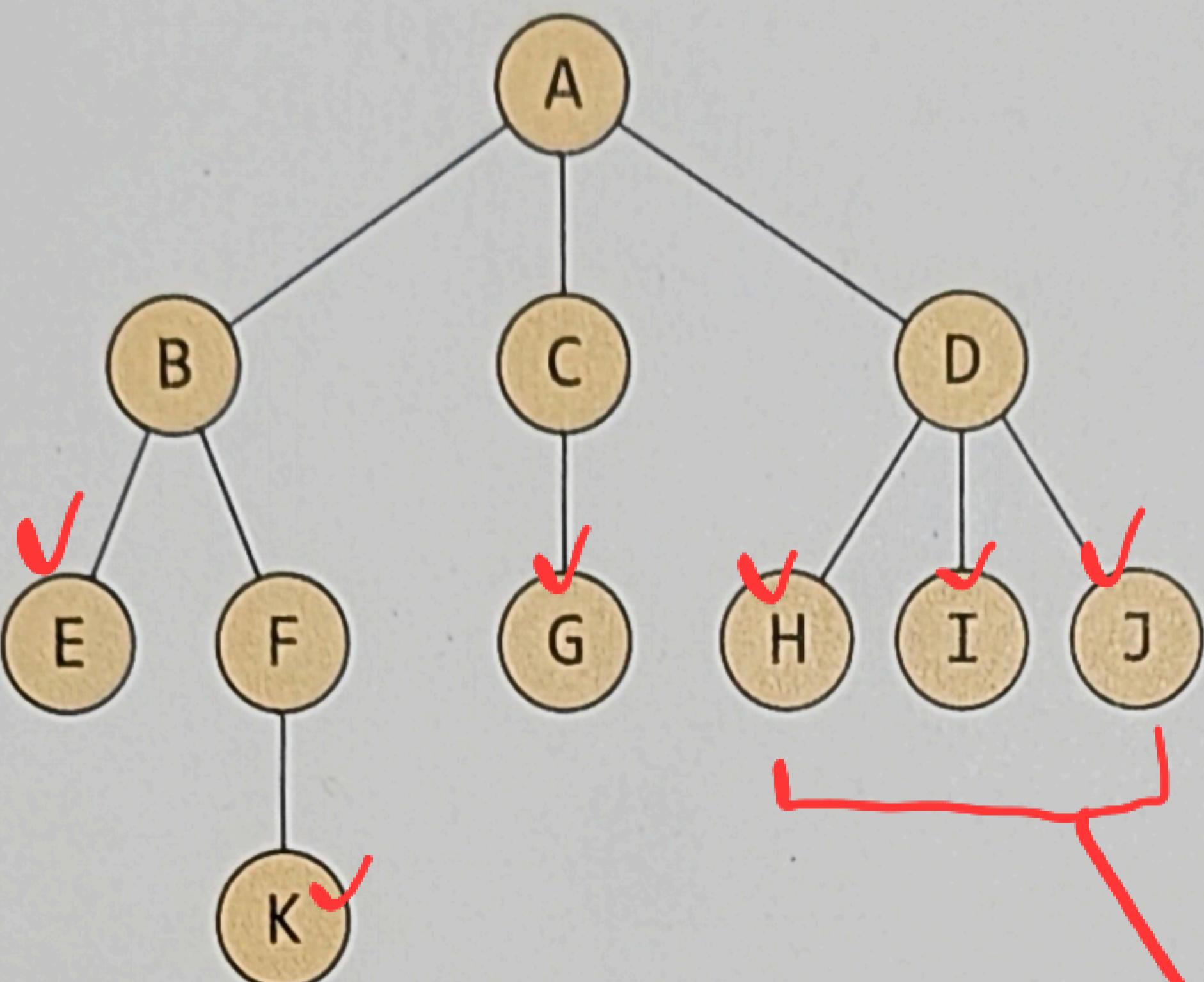


## 연습문제

### 8.1 다음 트리에 대한 각 물음에 답하라.

**정점의 차수:** 정점이 연결된 자식 노드의 수.

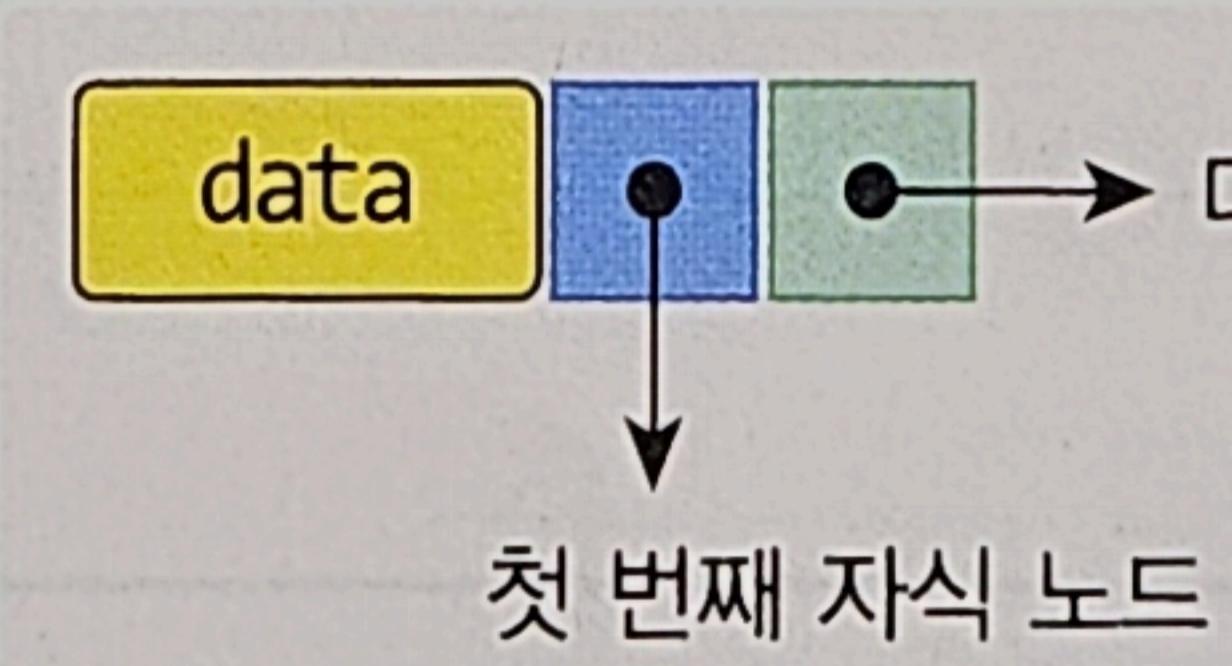
**트리의 차수:** 트리 내에서 가장 높은 정점의 차수.



자식이 없는 노드

- (1) 루트 노드: A
- (2) 단말 노드: E, K, G, H, I, J
- (3) D의 부모 노드: A
- (4) D의 형제 노드: B, C
- (5) B의 자식 노드: E, F
- (6) B의 자손 노드: E, F, K
- (7) K의 조상 노드: F, B, A
- (8) B의 레벨: 1
- (9) D의 차수: 3
- (10) 트리의 높이: 4
- (11) 트리의 차수: 3
- (12) 이 트리는 이진트리인가? X

### 8.2 다음과 같은 구조의 노드를 이용해 위 문제의 일반 트리를 표현해 보라. 화살표가 없는 링크는 None을 가리킨다고 가정한다.



$k = \text{TNode}('K', \text{None}, \text{None}, \text{None})$

$j = \text{TNode}('J', \text{None}, \text{None}, \text{None})$

$i = \text{TNode}('I', \text{None}, \text{None}, \text{None})$

$h = \text{TNode}('H', \text{None}, \text{None}, \text{None})$

$g = \text{TNode}('G', \text{None}, \text{None}, \text{None})$

$f = \text{TNode}('F', k, \text{None}, \text{None})$

$e = \text{TNode}('E', \text{None}, \text{None}, \text{None})$

$d = \text{TNode}('D', h, i, j)$

$c = \text{TNode}('C', g, \text{None}, \text{None})$

$b = \text{TNode}('B', e, f, \text{None})$

$\text{root} = \text{TNode}('A', b, c, d)$

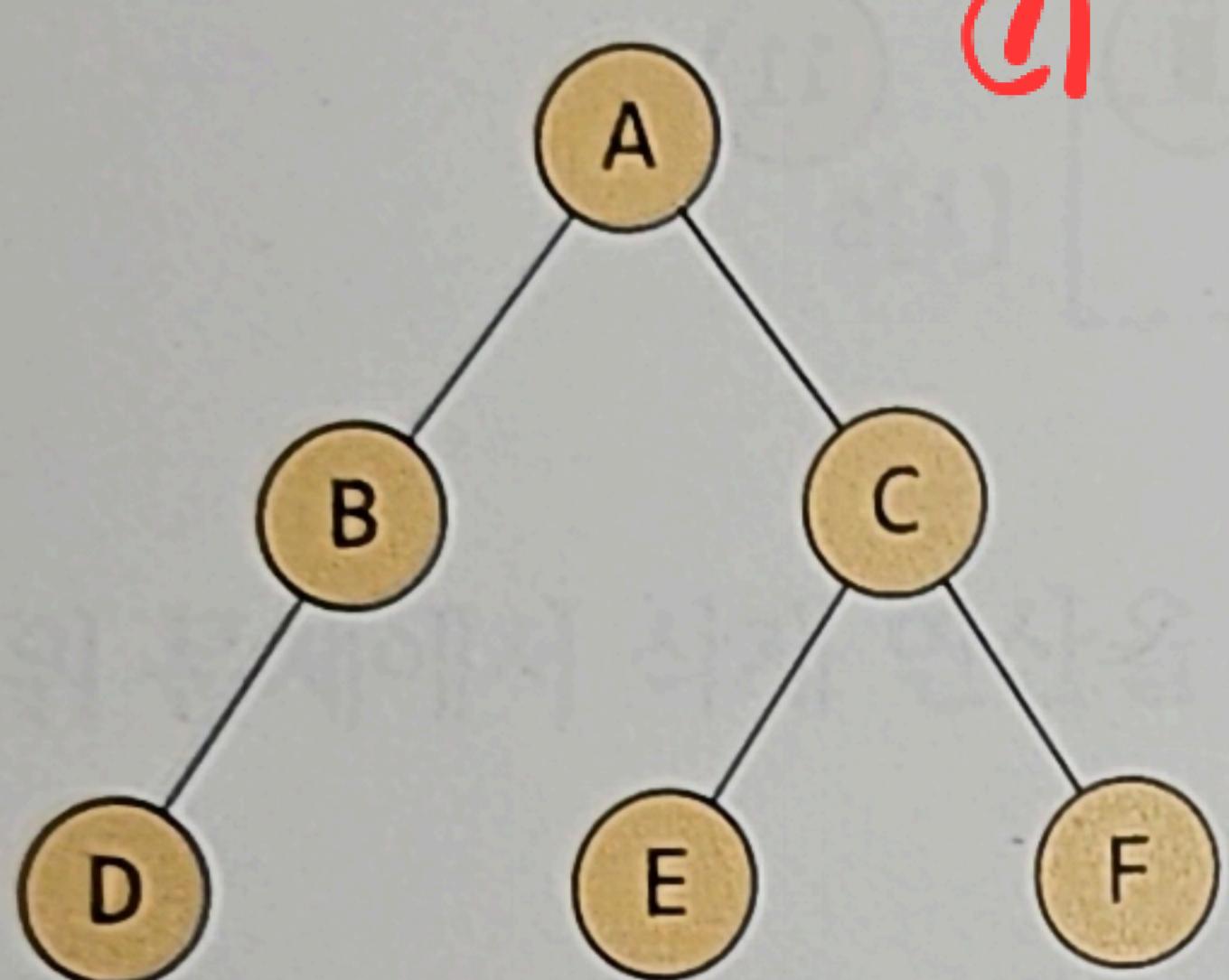
### 8.3 다음 중 트리에 대해 설명으로 옳지 않은 것은?

- ① 계층적인 구조를 갖고 있다. o
- ② 모든 노드는 루트 노드에서 출발한다. o
- ③ 선형적인 자료를 나타내기에 적합하다. ✓
- ④ 트리를 합하면 포리스트가 된다.

## 8.4 다음 중 이진트리에 대한 설명으로 적절하지 않은 것은?

- ① 모든 노드의 차수가 2 이하이다.
- ② 왼쪽 자식과 오른쪽 자식은 정확히 구분되어야 한다.
- ③ 일반적인 이진트리는 배열 구조로 표현할 수 없다.
- ④ 링크를 이용해 노드들을 연결할 수 있다.

## 8.5 다음 트리를 배열 표현법과 링크 표현법으로 각각 나타내라.



①: [A, B, C, D, None, E, F]

② A → B, C

B → D, None

C → E, F

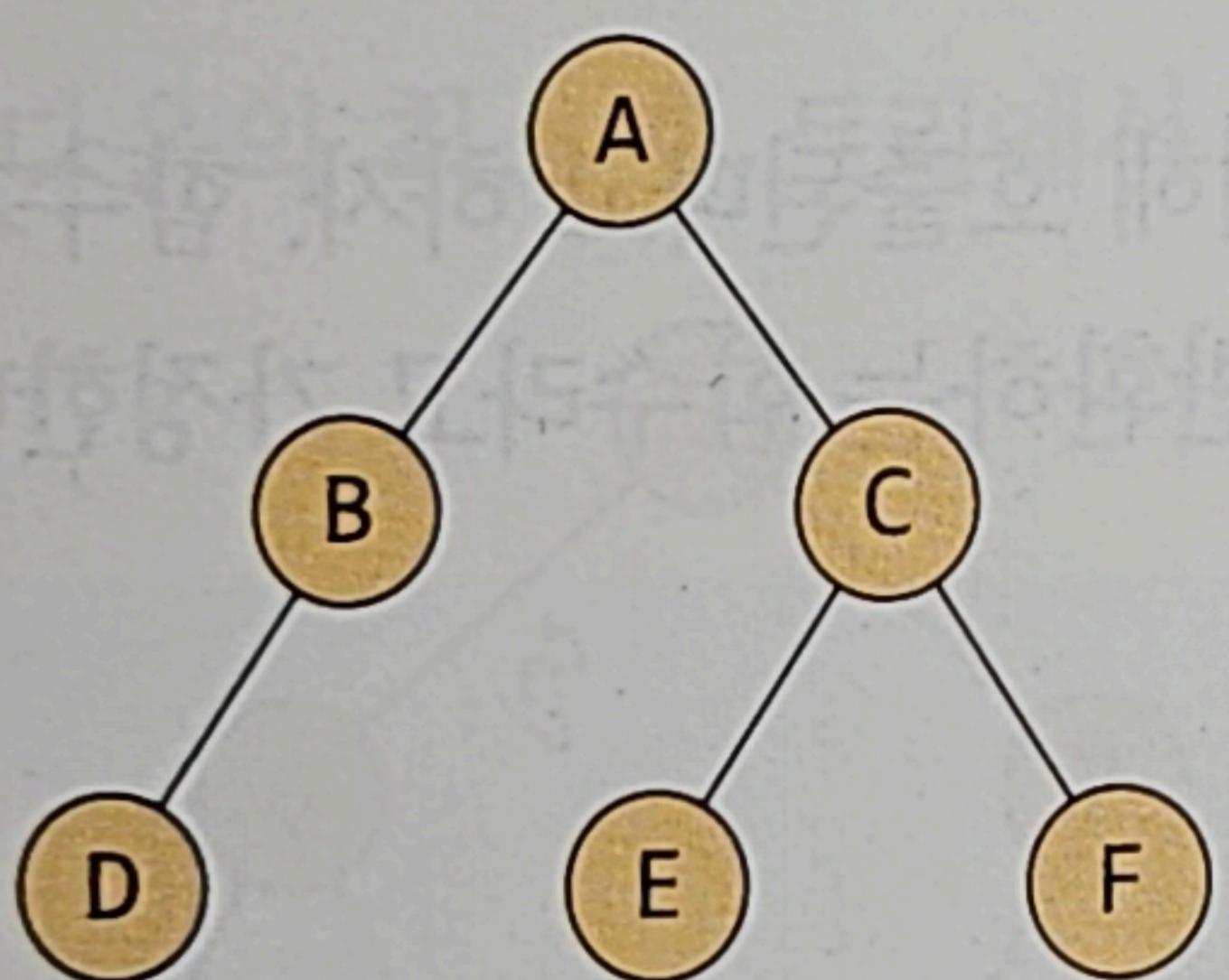
D → None, None

E → None, None

F → None, None

※ 전위, 중위, 후위

## 8.6 다음 트리에 대한 중위 순회 결과는?



- ① A B D C E F
- ② A B C D E F
- ③ D B E C F A
- ④ D B A E C F

## 8.7 문제 8.5의 트리를 전위 순회로 운행할 경우 다섯 번째로 탐색되는 노드은?

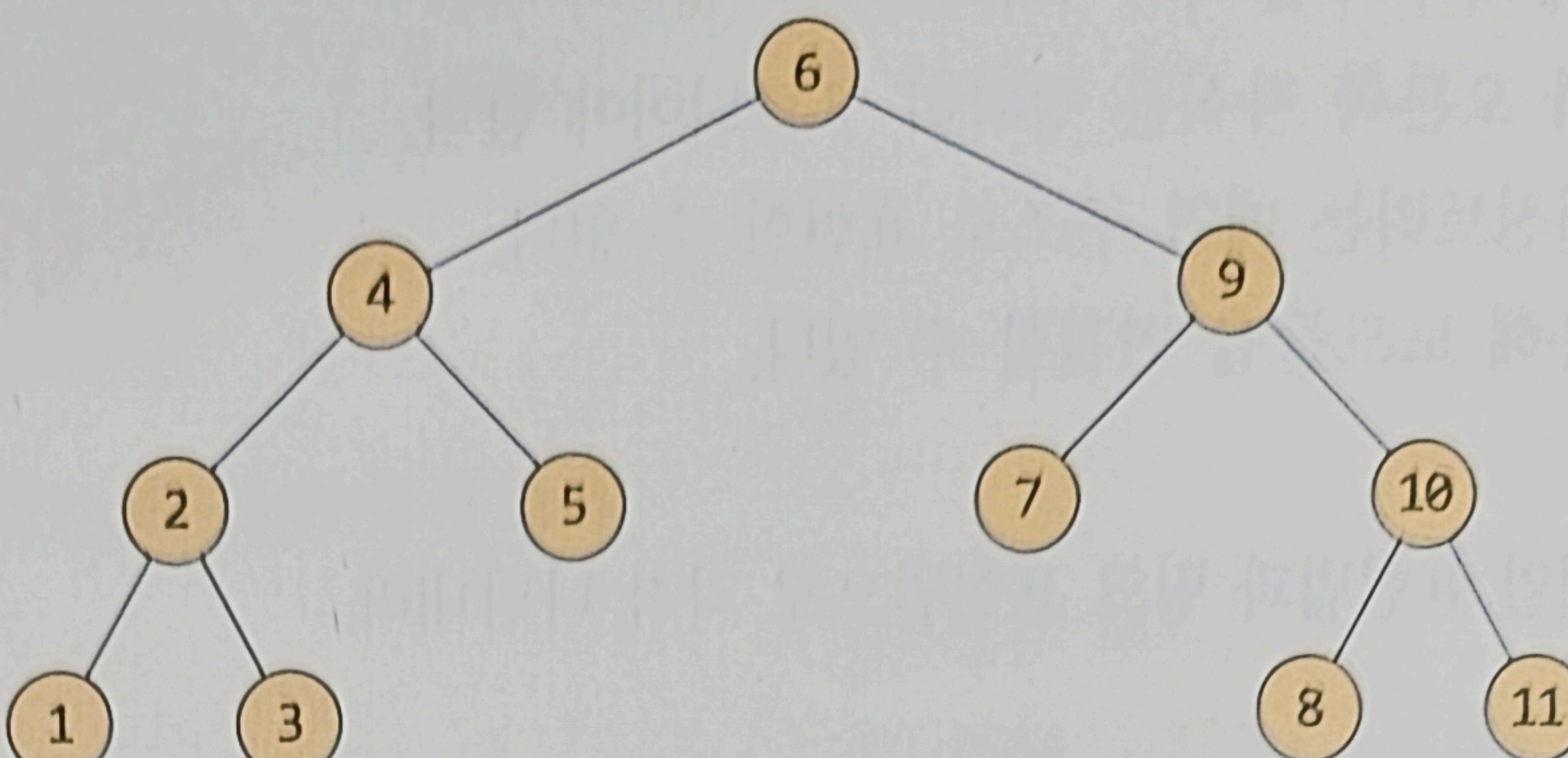
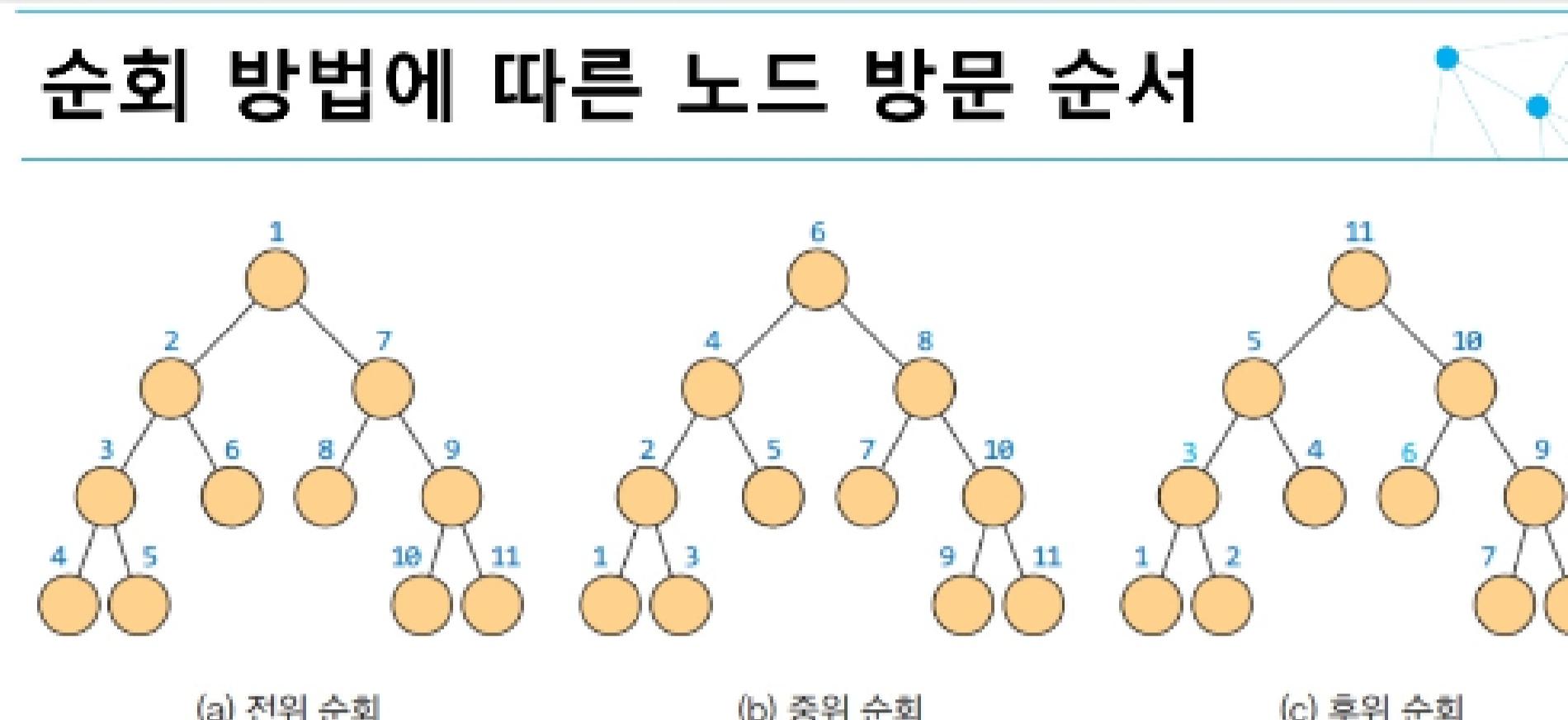
- ① C
- ② D
- ③ E
- ④ F

A, B, D, C, E, F

## 8.8 같은 수의 노드가 저장되는 경우, 가장 높이가 작아지는 트리는?

- ① 경사트리
- ② 이진트리
- ③ 이진탐색트리

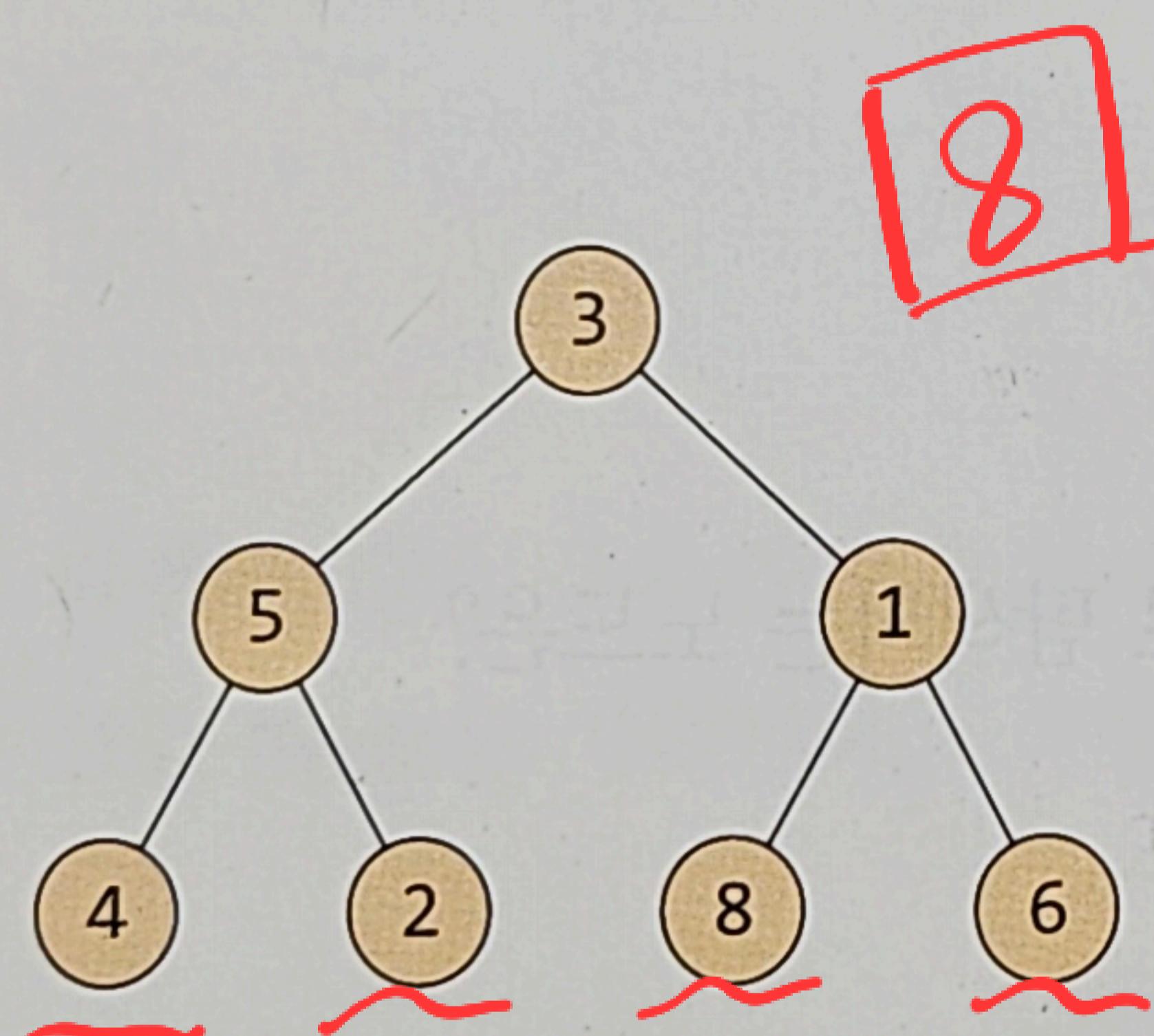
### 8.9 다음의 이진트리에 대하여 다음 질문에 답하라.



[ 6, 4, 9, 2, 5, 7, 10, 1, 3, None, None, None, None, 8, 11 ]

- (1) 이 트리를 1차원 배열로 표현하라.
  - (2) 이 트리를 전위 순회한 결과를 써라. 6, 4, 2, 1, 3, 5, 9, 7, 10, 8, 11
  - (3) 이 트리를 후위 순회한 결과를 써라. 1, 3, 2, 5, 4, 7, 8, 11, 10, 9, 6
  - (4) 이 트리를 중위 순회한 결과를 써라. 1, 2, 3, 4, 5, 6, 7, 9, 8, 10, 11
  - (5) 이 트리를 레벨 순회한 결과를 써라.
- 6, 4, 9, 2, 5, 7, 10, 1, 3, 8, 11

### 8.10 다음과 같은 함수가 아래에 표시된 이진트리의 루트 노드에 대해 호출된다고 하자. 함수가 반환하는 값은 무엇인가? 단, max(a,b)는 a와 b 중에 더 큰 값을 반환하는 함수라고 가정한다.



```
def mystery( p ) :
    if p == None :
        return 0
    elif p.left == None and p.right == None :
        return p.data
    else :
        return max(mystery(p.left), mystery(p.right))
```

### 8.11 다음 중 힙을 이용한 정렬이 특히 유용하게 사용될 수 있는 경우는?

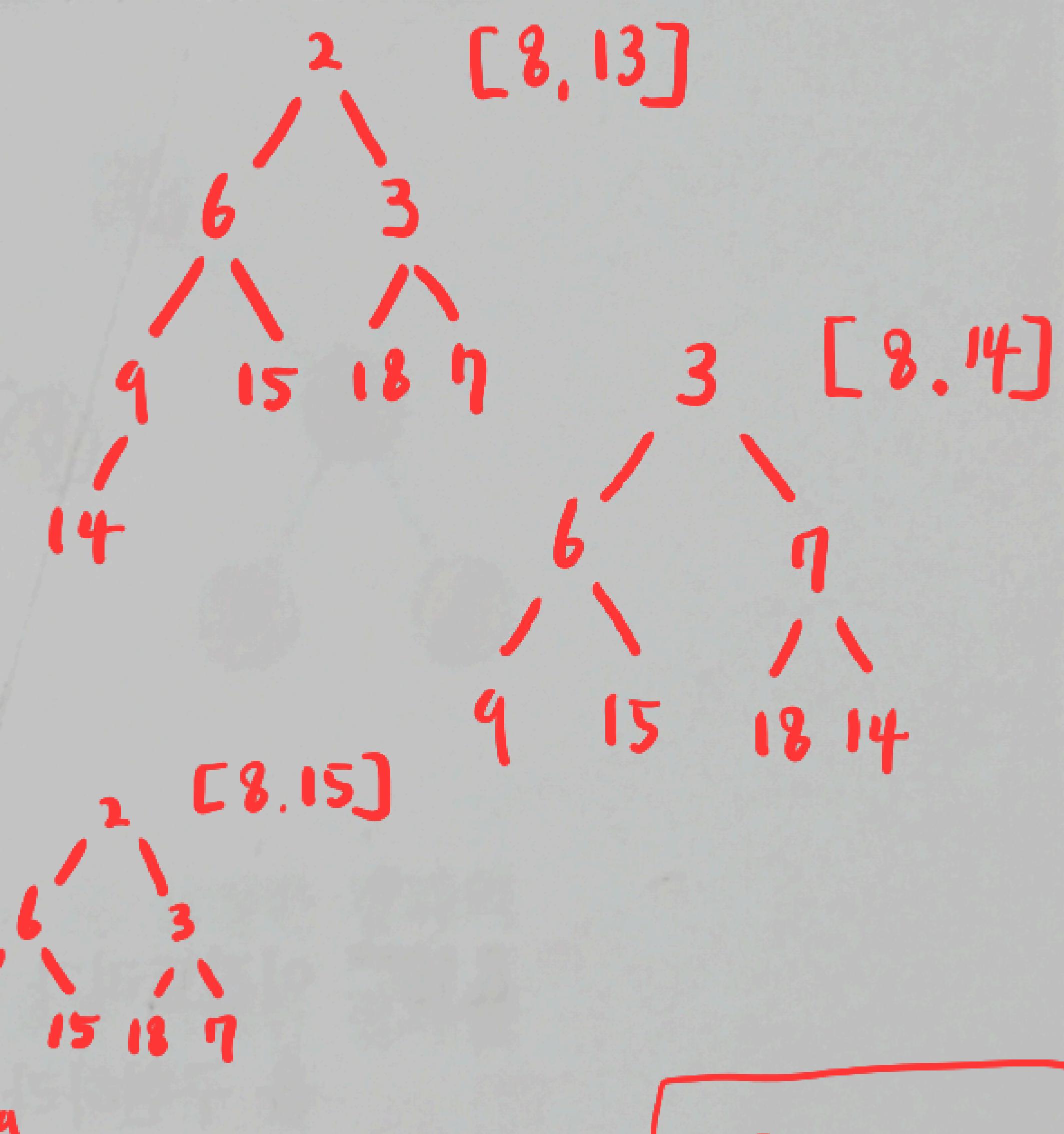
- ① 데이터 100개 중에서 가장 큰 5개만 뽑고자 할 때
- ② 비교적 데이터의 개수가 적을 때
- ③ 정렬의 대상이 되는 레코드의 크기가 클 때
- ④ 데이터가 역순으로 정렬되어 있을 때

8.12 최소 힙에서 가장 작은 데이터가 있는 노드는?

- ① 마지막 노드
- ② 첫 번째 노드
- ③ 중간 노드
- ④ 알 수 없다.

8.13 최소 힙을 구현한 배열의 내용이 다음과 같을 때 해당하는 힙 트리를 그려라. 단, 인덱스 0은 사용하지 않았다.

	0	1	2	3	4	5	6	7	8
a[i]		2	6	3	9	15	18	7	14



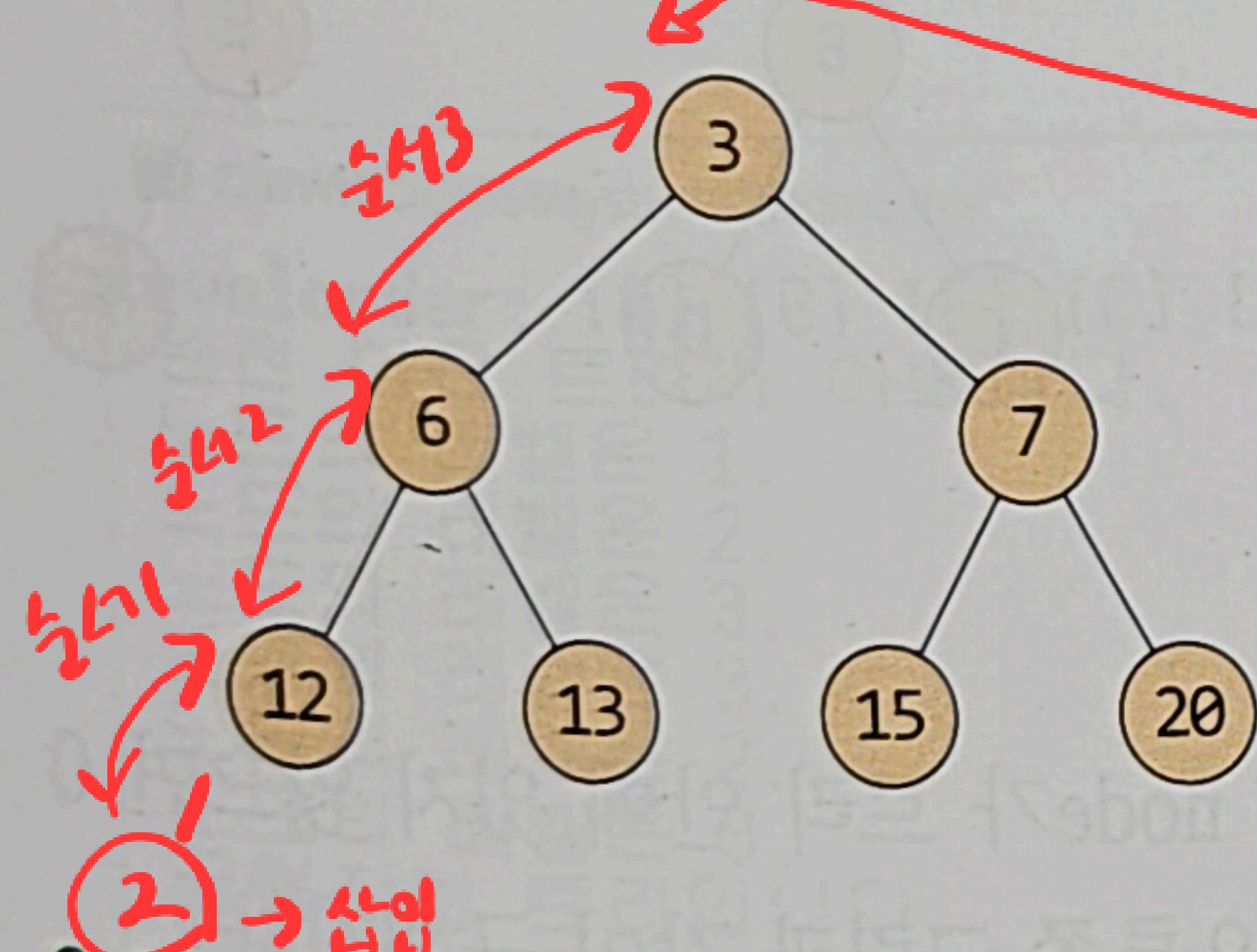
8.14 위 문제에서 삭제 연산을 한번 수행한 후의 배열의 내용을 적어라.

[3, 6, 7, 9, 15, 18, 14]

8.15 문제 8.13의 힙에서 데이터 7을 삽입한 후의 배열의 내용을 적어라.

[2, 6, 3, 7, 15, 18, 7, 14, 9]

8.16 다음의 최소 힙 트리에서 답하라.



(1) 2를 삽입하였을 경우, 힙 트리를 재구성하는 과정을 보여라.

(2) 삭제 연산이 한번 이루어진 다음에 힙을 재구성하는 과정을 보여라.

8.17 다음과 같은 숫자가 순서대로 주어졌을 때 다음 물음에 답하시오.

10, 40, 30, 5, 12, 6, 15, 9, 60

(1) 위의 숫자들을 순차적으로 읽어서 최대 힙 트리를 구성하라. 공백 트리에서부터 최대 힙 트리가 만들어지는 과정을 보여라.

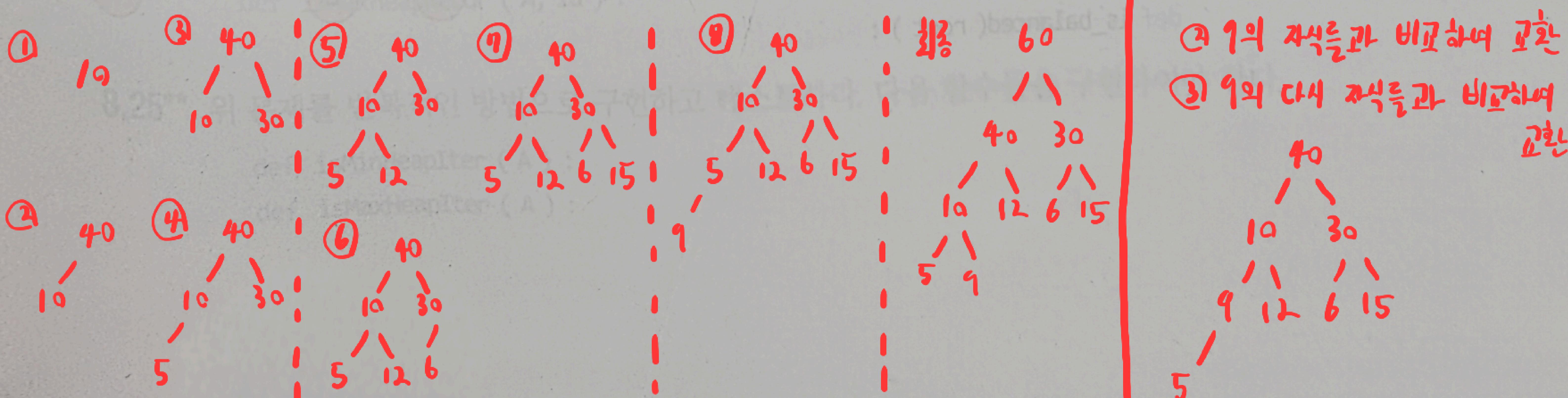
(2) 구성된 최대 힙 트리가 저장된 배열의 내용을 표시하라. [60, 40, 30, 10, 12, 6, 15, 5, 9]

(3) 구성된 최대 힙 트리에서 최댓값을 제거한 다음 재정비하는 과정을 설명하라.

① 60 제거 후 9를 루트로

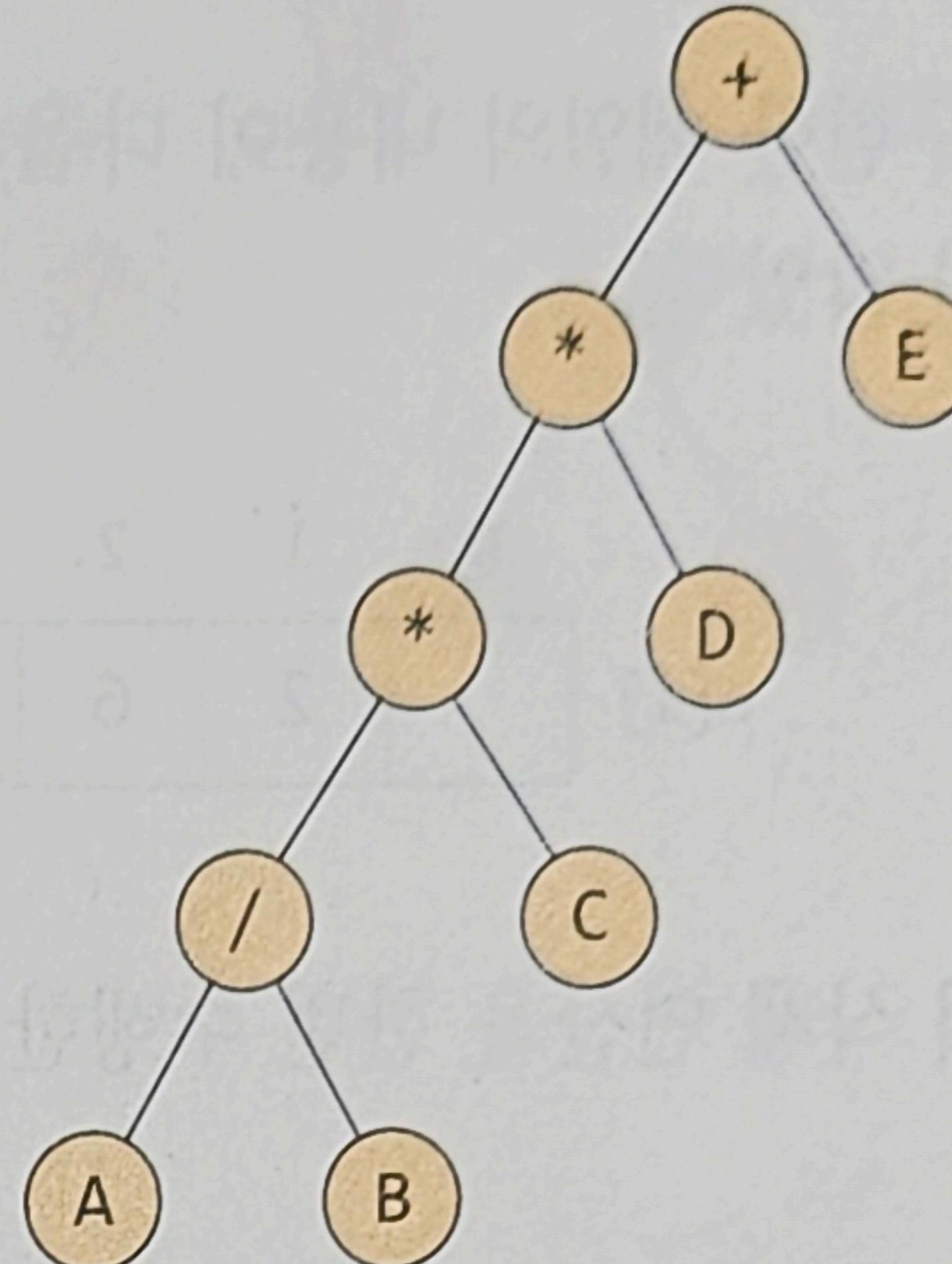
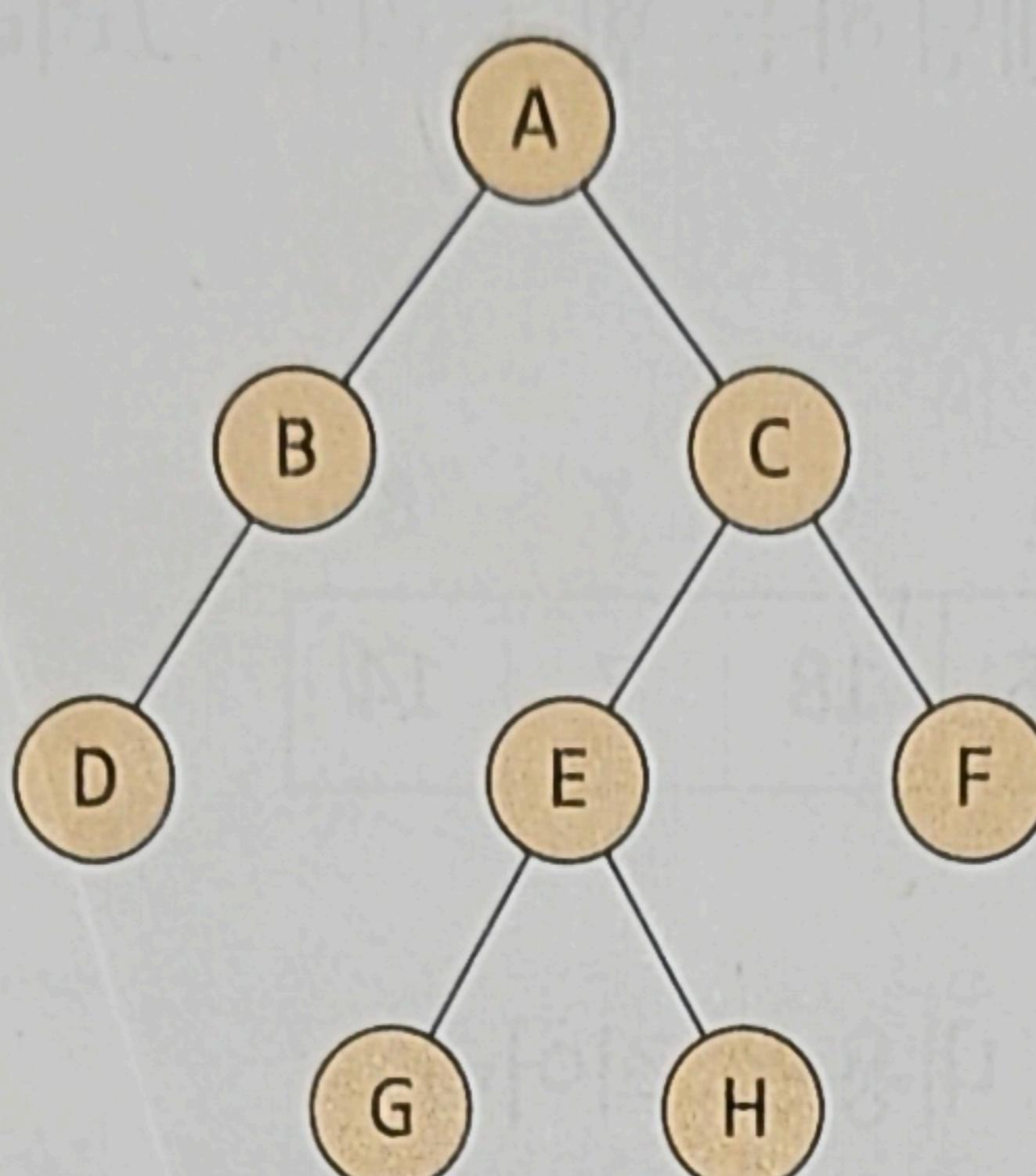
② 9의 자식들과 비교하여 교환

③ 9의 대체 자식들과 비교하여 교환



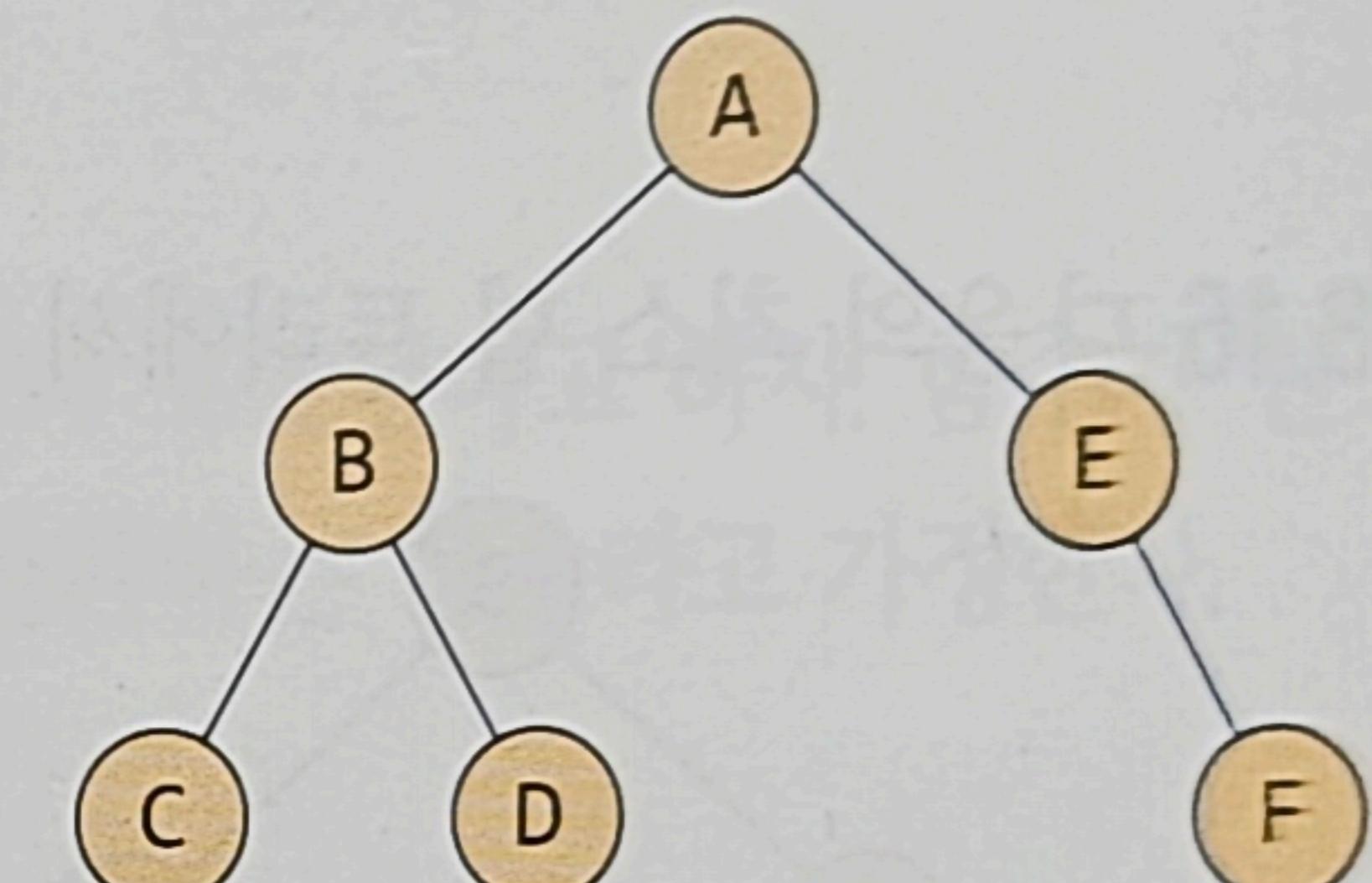
## 11Week 05\_13.py

- ★ 8.18\* 그림과 같은 트리를 연결된 구조로 표현하고 네 가지 순회 방법으로 노드를 방문한 결과를 출력하는 프로그램을 작성하라. 또한 각 트리의 노드의 개수와 단말 노드의 개수 및 트리의 높이를 구해 출력하라. 8.3절의 함수들과 테스트 코드를 이용하면 된다.



- 8.19\*\* 이진트리가 완전이진트리인지를 검사하는 다음 연산을 구현하라. 오른쪽 트리는 완전이진트리가 아니므로 False가 반환되어야 한다.

```
def is_complete_binary_tree( root ) :
```

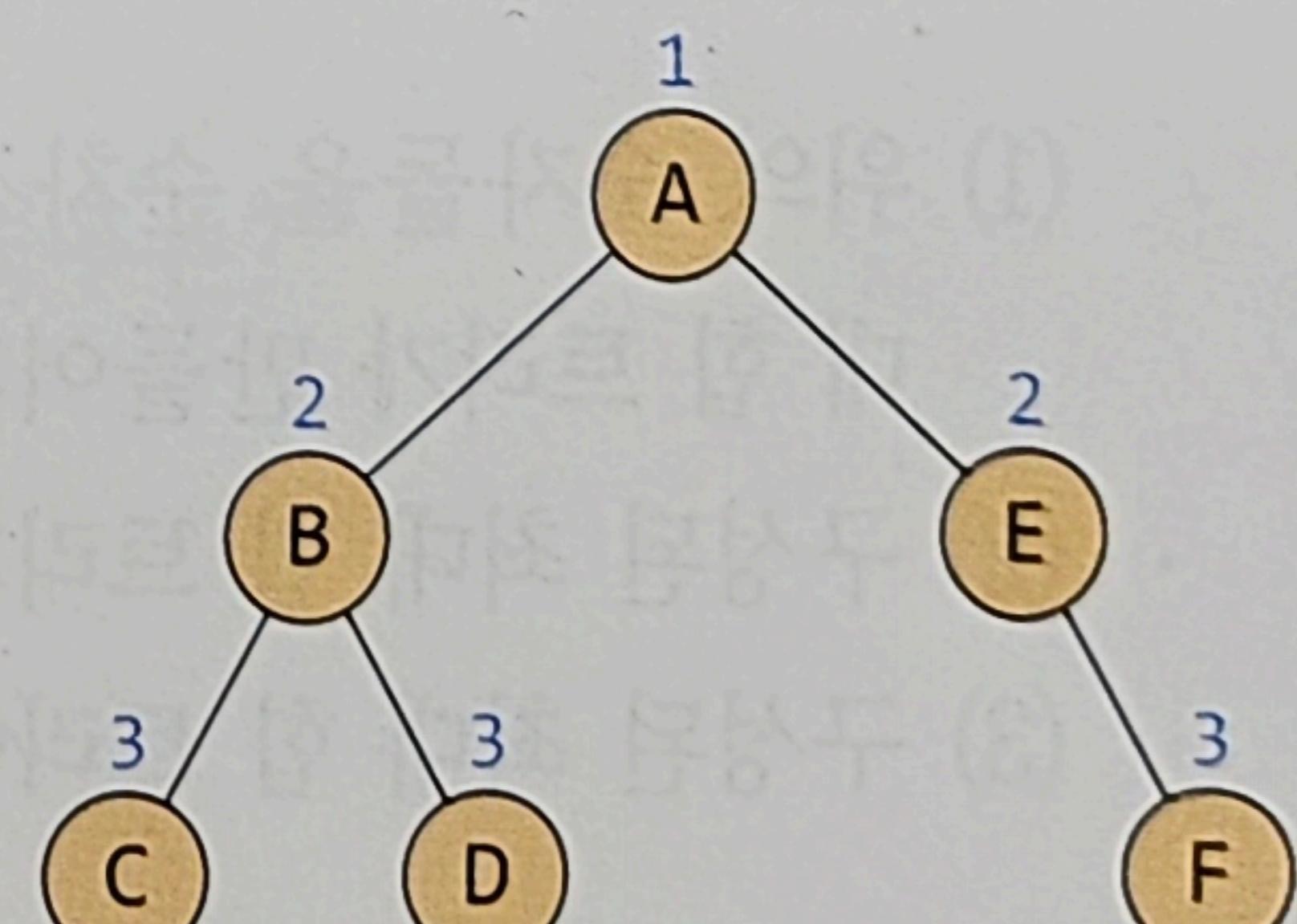


- 8.20\*\* 임의의 node의 레벨을 구하는 연산을 구현하라. 만약 node가 트리 안에 있지 않으면 0을 반환하라. 위의 이진트리에 대한 각 노드의 레벨은 오른쪽 그림과 같이 구해진다.

```
def level( root, node ) :
```

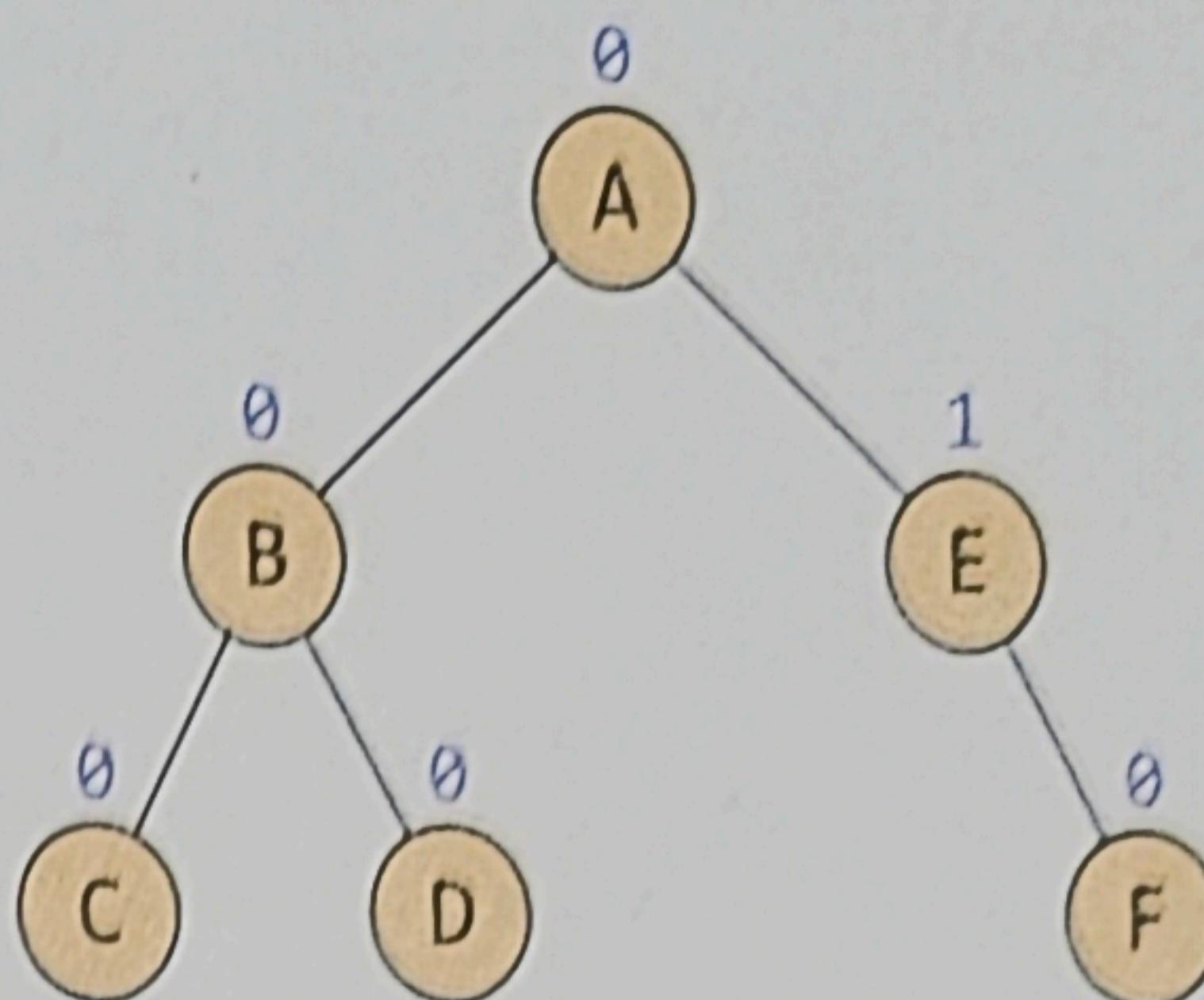
- 8.21\*\* 이진트리의 모든 노드에서 왼쪽 서브트리와 오른쪽 서브트리의 높이의 차이가 2보다 작으면 이 트리를 “균형 잡혀 있다(balanced)”라고 한다. 현재 이진트리가 균형 잡혀 있는지를 검사하는 다음 연산을 구현하라. 위의 트리는 높이의 차이가 1 이하로 균형 잡혀 있다.

```
def is_balanced( root ) :
```



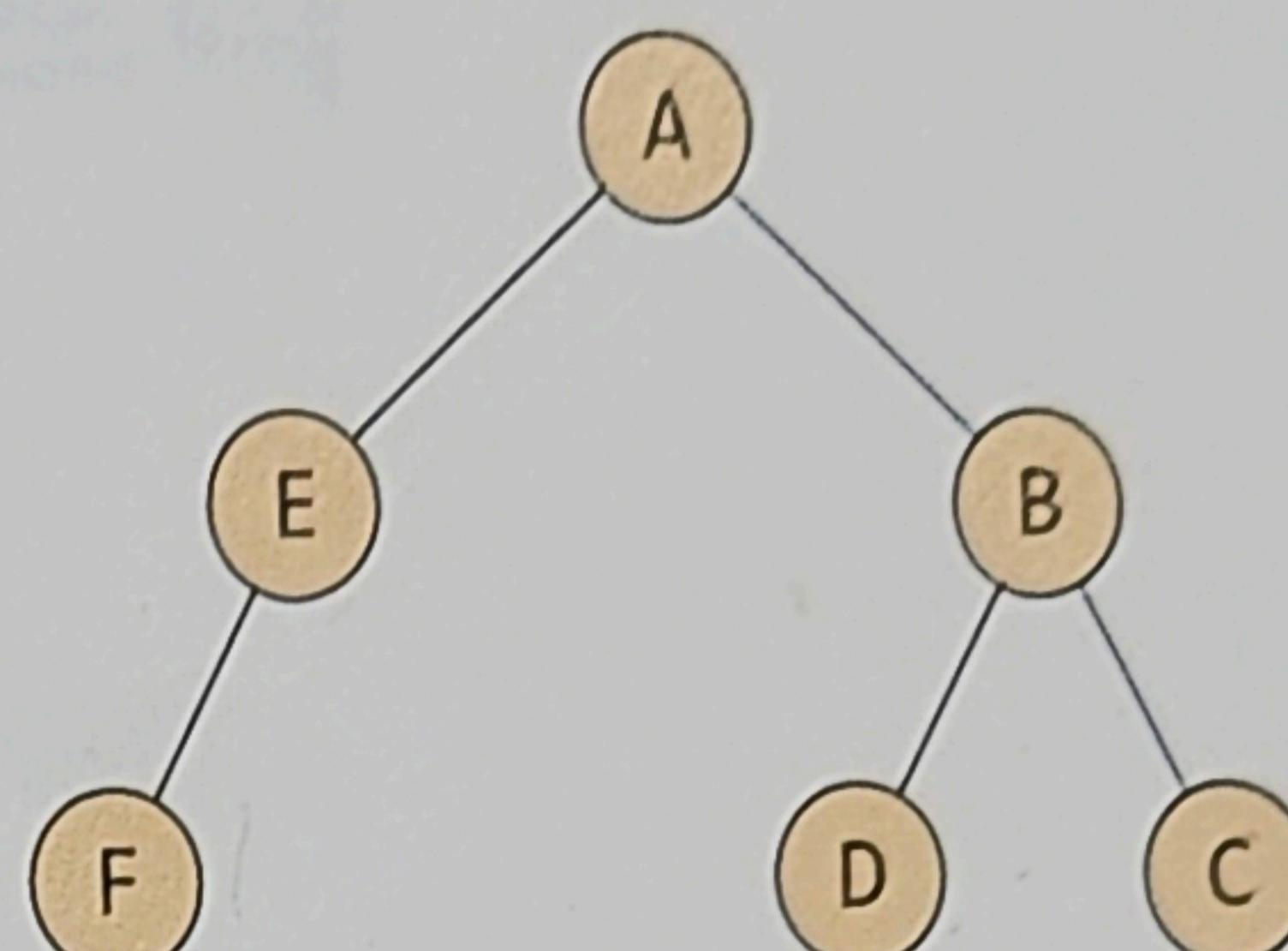
- ~~8.22\*\*~~ 이진트리에서 경로의 길이(path length)를 루트에서부터 모든 자식 노드까지의 경로의 길이의 합이라고 하자. 경로의 길이를 구하는 다음 연산을 구현하라. 위의 트리에서 경로의 길이는  $0+1+1+2+2+2=8$ 이다.

```
def path_length( root ) :
```



- ~~8.23\*\*~~ 이진트리를 좌우로 대칭시키는 다음 연산을 구현하라. 위의 그래프에 대한 대칭 연산 결과는 오른쪽과 같다.

```
def reverse( root ) :
```



※ 8.19 ~ 8.23을 구현하고 주어진 이진트리에 대해 모든 함수를 순서적으로 실행한 결과의 예는 다음과 같다. 단, 맨 앞과 마지막에 트리의 형태를 알 수 있도록 레벨 순회한 결과를 추가하였다.

```
C:\WINDOWS\system32\cmd.exe
levelorder: [A] [B] [E] [C] [D] [F]
완전 이진 트리가 아닙니다.
노드의 레벨은 1.
노드의 레벨은 2.
노드의 레벨은 3.
노드의 레벨은 3.
노드의 레벨은 2.
노드의 레벨은 3.
균형잡힌 트리입니다.
전체 경로의 길이는 8입니다.

트리의 좌우를 교환합니다.
levelorder: [A] [E] [B] [F] [D] [C]
계속하려면 아무 키나 누르십시오 . . .
```

- ~~8.24\*\*~~ 배열로 표현된 완전이진트리 A가 힙 조건을 만족하는지를 검사하는 다음 함수를 순환적인 방법으로 구현하고 테스트하라. 다음 함수들을 구현하여야 한다.

```
def isMinHeapRecur ( A, id ) :
def isMaxHeapRecur ( A, id ) :
```

- ~~8.25\*\*~~ 위 문제를 반복적인 방법으로 구현하고 테스트하라. 다음 함수들을 구현하여야 한다.

```
def isMinHeapIter ( A ) :
def isMaxHeapIter ( A ) :
```

## | 연습문제 |

**9.1** 이진탐색트리의 삽입 연산에 대한 설명으로 옳지 않은 것은?

- ① 루트에서부터 탐색 연산을 먼저 진행해야 한다.
- ② 순환 구조와 반복 구조로 모두 구현할 수 있다.
- ③ 탐색이 성공하면 중복된 노드가 있는 것이므로 삽입하지 않는다.
- ④ 다음 탐색 위치가 공백(None)이면 삽입 연산은 실패한 것이다.

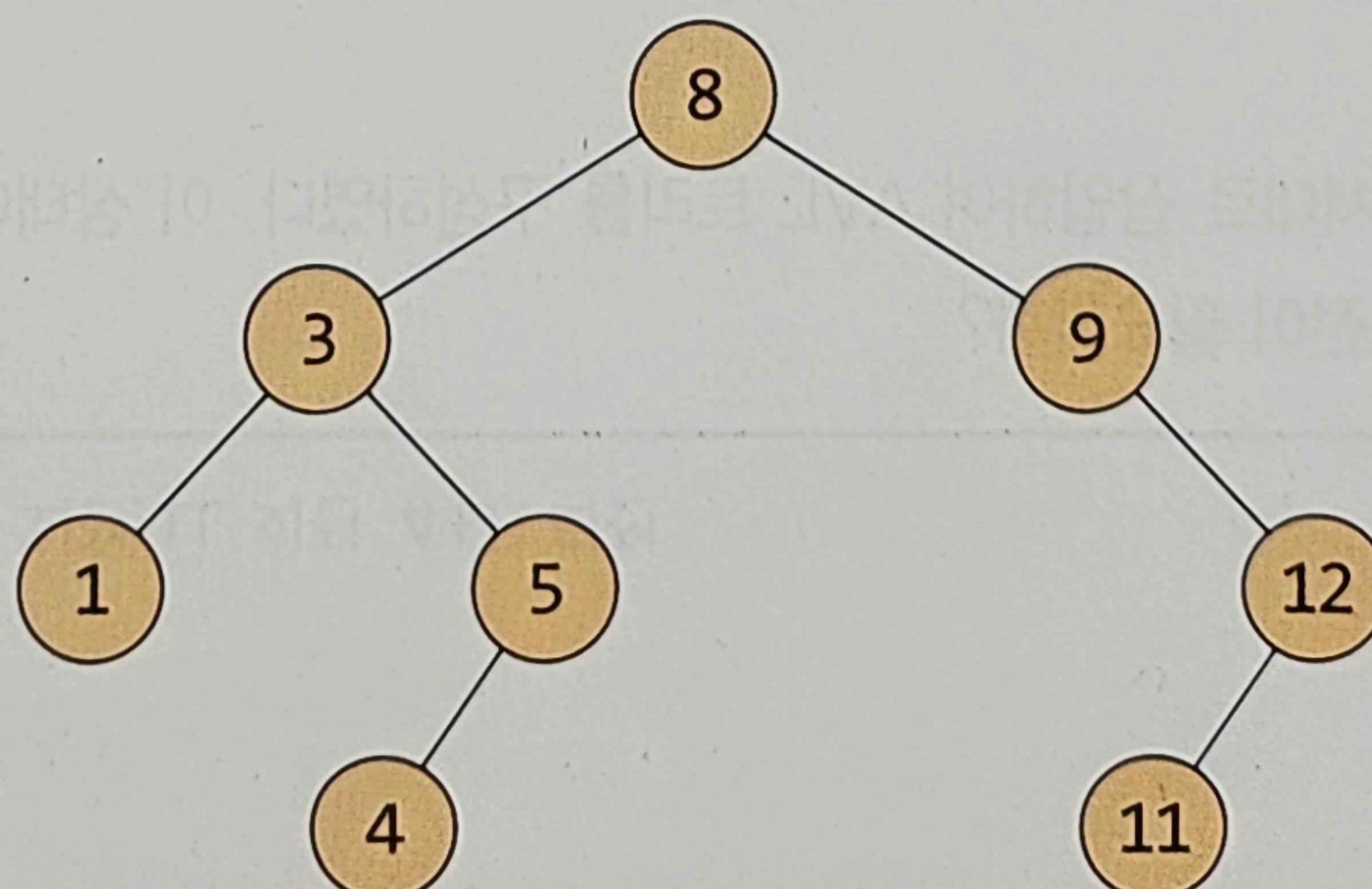
**9.2** 이진탐색트리의 탐색 연산에 대한 설명으로 옳지 않은 것은?

- ① 루트 노드부터 탐색을 시작한다.
- ② 키값이 노드의 값보다 작으면 왼쪽 서브트리로 탐색을 진행한다.
- ③ 서브트리가 공백 노드이면 탐색은 실패한 것이다.
- ④ 반복 구조로는 구현할 수 없다.

**9.3** 이진탐색트리의 삭제 연산은 세 가지 경우로 나누어 처리된다. 다음 중 이에 포함되지 않는 경우는?

- ① 삭제하려는 노드가 루트 노드일 경우
- ② 삭제하려는 노드가 단말 노드일 경우
- ③ 삭제하려는 노드가 하나의 서브트리만 가지고 있는 경우
- ④ 삭제하려는 노드가 두 개의 서브트리 모두 가지고 있는 경우

\* 다음과 같은 이진탐색트리가 주어졌다. 물음에 답하라(9.4~9.7).



9.4 이 트리에 7을 추가하면 트리는 어떻게 될까? 그림으로 그려라. 7은 5의 왼쪽 자식 노드

9.5 이 트리에서 4를 탐색할 때 거치는 노드들을 순서대로 나열하라.

$8 \rightarrow 3 \rightarrow 5 \rightarrow 4$

9.6 이 트리에서 노드 9를 삭제하면 트리는 어떻게 될까?

12가 9의 자리로 간다. 11이 12의 자리에 차운다.

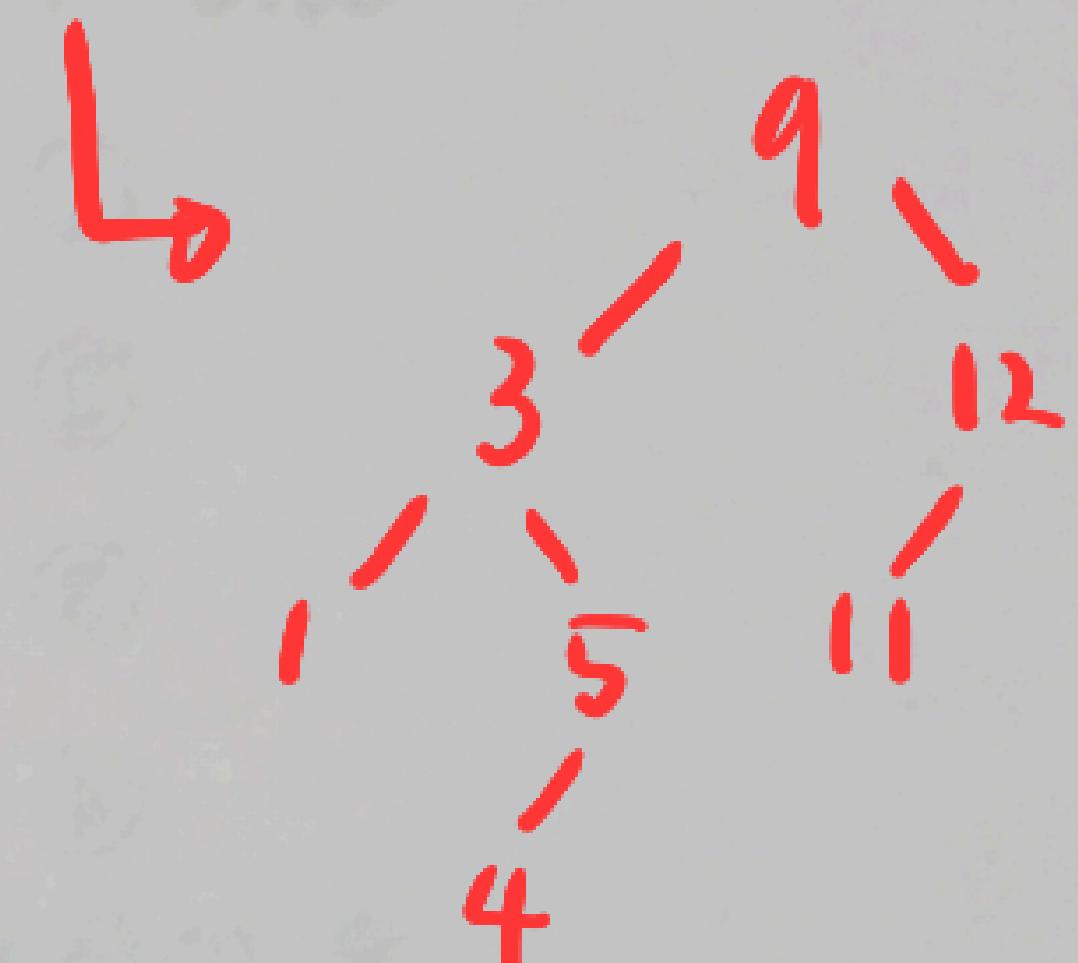
9.7 이 트리에서 노드 8을 삭제하면 트리는 어떻게 될까?

9가 8의 자리로, 12는 9의 자리, 11은 12의 왼쪽 자식으로

9.8 공백상태인 이진탐색트리에 다음과 같은 순서로 노드들이 추가된다.

11, 6, 8, 19, 4, 10, 5, 17, 43, 49, 31

생성되는 이진탐색트리를 그려라.



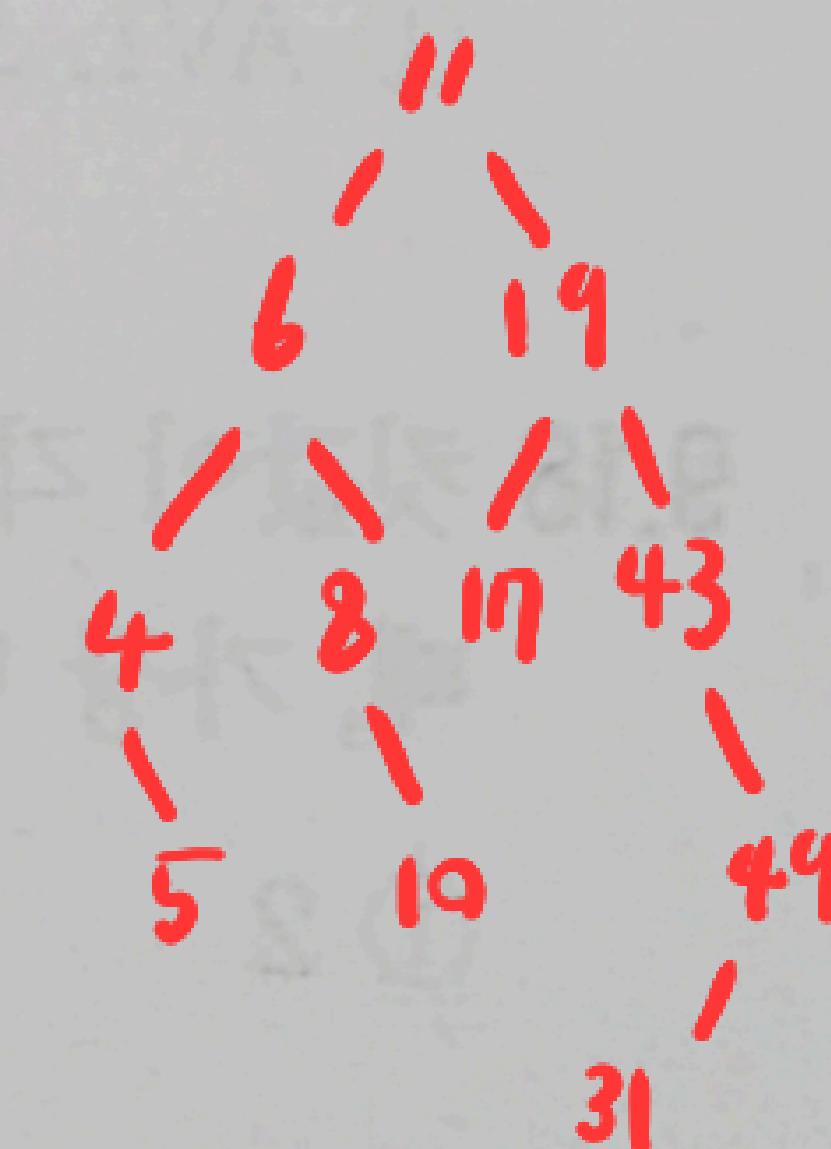
9.9 위 문제에서 생성되는 이진탐색트리를 다음의 방법으로 순회하였을 때 방문하는 노드를 순서대로 적어라.

(1) 전위순회 11, 6, 4, 5, 8, 10, 14, 17, 43, 31, 49

(2) 중위순회 4, 5, 6, 8, 10, 11, 14, 17, 31, 43, 49

(3) 후위순회 5, 4, 10, 8, 6, 17, 31, 44, 43, 19, 11

(4) 레벨순회 11, 6, 14, 4, 8, 17, 43, 5, 19, 31, 49



9.10 이진탐색트리에서 어느 단말 노드를 삭제하려고 한다. 삭제가 끝나면 트리의 루트 노드가 변경되는 상황은 언제 발생하나? 루트노드가 단말노드 일 때 루트 노드가 변경됨.

즉, 루트노드가 자식노드를 가지 않을 때 발생한다.

9.11 이진탐색트리에서 자식이 하나인 어느 노드를 삭제하려고 한다. 삭제가 끝나면 루트 노드가 변경되는 상황은 언제 발생하나? 루트노드가 자식을 하나만 가질 때 루트노드가 변경됨.

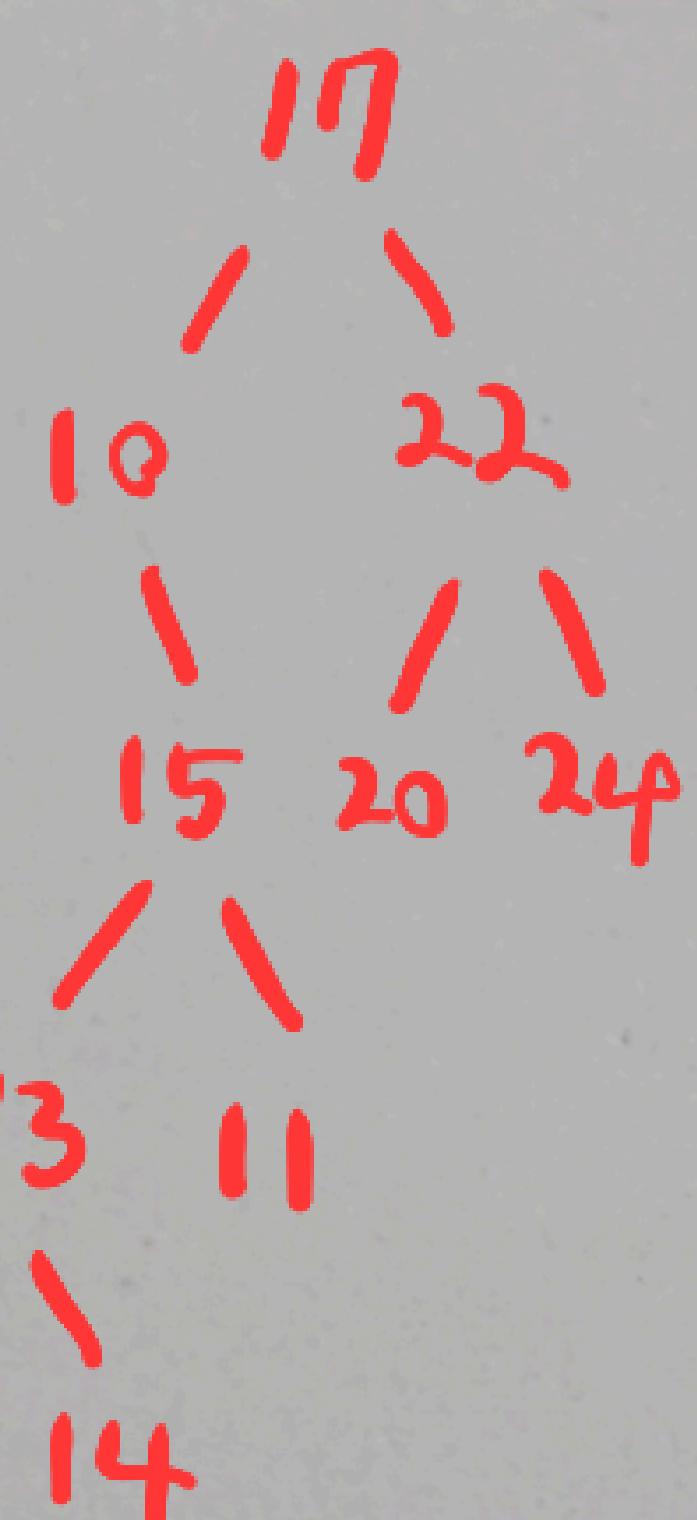
이 경우, 자식노드가 새로운 루트가 됩니다.

9.12 다음 데이터를 순서대로 이진탐색트리에 입력했을 때 단말 노드의 개수는?

17, 10, 22, 15, 13, 24, 20, 11, 14

11, 14, 20, 24

4개



9.13 배열에 정렬된 값이 들어 있는 경우에 이진탐색이라는 효과적인 탐색기법을 사용할 수 있다. 이진탐색을 이용한 탐색과 이진탐색트리를 이용한 탐색의 장단점을 설명하라.

이진탐색의 장점: 배열에 저장된 값이 정렬되어 있어 탐색 속도가 빠르다 ( $O(\log n)$ ).

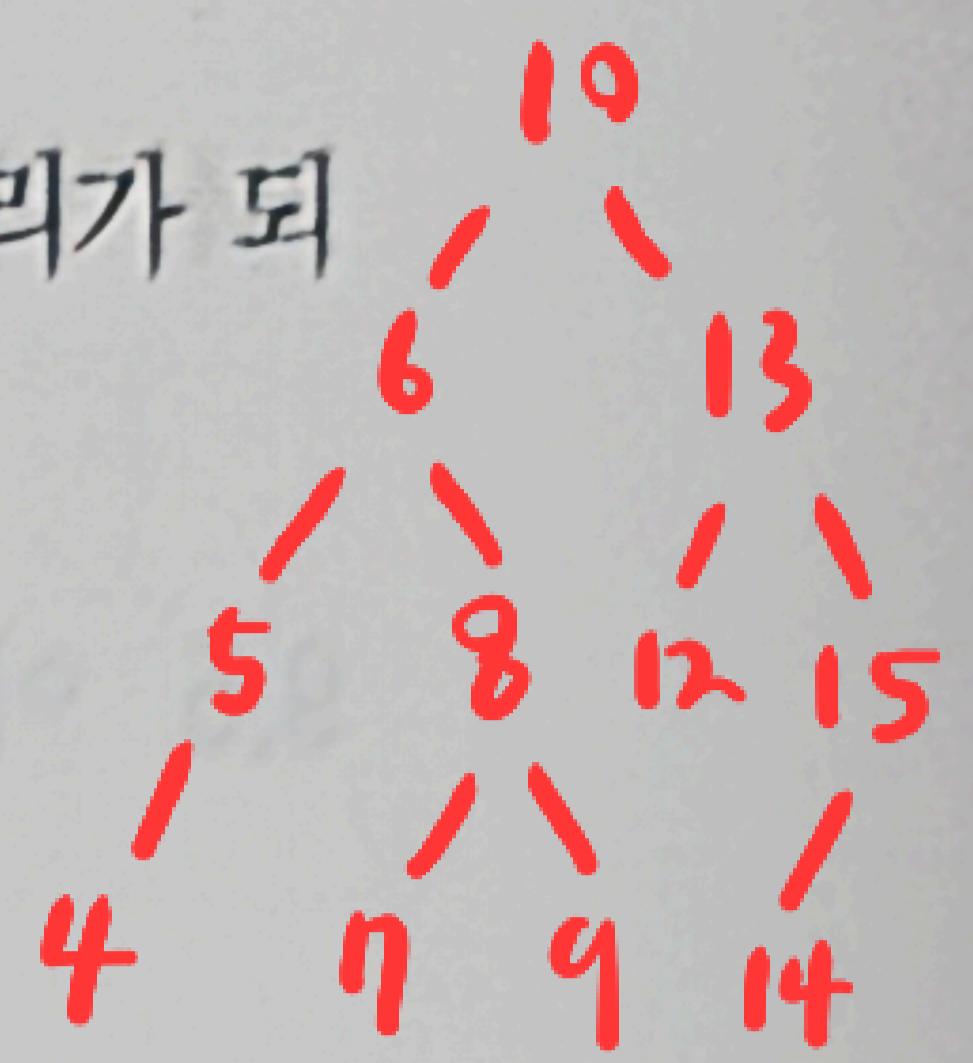
이진탐색의 단점: 삽입과 삭제가 비효율적이다.

이진탐색트리의 장점: 동적 데이터의 삽입과 삭제가 용이하다.

이진탐색트리의 단점: 균형이 맞지 않으면 탐색 속도가 느려질 수 있다 (최악의 경우  $O(n)$ ).

9.14 다음 데이터들이 어떤 순서로 이진탐색트리에 입력되었을 경우, 가장 균형 잡힌 트리가 되는가? 여러 가지 답이 가능하다.

10, 5, 6, 13, 15, 8, 14, 7, 12, 4



9.15 위 문제의 데이터가 어떤 순서로 입력되었을 경우에 가장 불균형한 이진탐색트리가 되는가?

4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15

9.16 다음 중 AVL 트리에 대한 설명으로 옳지 않은 것은?

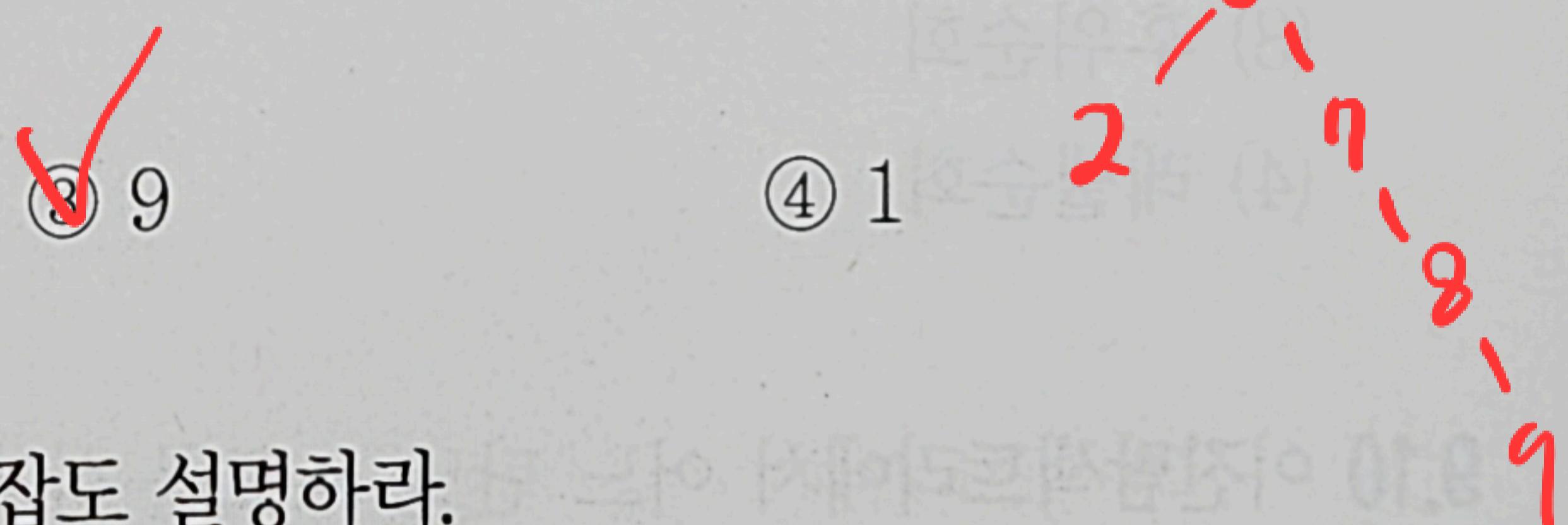
- ① Adelson-Velskii와 Landis에 의해 제안되었다.
- ② 평균, 최선, 최악 시간 복잡도로  $O(\log n)$ 을 보장한다.
- ③ 모든 노드에서 왼쪽과 오른쪽 서브트리의 높이 차가 1을 넘지 않는다.
- ④ 탐색 연산과 삽입 연산에서 균형이 깨질 수 있다.

9.17 다음 중 탐색트리의 균형화 기법이 아닌 것은?

- ① AVL 트리
- ② 2-3 트리
- ③ Red-Black 트리
- ④ 포화이진트리

9.18 키값이 각각 6, 7, 2, 8, 9, 1, 5, 4, 3인 레코드가 순서대로 공백 AVL 트리에 입력된다고 할 때, 가장 먼저 균형이 깨지는 입력은?

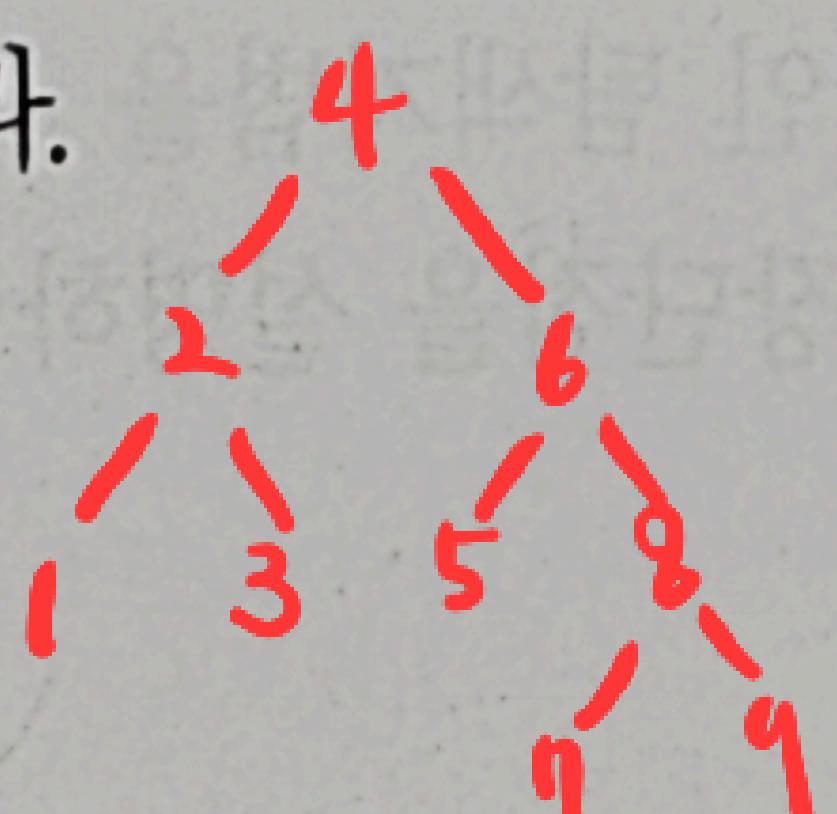
- ① 2
- ② 8
- ③ 9
- ④ 1



9.19 균형 이진탐색트리에서 다음 연산의 시간 복잡도 설명하라.

- (1) 임의의 키값의 노드 탐색 연산  $O(\log n)$
- (2) 노드 삽입 연산  $O(\log n)$
- (3) 노드 삭제 연산  $O(\log n)$
- (4) 트리의 최솟값이나 최댓값을 가진 노드 탐색 연산  $O(\log n)$
- (5) 트리의 노드 수 계산  $O(n)$
- (6) 단말 노드의 수 계산  $O(n)$
- (7) 트리의 높이 계산  $O(1)$

9.20\* 공백상태인 AVL 트리에 다음과 같은 순서로 노드들이 추가된다. 생성되는 AVL 트리를 그려라.



1, 2, 3, 4, 5, 6, 7, 8, 9

~~9.21\*~~ 이진탐색트리에서 최대 키와 최소 키를 가진 노드를 탐색하는 함수를 순환 구조를 이용하여 구현하라.

~~9.22\*~~ 반복 구조를 이용해 이진탐색트리의 삽입 연산을 구현하라.

~~9.23\*~~ 숫자들이 들어 있는 이진탐색트리를 중위 순회하면 정렬된 숫자가 얻어진다. 이를 이용하여 다음 배열에 들어 있는 숫자들을 정렬시키는 함수를 작성하여 보라. 배열에 들어 있는 숫자들을 이진탐색트리에 추가한 후에 트리를 중위 순회하면서 숫자들을 출력한다. 단 숫자들은 중복되지 않는다고 가정하자.

0	1	2	3	4	5	6	7	8	9	10
11	3	4	1	56	5	6	2	98	32	23

~~9.24\*\*~~ 이진탐색트리를 사용하여 우선순위 큐를 구현할 수도 있다. 이진탐색트리를 이용하여 우선순위 큐를 구현하고 삽입, 삭제, 탐색 연산의 시간 복잡도를 설명하라.

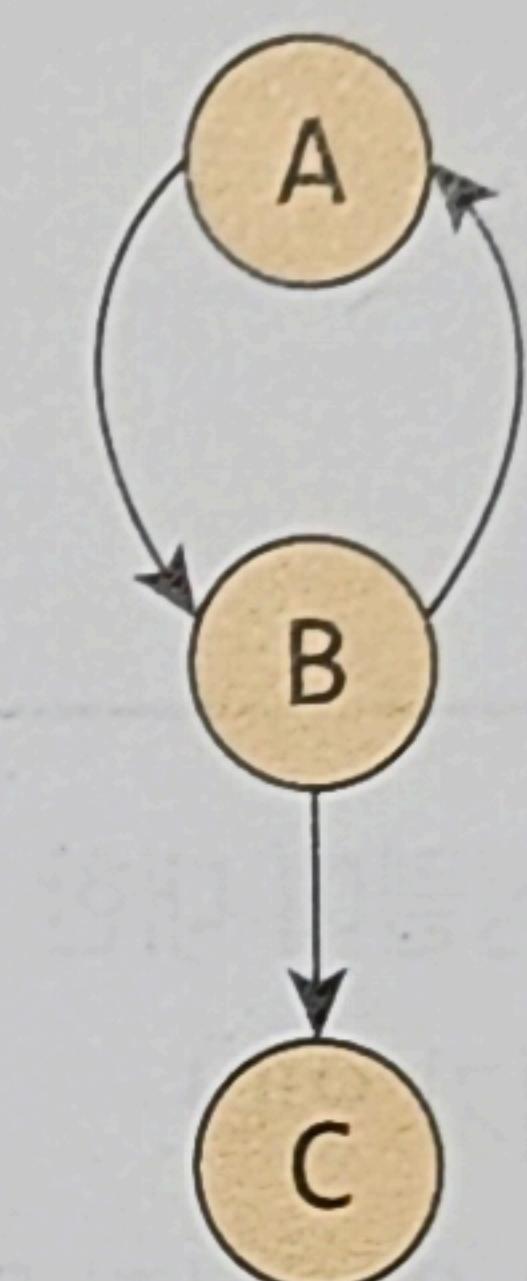
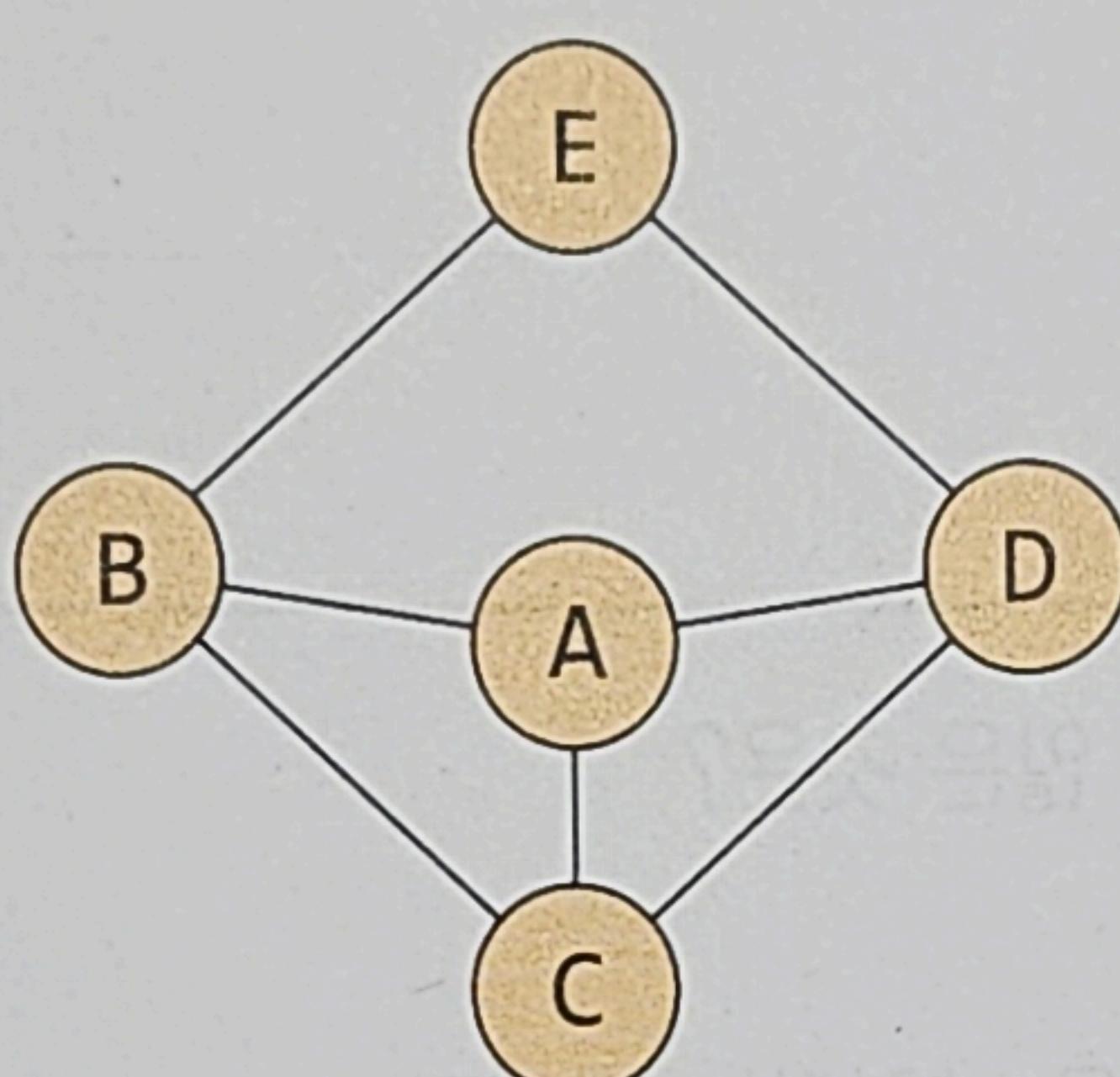
## 연습문제

10.1 다음 중 그래프에 대해 설명으로 옳지 않은 것은?

- ① 연결되어 있는 객체 간의 관계를 표현할 수 있다.
- ② 가장 일반적인 자료구조 형태이다.
- ③ 계층적인 구조의 자료를 나타내기에 적합하다.
- ④ 그래프와 관련된 다양한 문제를 연구하는 학문이 그래프 이론이다.

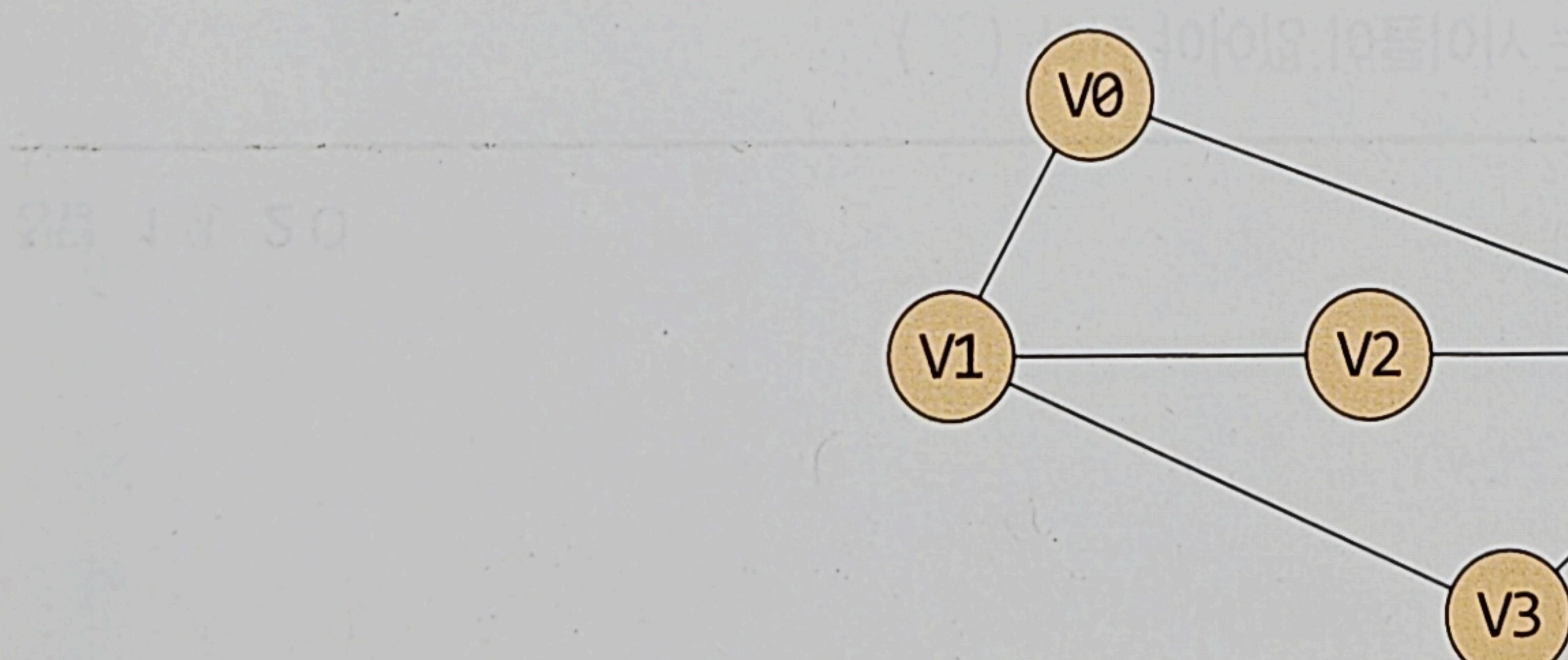
10.2 다음의 그래프의 부분 그래프를 3개 그려보라.

- 1) B-A  
2) D-C  
3) E-B-D



- 1) A  $\leftrightarrow$  B  
2) B  $\rightarrow$  C  
3) A  $\leftrightarrow$  B  $\rightarrow$  C

10.3 다음의 무방향 그래프를 인접 행렬과 인접 리스트로 각각 표현하라.

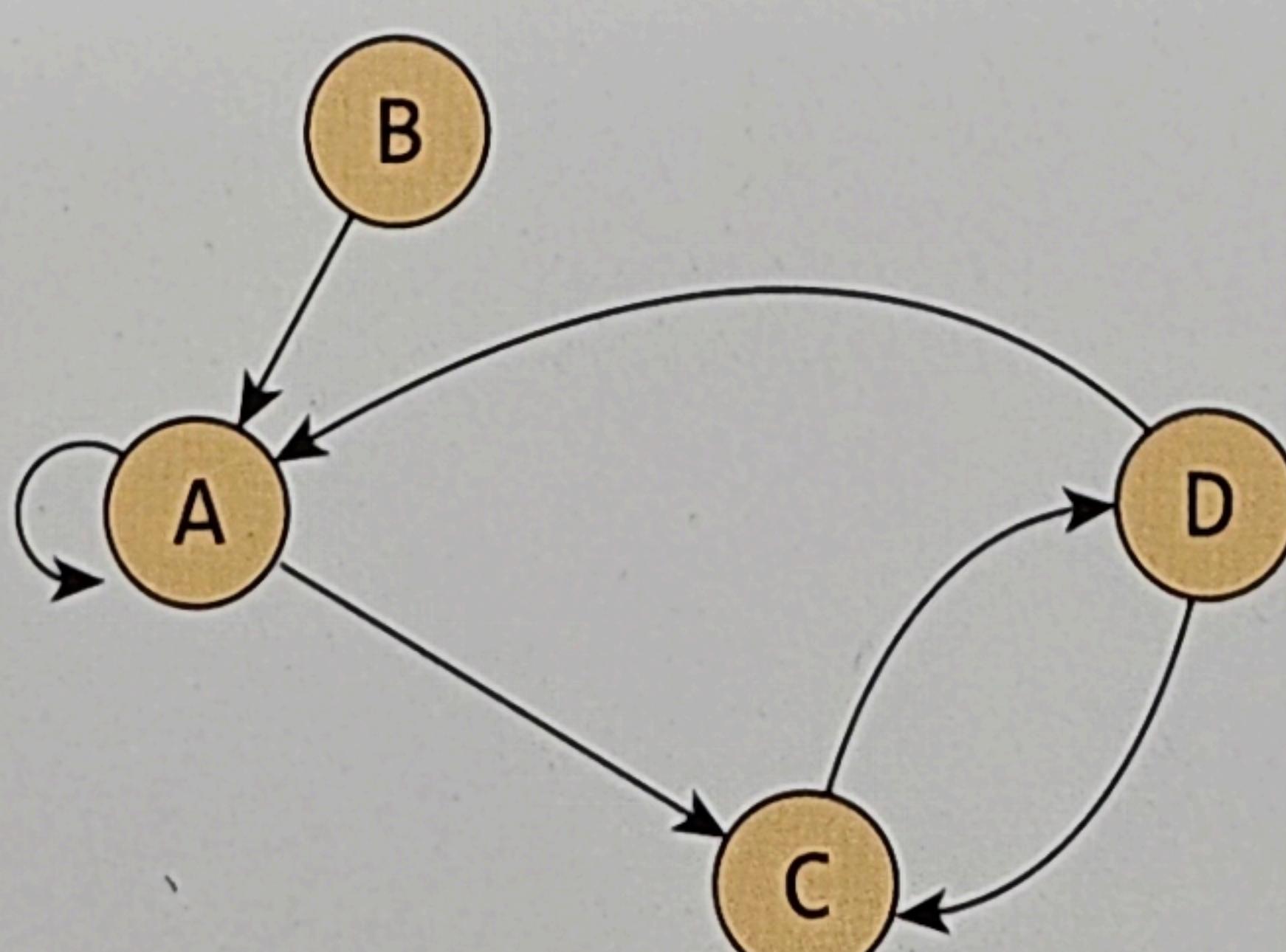


$V_0$	$V_1$	$V_2$	$V_3$	$V_4$
0	1	1	0	1
1	0	1	0	1
1	1	0	0	1
0	0	0	1	0

인접 행렬

인접 리스트
$V_0 : V_1, V_4$
$V_1 : V_0, V_2, V_3$
$V_2 : V_1, V_4$
$V_3 : V_1, V_4$
$V_4 : V_0, V_2, V_3$

10.4 다음의 방향 그래프를 인접 행렬과 인접 리스트로 각각 표현하라.



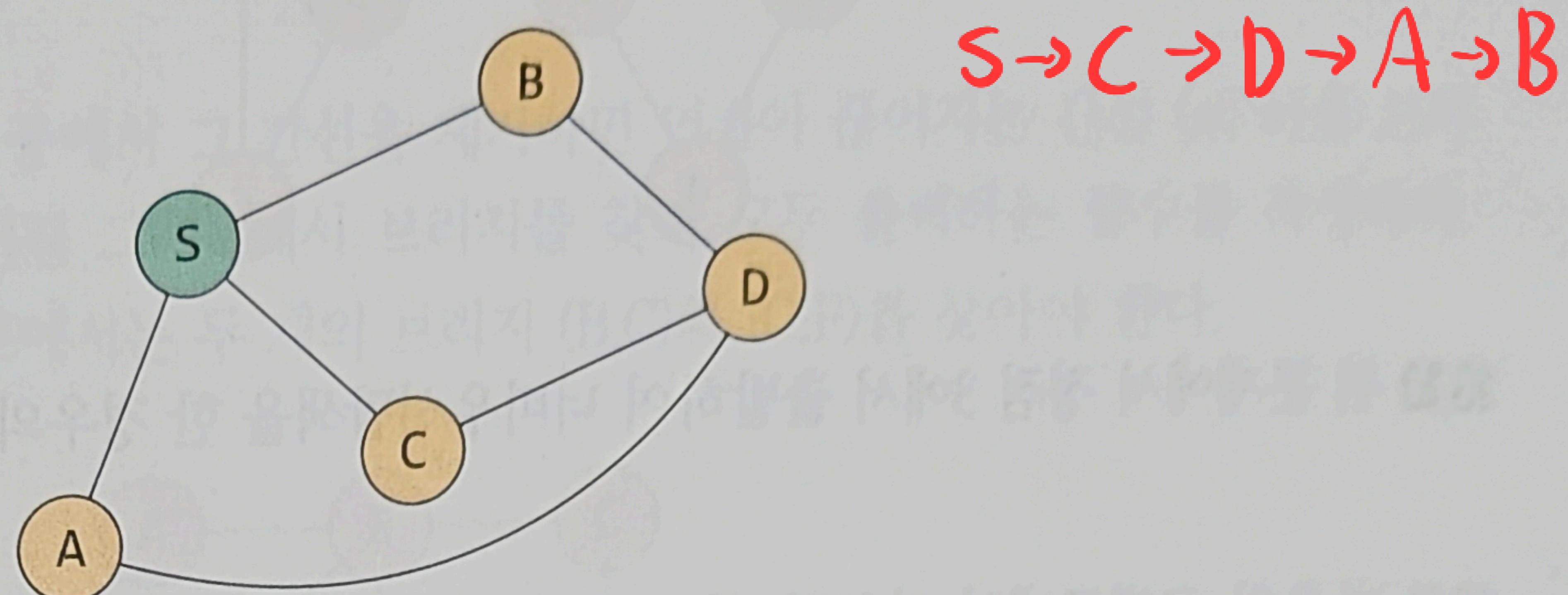
$A$	$B$	$C$	$D$
1	1	0	1
0	0	0	0
1	0	0	1
0	0	1	0

$A$	$: A, B, D$
$C$	$: A, D$
$D$	$: C$

**10.5** 그레프의 깊이우선탐색에 대한 설명으로 옳지 않은 것은?

- ① 그레프의 연결 요소를 구하기 위해 사용할 수 있다.
- ② 신장 트리를 구하기 위해 사용할 수 있다.
- ③ 최소신장트리를 구하는 크루스칼 알고리즘에서 사용한다.
- ④ 그레프의 임의의 노드에서 깊이우선탐색을 시작할 수 있다.

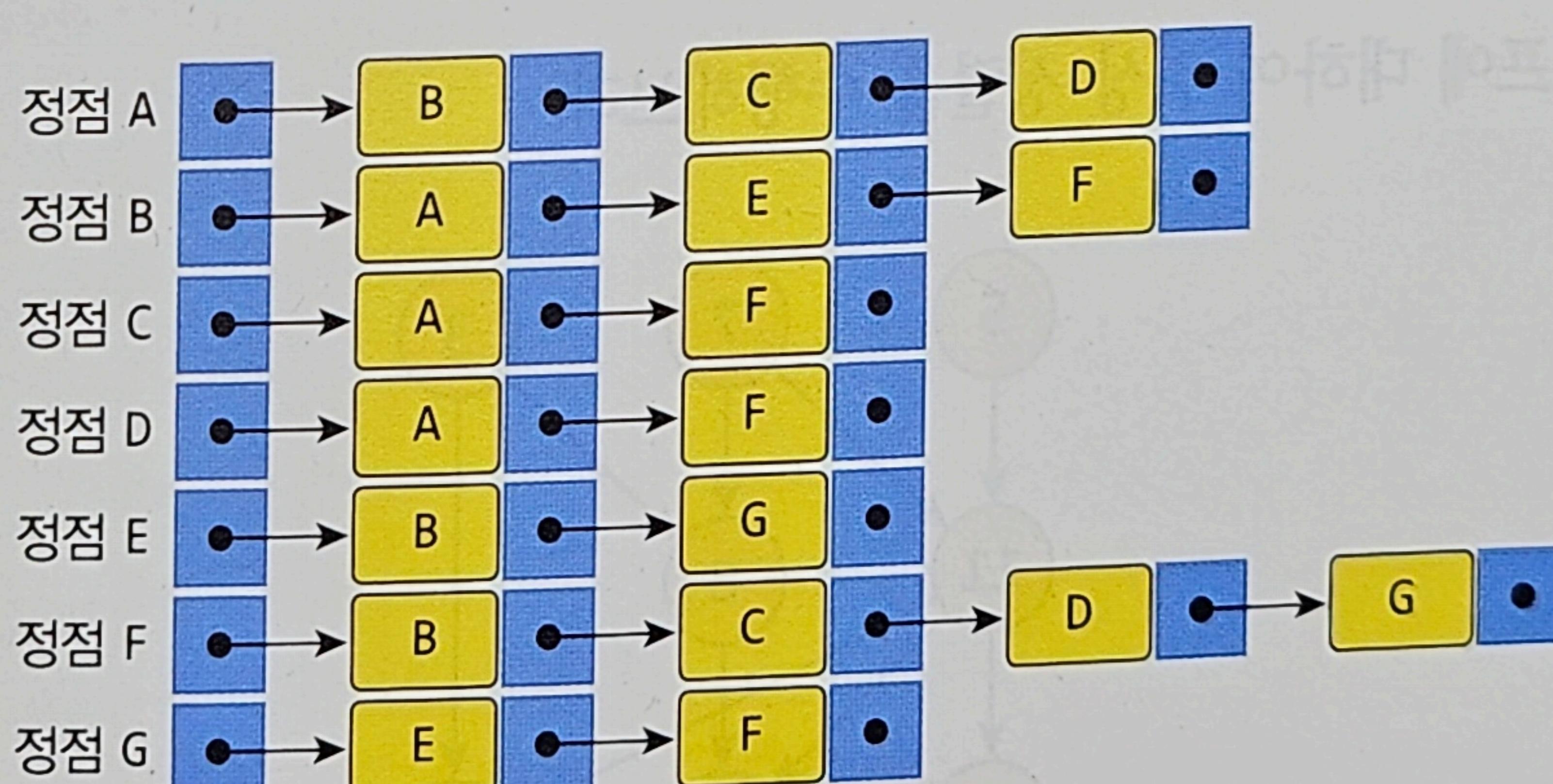
**10.6** 다음 그레프를 정점 S에서 출발하여 깊이우선탐색을 하는 경우의 방문순서를 표시하라.



**10.7** 위 문제의 그레프를 정점 S에서 출발하여 너비우선탐색을 하는 경우의 방문순서를 표시 하라.

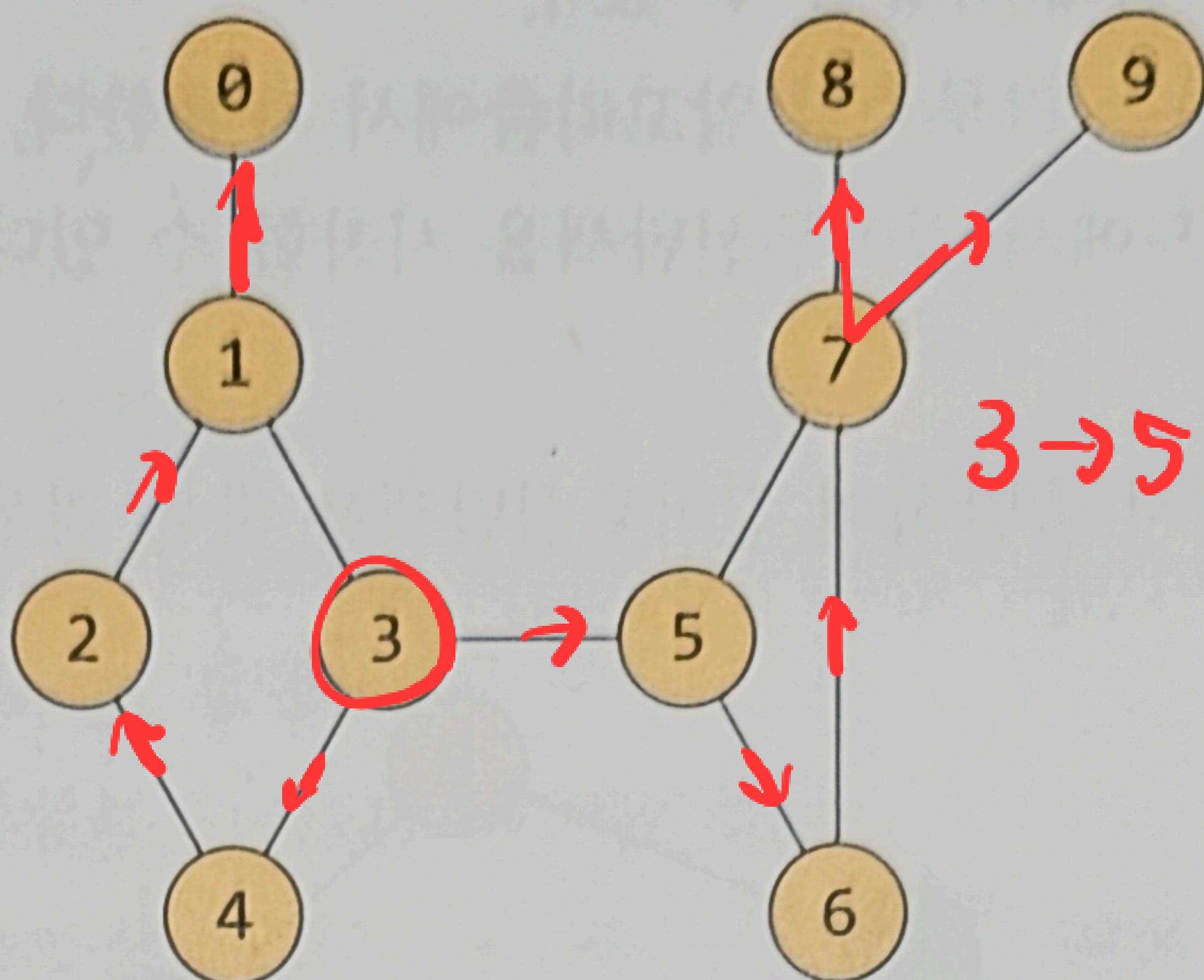
$S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$

**10.8** 다음은 어떤 그레프를 인접 리스트로 표현한 것이다. 이 그레프를 정점 A에서부터 깊이우선탐색할 때, 정점이 방문되는 순서로 옳은 것은? 단, 링크가  $\cdot$ 인 것은 None을 의미한다.



- ① A → B → C → D → F → G → E
- ② A → D → C → B → F → E → G
- ③ A → B → C → D → E → F → G
- ④ A → B → E → G → F → C → D

- 10.9 다음의 그래프가 인접 행렬로 표현되어 있다고 가정하고, 정점 3에서 출발하여 깊이우선탐색을 한 경우의 방문순서를 적어라.



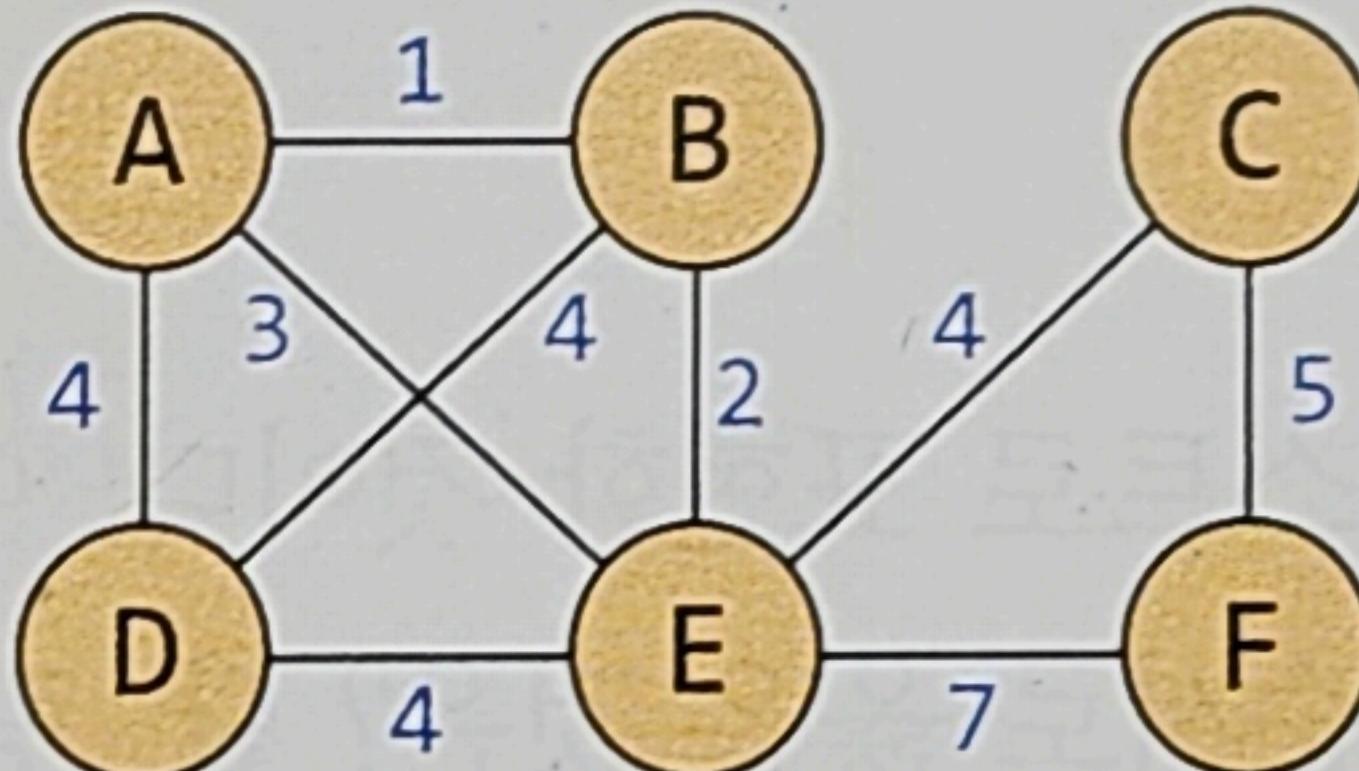
$3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 0$

- 10.10 위 문제에서 정점 3에서 출발하여 너비우선탐색을 한 경우의 방문순서를 적어라.

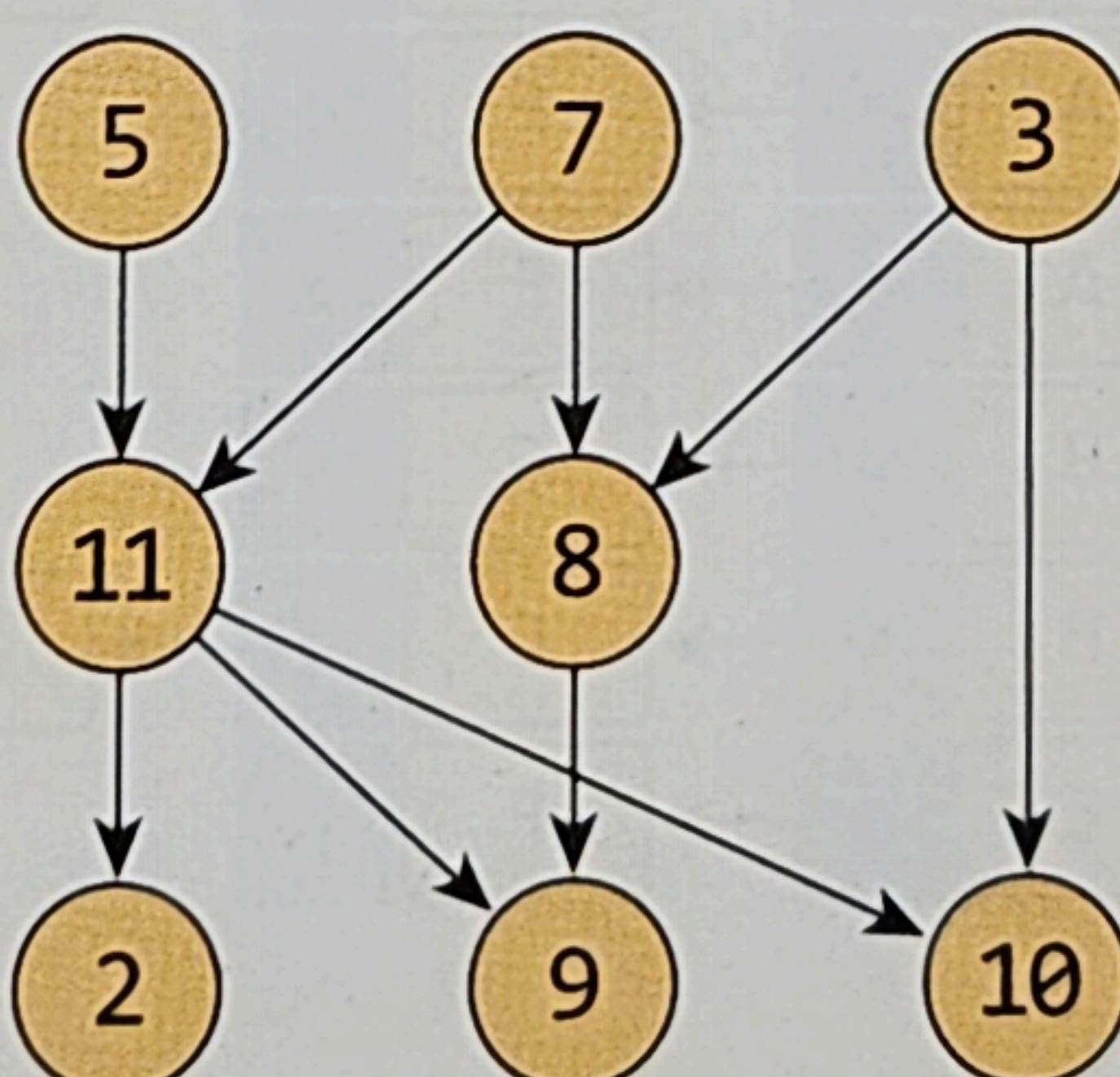
$3 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 8 \rightarrow 9$

- 10.11 다음의 그래프에서 가능한 신장 트리를 두 개 구해보라. 각 신장 트리의 간선의 가중치를 모두 더해보라.

$$16 + 16 = 32$$



- 10.12 다음의 그래프에 대하여 위상 정렬을 수행해보라.



$3 \rightarrow 5 \rightarrow 7 \rightarrow 11 \rightarrow 8 \rightarrow 2 \rightarrow 10 \rightarrow 9$

- 10.13\* 인접 리스트로 표현된 그래프에 대한 깊이우선탐색 함수를 구현하라. 알고리즘은 코드 10.1을 참고하고, 인접 리스트로는 파이썬 리스트, 큐로는 파이썬의 queue 모듈의 Queue 클래스를 사용하라.

~~10.14\*~~ 인접 행렬로 표현된 그래프에 대한 너비우선탐색 함수를 구현하라. 너비우선탐색으로 구현된 코드 10.3을 참고하라.

~~10.15\*\*~~ 코드 10.7의 연결 성분 검사에서는 너비우선탐색이 사용되었다. 깊이우선탐색을 이용하여 연결 성분 검사 알고리즘을 다시 구성하라.

~~10.16\*~~ 코드 10.10을 참고하여 너비우선탐색을 이용해 신장 트리를 구하는 함수를 구현하라.

~~10.17\*\*~~ 연결된 그래프의 간선들 중에서 그 간선을 제거하면 연결이 끊어지는 간선  $(u, v)$ 을 브리지(bridge)라고 한다. 주어진 그래프에서 브리지를 찾아 모두 출력하는 함수를 작성하라. 예를 들어, 다음의 그래프에서는 두 개의 브리지  $(B, C)$ 와  $(C, F)$ 를 찾아야 한다.

