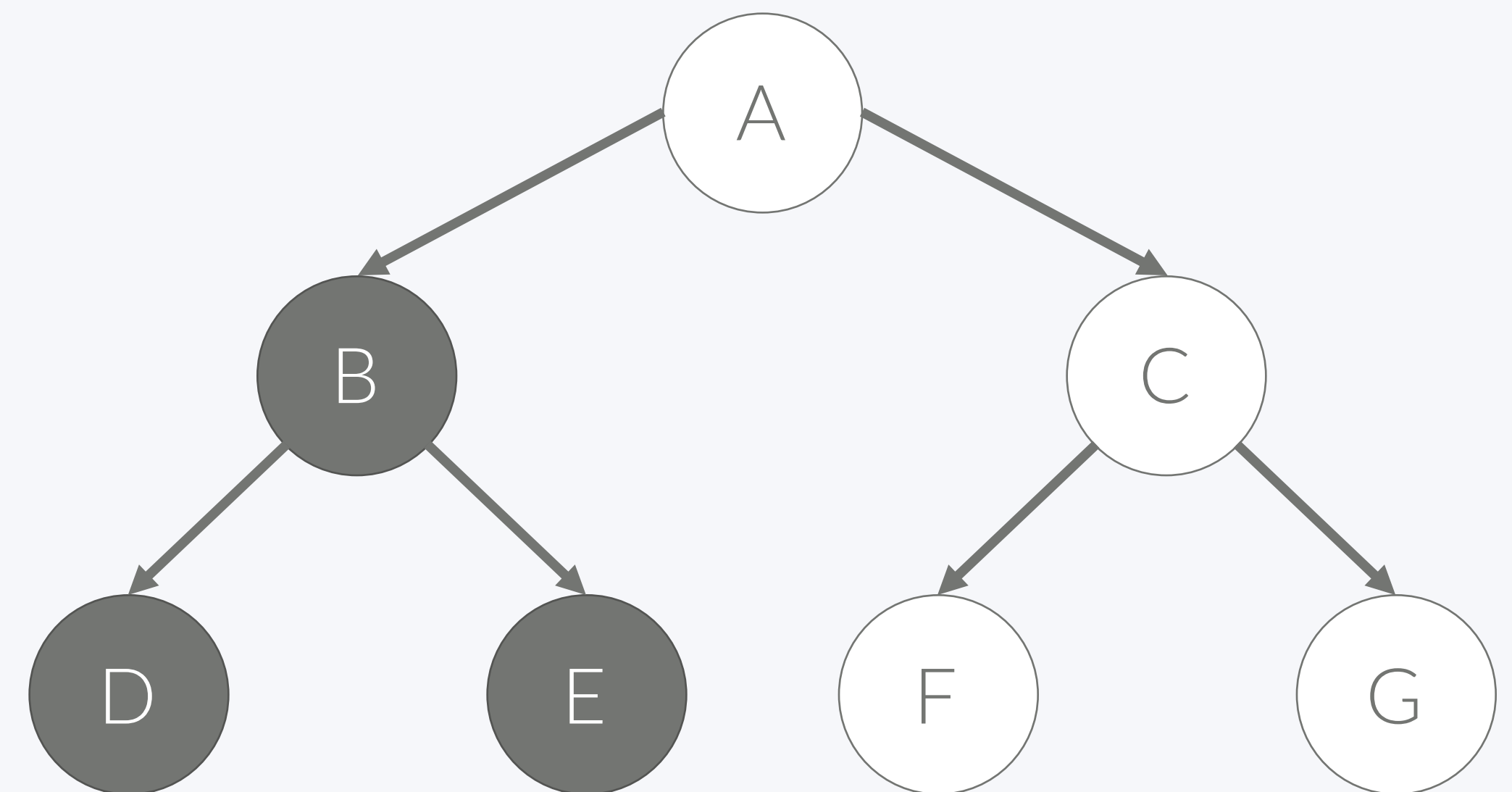


포스트오더

56

Postorder

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

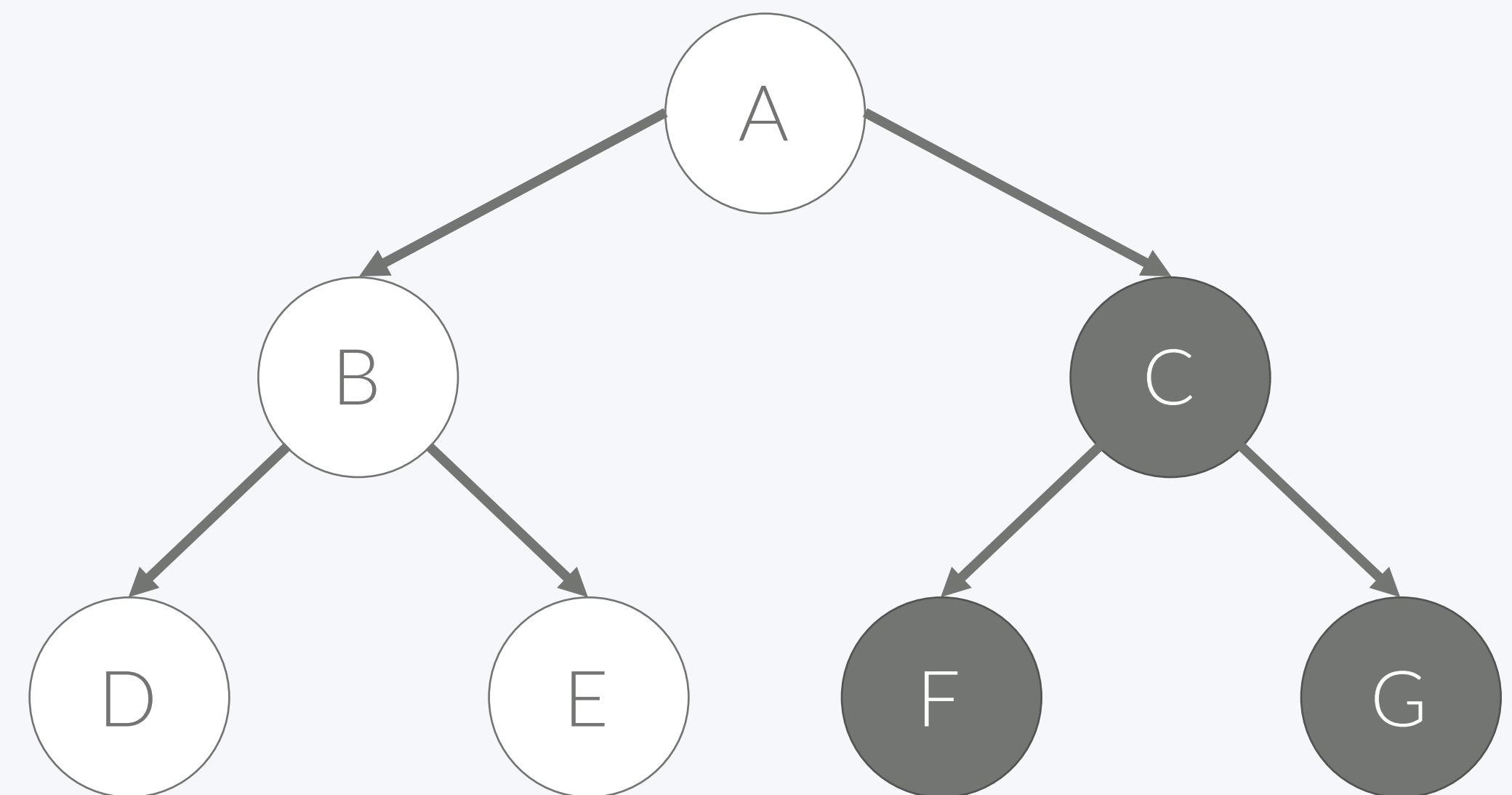


포스트오더

57

Postorder

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

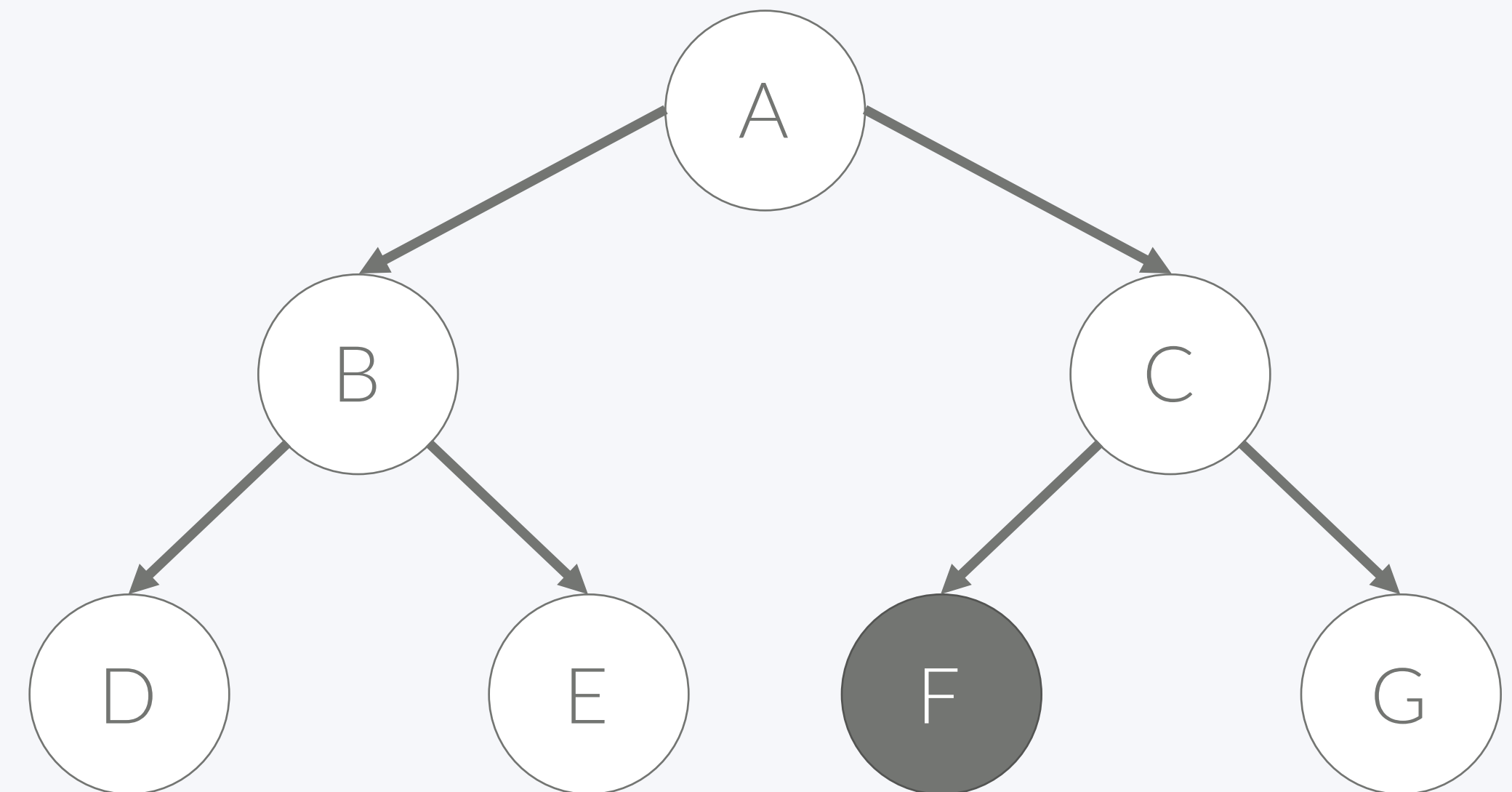


포스트오더

58

Postorder

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

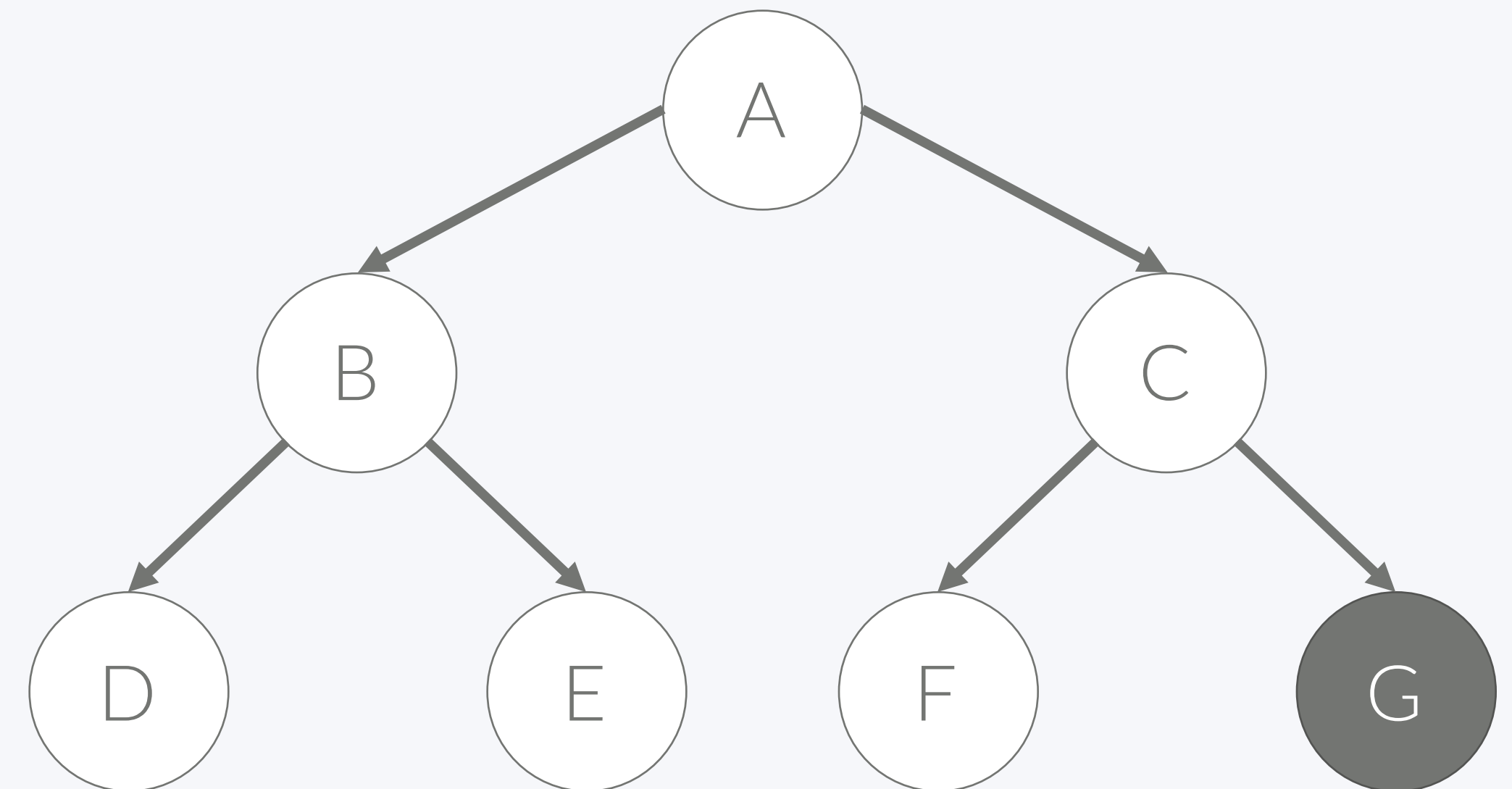


포스트오더

59

Postorder

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

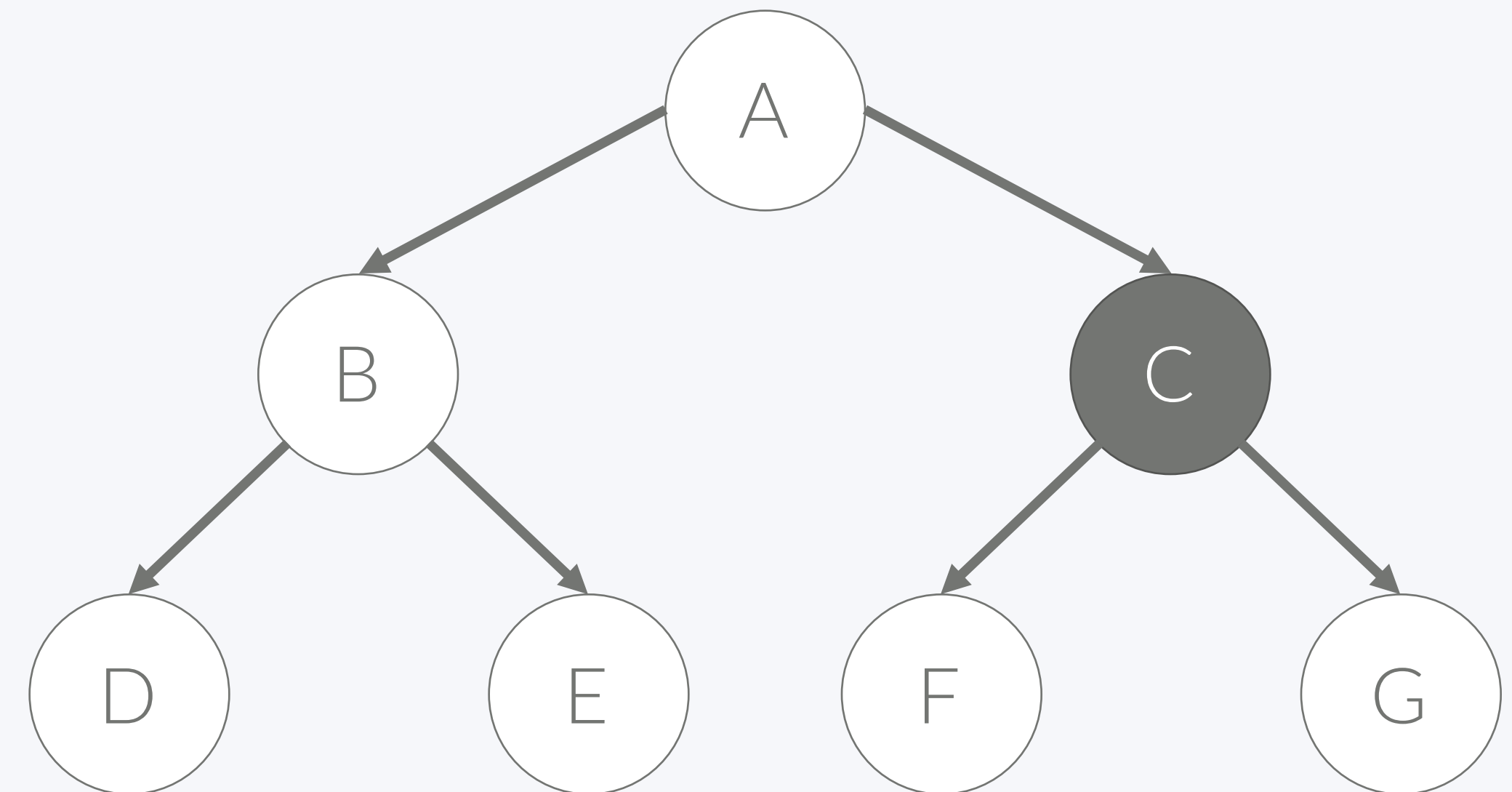


포스트오더

60

Postorder

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

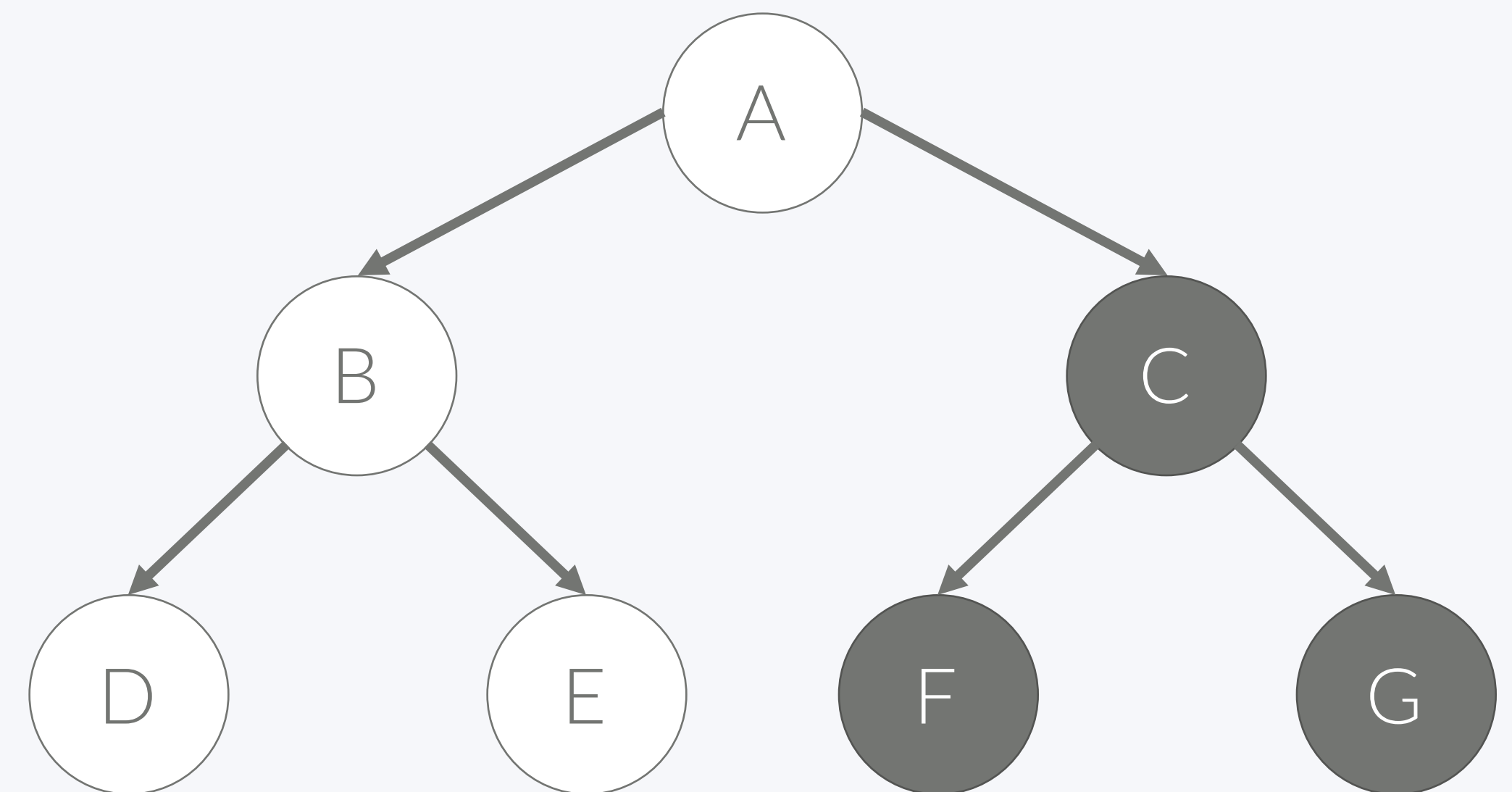


포스트오더

61

Postorder

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

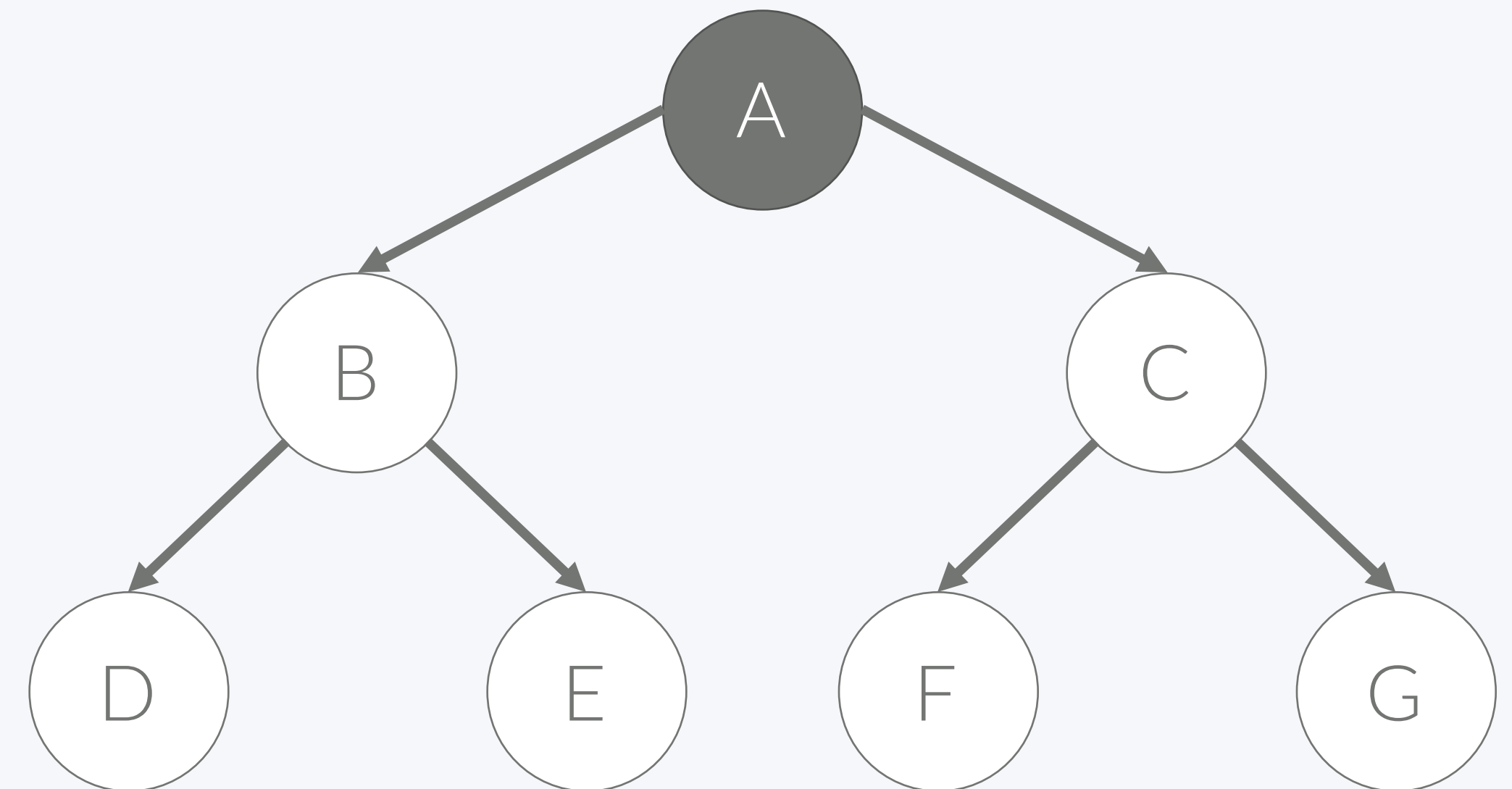


포스트오더

62

Postorder

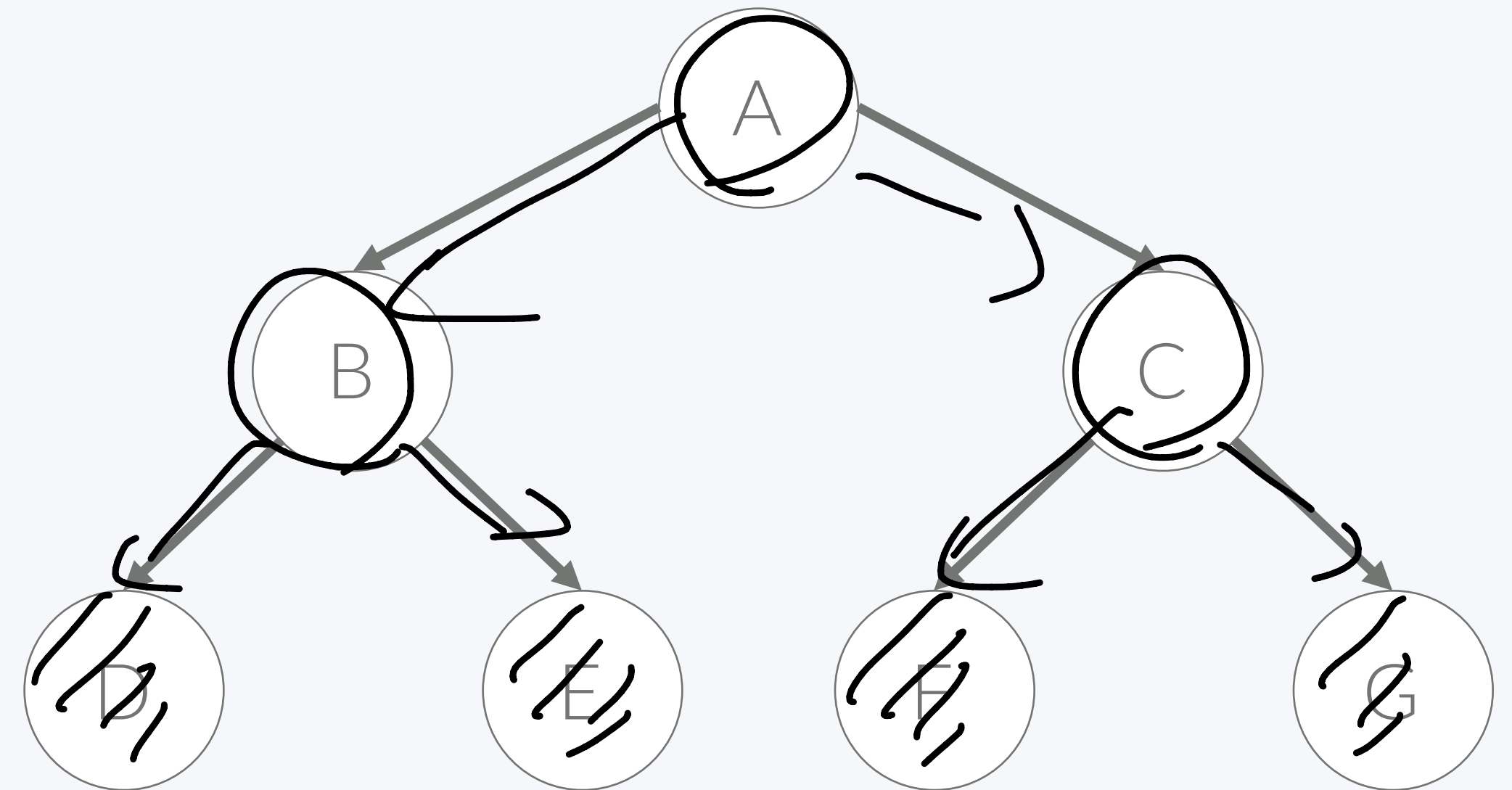
- 왼쪽 자식 포스트오더
 - DEB
- 오른쪽 자식 포스트오더
 - FGC
- 노드 방문
 - A



포스트오더

Postorder

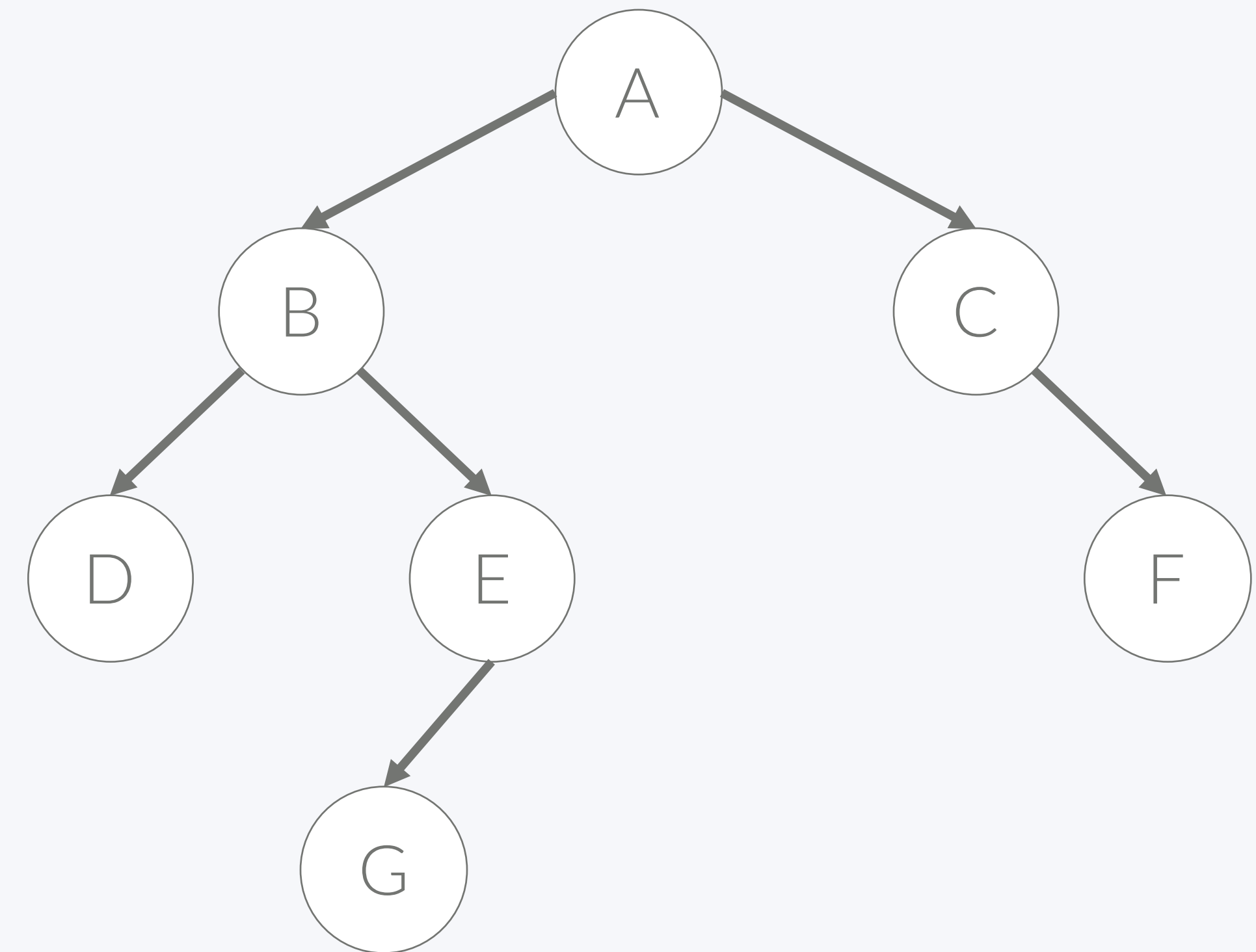
- DEBFGCA



프리오더

Pre-order

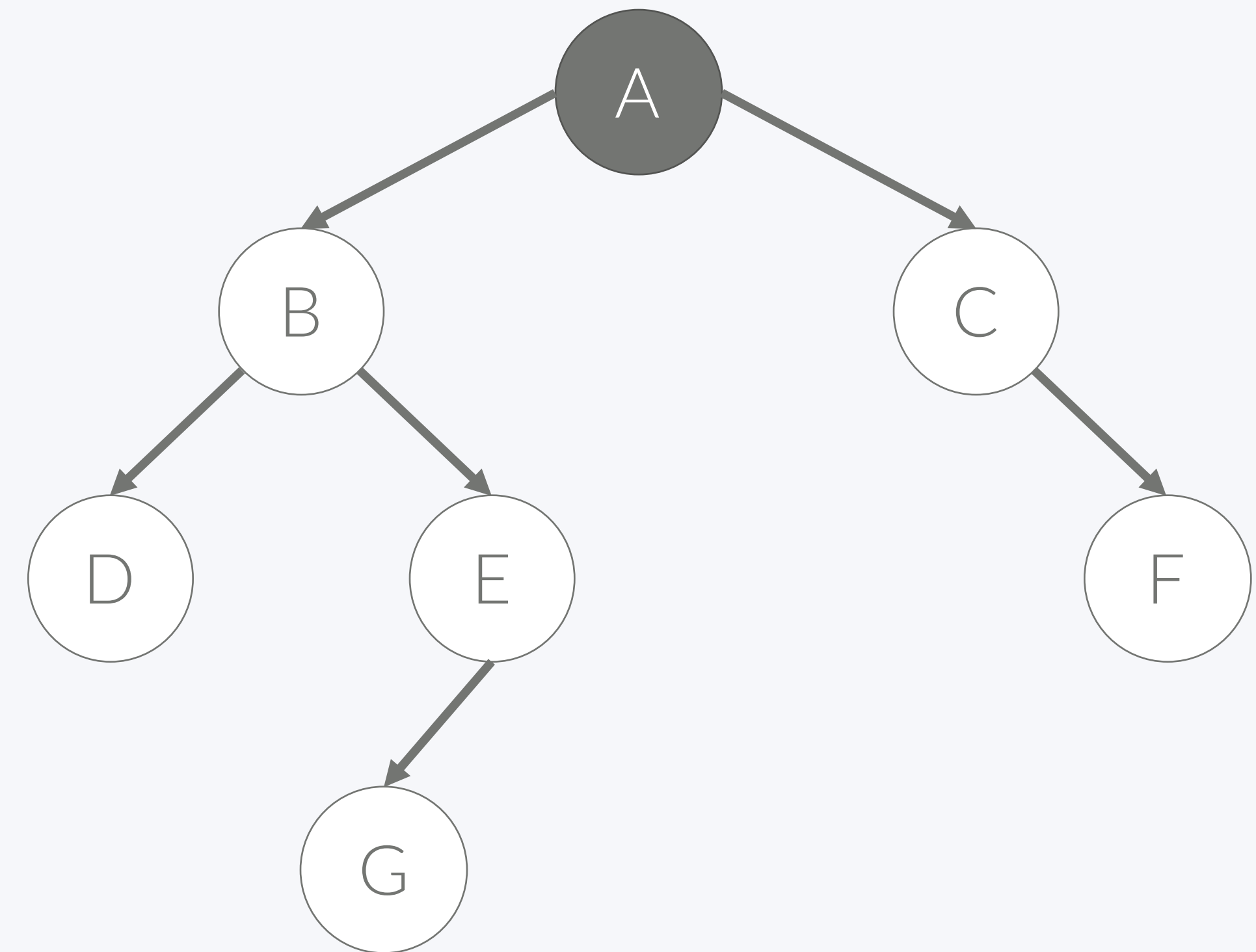
- 노드 방문
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

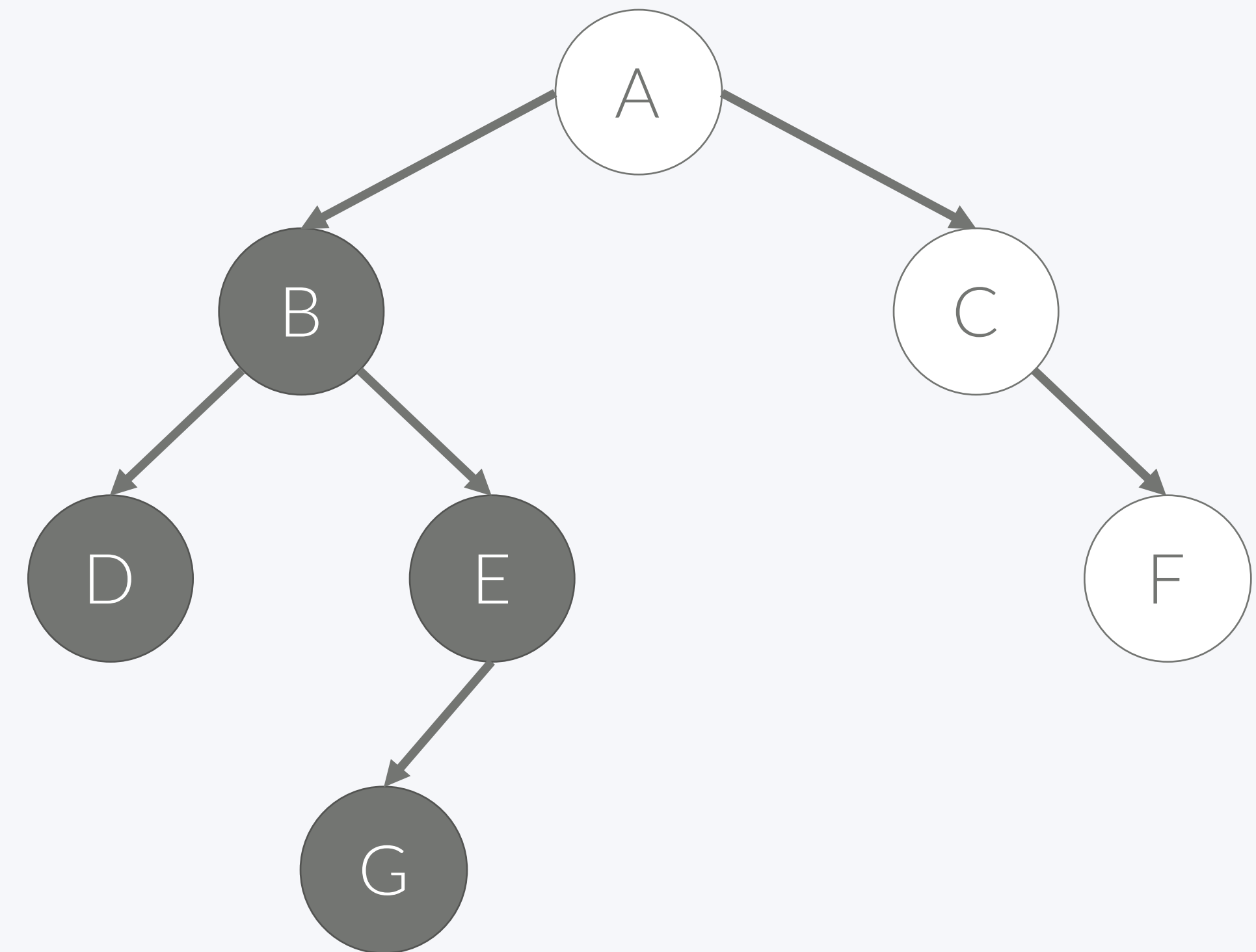
- 노드 방문
 - A
- 왼쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

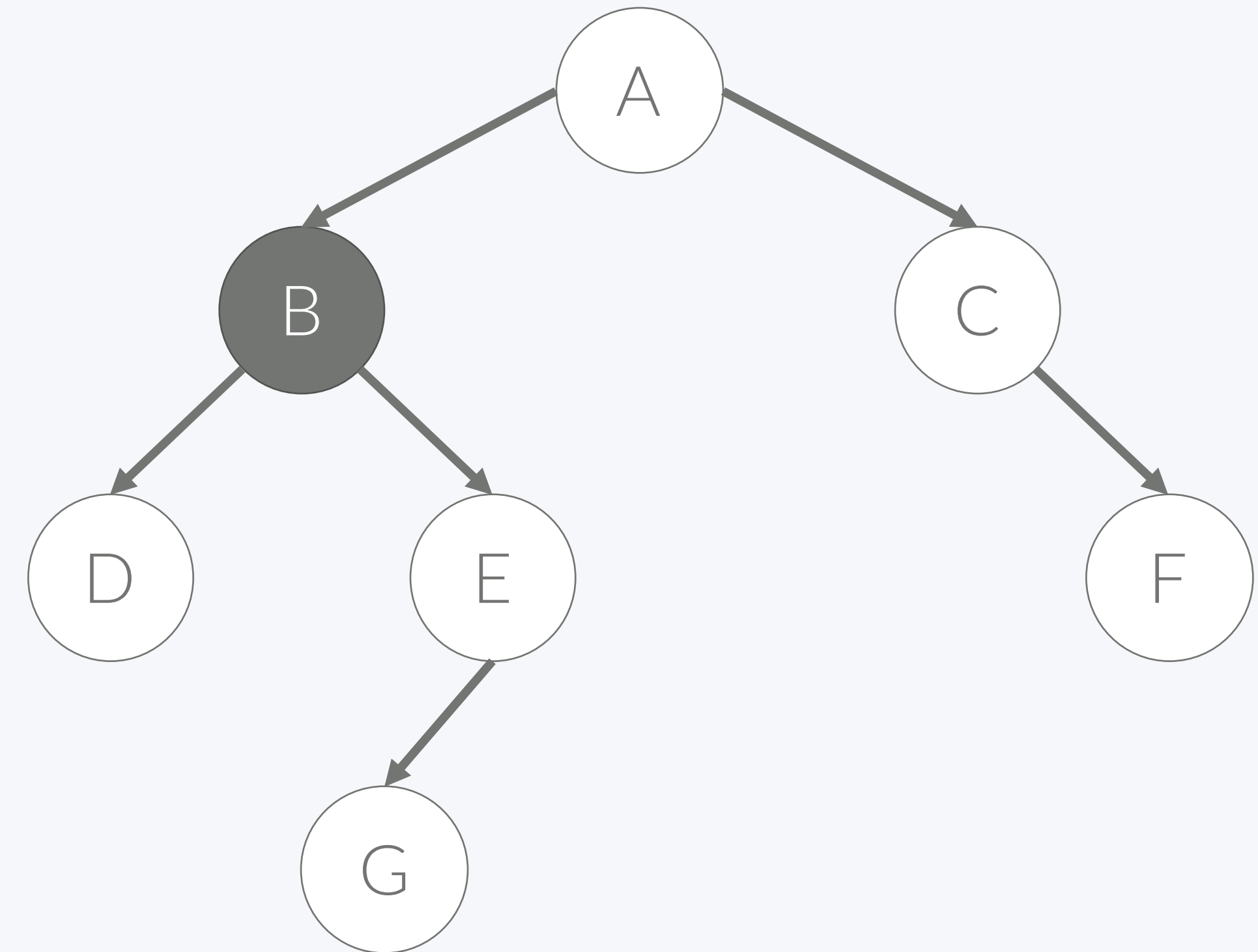
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
- 오른쪽 자식 프리오더



프리오더

Pre-order

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

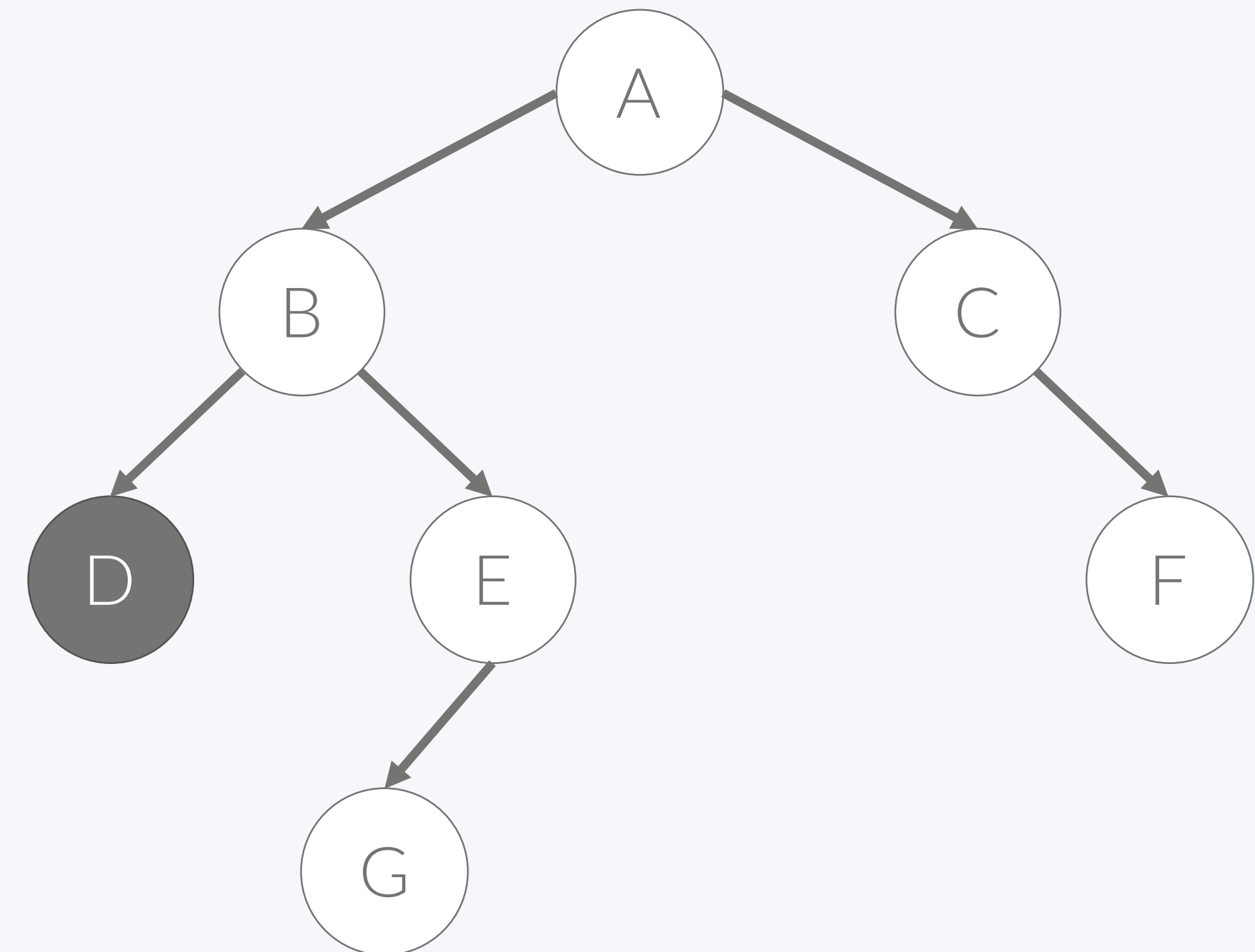


프리오더

68

Pre-order

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

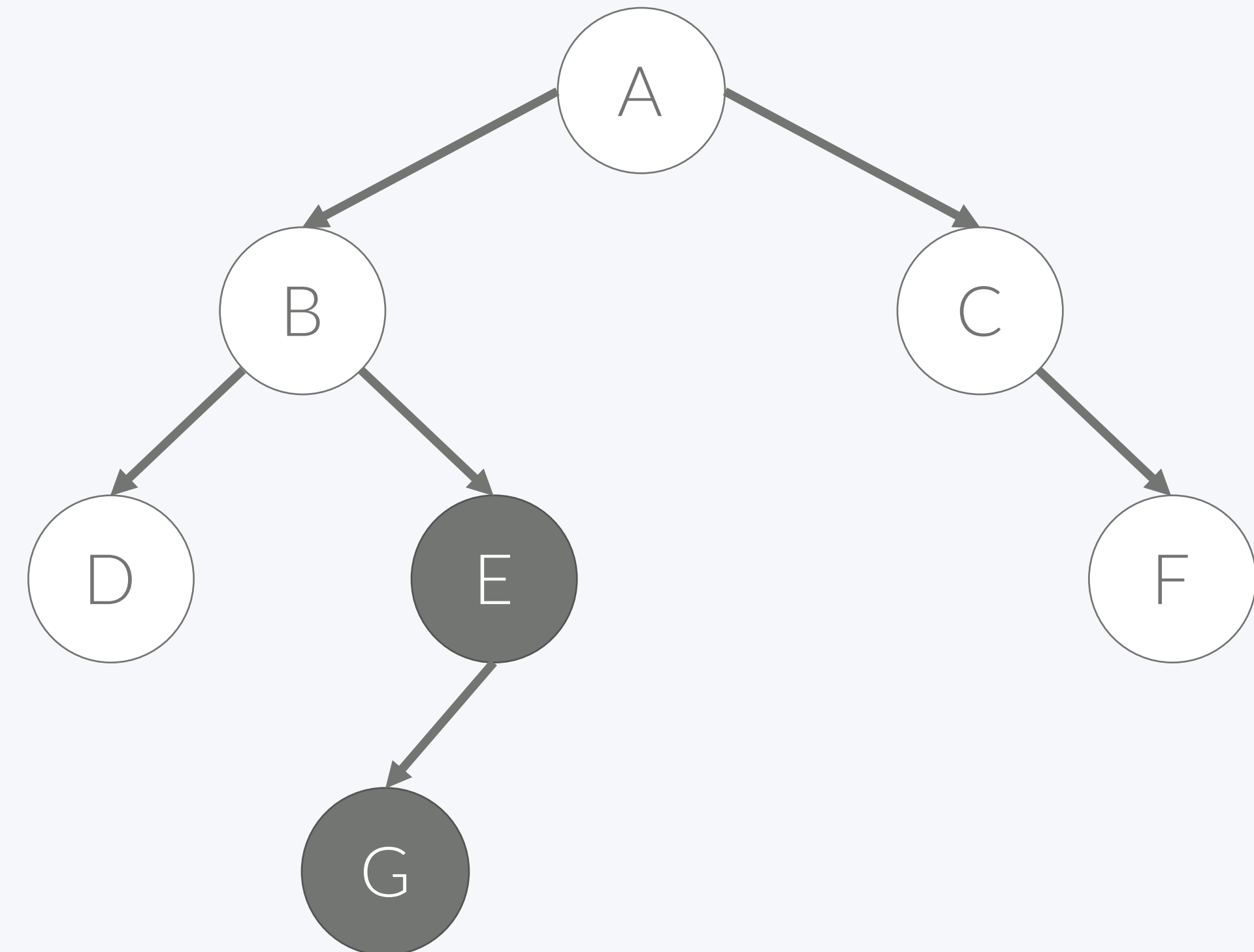


프리오더

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

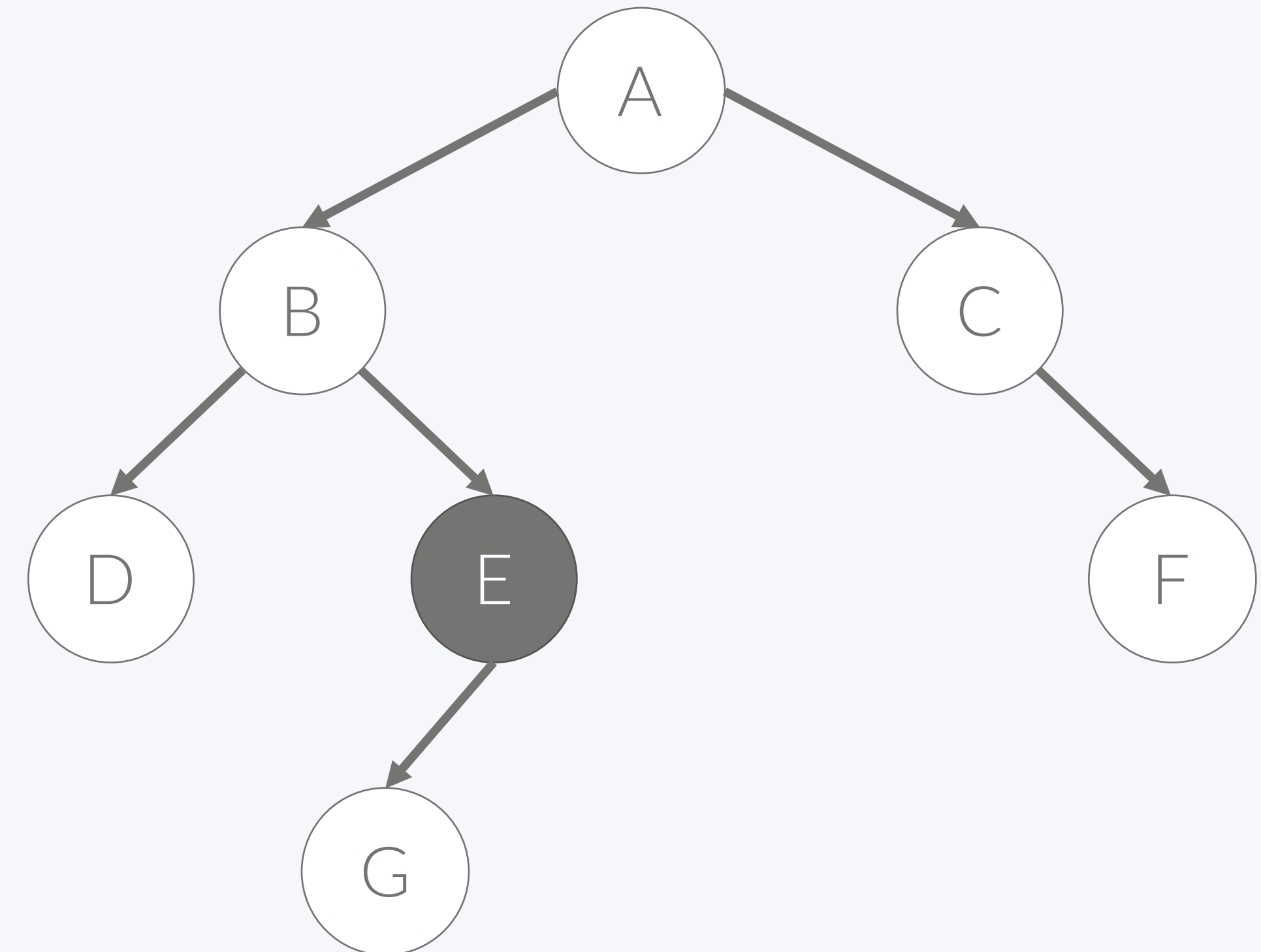


프리오더

70

Pre-order

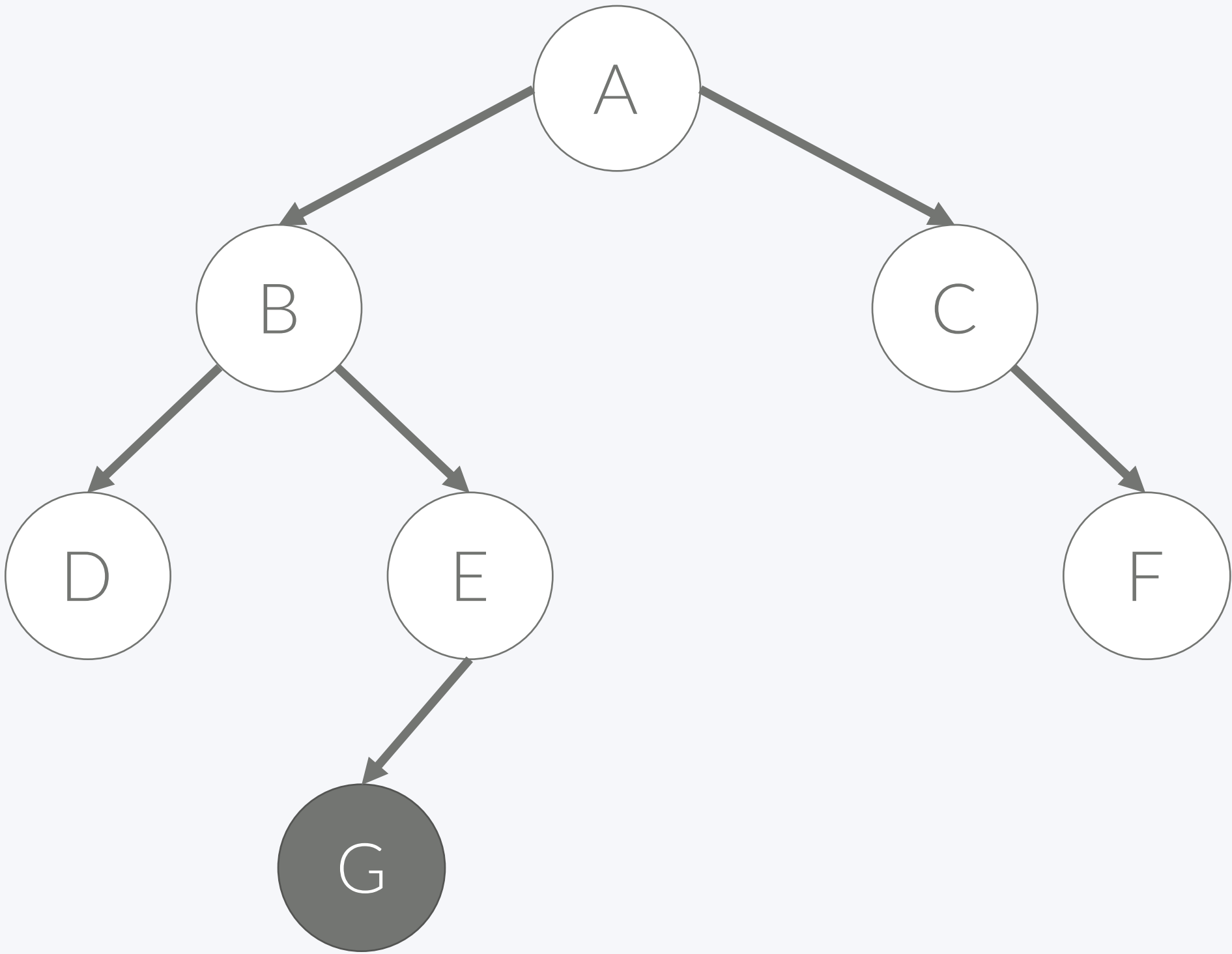
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
- 오른쪽 자식 프리오더
 - 노드 방문
 - E
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더



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

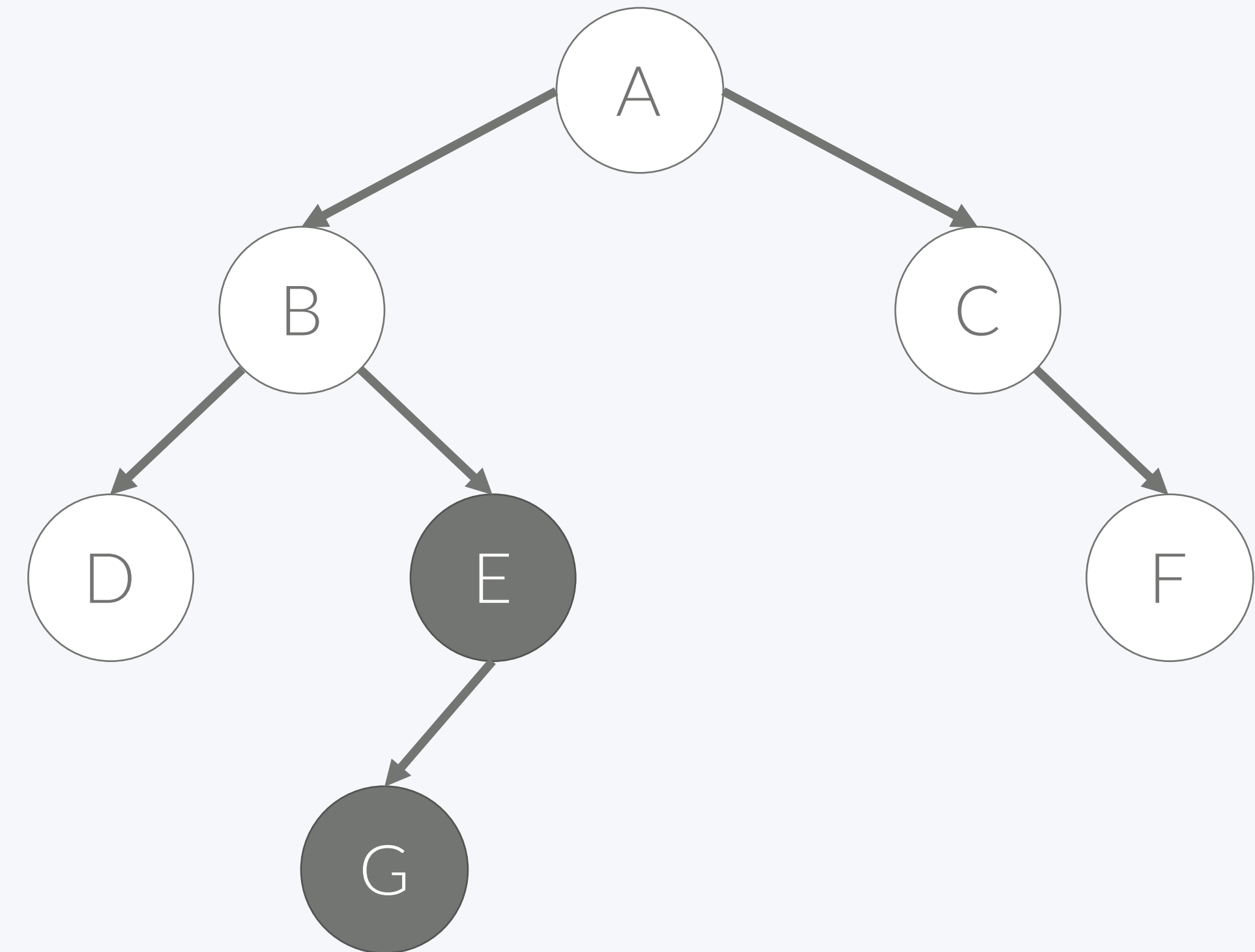


프리오더

72

Pre-order

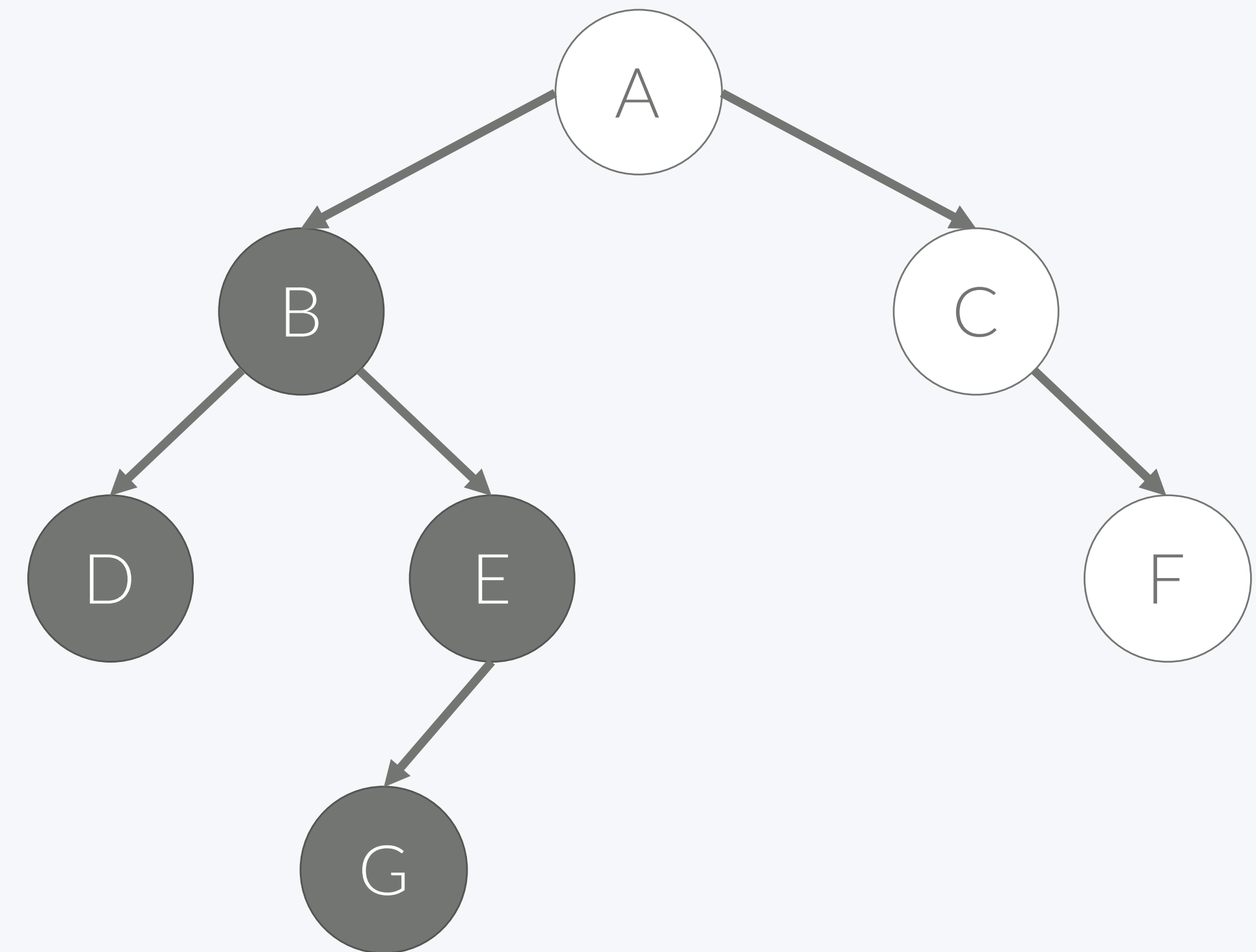
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - 노드 방문
 - B
 - 왼쪽 자식 프리오더
 - D
 - 오른쪽 자식 프리오더
 - EG
- 오른쪽 자식 프리오더



프리오더

Pre-order

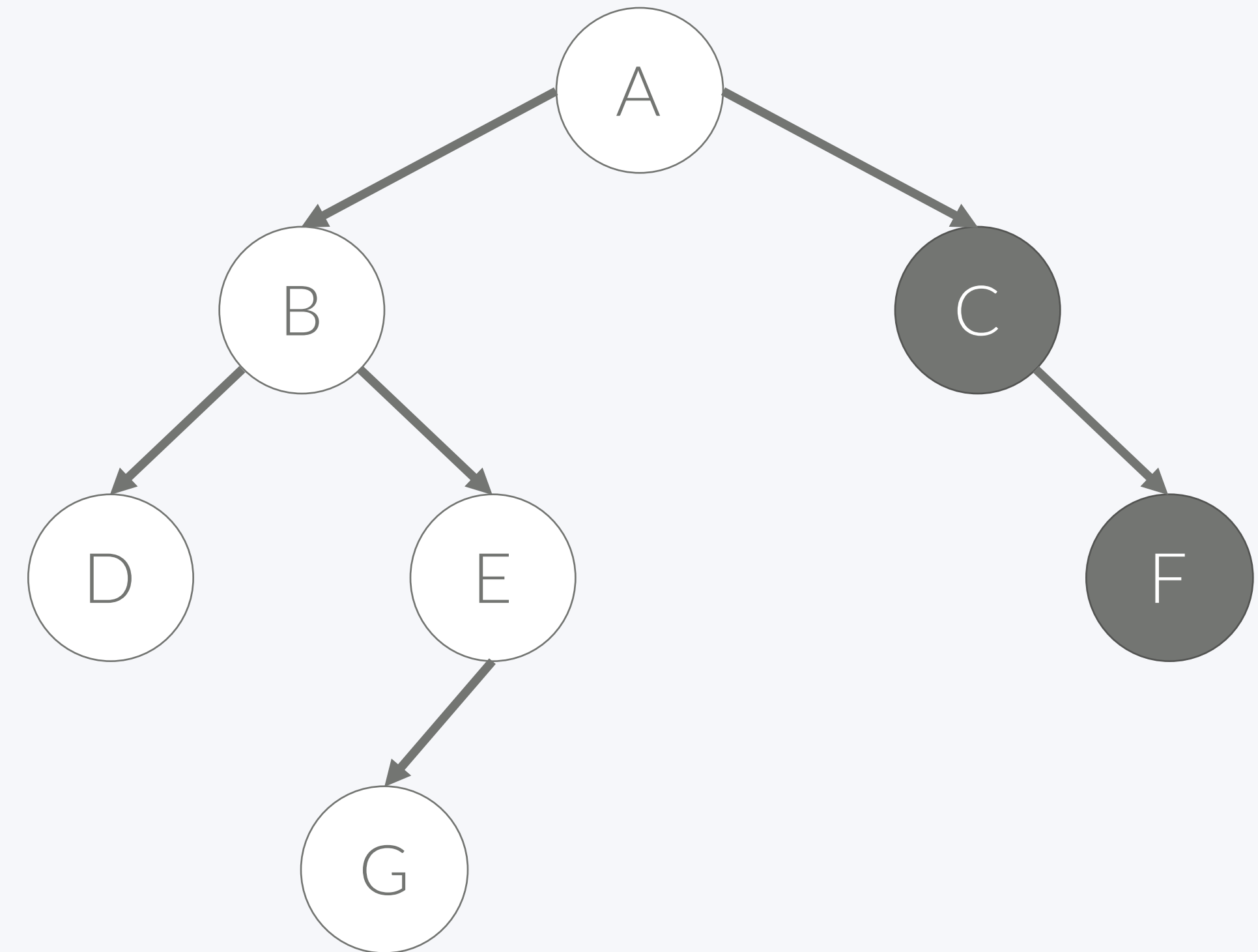
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더



프리오더

Pre-order

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

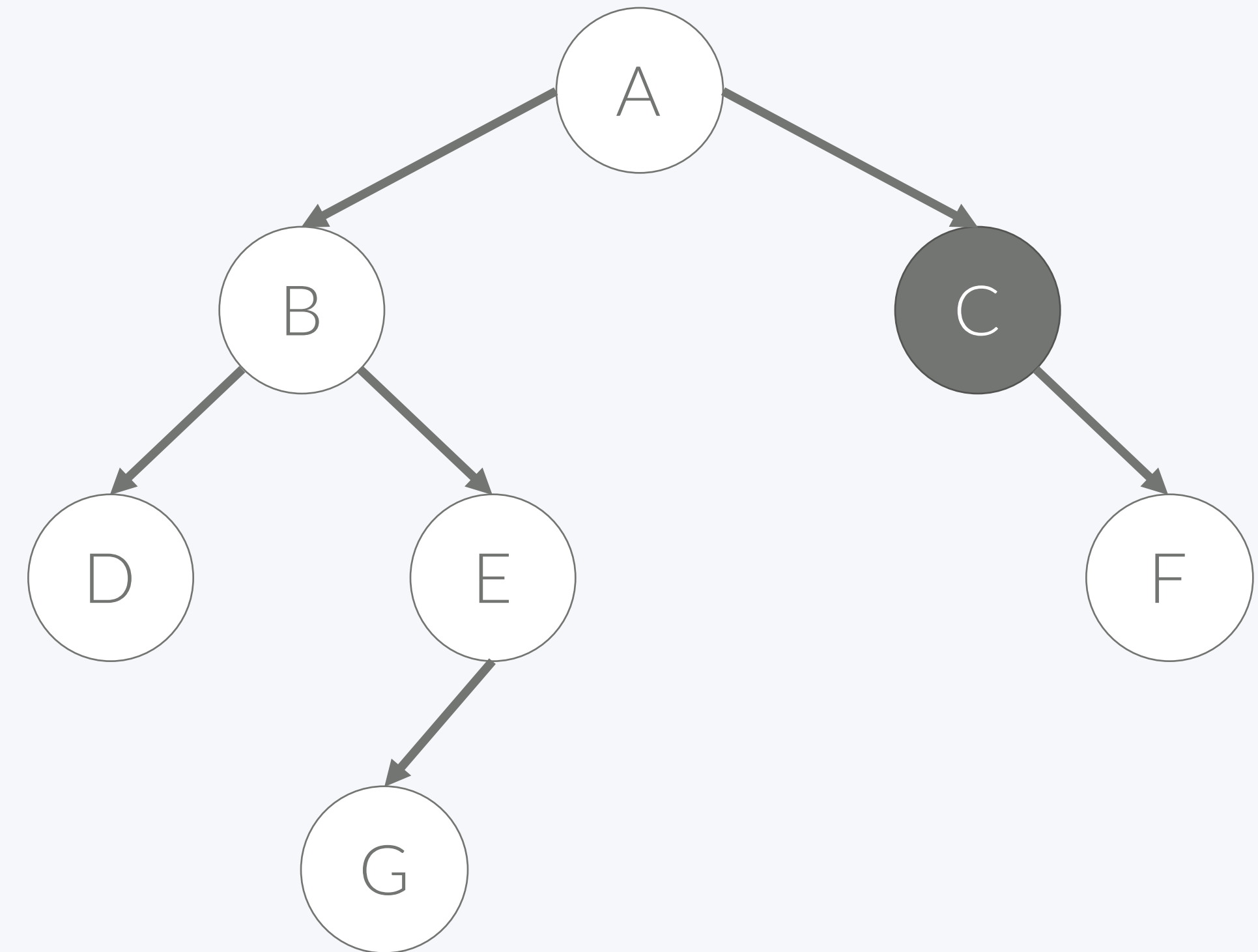


프리오더

75

Pre-order

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

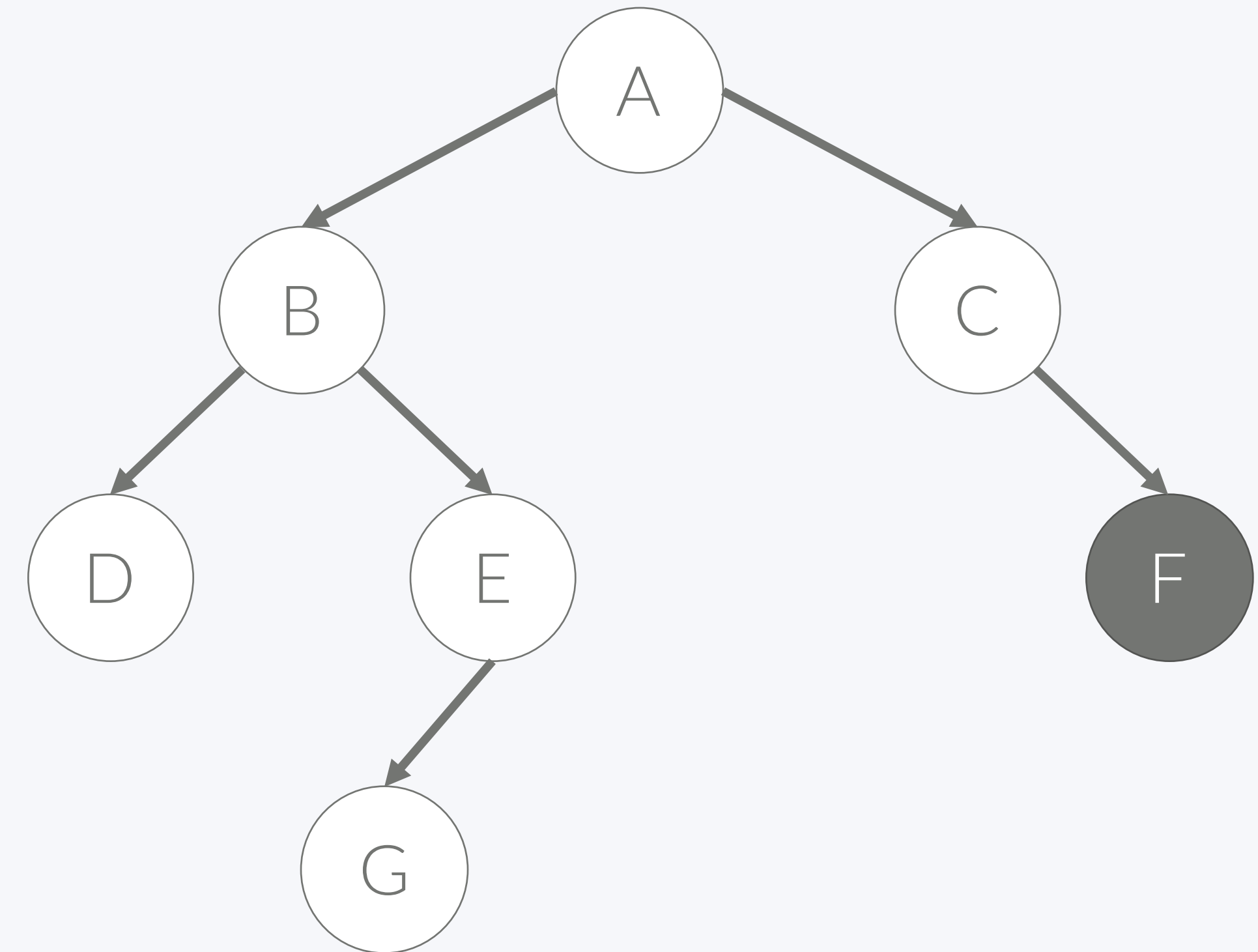


프리오더

76

Pre-order

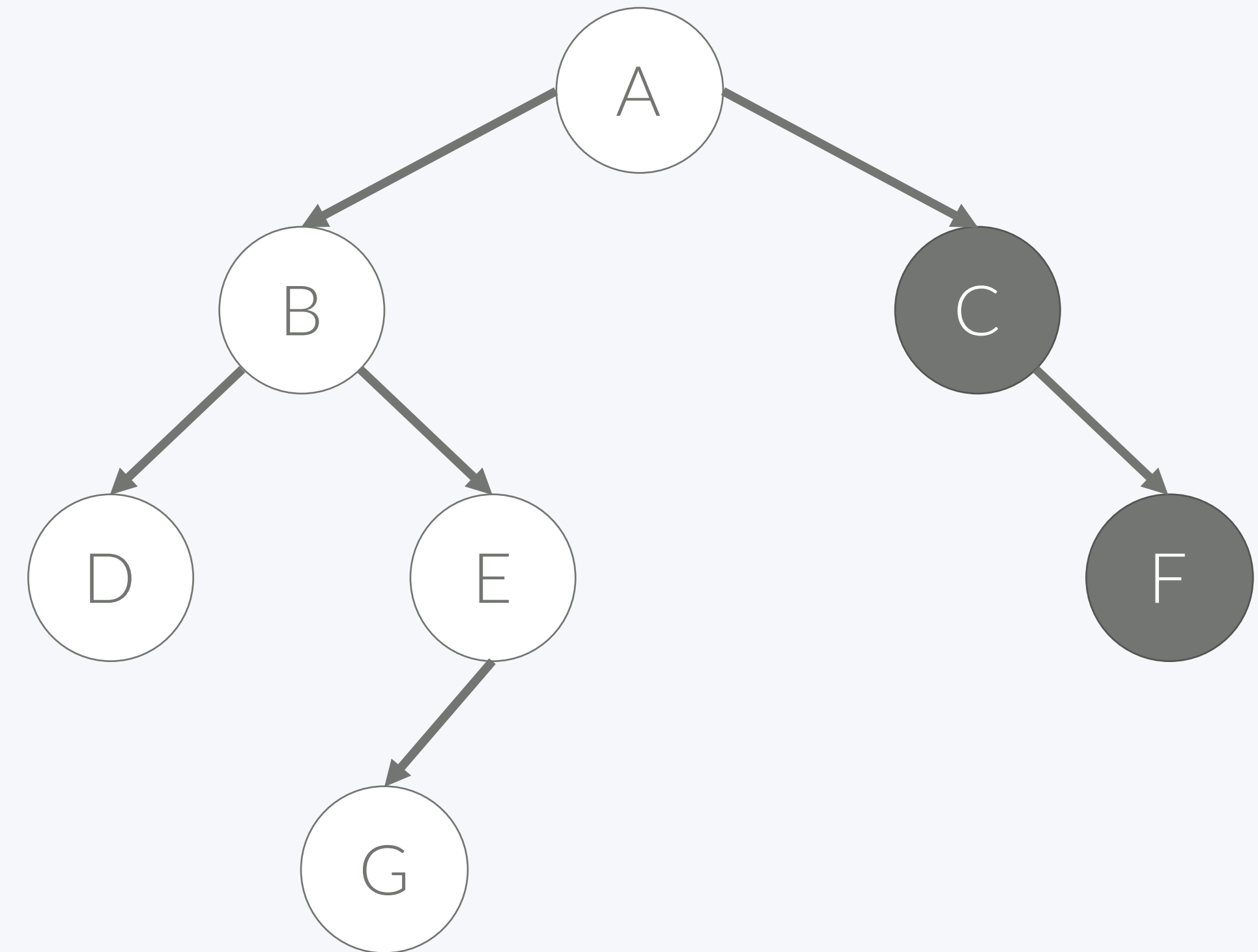
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더
 - 노드 방문
 - C
 - 왼쪽 자식 프리오더
 - 오른쪽 자식 프리오더
 - F



프리오더

Pre-order

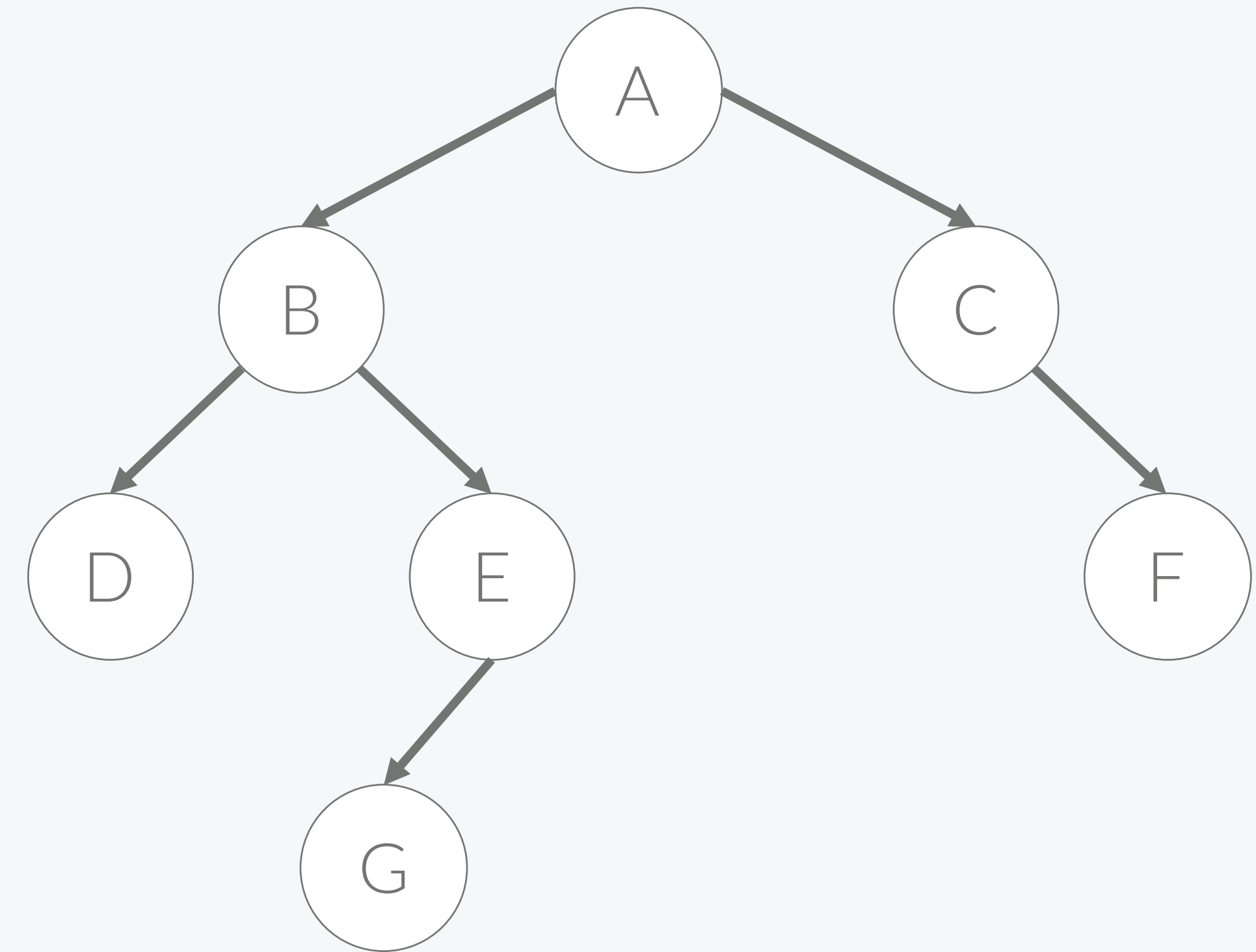
- 노드 방문
 - A
- 왼쪽 자식 프리오더
 - BDEG
- 오른쪽 자식 프리오더
 - CF



프리오더

Pre-order

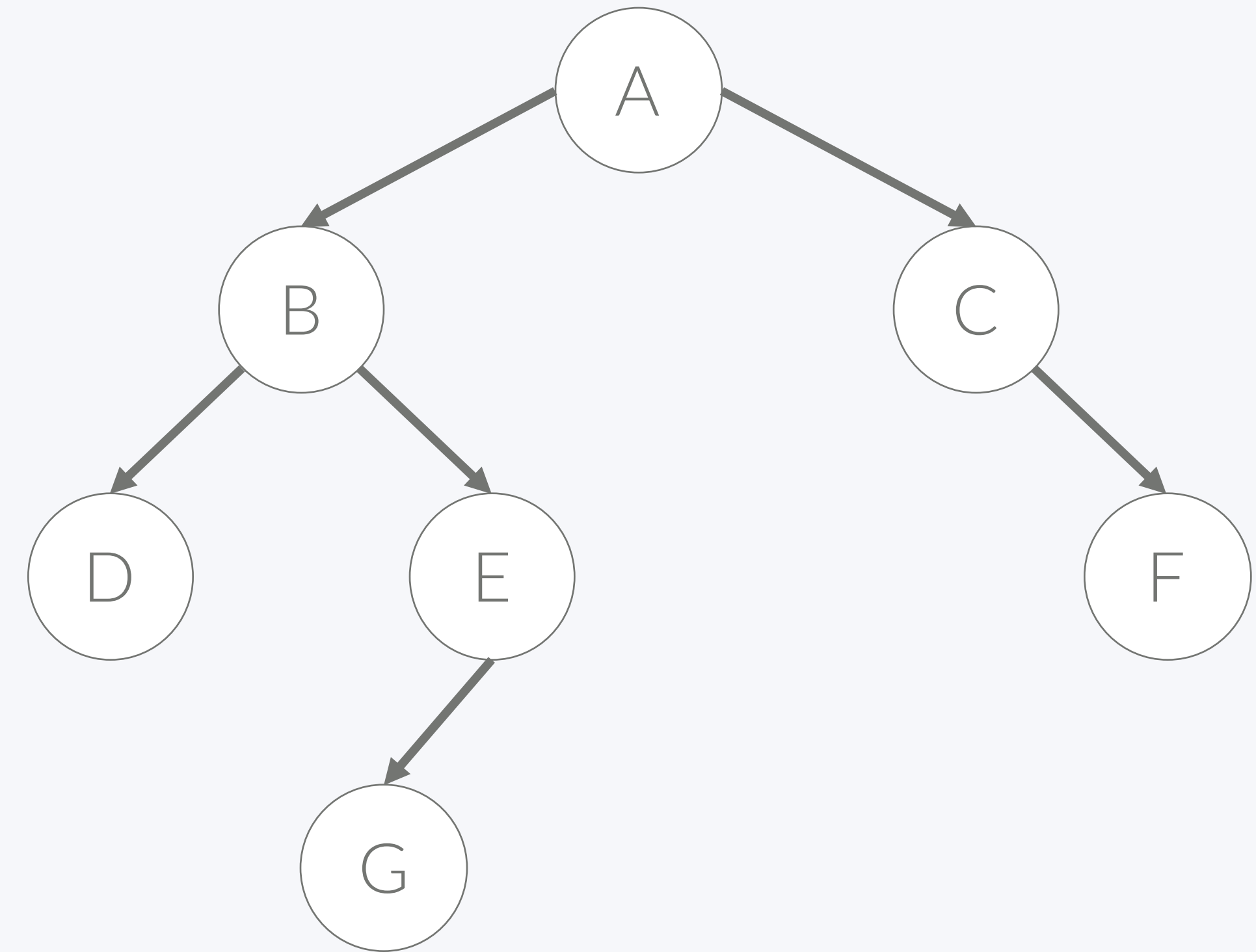
- ABDEGCF



인오더

In-order

- 왼쪽 자식 인오더
- 노드 방문
- 오른쪽 자식 인오더

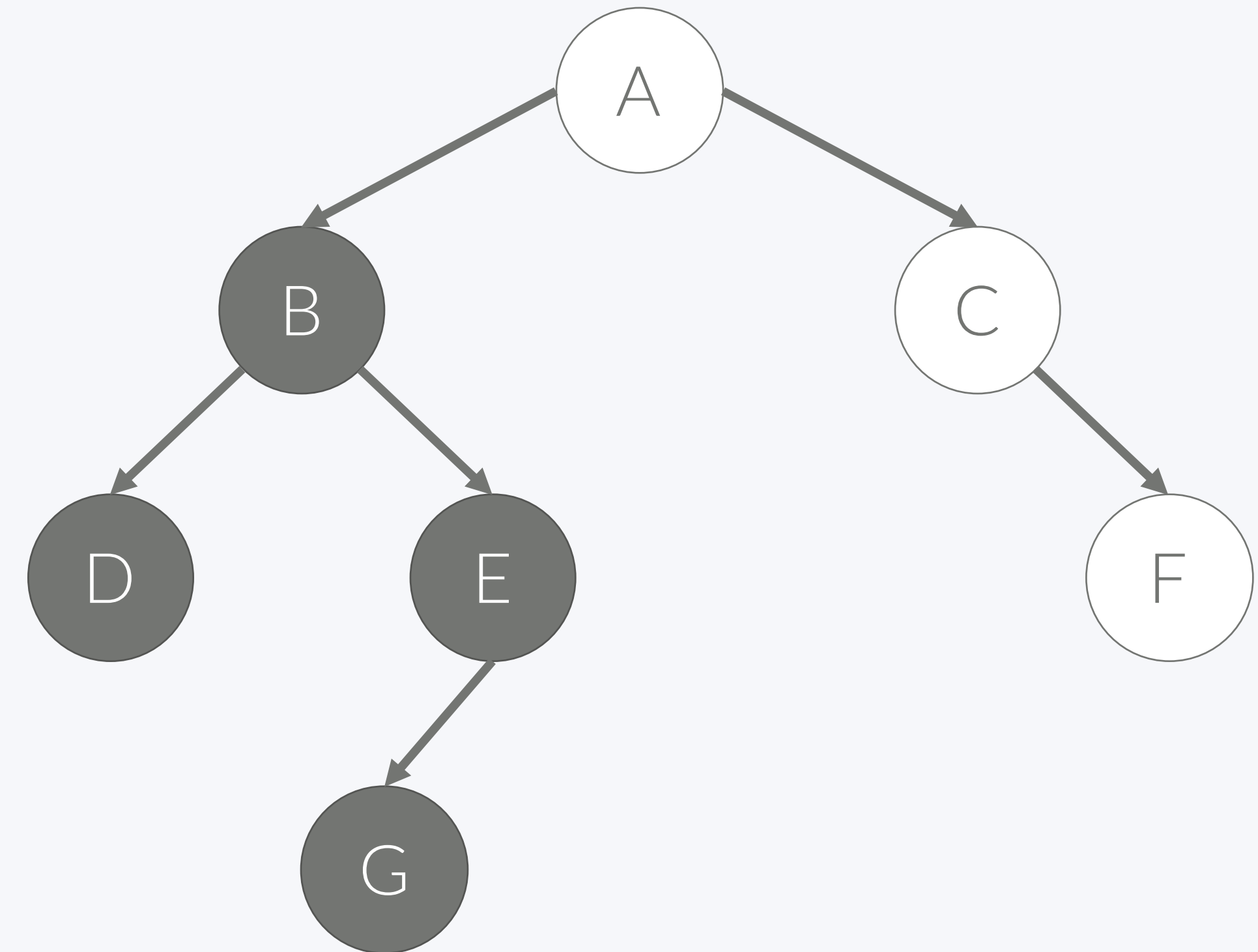


인오더

In-order

80

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

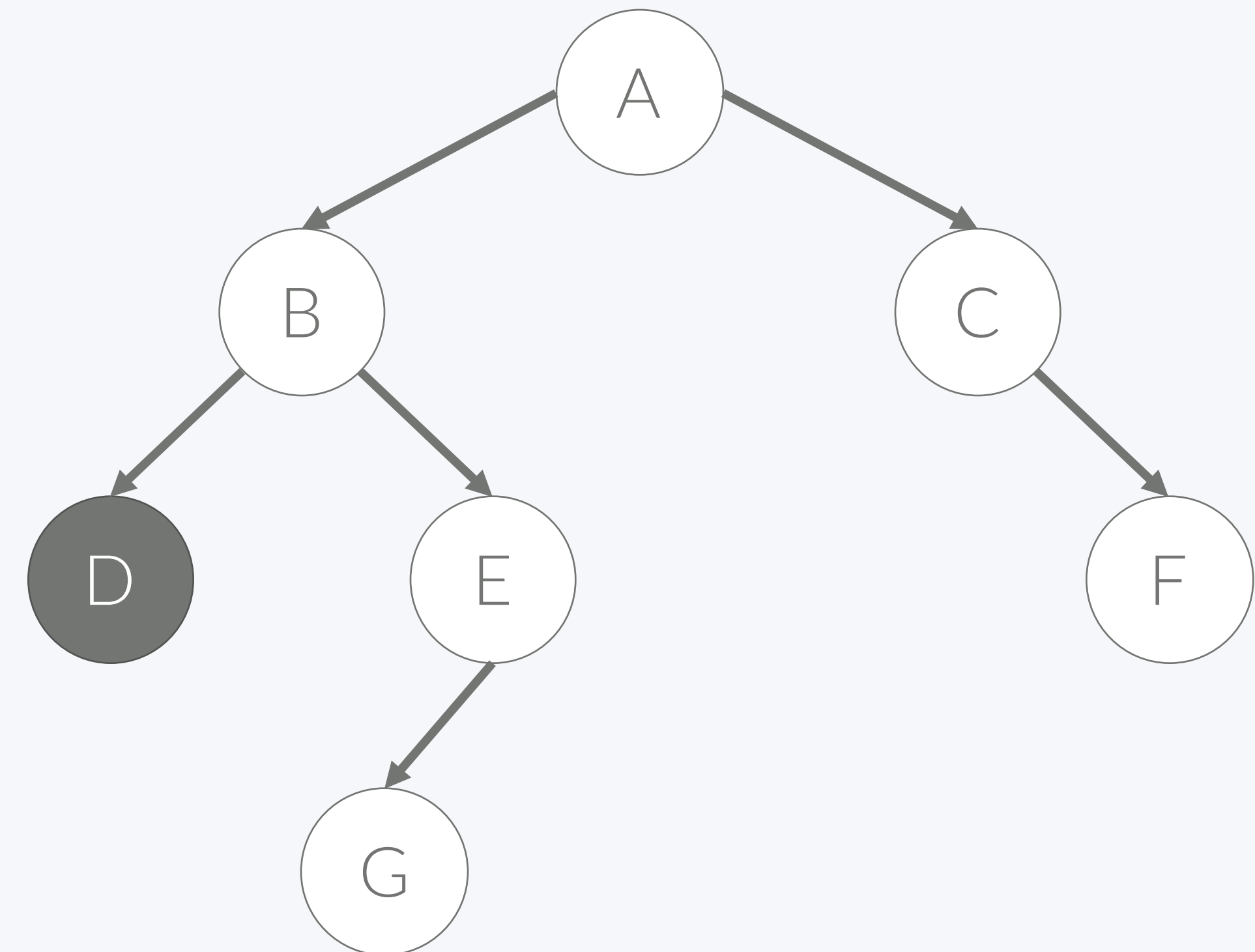


인오더

In-order

81

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

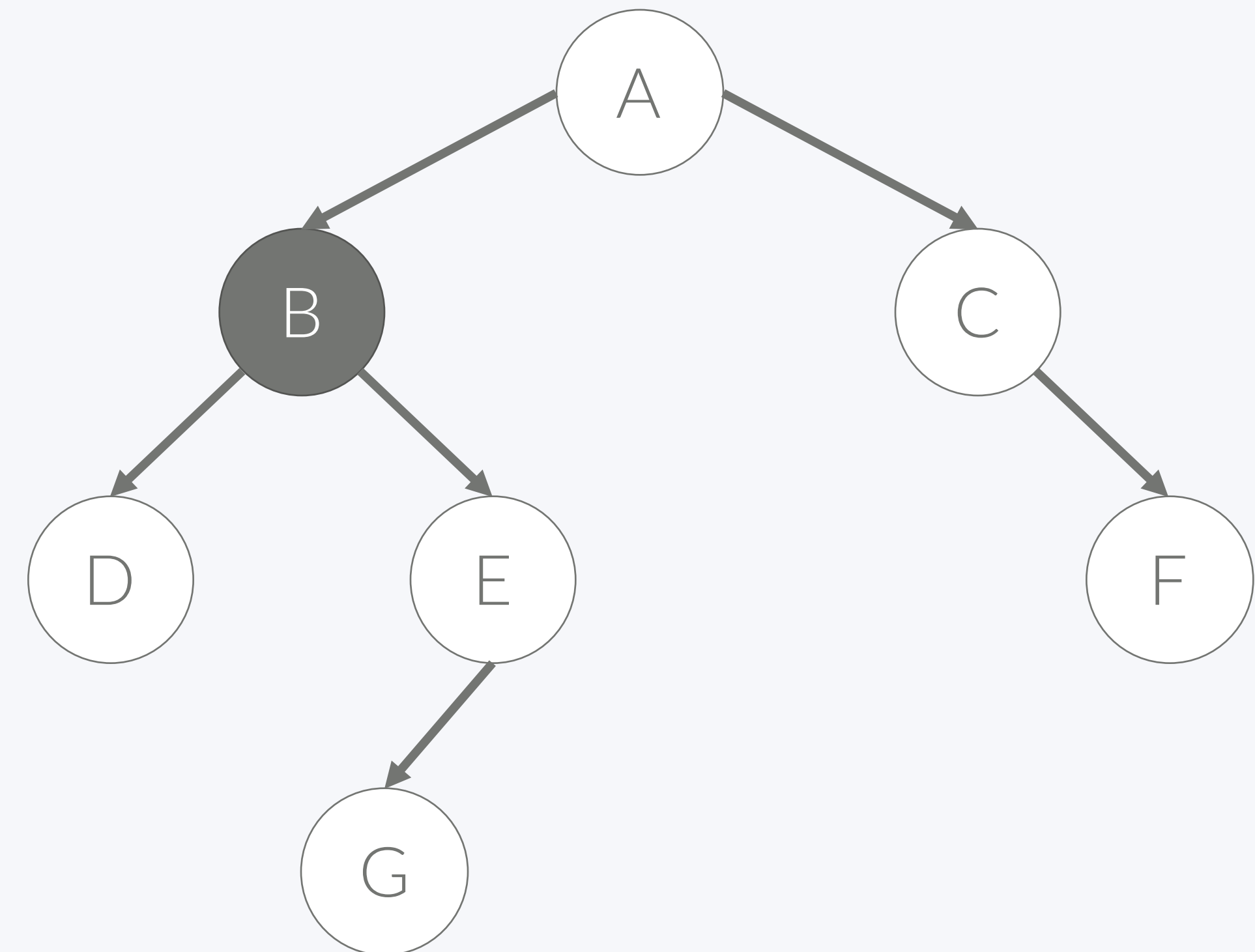


인오더

In-order

82

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

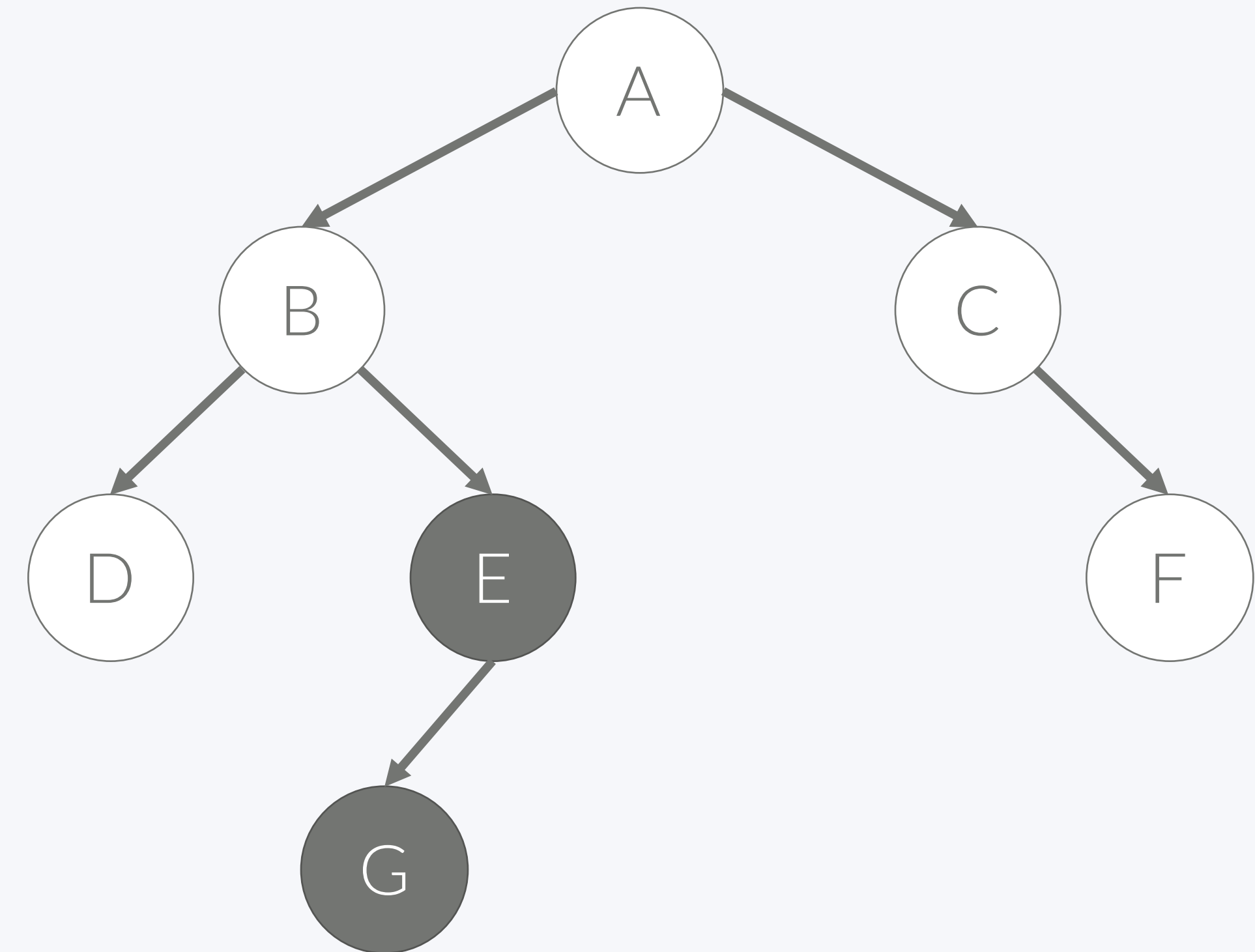


인오더

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

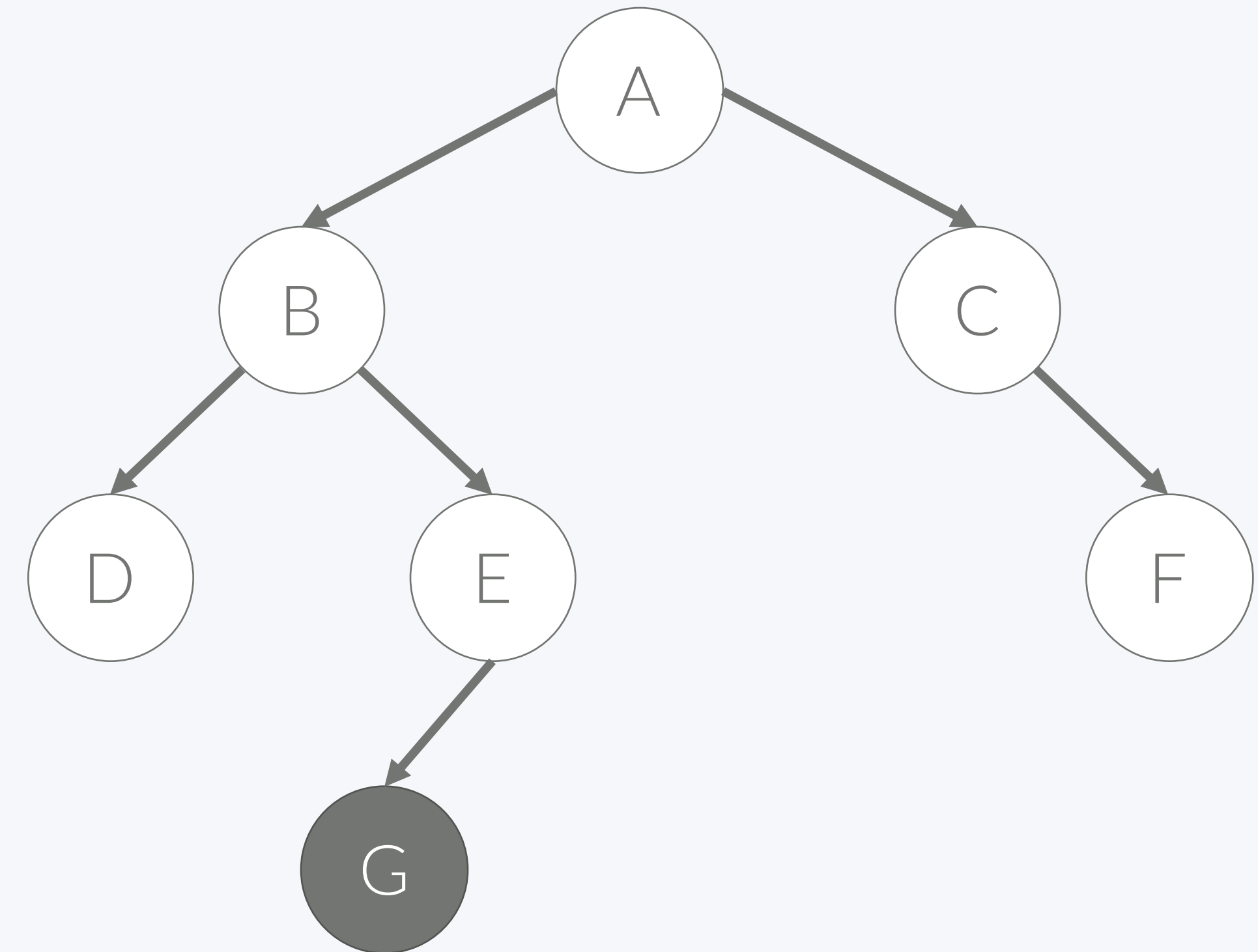


인오더

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

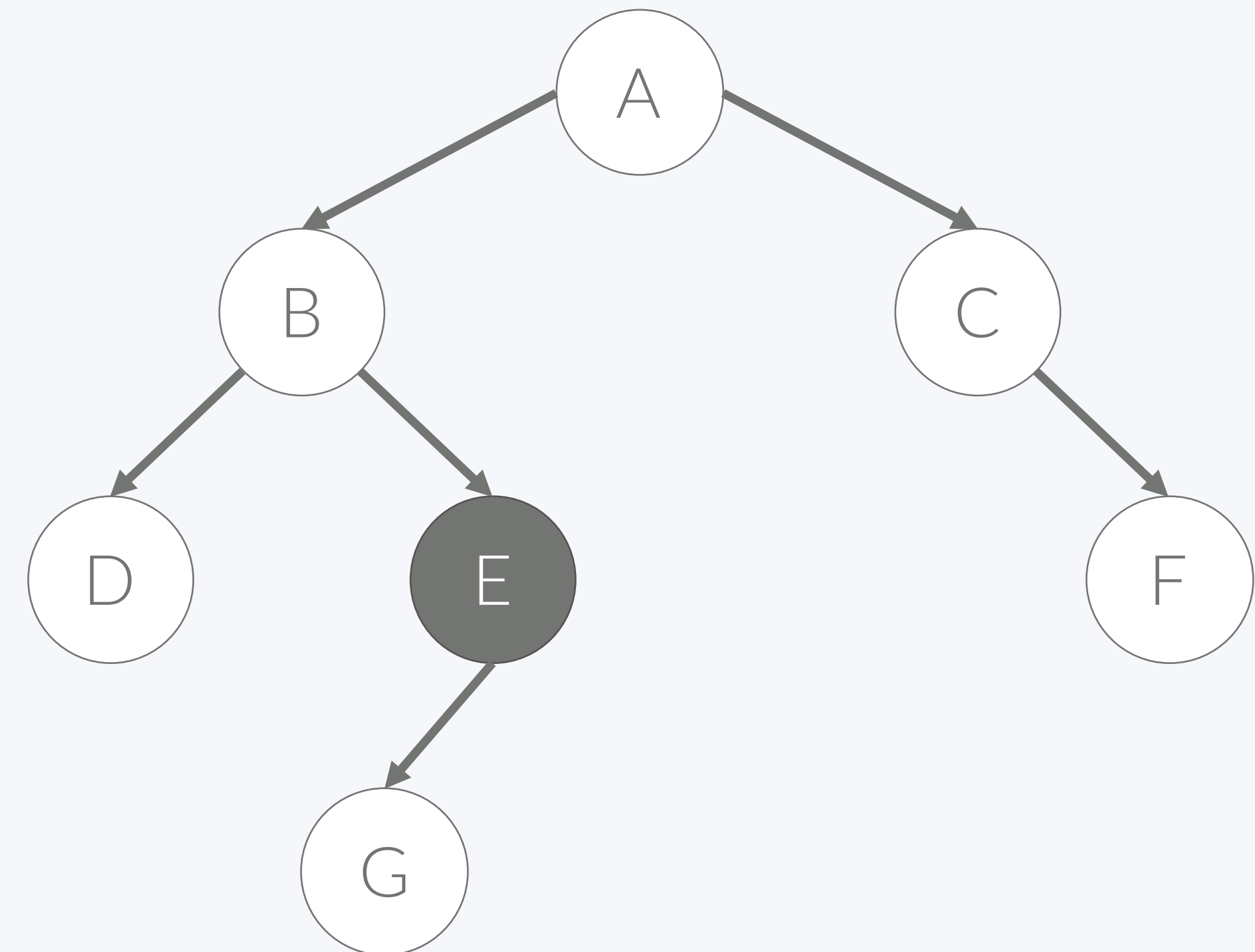
- G

- 노드 방문

- E

- 오른쪽 자식 인오더

- 노드 방문

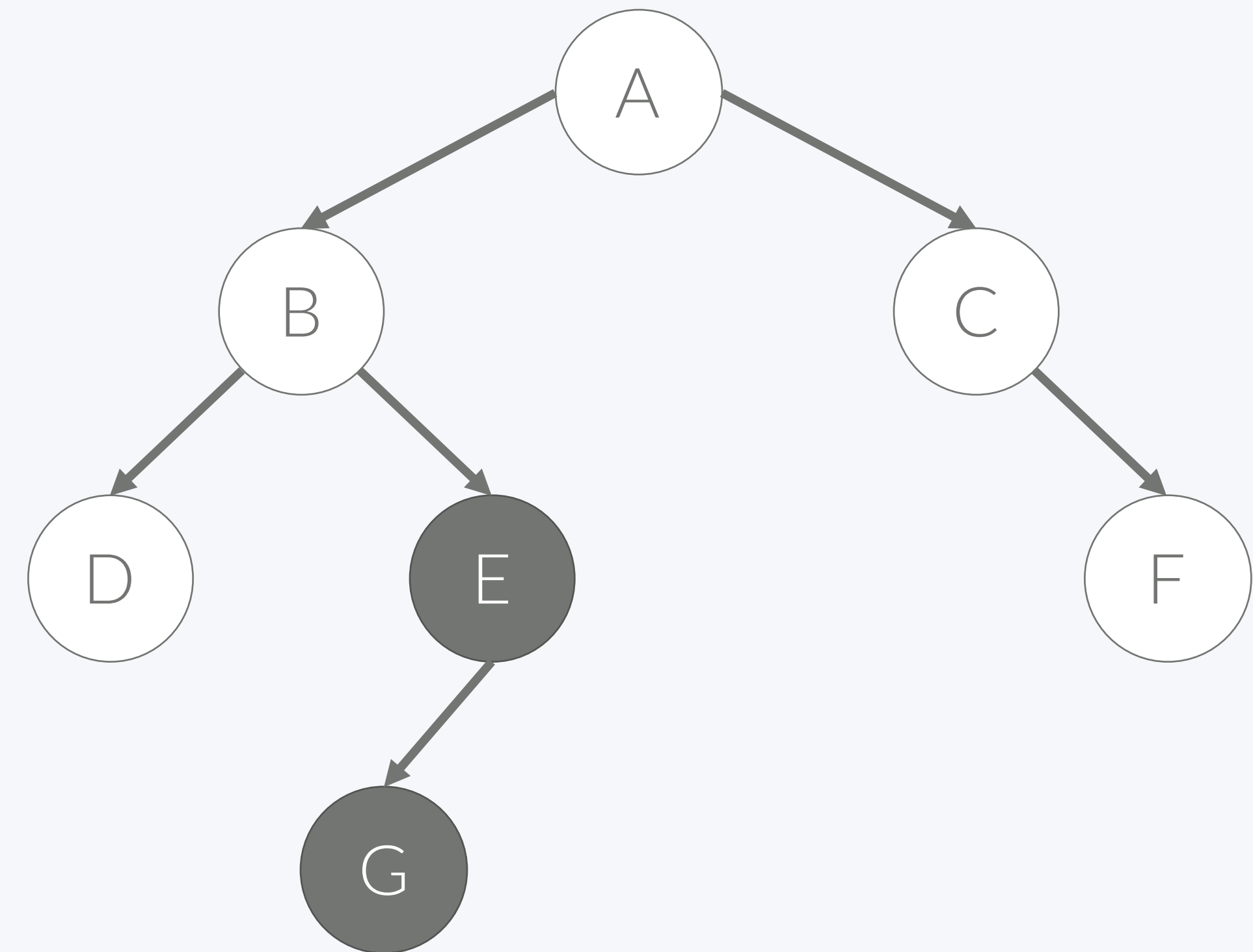


인오더

In-order

86

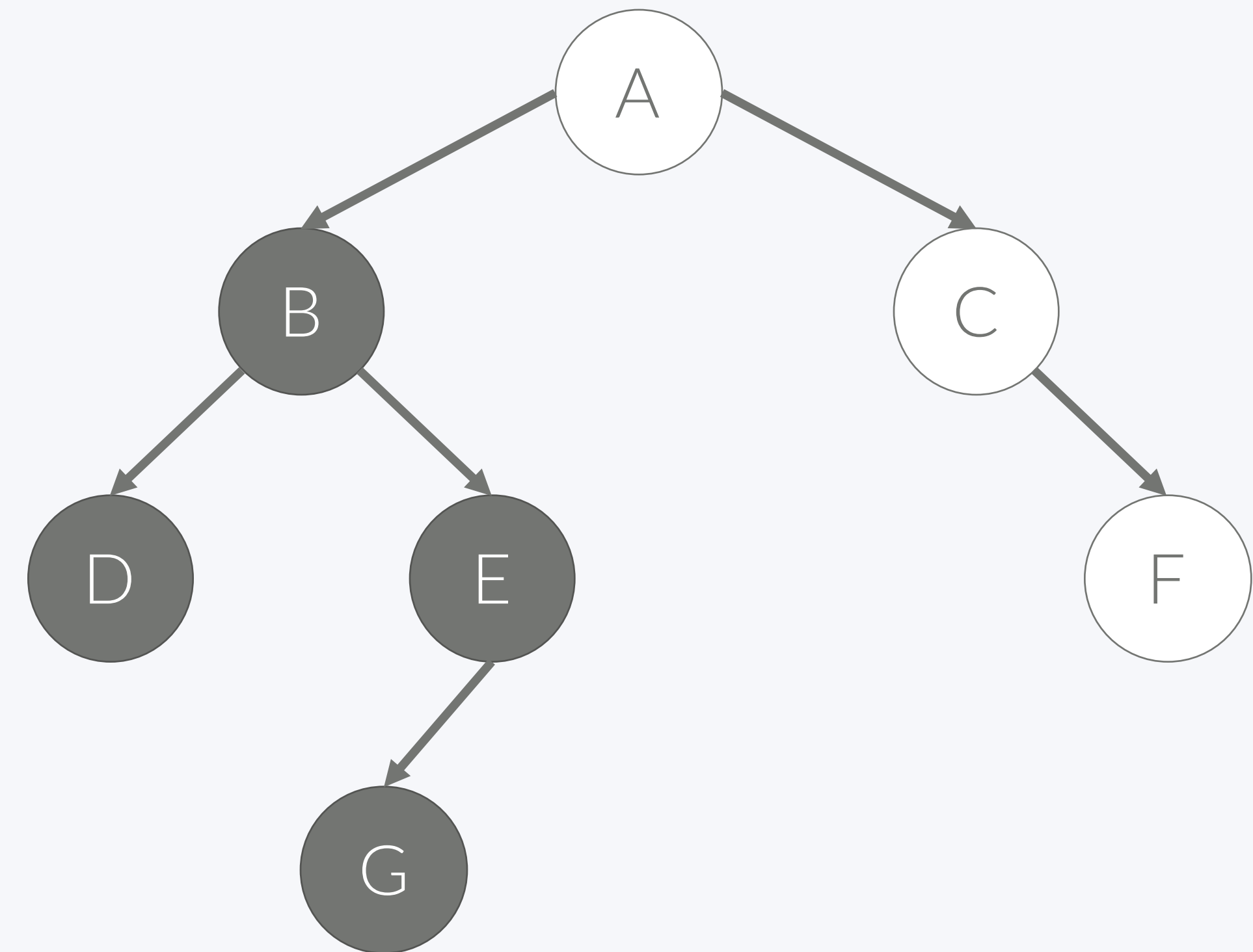
- 왼쪽 자식 인오더
 - 왼쪽 자식 인오더
 - D
 - 노드 방문
 - B
 - 오른쪽 자식 인오더
 - GE
- 노드 방문
- 오른쪽 자식 인오더



인오더

In-order

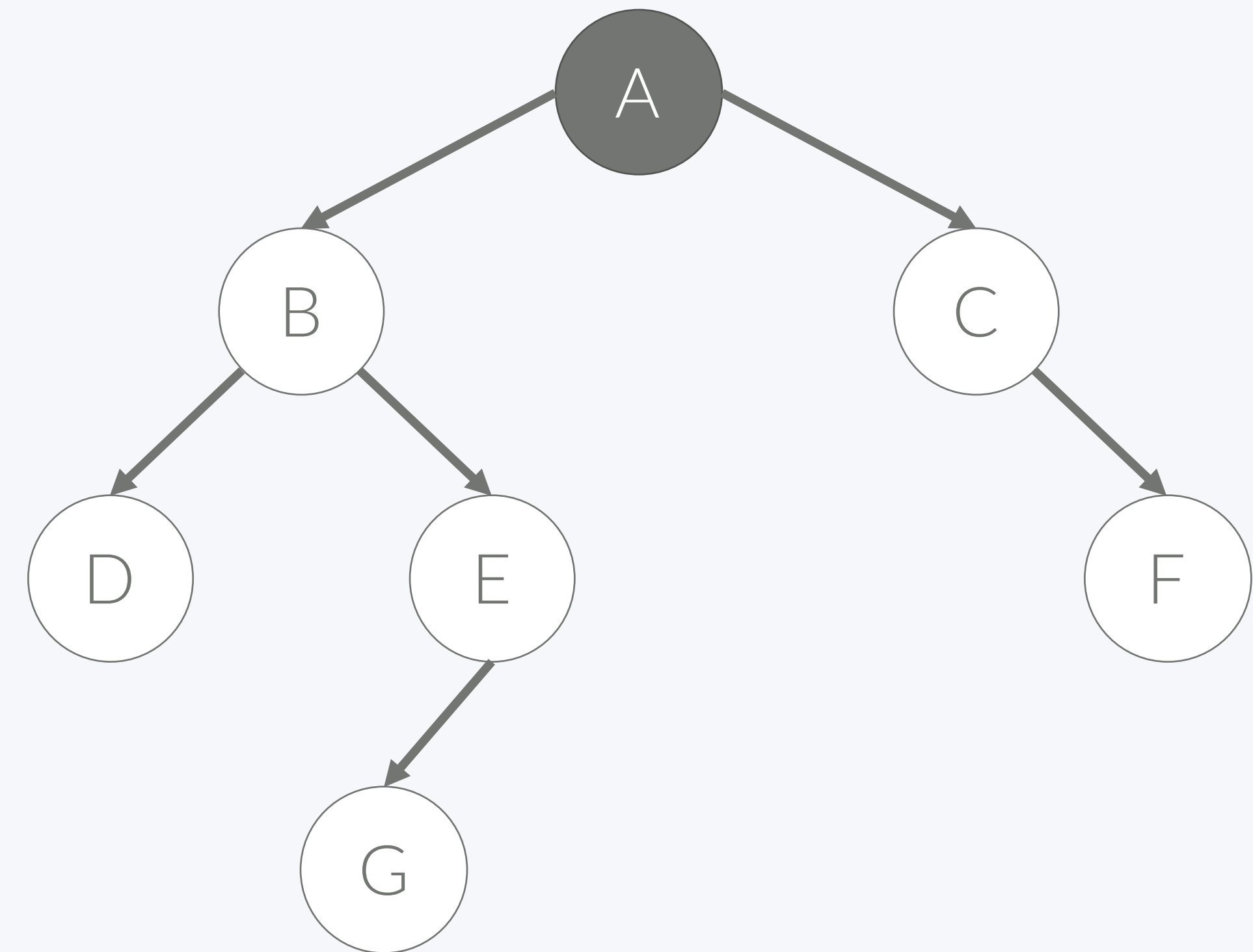
- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
- 오른쪽 자식 인오더



인오더

In-order

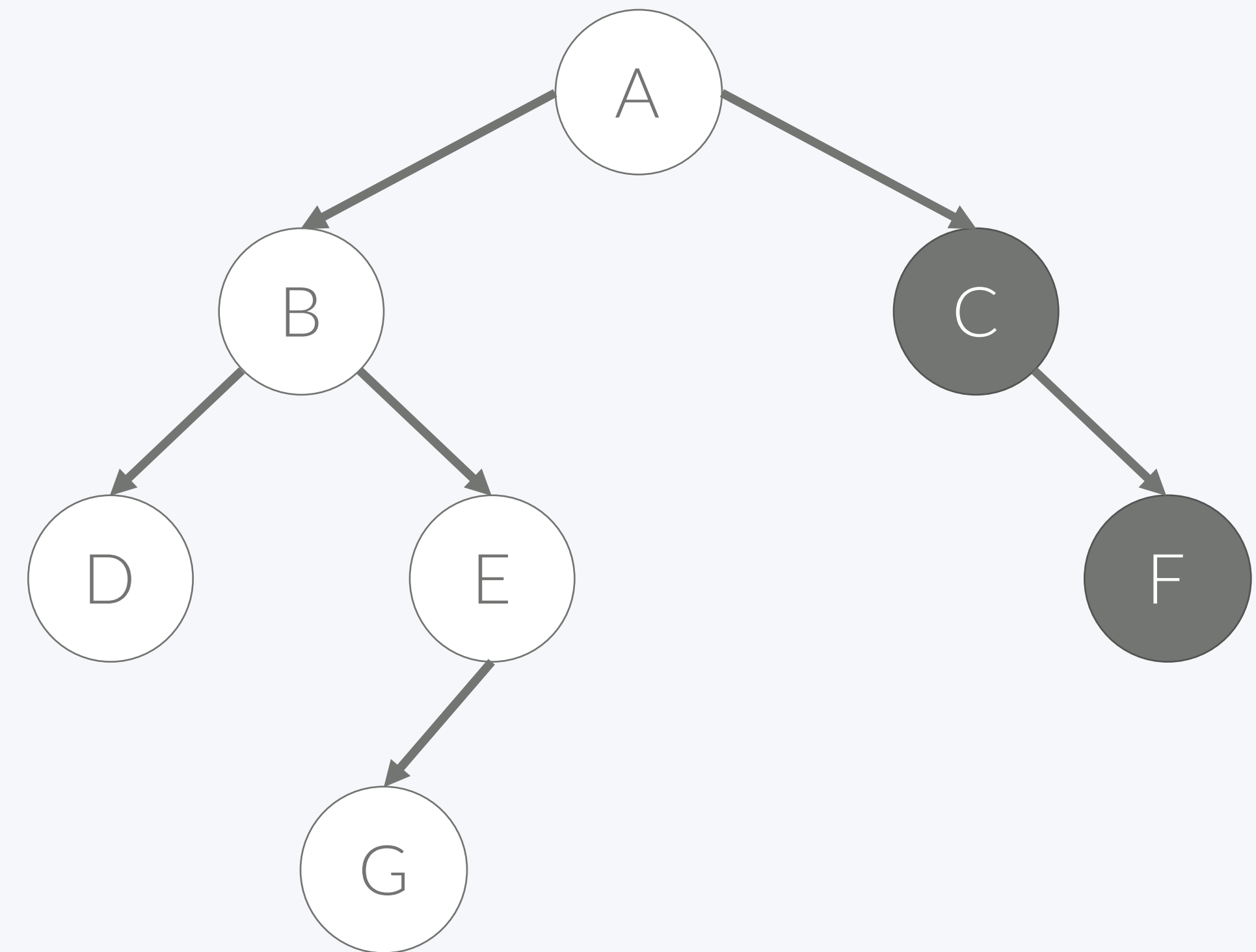
- 왼쪽 자식 인오더
 - DBGE
- **노드 방문**
 - A
- 오른쪽 자식 인오더



인오더

In-order

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

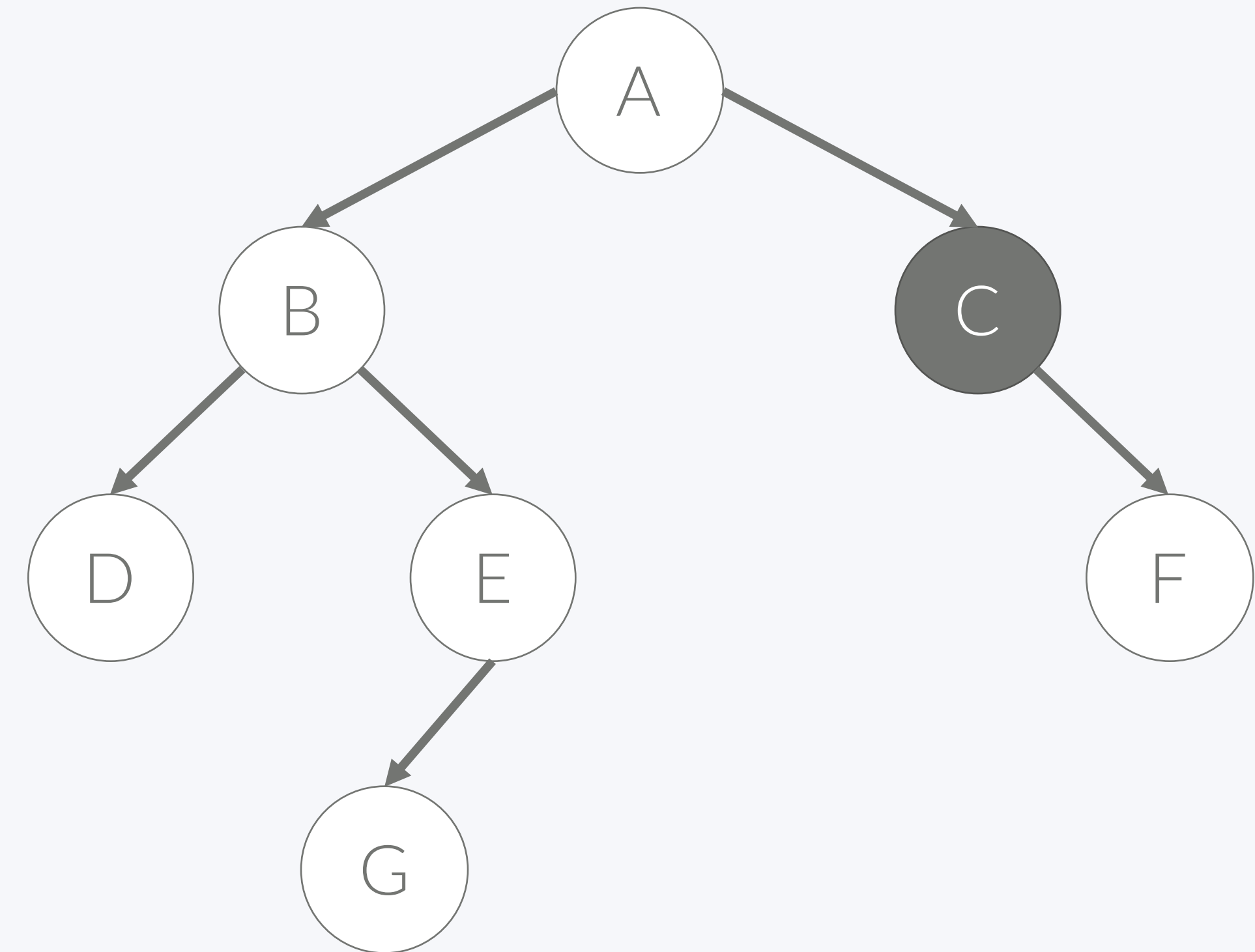


인오더

90

In-order

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

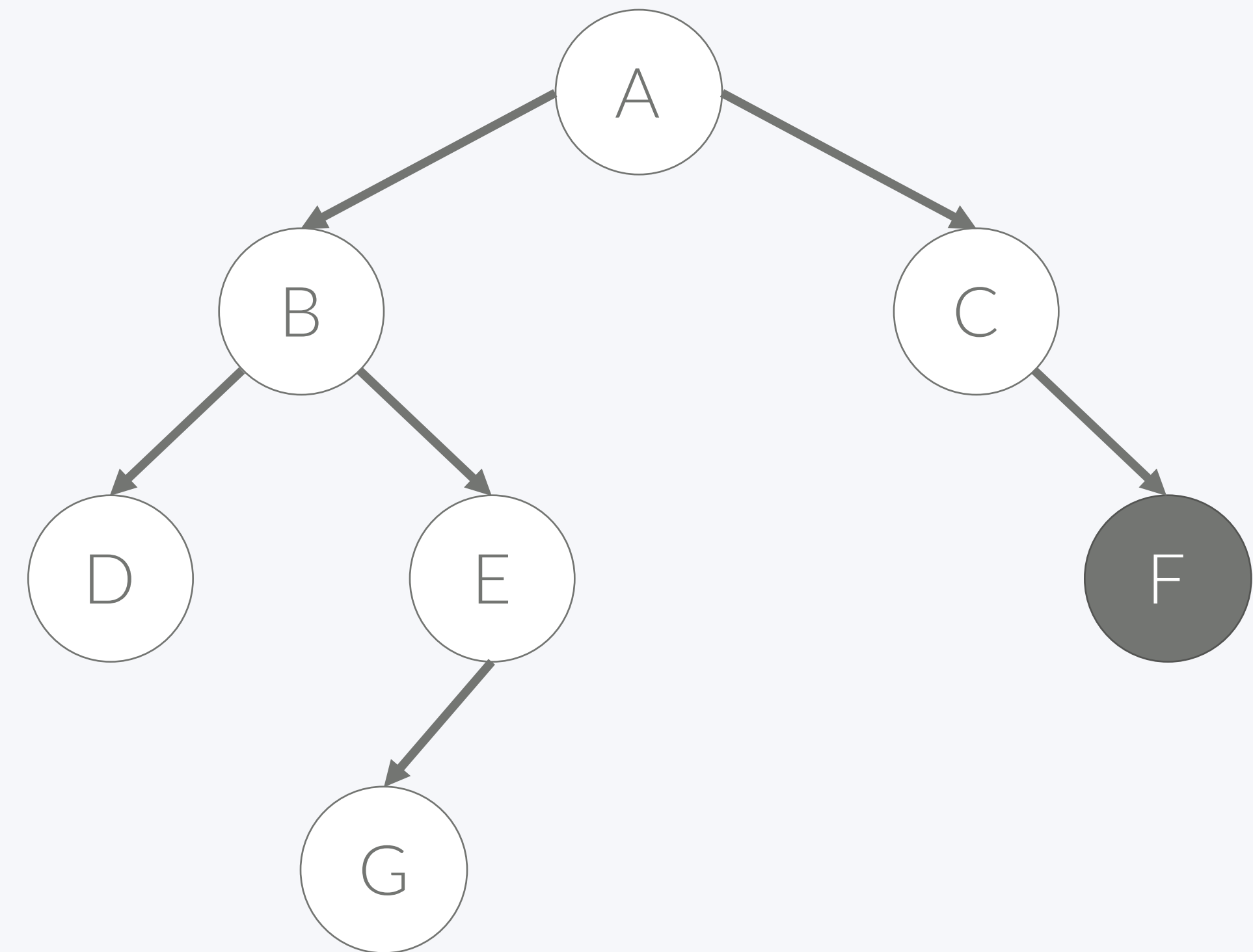


인오더

91

In-order

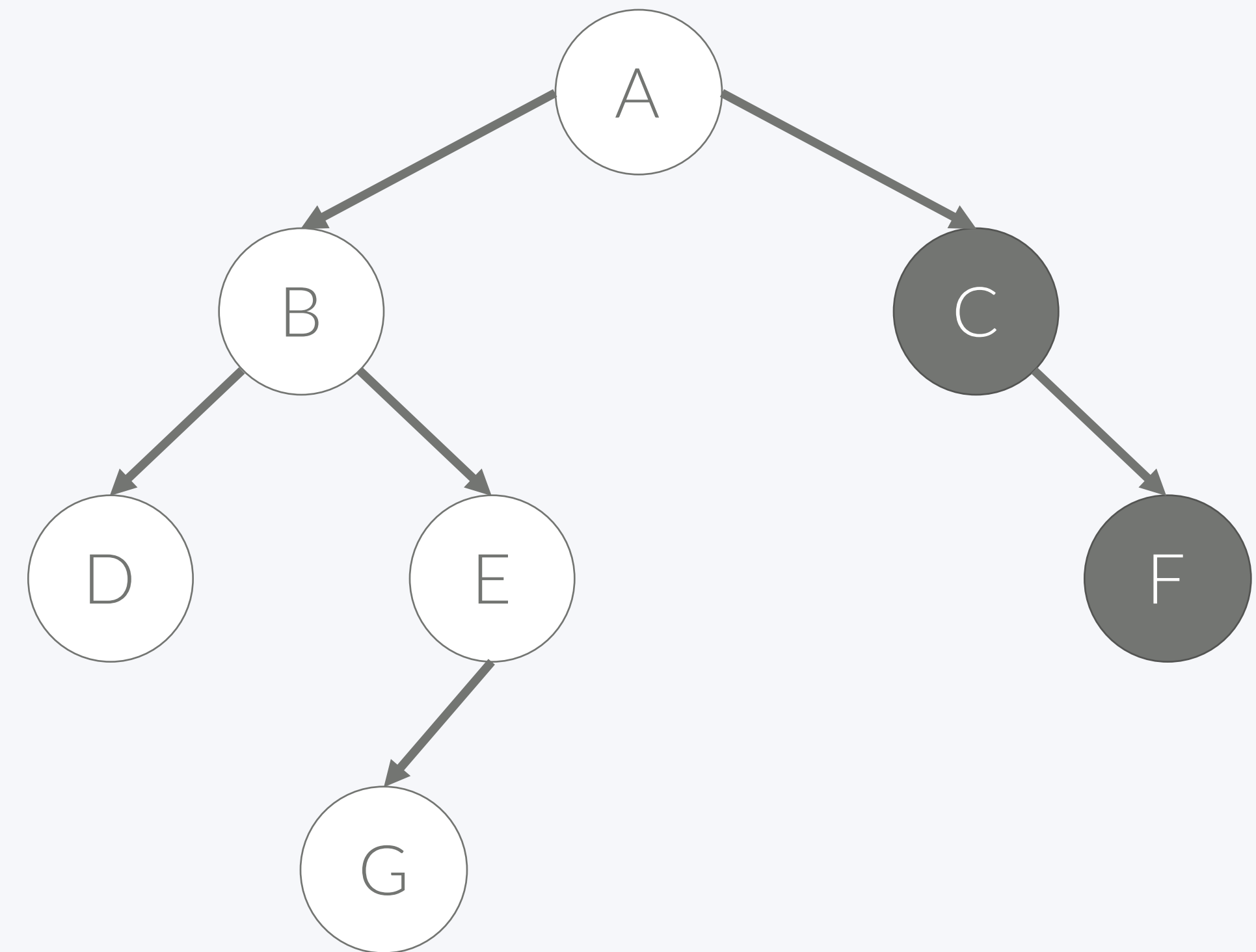
- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - 왼쪽 자식 인오더
 - 노드 방문
 - C
 - 오른쪽 자식 인오더
 - F



인오더

In-order

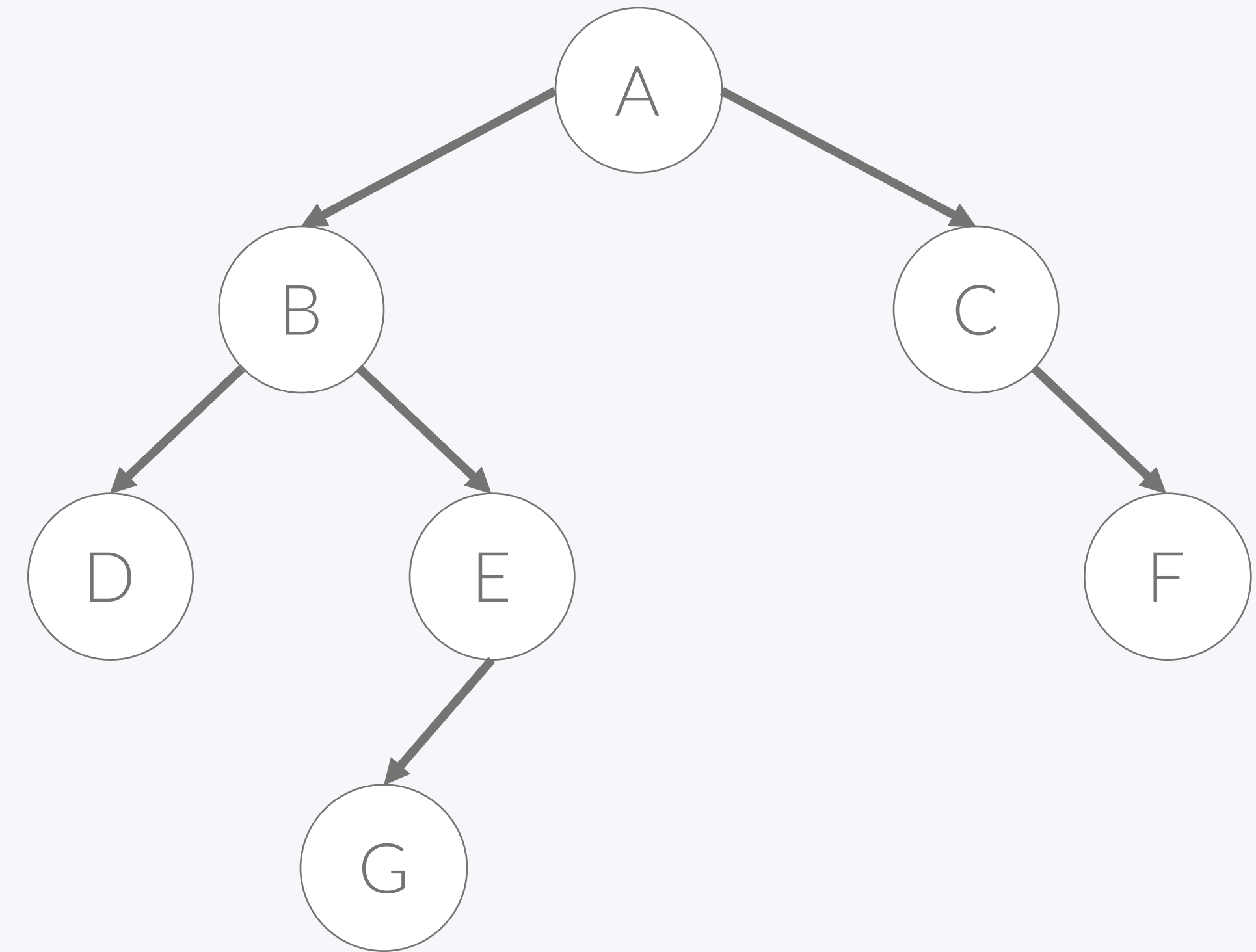
- 왼쪽 자식 인오더
 - DBGE
- 노드 방문
 - A
- 오른쪽 자식 인오더
 - CF



인오더

In-order

- DBGEACF

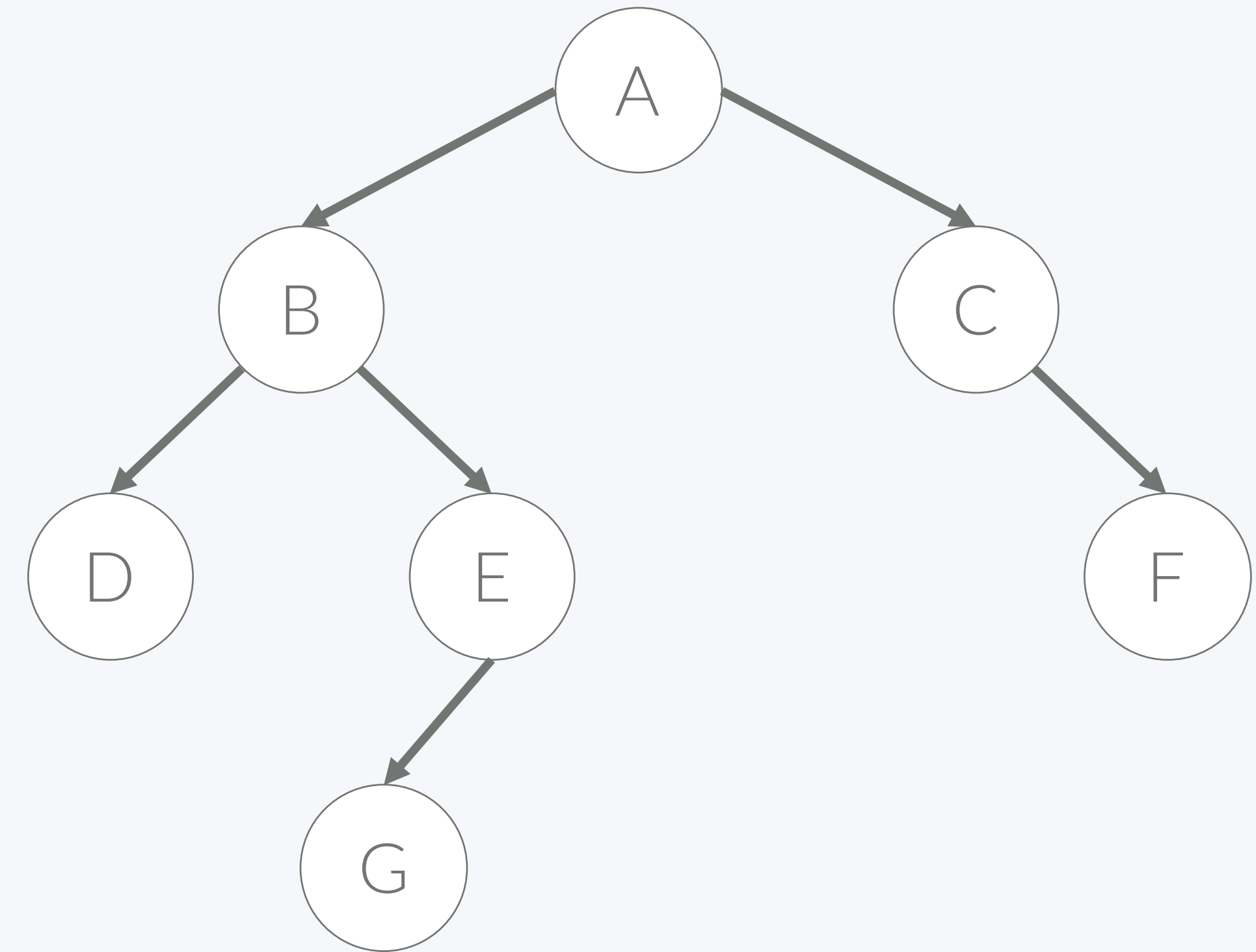


포스트오더

94

Post-order

- 왼쪽 자식 포스트오더
- 오른쪽 자식 포스트오더
- 노드 방문

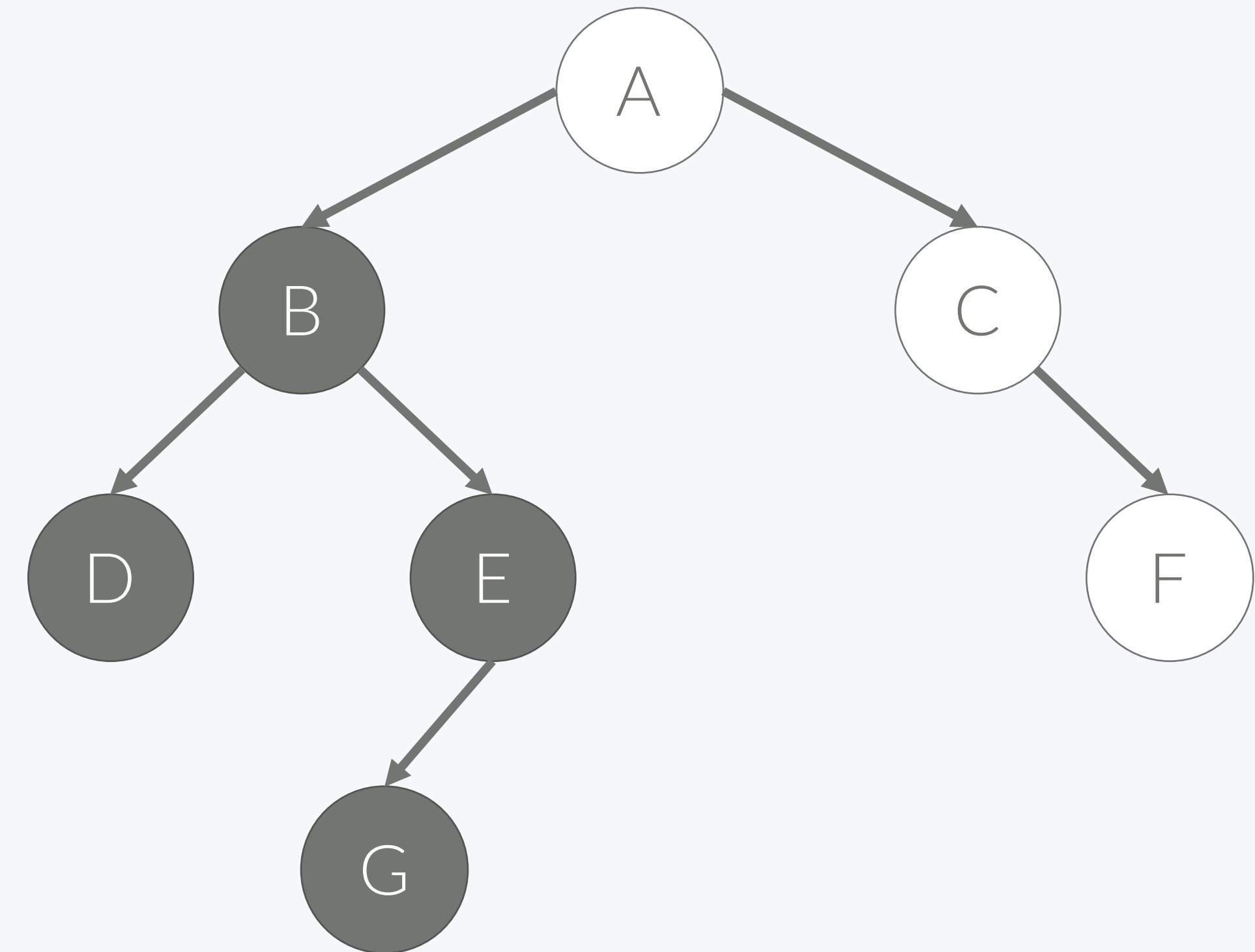


포스트오더

95

Post-order

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

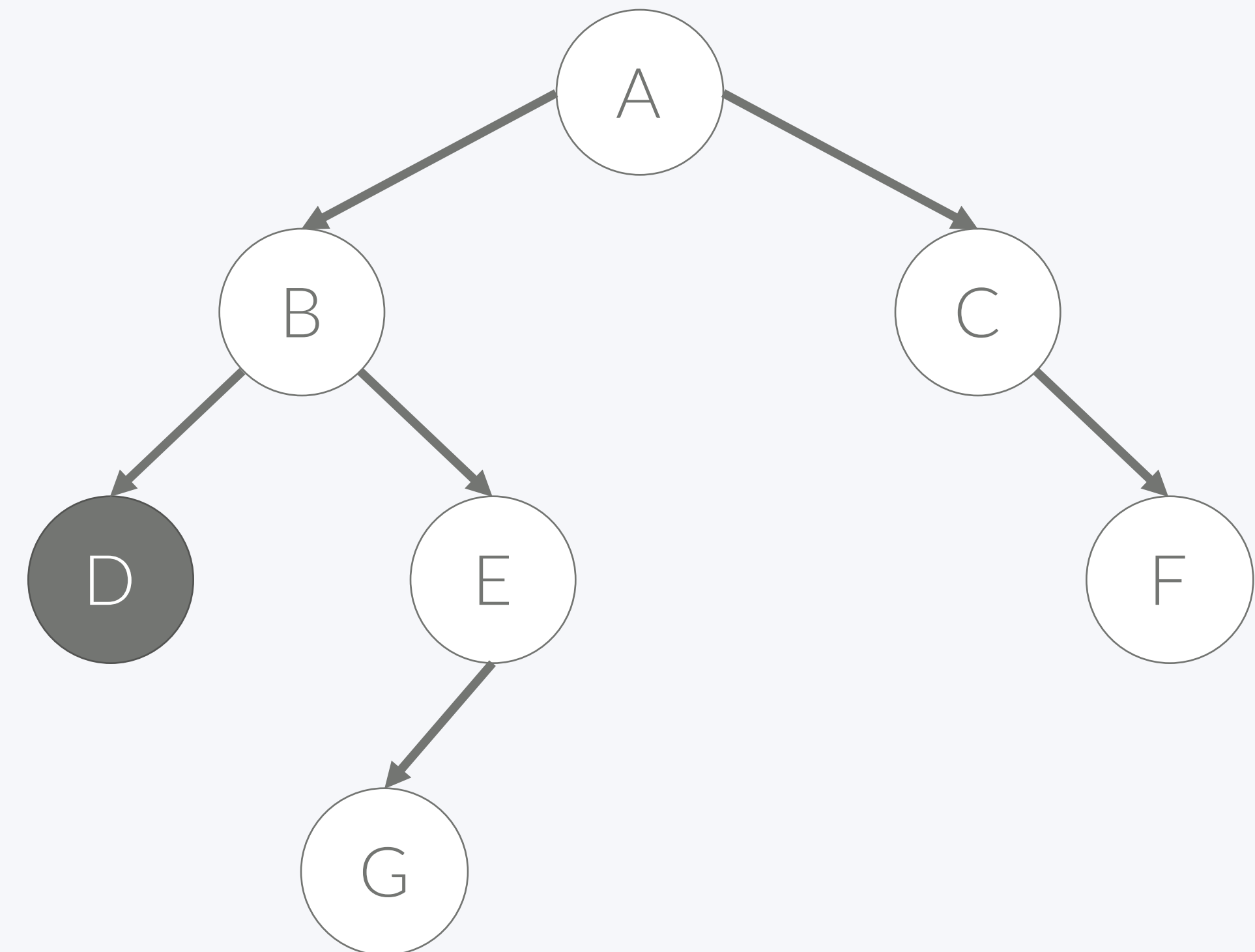


포스트오더

96

Post-order

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

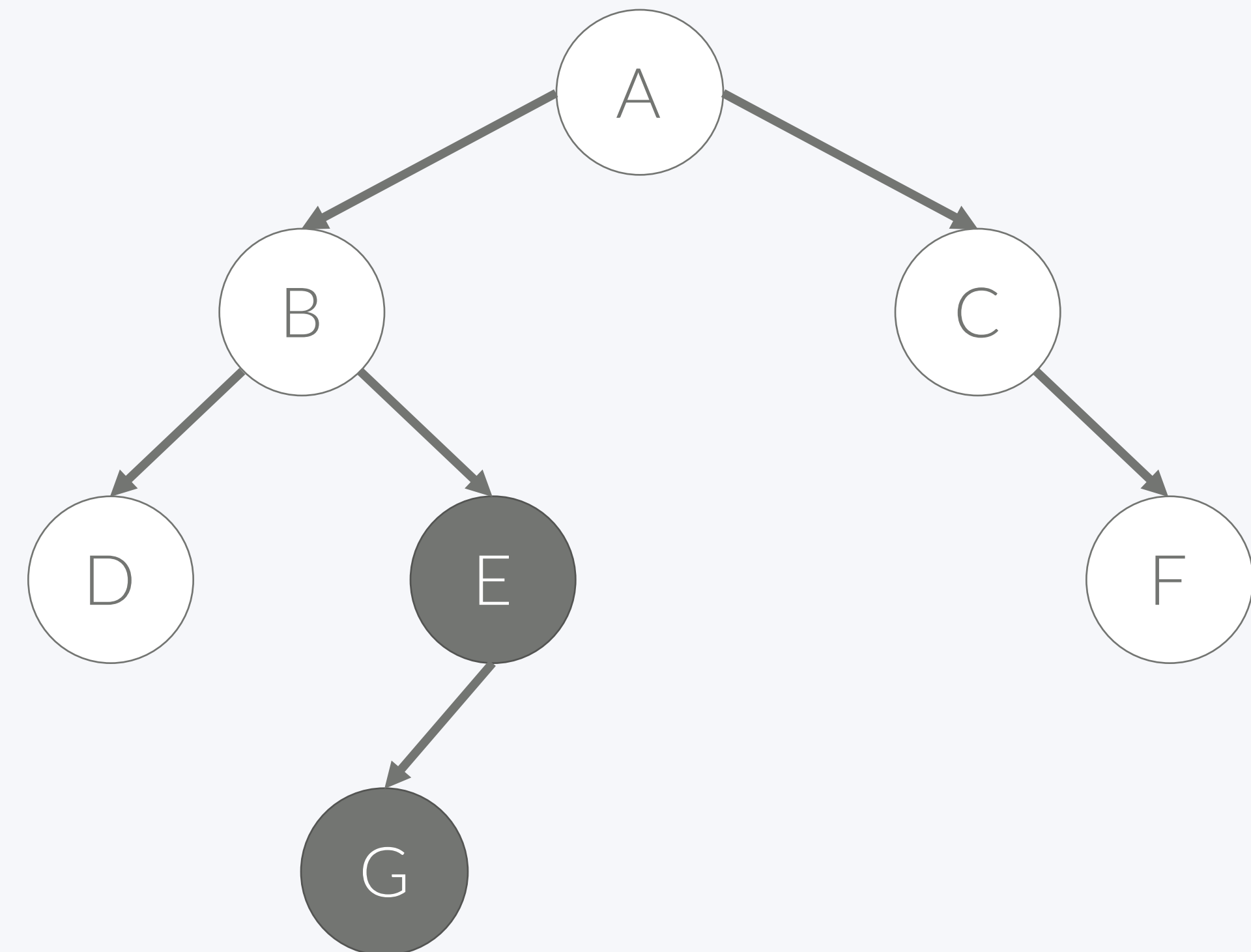


포스트오더

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

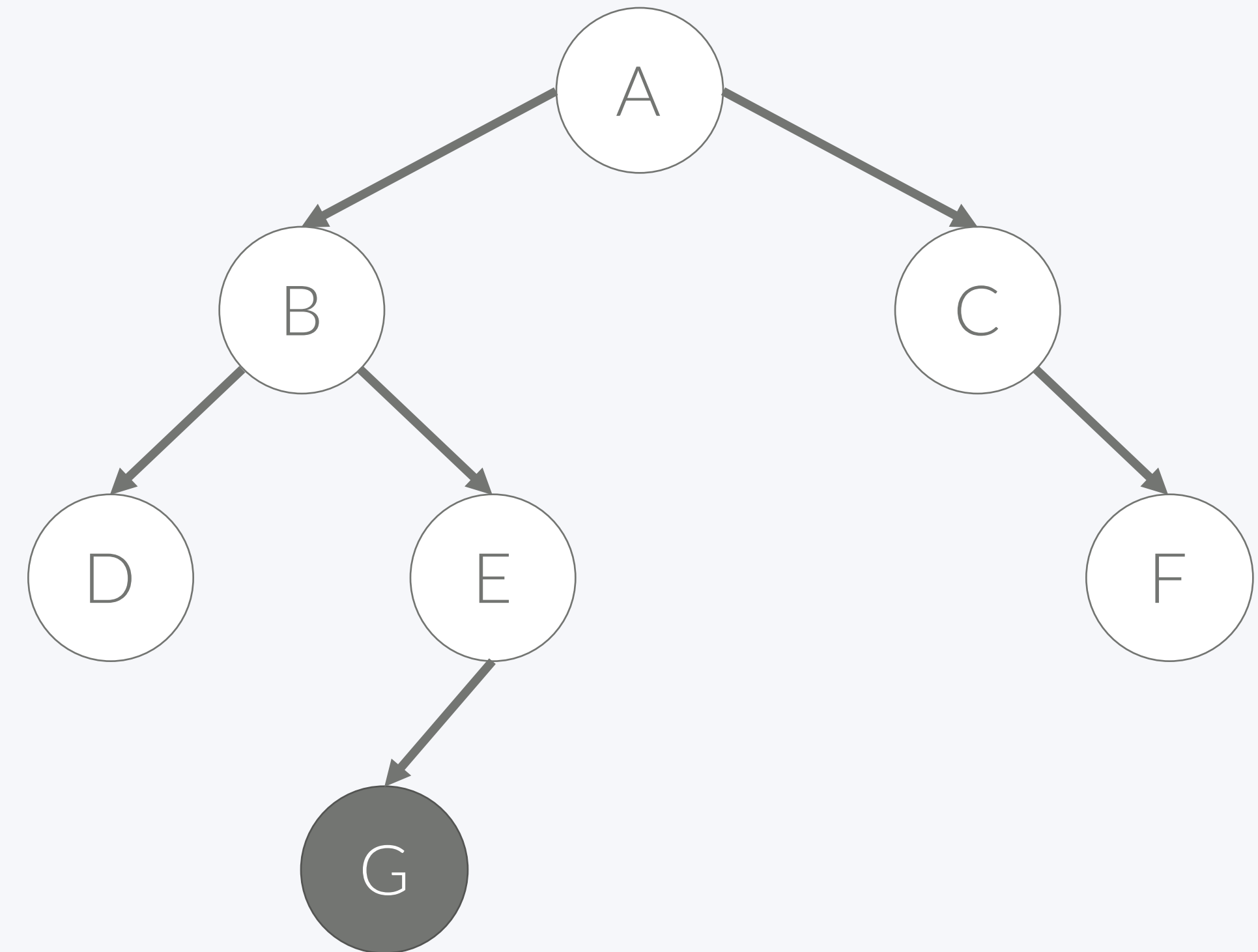


포스트오더

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

Post-order

- 왼쪽 자식 포스트오더

- 왼쪽 자식 포스트오더
 - D

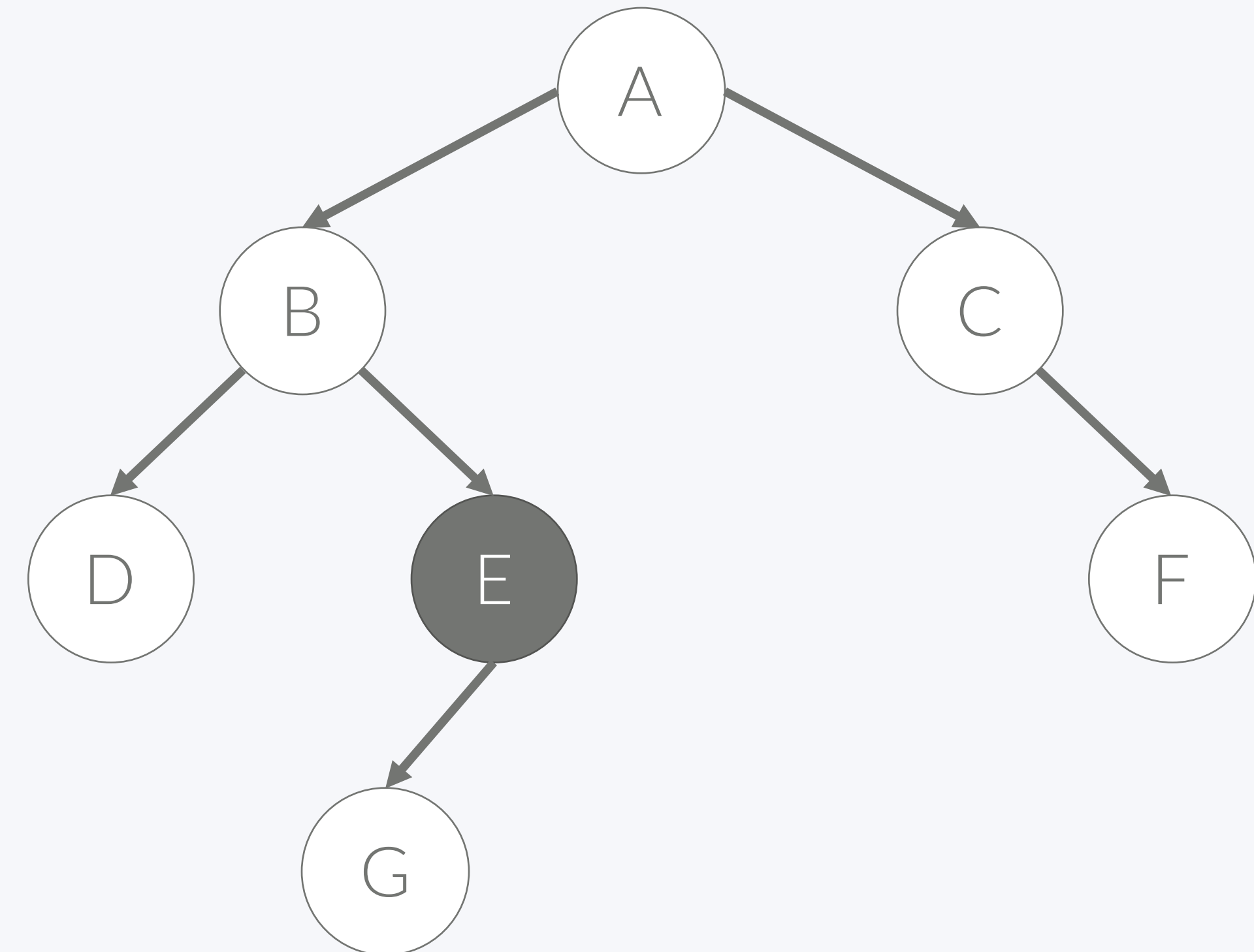
- 오른쪽 자식 포스트오더

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

- 노드 방문

- 오른쪽 자식 포스트오더

- 노드 방문

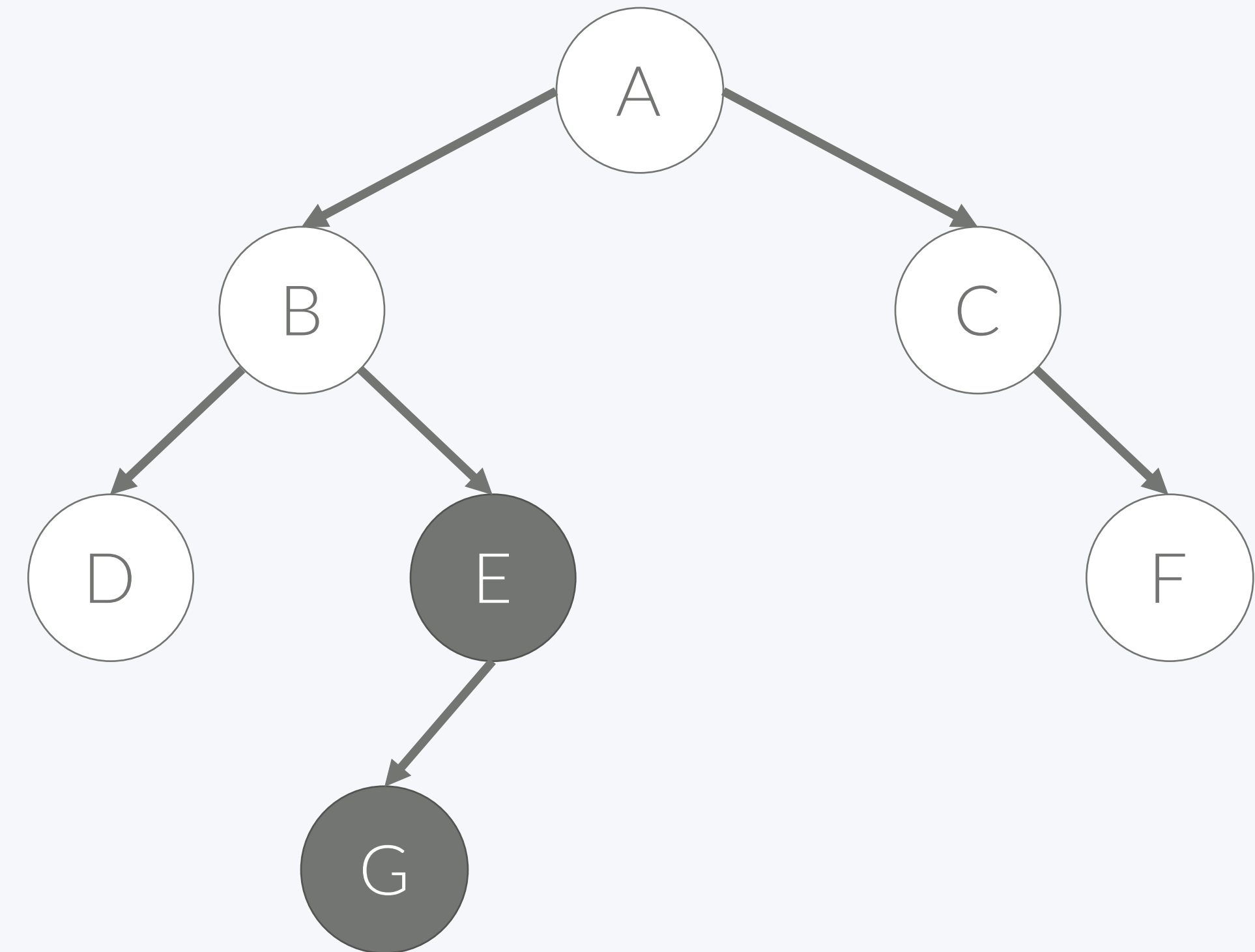


포스트오더

100

Post-order

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

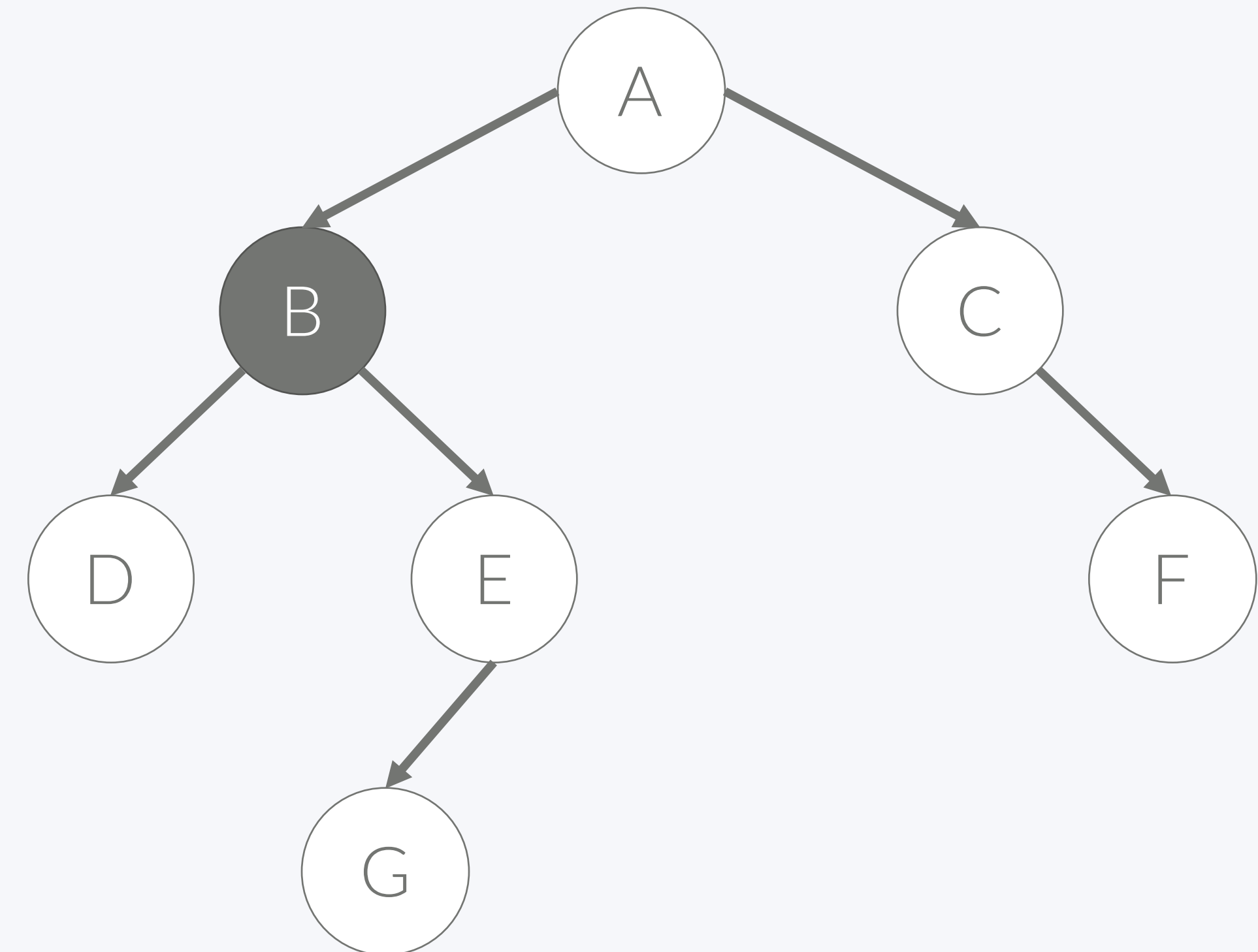


포스트오더

101

Post-order

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

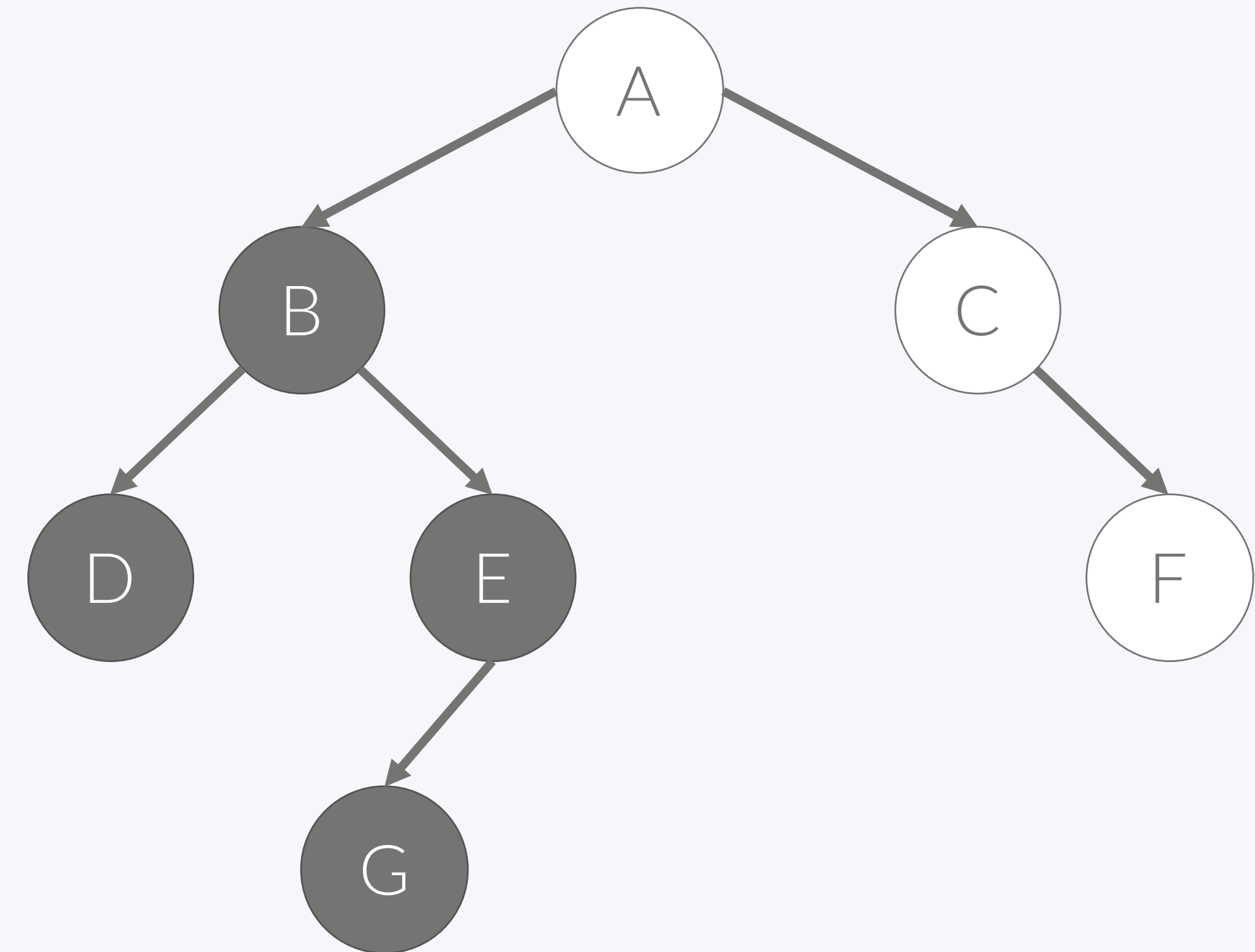


포스트오더

102

Post-order

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

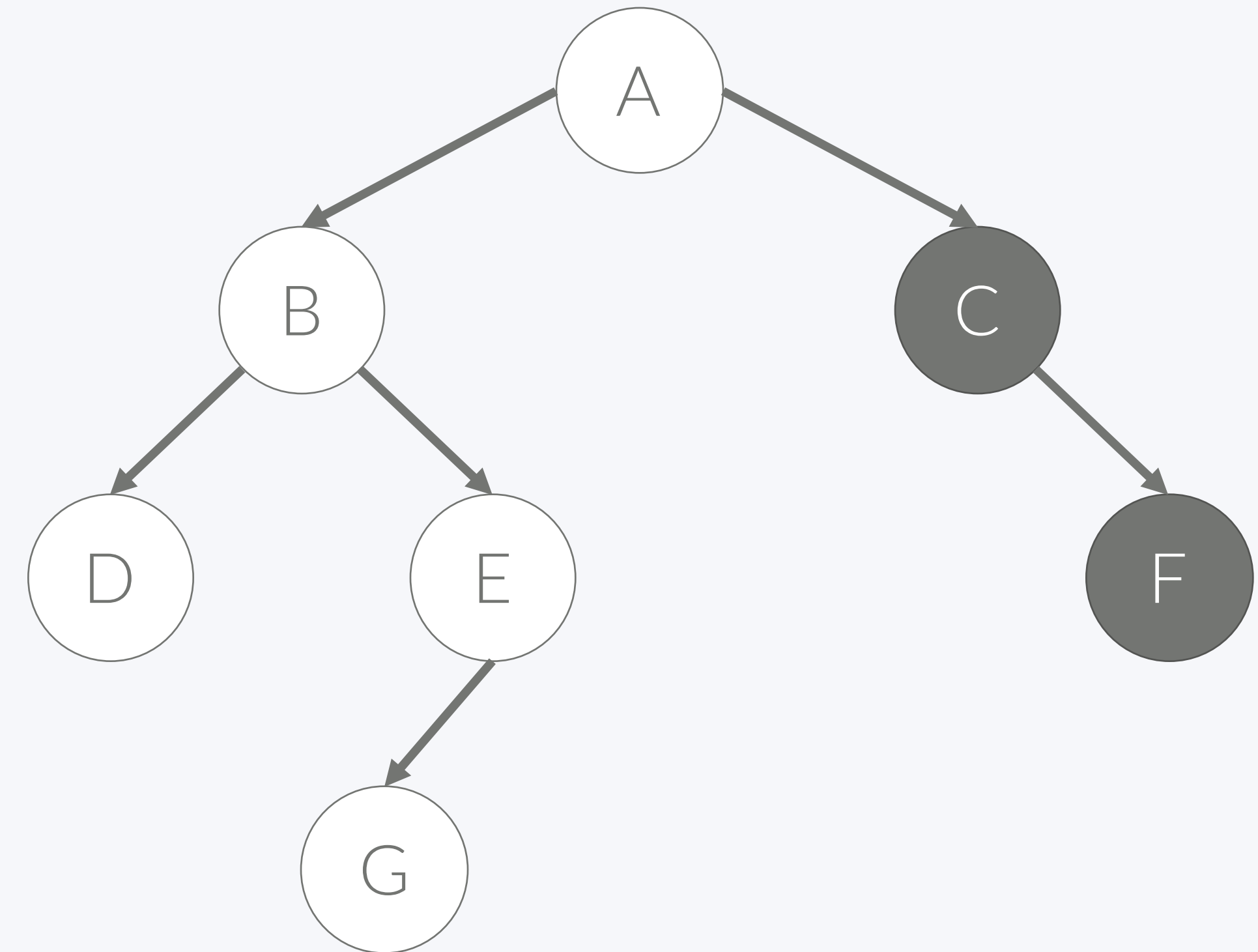


포스트오더

103

Post-order

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

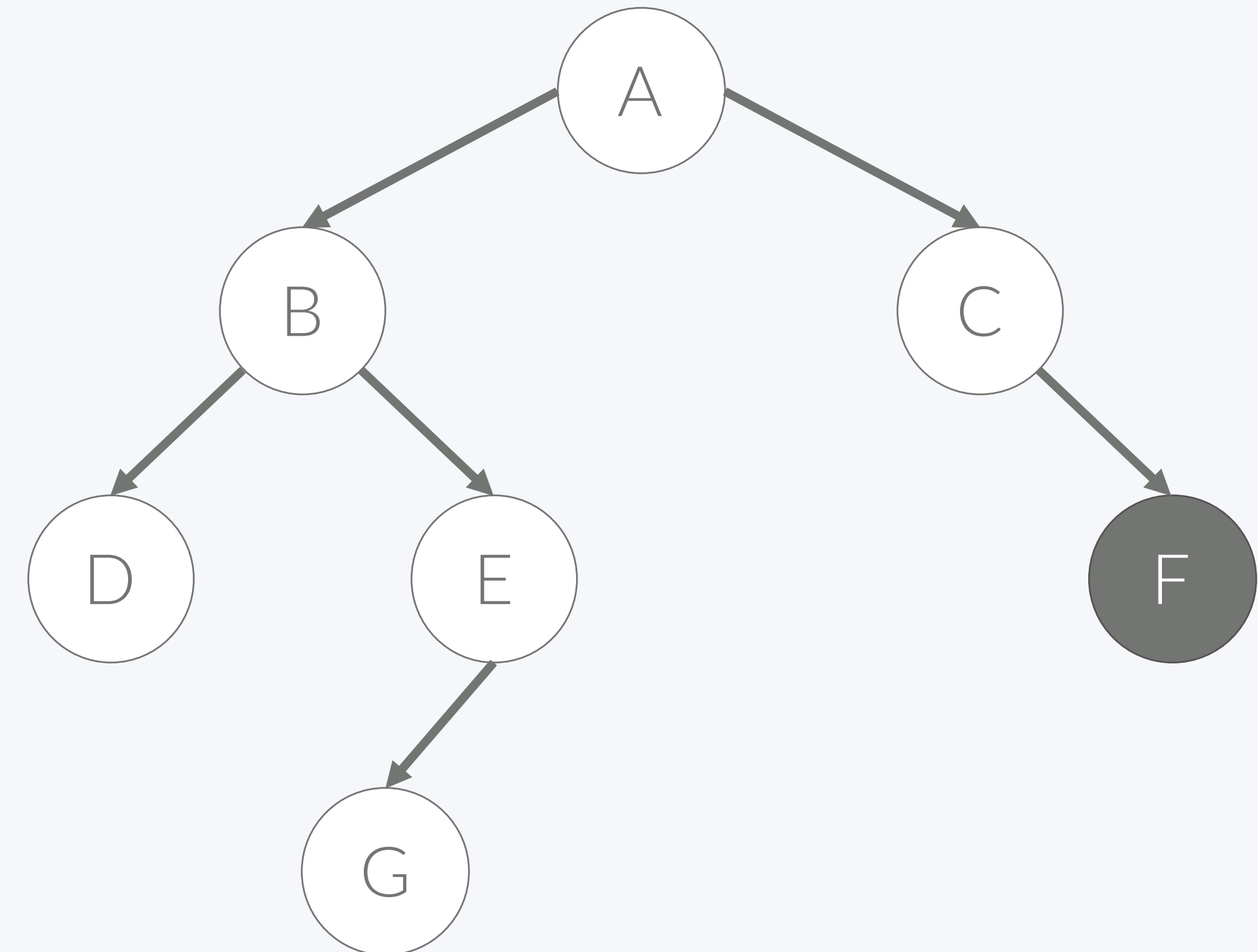


포스트오더

104

Post-order

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

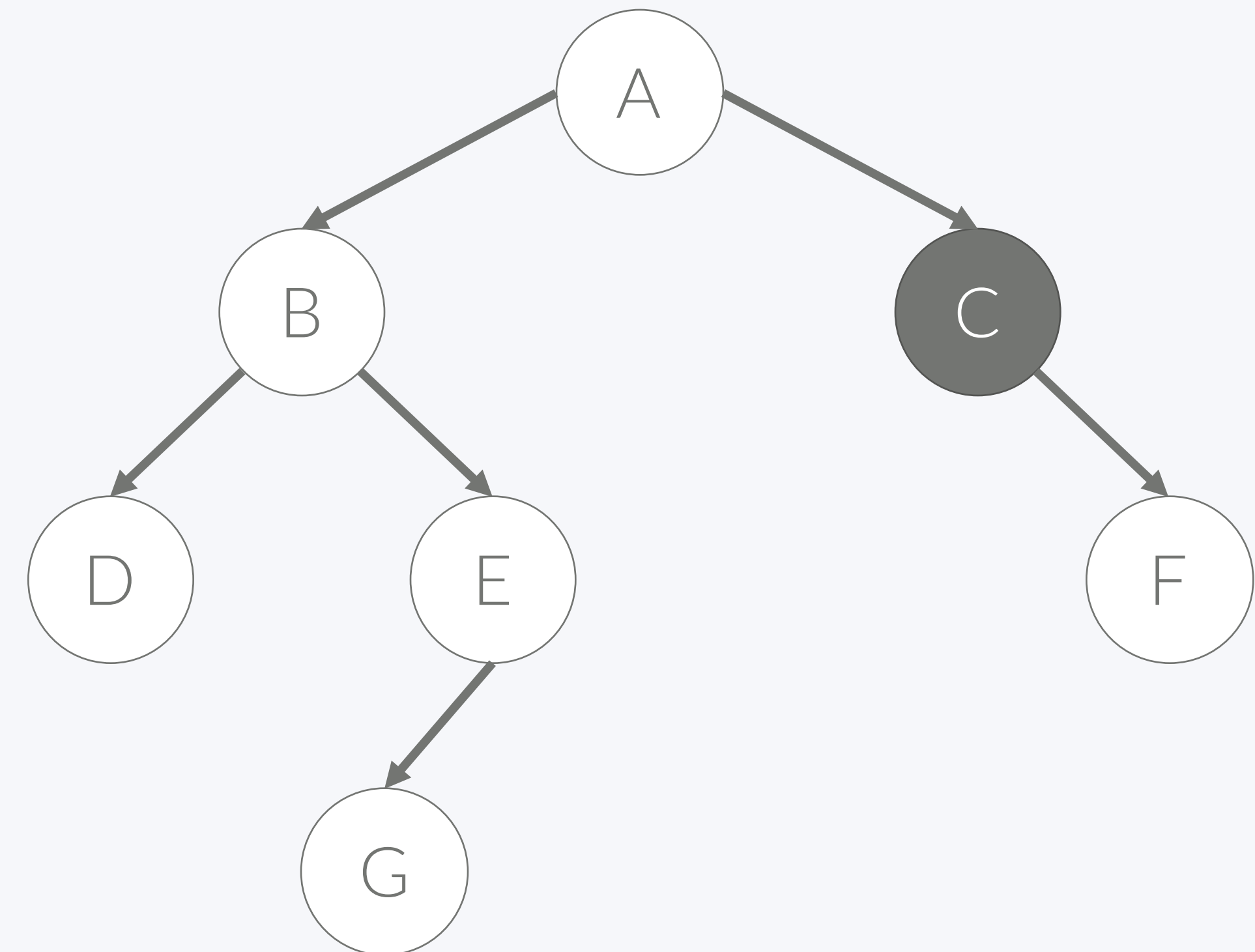


포스트오더

105

Post-order

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

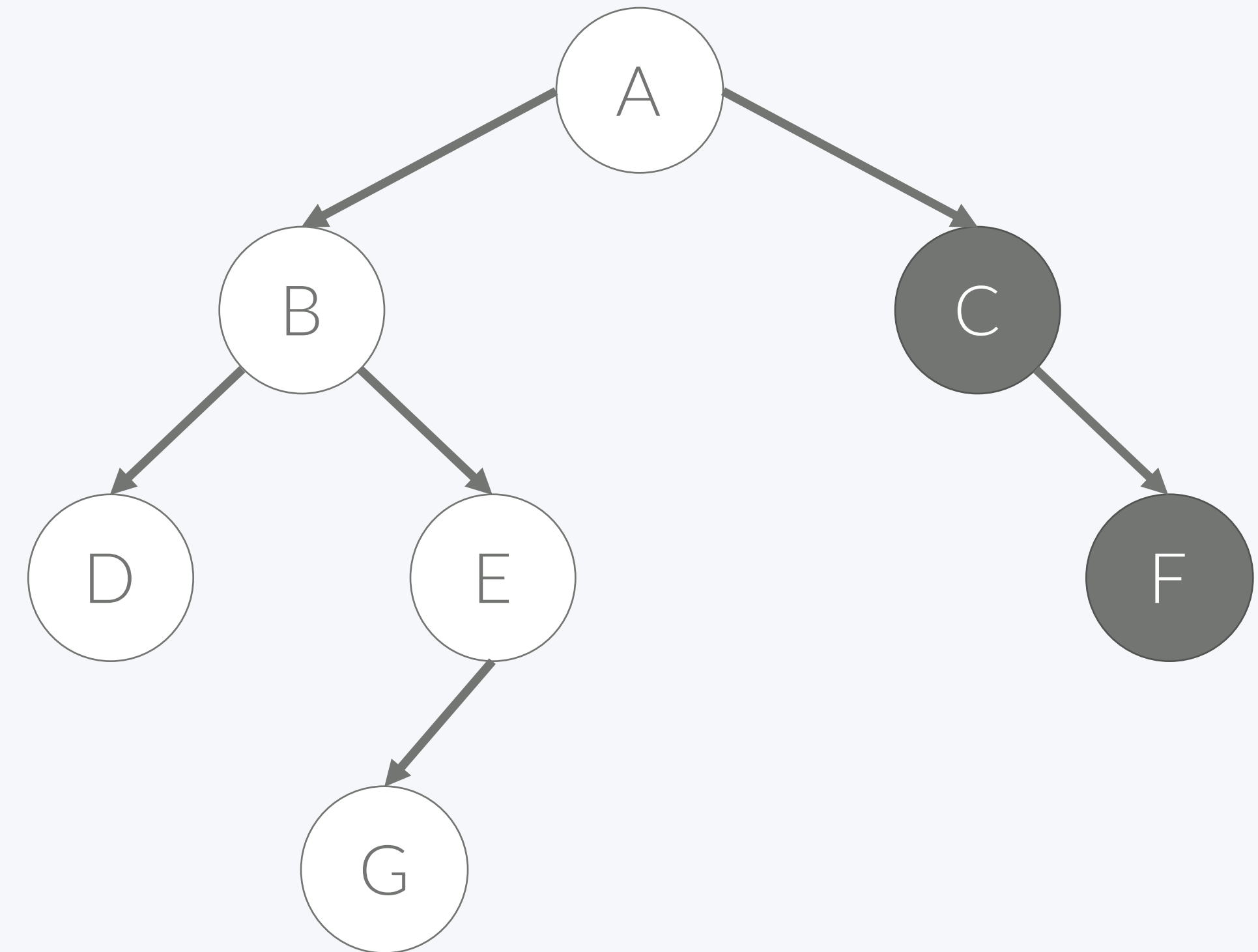


포스트오더

106

Post-order

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

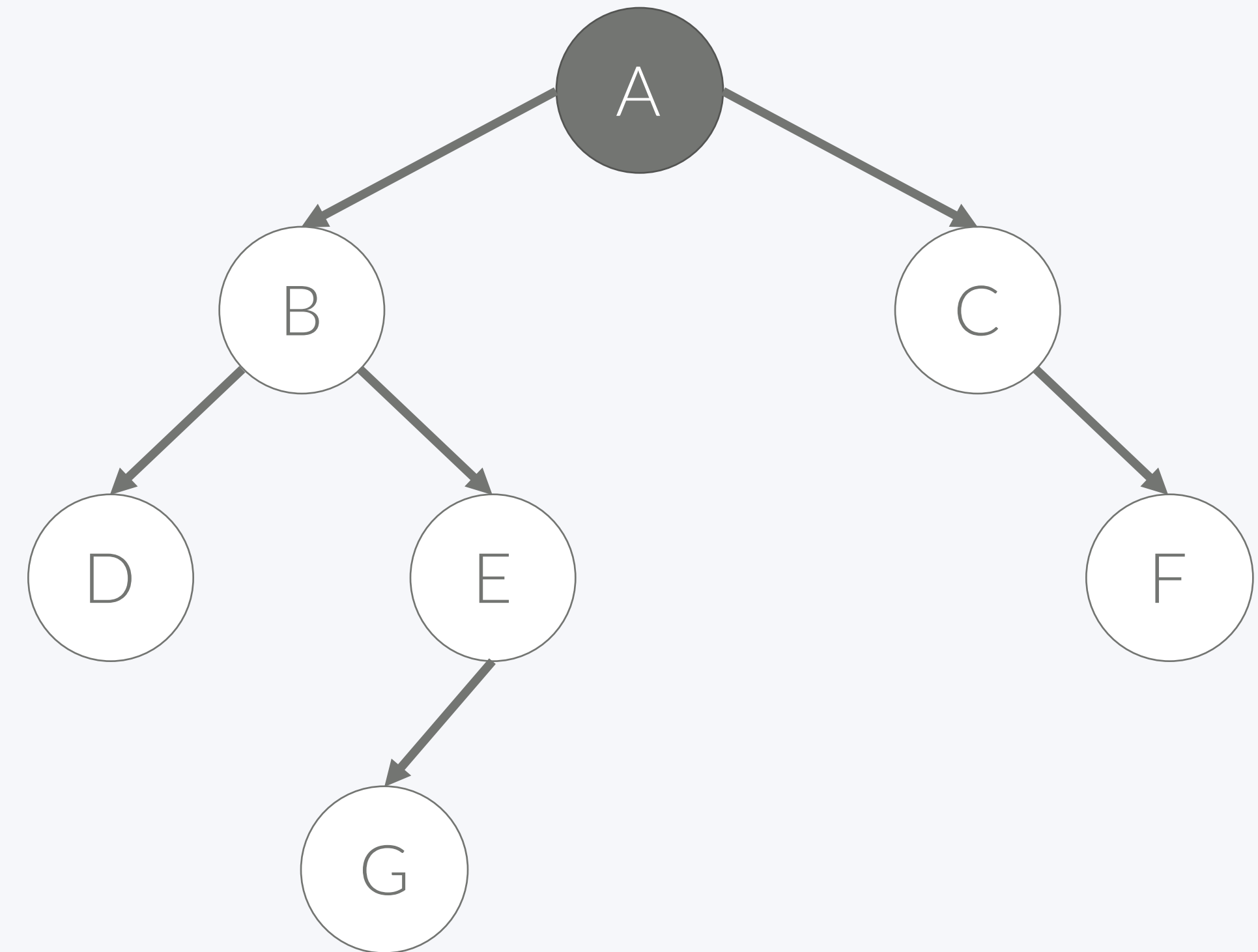


포스트오더

107

Post-order

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

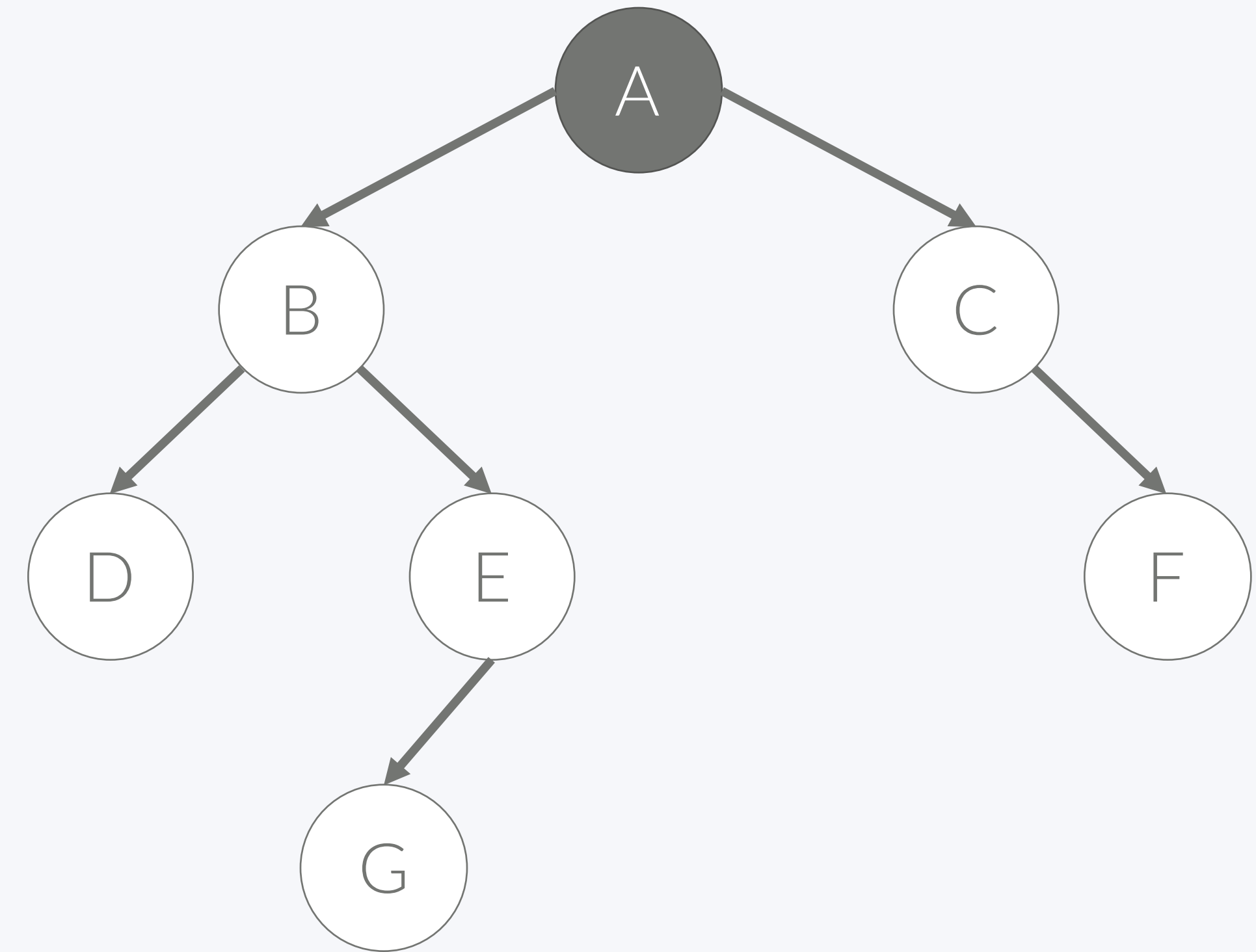


포스트오더

Post-order

- DGEBFCA

108



트리 순회

109

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

- 이진 트리의 프리오더, 인오더, 포스트오더 순서를 출력하는 문제

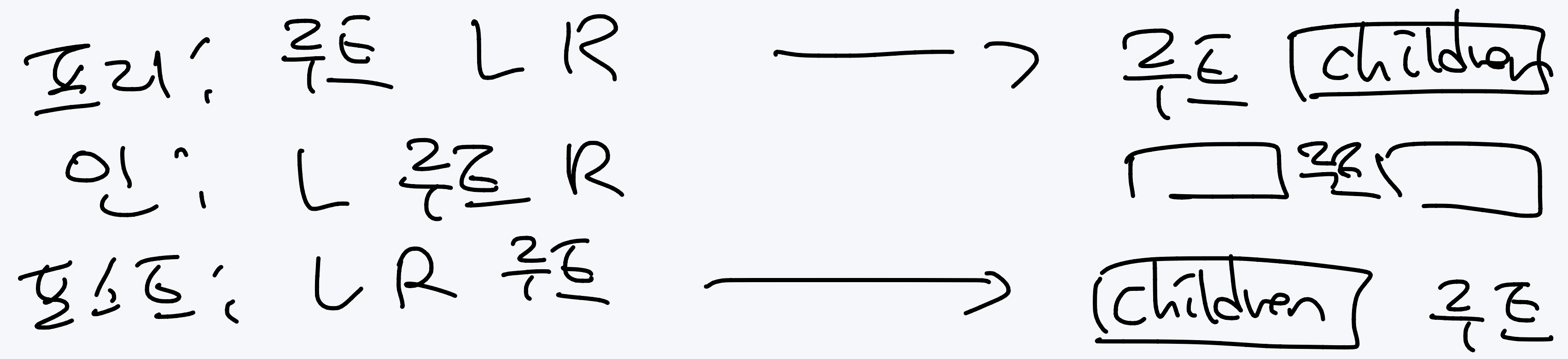
트리 순회

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

- C++: <https://gist.github.com/Baekjoon/bcca75d6bde538e0be3a>
- Java: <https://gist.github.com/Baekjoon/a0fbbde7274d2c125fcd>

$A[i][0]$: i 의 왼쪽
 $A[i][1]$: i 의 오른쪽
 $A[i][2]$: i 의 부모
(char) (i+65)

A ~ Z
65 ~ 65+26-1
↓
0
↓
25



트리의 탐색

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

DFS/BFS

최단거리

112

도시 N 개 도시 $N-1$ 개

연결 그래프

임의의 두 도시 사이에

경로가 항상

존재

두 도시 사이에
경로 1개

반항 X

트리의 부모 찾기

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

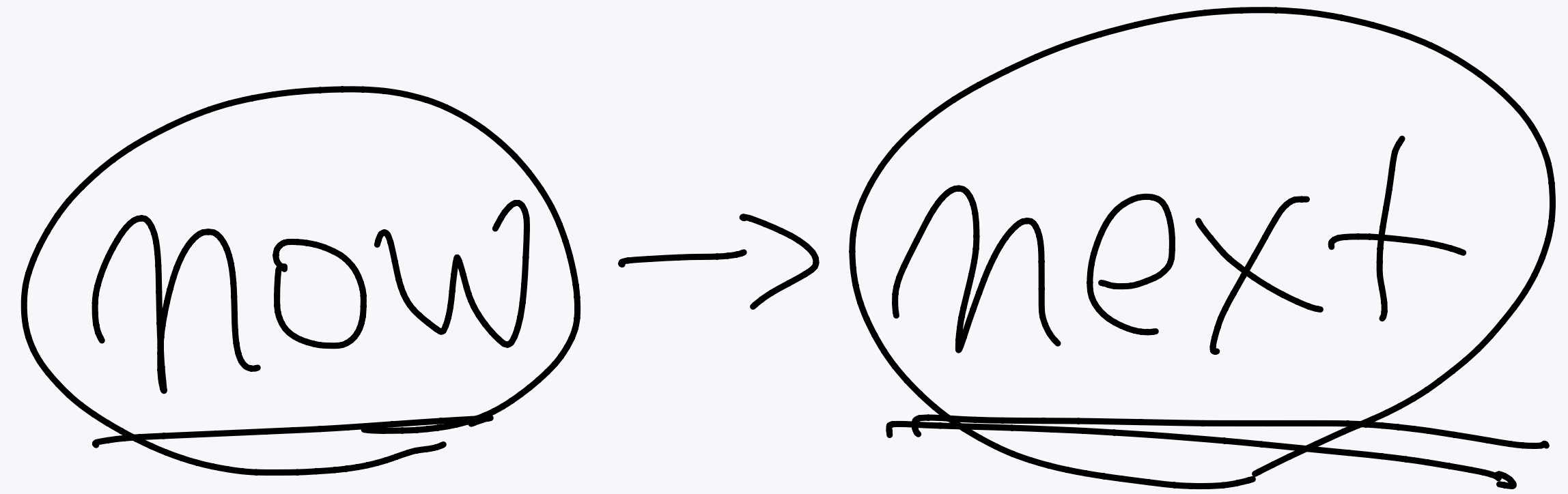
- 그래프로 트리를 입력받고

루트를 1이라고 정했을 때

- 각 노드의 부모를 찾는 문제

- BFS 탐색으로 해결할 수 있다.

DFS.



① ~~이~~ 간선 준거

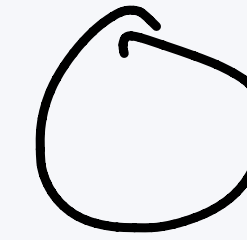
② $check[next] = false$
next를 아직 방문x

$parent[next] = now;$
 $depth[now] + 1$

트리의 부모 찾기

114

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



```
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);
        }
    }
}
```

BFS

$dist[y] = dist[x] + 1$
 $from[y] = x;$

트리의 부모 찾기

115

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

- C++: <https://gist.github.com/Baekjoon/f952d0f8fc8339ae0978>
- Java: <https://gist.github.com/Baekjoon/39384d6e37fd6af760bcb34019a8c082>

트리의 지름

트리의 지름

Diameter

- 트리에 존재하는 모든 경로 중에서 가장 긴 것의 길이를 트리의 지름이라고 한다
- 트리의 지름은 탐색 2번으로 구할 수 있다.
 1. 루트에서 모든 정점까지의 거리를 구한다. 이 때, 가장 먼 거리였던 정점을 A라고 한다.
 2. A를 루트라고 하고 모든 정점까지의 거리를 구한다. 이 때 구한 가장 먼 거리가 지름이다.

루트 \Rightarrow (거리)

A \Rightarrow (거리)

먼 거리

$\frac{2}{2}$

트리의 지름

118

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

- 트리의 지름을 구하는 문제

트리의 지름

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

- C++: <https://gist.github.com/Baekjoon/b59d721faa5029700adc>
- Java: <https://gist.github.com/Baekjoon/fd72ae15dfb03543b758>

트리의 지름

120

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

- 트리의 지름을 구하는 문제

트리의 지름

121

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

- C++: <https://gist.github.com/Baekjoon/f5c071f4a0637e002cdbb08555ae5c18>