

# Algorithm

<http://smartlead.hallym.ac.kr>

**Instructor: Jin Kim**  
**010-6267-8189(033-248-2318)**  
**[jinkim@hallym.ac.kr](mailto:jinkim@hallym.ac.kr)**

**Office Hours:**



# 중간과제 모음

**<http://smartlead.hallym.ac.kr>**

**Instructor: Jin Kim**  
**010-6267-8189(033-248-2318)**  
**[jinkim@hallym.ac.kr](mailto:jinkim@hallym.ac.kr)**

**Office Hours:**



답: 2번

## 문제 1

다음 이진 트리에 관한 설명 중 바르게 기술된 사항은  
트리의 레벨은 0으로 간주

1) 깊이가  $k$ 인 이진 트리에서의 최대 노드 수는  $2^k - 1$ 이다.

루트의 레벨 0 :  $2^{k+1} - 1$

2) 이진 트리에서 레벨의  $i$ 의 최대 노드 수는  $2^i$ 이다.

루트의 레벨 0 :  $2^i$

3) 이진 트리  $T$ 에 대해서  $n_0$ 는 단말 노드 수,  $n_2$ 는 차수가 2인 노드 수라고 하면  $n_0 = n_2 - 1$ 이다.

$n_0 = n_2 - 1$ 이 아니고  $n_0 = n_2 + 1$

4) 노드가  $n$ 개, 차수가  $k$ 인 트리에서 널 링크의 수는  $n(k-1)$ 이다

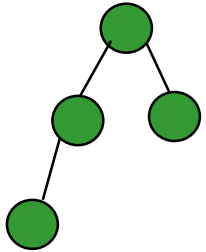
$n(k-1)$ 이 아니고  $n(k-1) + 1$

# 문제 1

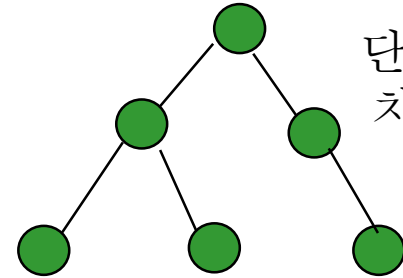
## ◆ 풀이

1) 깊이가  $k$ 인 이진 트리의 최대노드 수는  $2^{k+1}-1$ 이다

3)  $n_0$ (리프의 수) =  $n_2$ (차수 2인 노드의 수) + 1이다.



단말노드 수 2 = 차수가 2인 노드의 수 (1) + 1



단말 노드의 수 3 =  
차수가 2인 노드의 수(2)+1

4) 전체 링크 수 - 널이 아닌 링크수 =  $nk - (n - 1)$

널 링크의 개수는  $n(k-1)+1$  이다.



답: 2

## 문제 2

노드의 좌우 순서가 중요한 의미를 갖는 순서 트리에 해당하는 것은?

1) 일반 트리 2)이진 트리 3)삼진 트리 4)사진 트리

이진 트리는 자식의 순서를 구분하므로 이진 트리는 순서 트리에 해당한다.

## 문제 3

답: 3번 15

이진트리에서 루트의 레벨이 1인 경우, 깊이가 4인 포화 이진 트리에 속하는 노드의 개수는?

- 1) 7
- 2) 8
- 3) 15
- 4) 16

루트의 레벨이 1인 경우:  $2^k - 1$

## 문제 3

◆ 답 3

◆ 풀이

깊이가 4인 포화 이진 트리 노드의 수를 구하려면 이진트리의 최대 노드의 수 공식을 사용( $2^{k+1}-1$ : 루트의 레벨이 0인 경우)  
루트의 레벨이 1 깊이가 4 따라서 노드의 수는 15개이다.

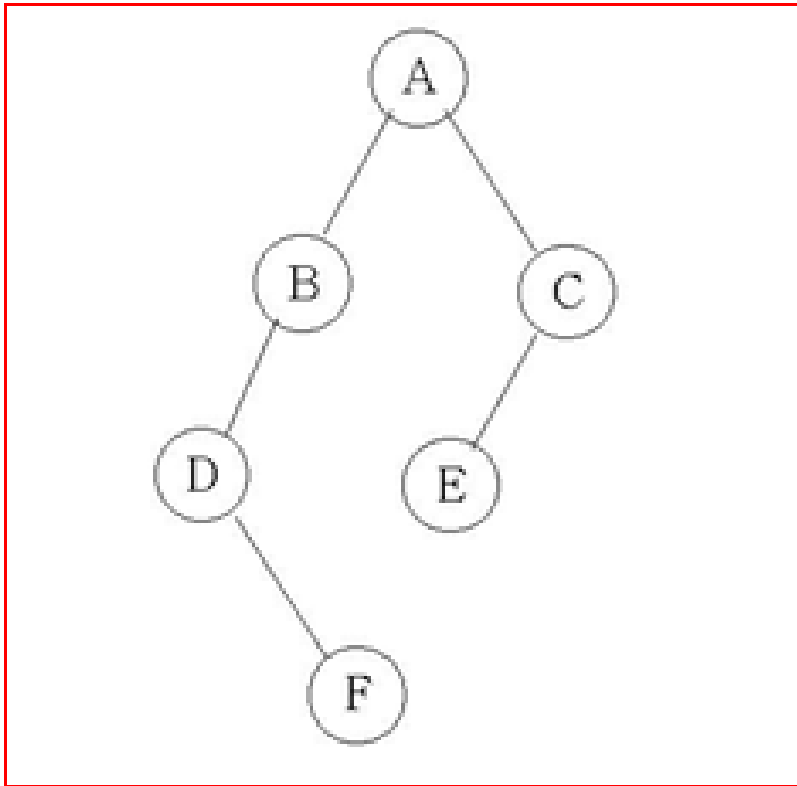


## 문제 4

다음 이진 트리의 중위 순회 결과를 바르게 기술한 항은?

중위 순회: left – root – right

답: D F B A E C





## 문제 5

◆ 답: 1

평행한 간선을 포함하는 그래프를 무엇이라고 하는가?

- 1) 다중그래프
- 2) 널 그래프
- 3) 가중 그래프
- 4) 단순 그래프

평행한 간선: 시작점과 끝점이 같은 간선

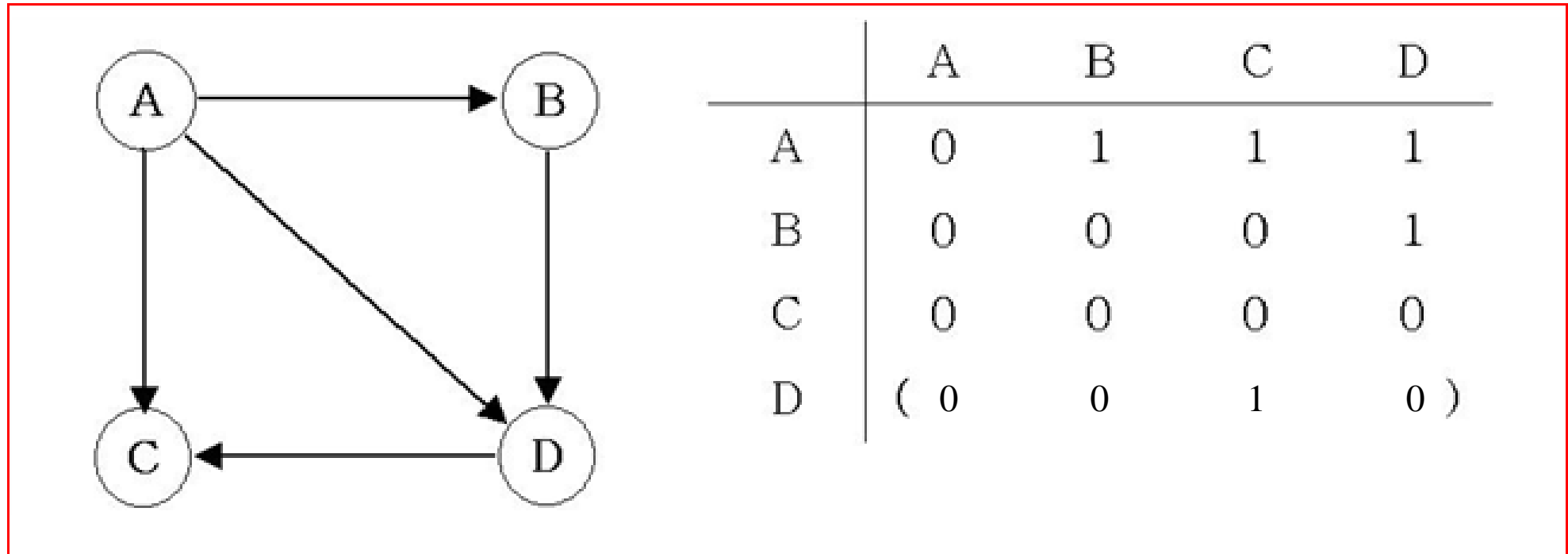
다중그래프는 정점과 정점사이에 여러개의 간선이 존재할 수 있다



## 문제 6

◆ 답: 0 0 1 0

다음 그래프를 인접행렬로 표현하라



정점D의 .간선을 보면 C로 나가는것 1개 이므로



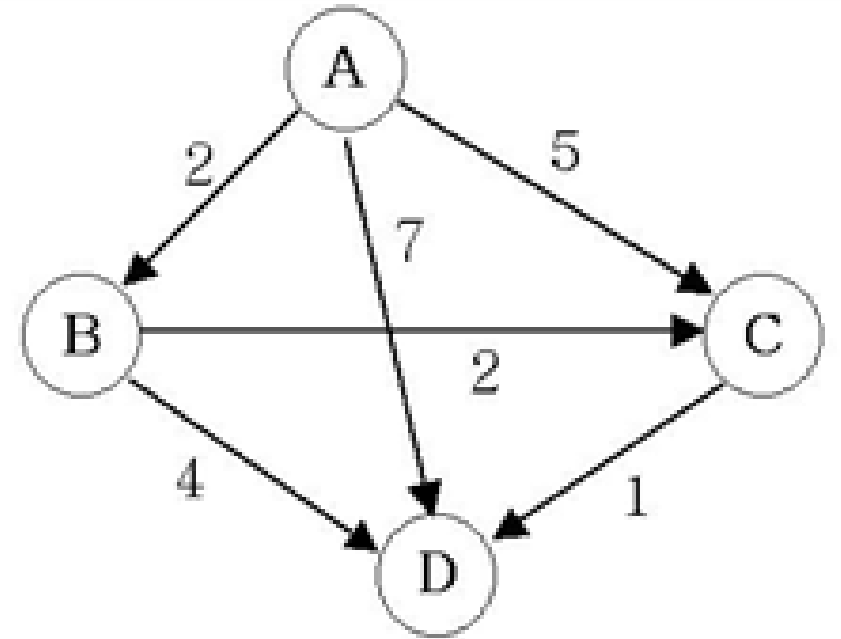
## 문제 7

◆ 답: 2

위 그래프에서 정점 C의  
진입 차수는 얼마인가?

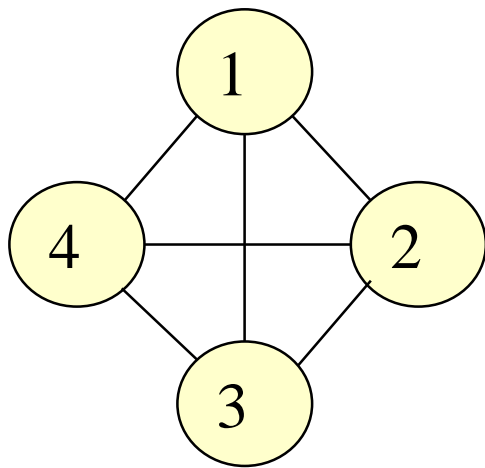
진입차수: 외부 노드에서 들어오는 간선의 수

정점 C로 들어오는 간선은 2개  
그러므로 진입 차수는 2

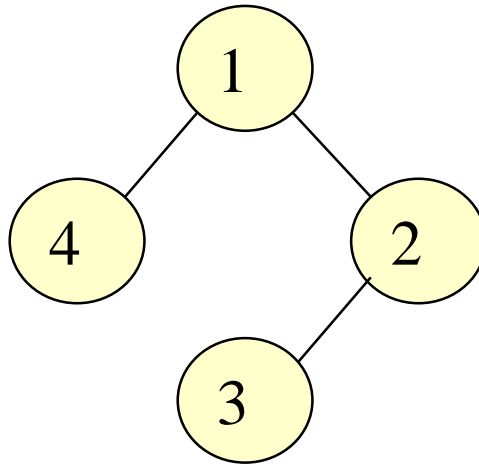


## 문제 8

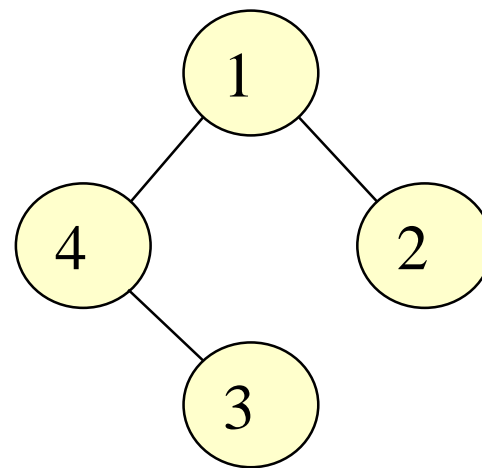
- ◆ 그래프의 간선이 모든 정점으로 구성된 트리를 무엇이라 하는가?
- ◆ 1) 완전 트리 2) 탐색 트리 3) 신장 트리 4) 선택 트리
- ◆ 답: 3) 신장 트리 :어떤 그래프의 부분 그래프
- ◆ 이유: 그래프를 통해서 그래프의 간선이 모든 정점으로 구성된 트리를 만들려면 아래의 그림처럼 만들어야 합니다. 그리고 그것을 우리는 신장 트리라고 합니다
- ◆ 신장 트리의 조건은 그래프의 모든 정점이 간선을 통해 이어 있어야 하고, 그래프가 사이클을 형성하지 않아야 합니다.



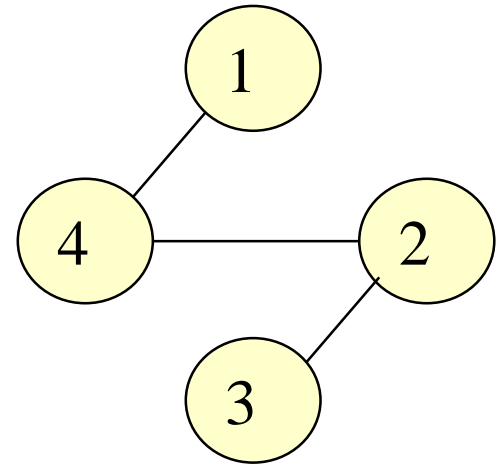
그래프



신장 트리1



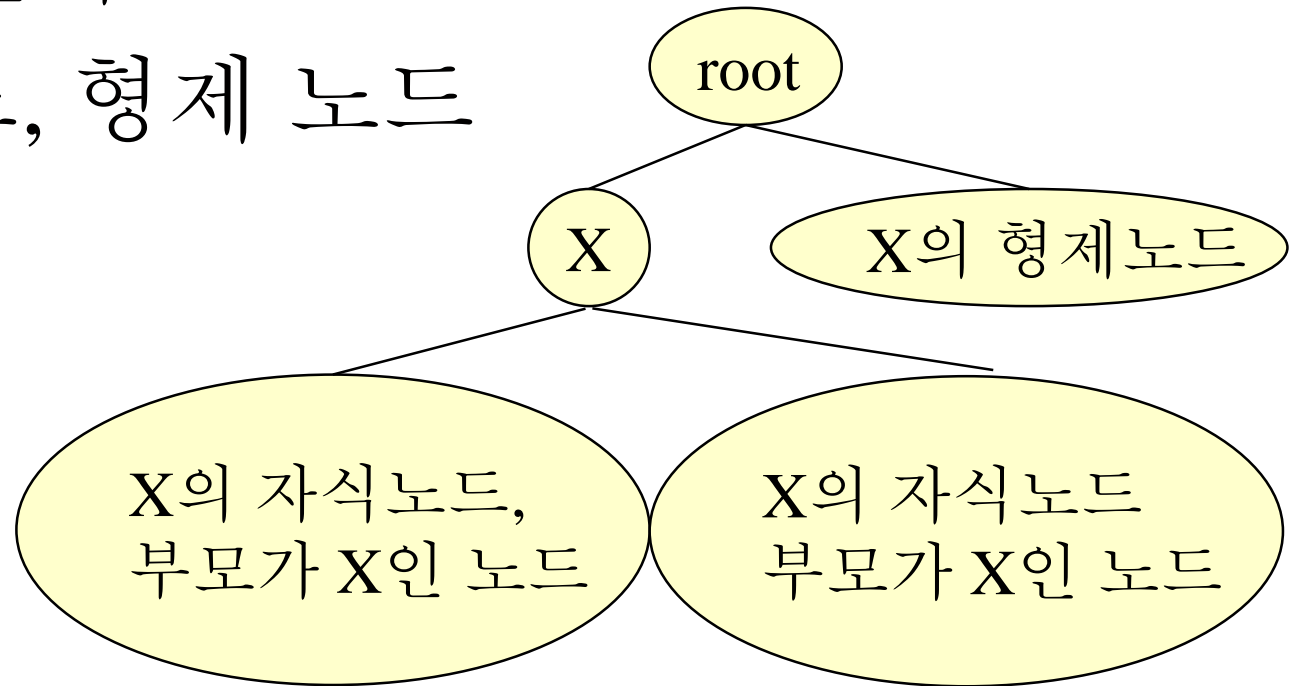
신장 트리2



신장 트리3

## 문제 9

- ◆ 어떤 노드  $X$ 의 서브트리들  $X$ 의(     노드)라 하고,  $X$ 는 이 자식의 (     노드)라 한다. 동일한 부모의 자식들은(     노드)라 한다.
- ◆ 답: 자식 노드, 부모 노드, 형제 노드



# 문제 10

## ◆ 답

깊이가  $k$ 이고, 노드 수가  $n$ 인 이진 트리가 만일 이 트리의 각 노드들이 깊이  $k$ 인 포화 이진 트리에서 1부터  $n$ 까지의 번호를 붙인 노드들과 1대 1로 일치하면, 이 트리는 ( 완전 이진 )트리이다.

포화이진트리는 모든 노드가 왼쪽과 오른쪽 자식으로 꽉 차있는 트리이기 때문에 깊이가 같고 노드들과 1:1로 일치하면 그 트리는 완전 이진 트리이다.  
(모든 포화 이진 트리는 완전 이진 트리이지만, 완전 이진 트리가 반드시 포화이진트리인 것은 아니다.)



# 문제 11

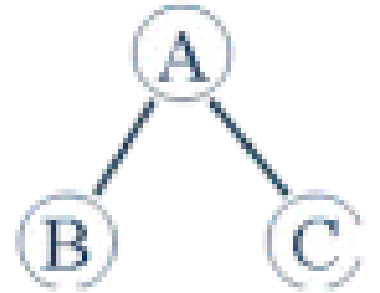
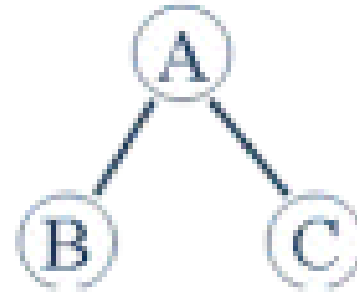
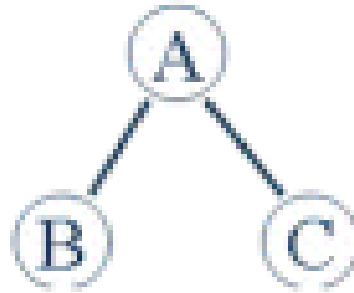
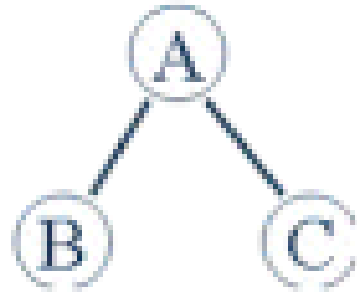
다음 이진 트리를 전위, 중위, 후위 순회한 결과를 적어라.

## ◆ 답

◆ 전위: **A B C**

◆ 중위: **B A C**

◆ 후위: **B C A**



전위 – 루트를 먼저 방문하고 재귀로 왼쪽 자식 노드와 오른쪽 자식 노드를 전위 순회한다.

중위 – 재귀로 왼쪽 자식 노드를 순회하고 루트를 순회한 후 오른쪽 자식 노드를 순회한다.

후위 – 재귀로 왼쪽 자식 노드를 순회하고 오른쪽 자식 노드를 순회한 후 노드로 순회한다.



# 문제 12

다음 중위순회 알고리즘에서 ( )를 채워라.

◆ 답

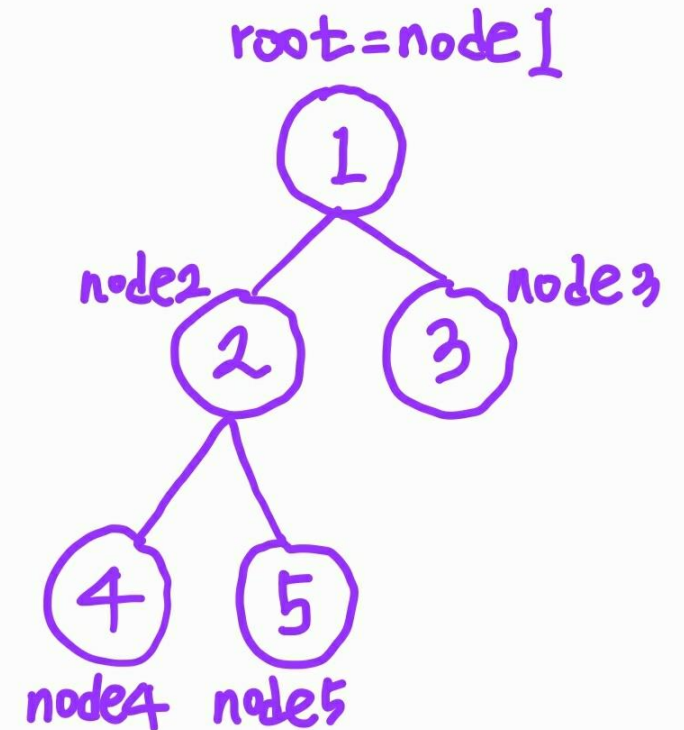
```
void INORDER(tree_pointer T){  
    if (T){  
        INORDER( T.left );  
        printf("%d", T->DATA);  
        INORDER( T.right );  
    }  
}
```

```
class tree_pointer{  
    int key;  
    tree_pointer left, right;  
    public tree_pointer(int item) {  
        key=item;  
        left=null;  
        right=null;  
    }  
}
```

```
class BinaryTree1{  
    tree_pointer root;  
    public BinaryTree1() {  
        root=null;  
    }  
    void INORDER(tree_pointer node) {  
        if(node==null) {return;}  
        INORDER(node.left);  
        System.out.print(" "+node.key);  
        INORDER(node.right);  
    }  
}
```

inorder  
4 2 5 1 3

```
public static void main(String[] args) {  
    BinaryTree1 tree = new BinaryTree1();  
  
    tree_pointer node1=new tree_pointer(1);  
    tree_pointer node2=new tree_pointer(2);  
    tree_pointer node3=new tree_pointer(3);  
    tree_pointer node4=new tree_pointer(4);  
    tree_pointer node5=new tree_pointer(5);  
  
    node2.left=node4; node2.right=node5;  
    node1.left=node2; node1.right=node3;  
    tree.root=node1;  
  
    System.out.println("inorder");  
    tree.INORDER(tree.root);  
}
```





# 문제 13

## ◆ 답

힙의 정의에서 “최대 트리는 각 노드의 키 값이(자식이 있다면)  
그 자식의 키 값보다 작지 않은 트리이다.  
( 최대 힙 )는 최대 트리인 ( 완전 이진 트리 )이다.

**완전 이진 트리**는 마지막 레벨을 제외하고 모든 레벨이 완전히 채워져 있으며, 마지막 레벨의 모든 노드는 가능한 한 가장 왼쪽에 있다.

**힙(Heap)**는 완전 이진 트리 기반으로 루트에 최소 값 혹은 최대 값이 위치하는 자료구조이다.  
루트에 최대 값이 오면 최대 힙, 최소 값이 오면 최소 힙이라고 부른다.

위 정의에 따라 최대 트리인 완전 이진 트리는 최대 힙이라고 말할 수 있다.



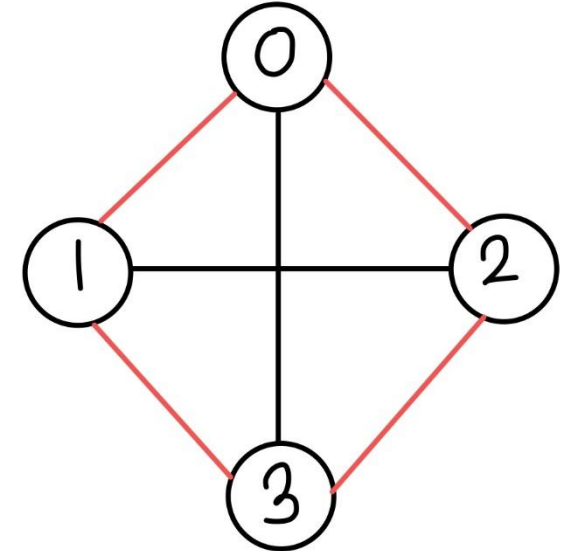
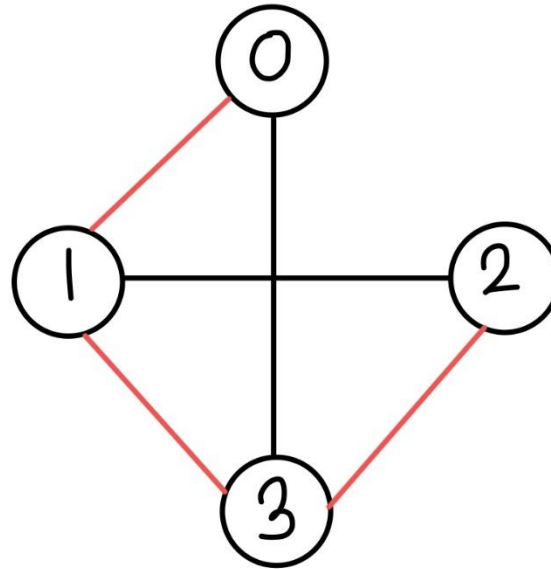
## 문제 14

### ◆ 답

한 경로상에 있는 모든 정점들이 서로 다를 때, 그 경로를 ( 단순 경로 )라 하고,  
또한 처음 정점과 마지막 정점이 같은 경로를 ( 사이클 )라 한다.

단순 경로는 경로의 처음과 끝이 다르고 중간에 중복 정점이 없다.

사이클은 경로의 처음과 끝이 같다.



## 문제 15

**Q.** 무방향 그래프에서 최대 간선의 수?

**A.**  $N$  : 정점의 개수라고 하면,

무방향 그래프에서 최대 간선의 수 =  $(N - 1) + (N - 2) + \dots + 2 + 1$

이 때,  $(N - 1) + (N - 2) + \dots + 2 + 1 = N * (N - 1) / 2$

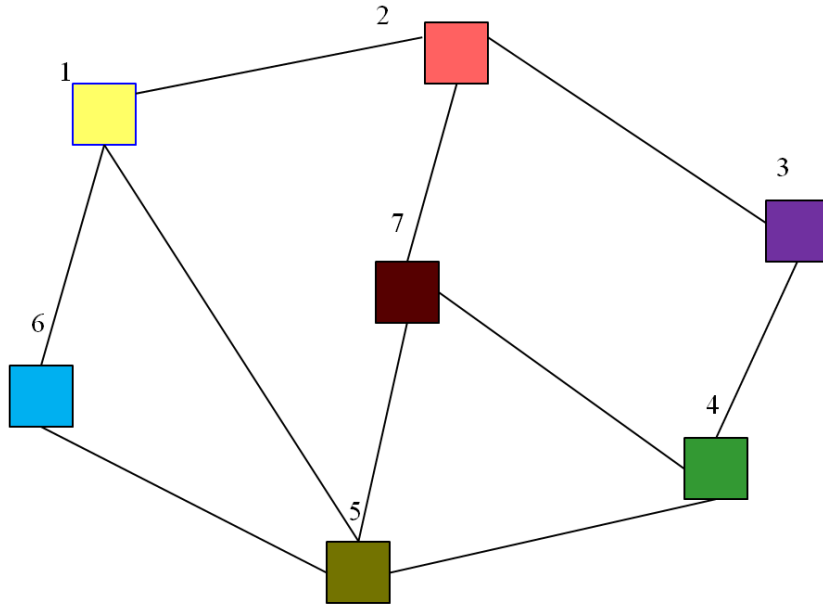
따라서, 무방향 그래프에서 최대 간선의 수 =  $N * (N - 1) / 2$  개



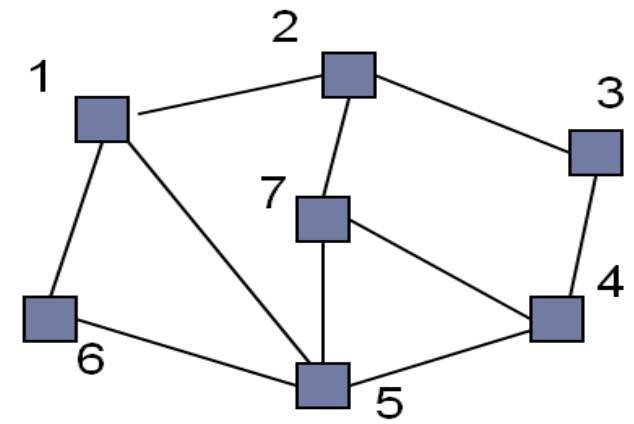
# 문제 16

Q. 다음 그래프를 인접 리스트로 표현하라. 인접 행렬로 표현하라.

A.



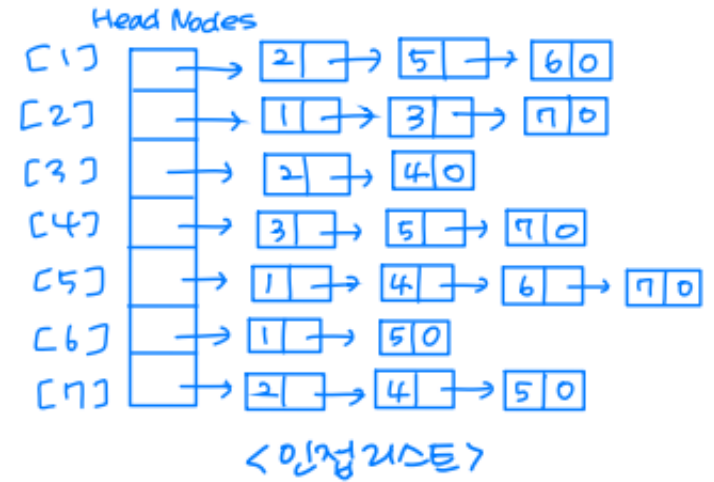
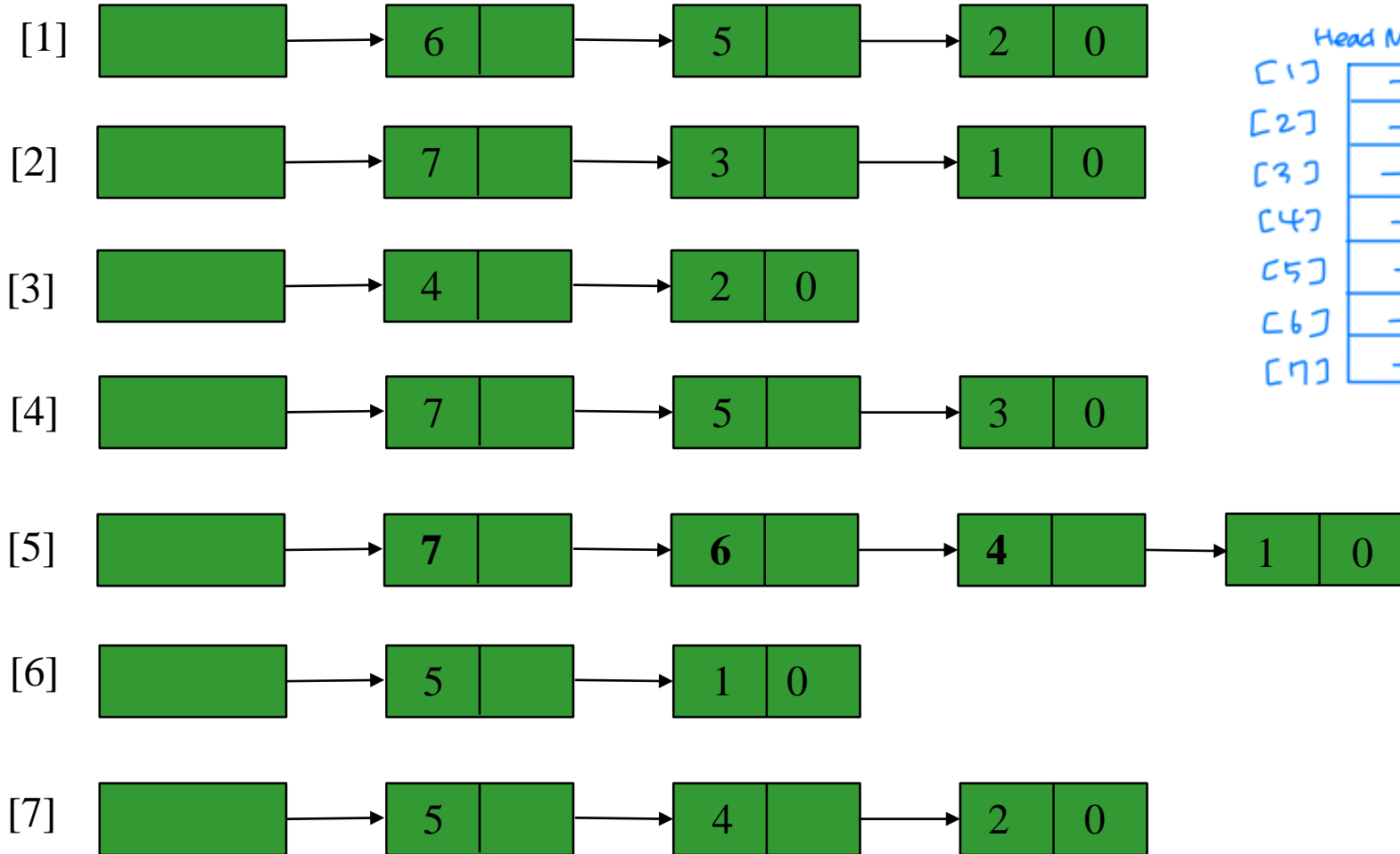
	1	2	3	4	5	6	7
1	0	1	0	0	1	1	0
2	1	0	1	0	0	0	1
3	0	1	0	1	0	0	0
4	0	0	1	0	1	0	1
5	1	0	0	1	0	1	1
6	1	0	0	0	1	0	0
7	0	1	0	1	1	0	0



# 문제 16( 두 개 모두 정답)

A.

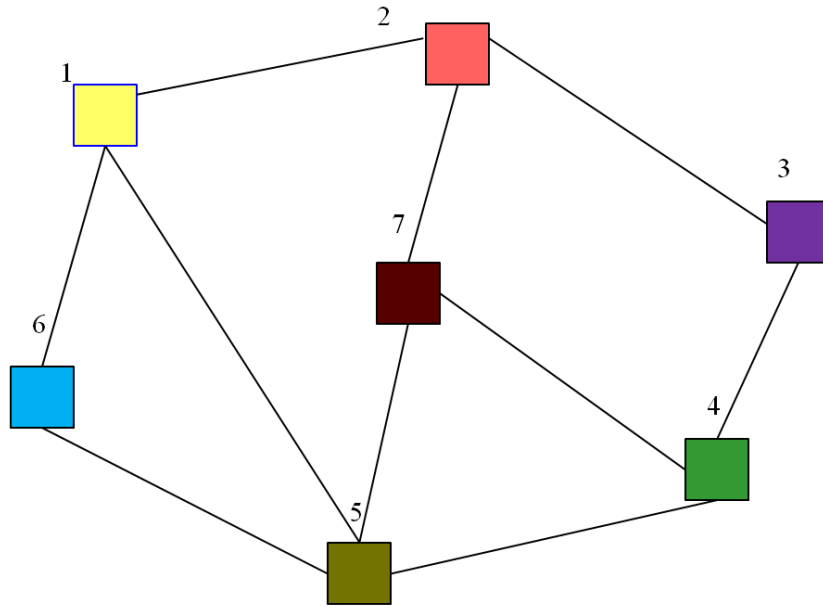
Head Nord



## 문제 17

Q. 16번 그래프를 정의하라.

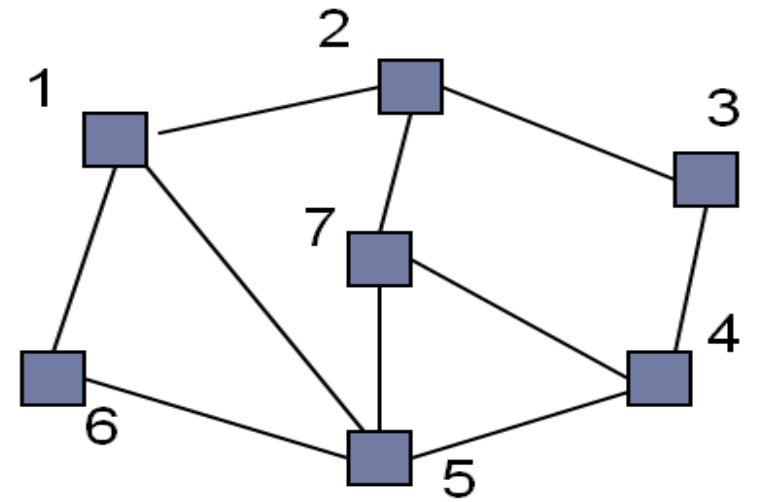
A.



위의 그래프를  $G_1$ 라 하자.

$G_1 = \{V, E\}$

$V(G_1) = \{1, 2, 3, 4, 5, 6, 7\}$ ,  $E(G_1) = \{(1, 2), (1, 5), (1, 6), (2, 3), (2, 7), (3, 4), (4, 5), (4, 7), (5, 6)\}$



## 문제 18

**Q. 다음 중 선형 자료구조가 아닌 것은?** ( 가. 리스트 나. 스택 다. 큐 라. 트리 )

**A. 라. 트리**

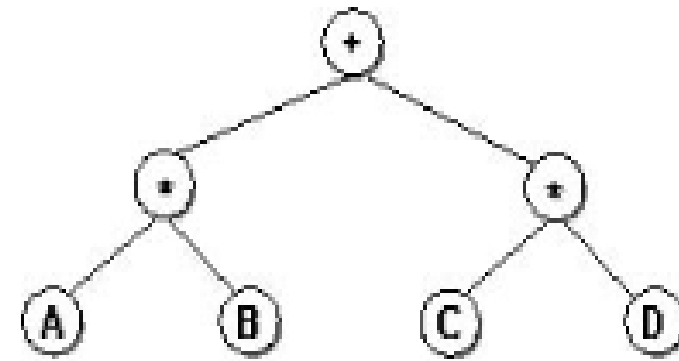
리스트, 스택, 큐는 데이터가 일직선상에 위치하며 선행자와 후행자가 있는 선형 자료구조이다.  
하지만 트리는 데이터가 선형으로 위치하지 않는, 계층 구조적인 비선형자료이다.



## 문제 19

Q. 다음의 수식 이진 트리를 후위 순회하여 수식에 대한 후위 표기식을 구하여라.

A.



후위 순회: LRP

순회 경로: + -> \* -> A(왼쪽 자식, 오른쪽 자식 없으므로 print, 다시 위로 올라감) -> \* -> B(왼쪽 자식, 오른쪽 자식 없으므로 print, 다시 위로 올라감) -> \*(왼쪽 자식, 오른쪽 자식 출력했으므로 print, 다시 위로 올라감) -> + -> \* -> C(print) -> \* -> D(print) -> \*(print) -> +(print)

후위 표기식: AB\*CD\*+ (답만 적으면 된다)





## 문제 20

**Q.** 다음 중 같은 개수의 노드를 트리로 저장하는 경우에 트리의 높이가 가장 높은 트리는 무엇인가?

( 가. 이진 트리   나. 포화 이진 트리   다. 완전 이진 트리   라. 편향 이진 트리 )

**A.** 라. 편향 이진 트리

편향 이진 트리는 자식 노드가 1개씩 존재하기 때문에 다른 트리가 같은 노드의 수를 가진다 하더라도 가장 높이가 높은 이진 트리가 된다.



## 문제 21

**Q.** 20개의 노드로 구성된 이진 트리에서 간선의 개수와 가능한 최대 높이와 최소 높이를 구하여라.

**A.** 간선의 개수 = 노드의 수 - 1 =  $20 - 1 = 19$

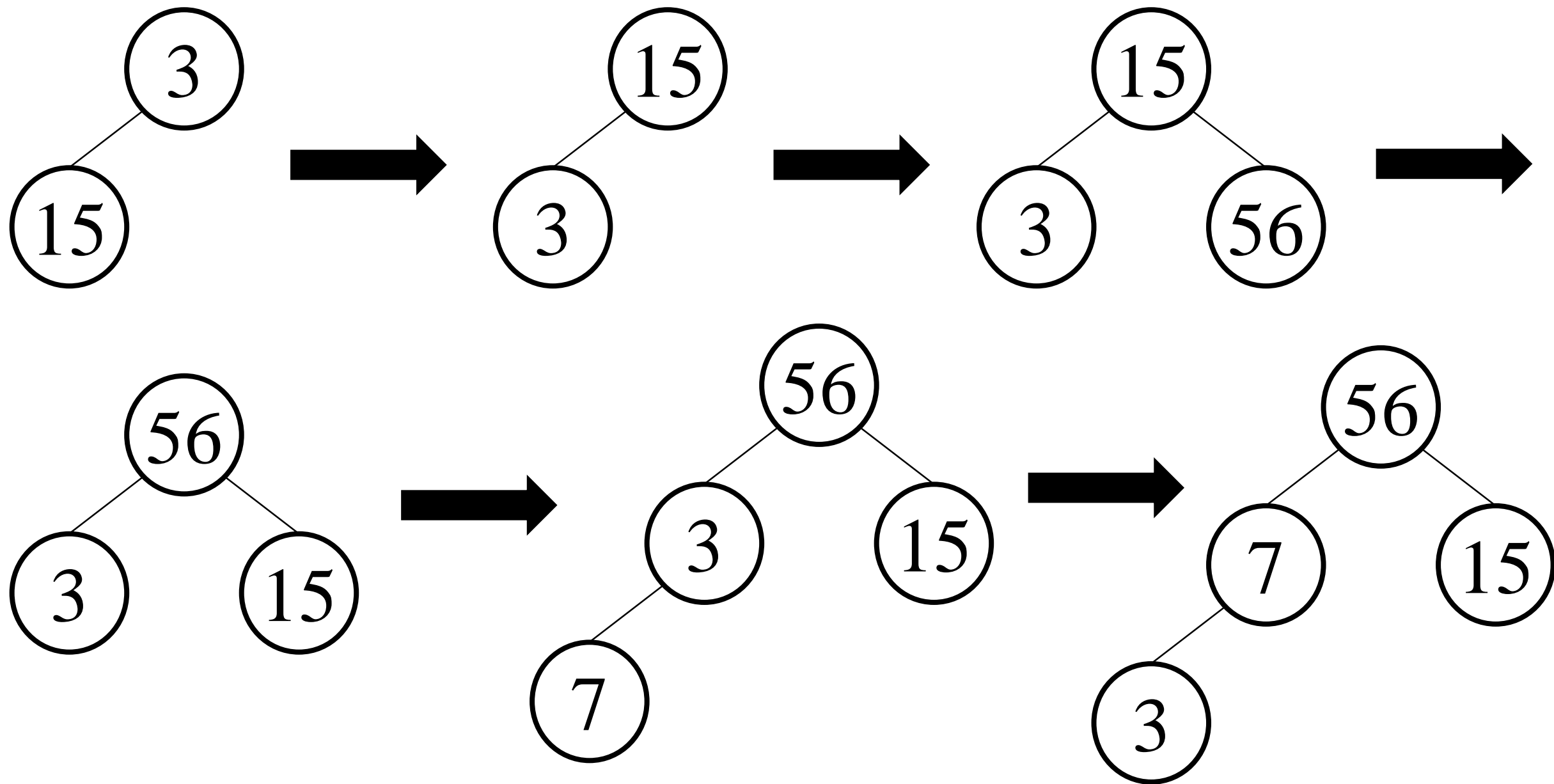
최대 높이는 노드의 수가 구해져 있기 때문에 최소 노드의 개수 구하는 방법에서 구할 수 있다.

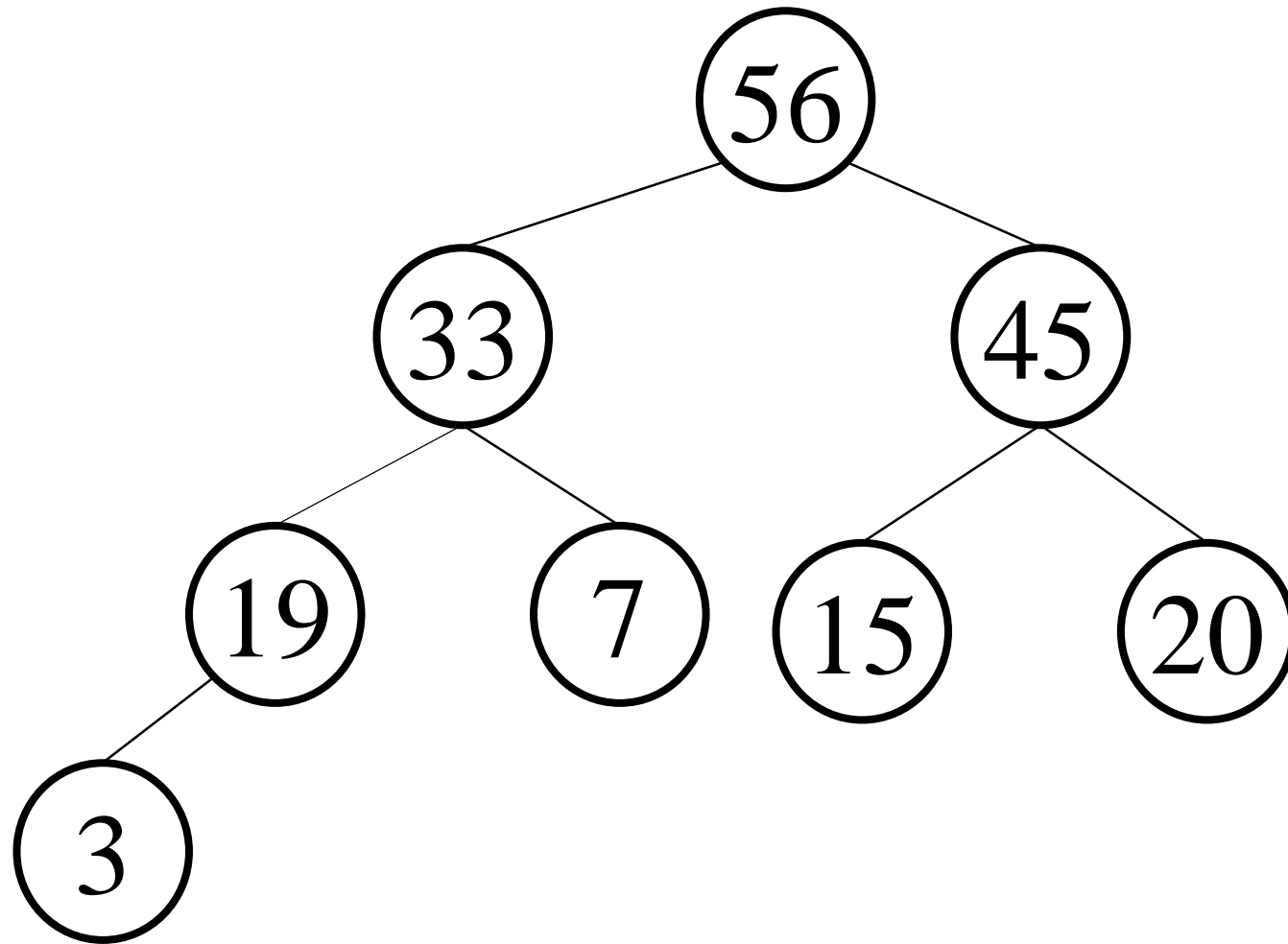
최소 노드의 수 =  $H - 1 = 20$  따라서 최대 높이는 19

최소 높이도 노드의 수가 정해져 있기 때문에 최대 노드의 개수 구하는 방법에서 구할 수 있다.

최대 노드의 수 =  $2^{h+1} - 1 = 20$ ,  $20 = 2^4 + 5$ , 높이가 3인 포화 이진 트리 + 단말 노드 5 인 트리가 된다. 따라서 최대 높이는 4







22번 답

## 문제 23

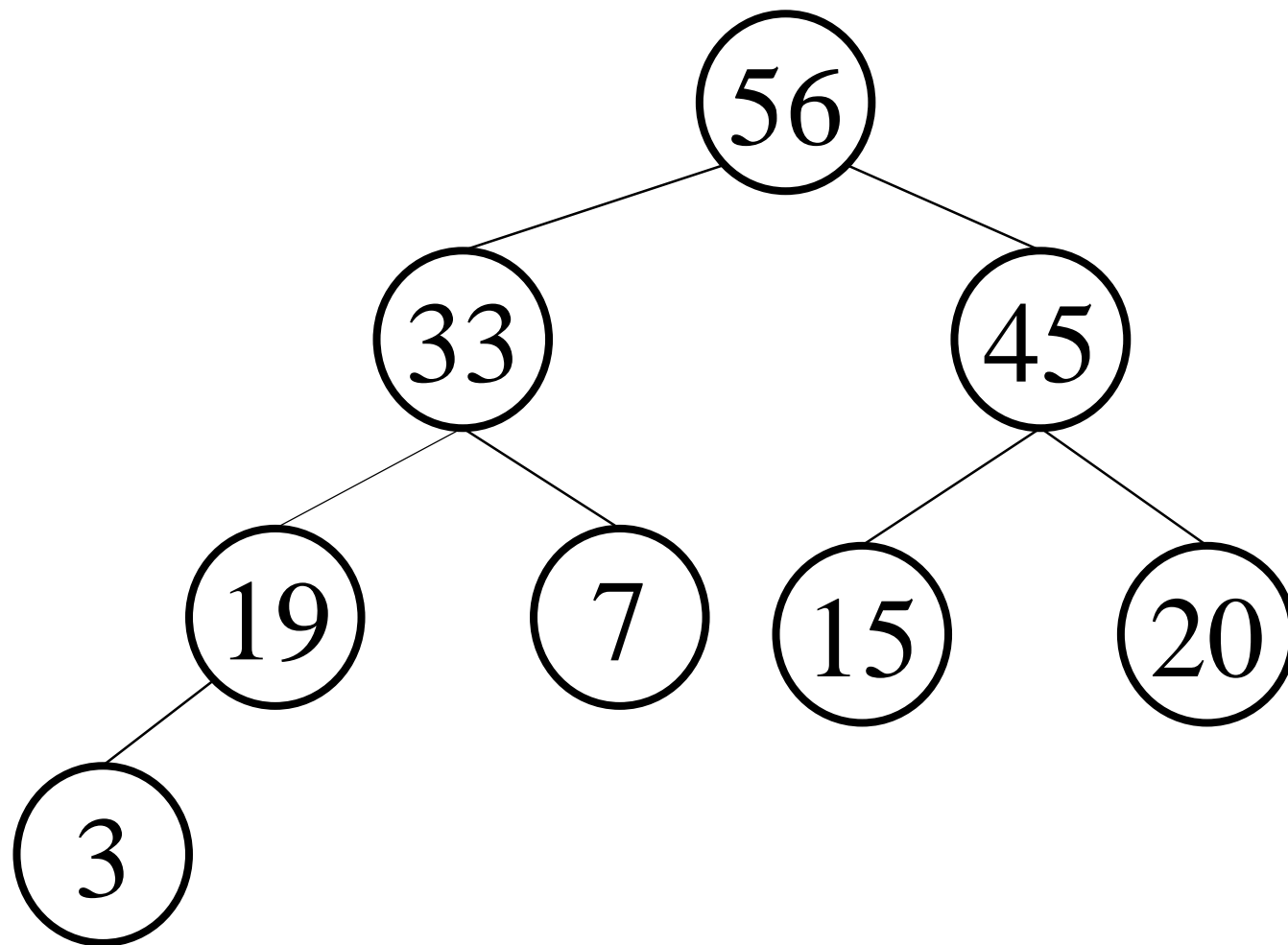
- ◆ 위의 최대 힙에 대해서 삭제연산을 수행하여 원소가 삭제되는 순서를 써라.

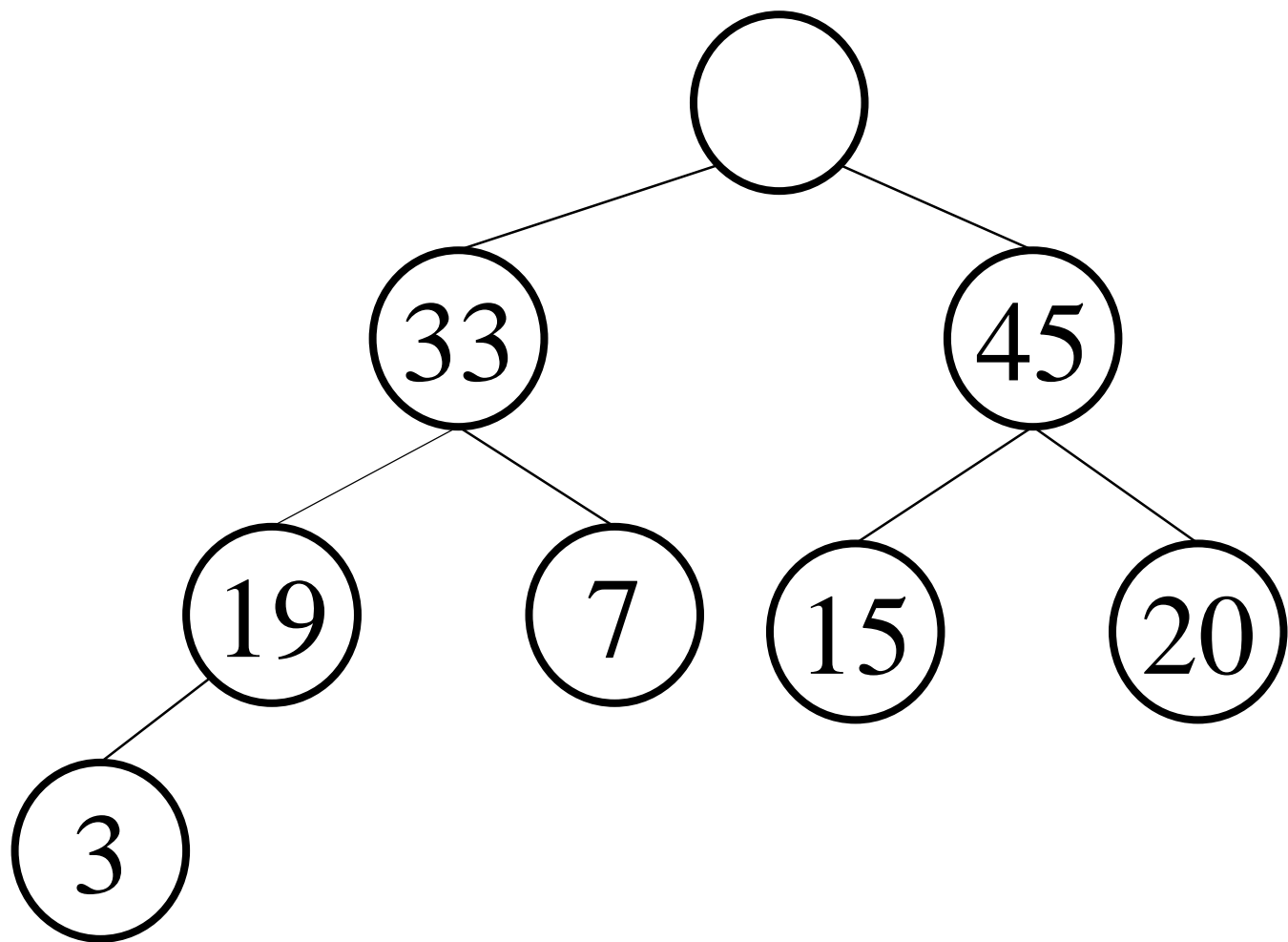
답 :  $56 - 45 - 33 - 20 - 19 - 15 - 7 - 3$

최대힙 = 최대트리 = 완전이진트리

순서

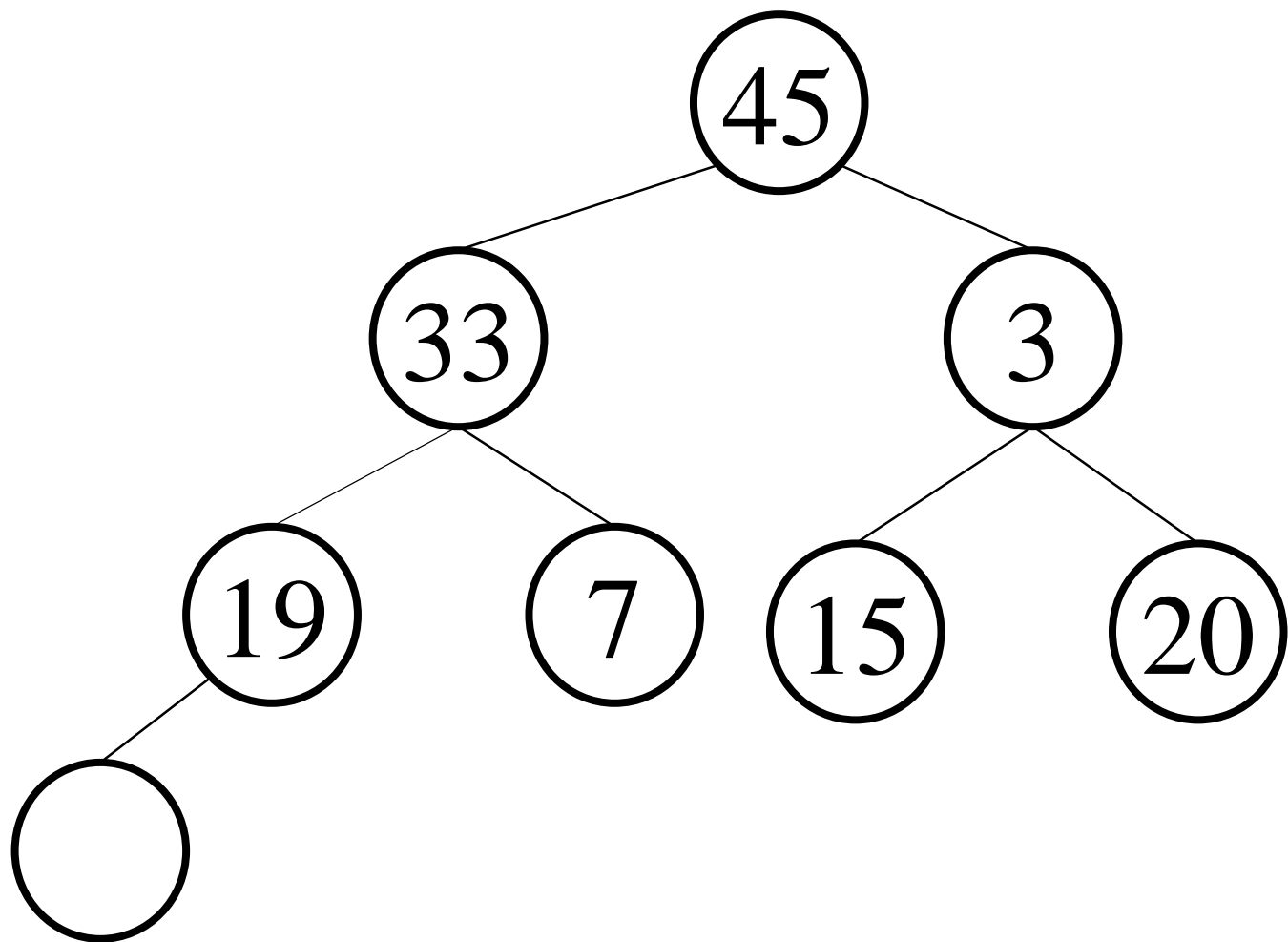
1. 루트노드 삭제
2. 최후미 단말노드를 루트노드에 올린 후
3. 자식노드들과 비교하며 가장 큰 노드를 루트노드에 가까운 자리로 옮긴다.





삭제된 원소

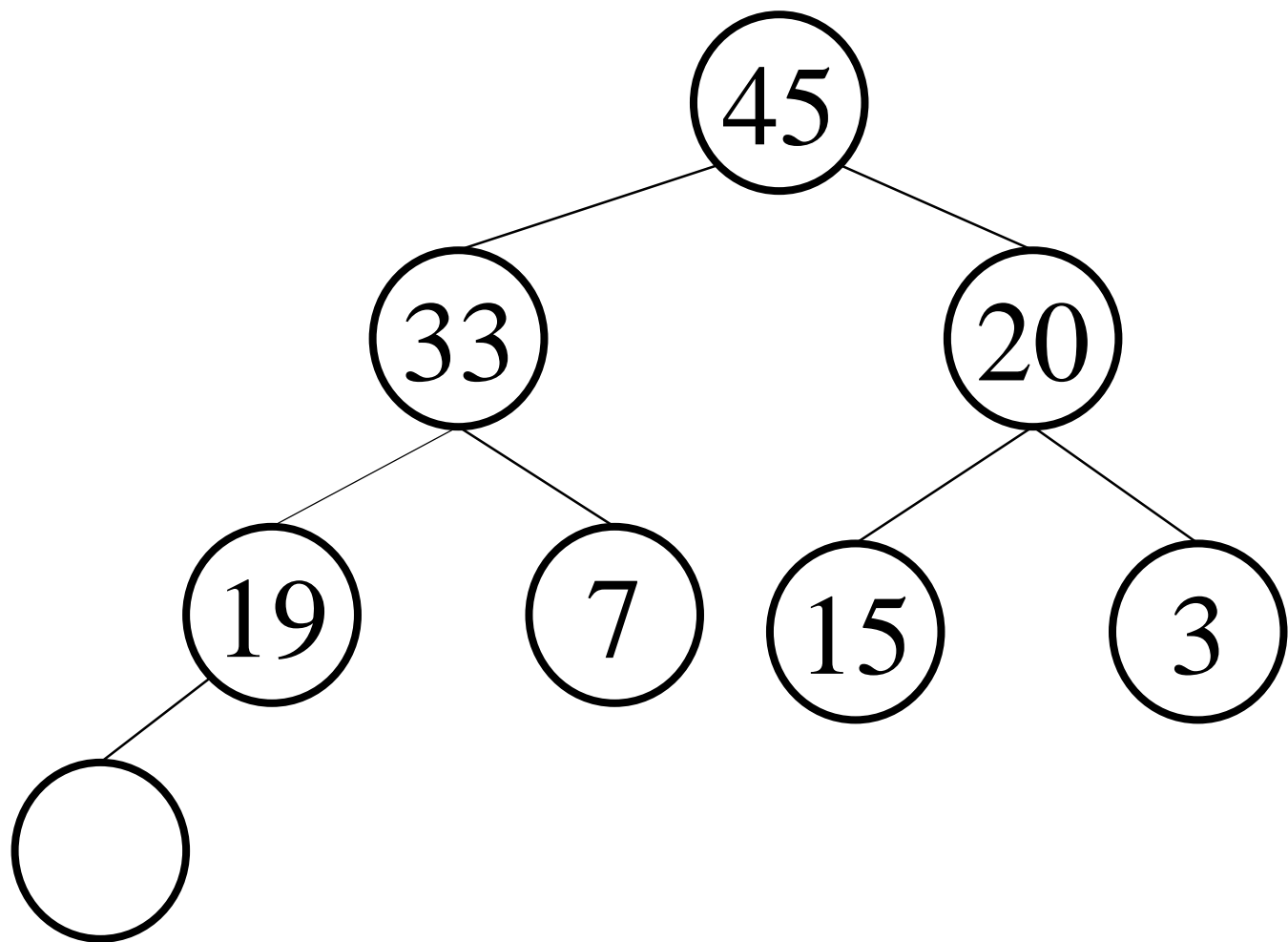
56



삭제된 원소

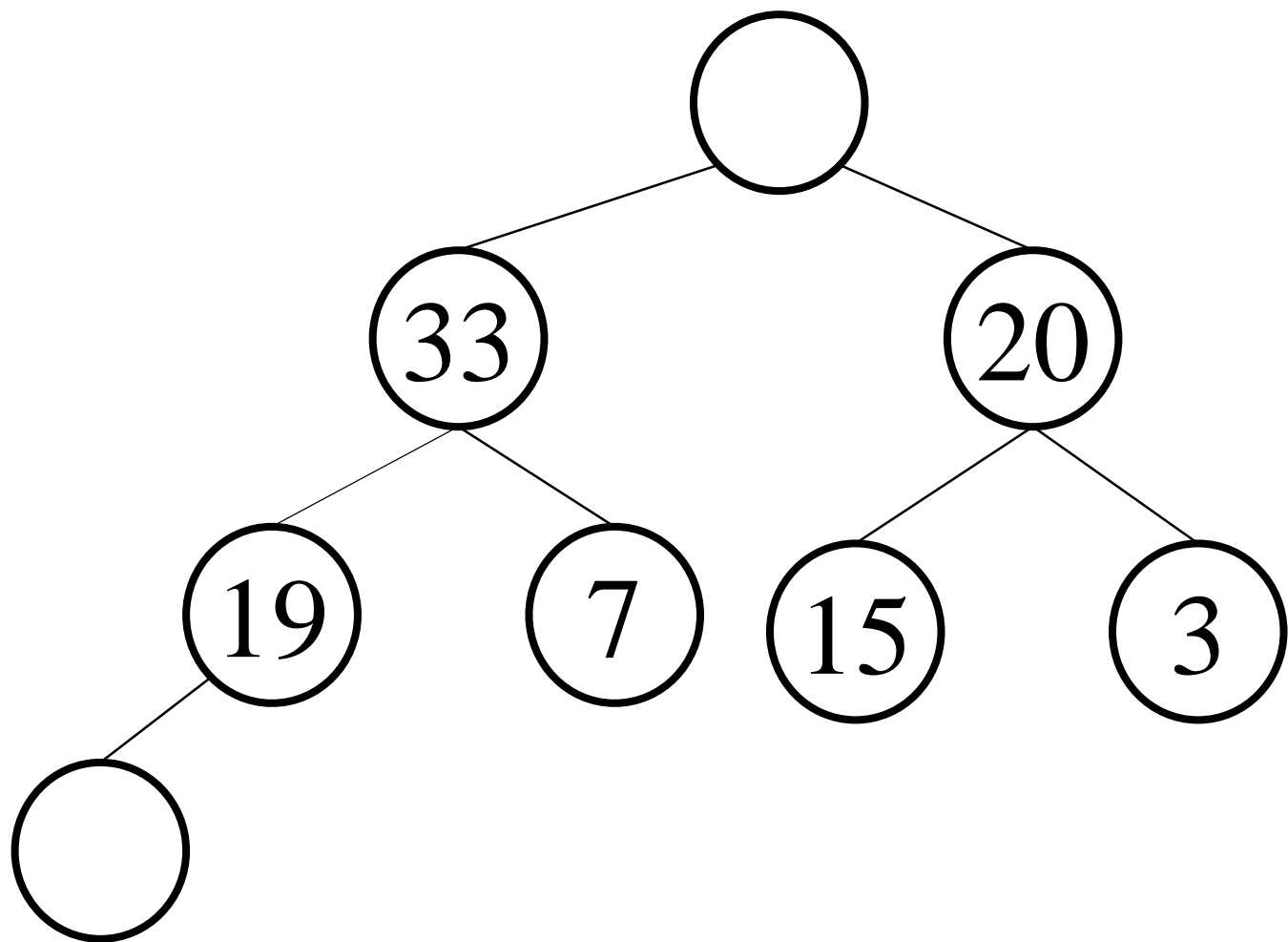
56





삭제된 원소

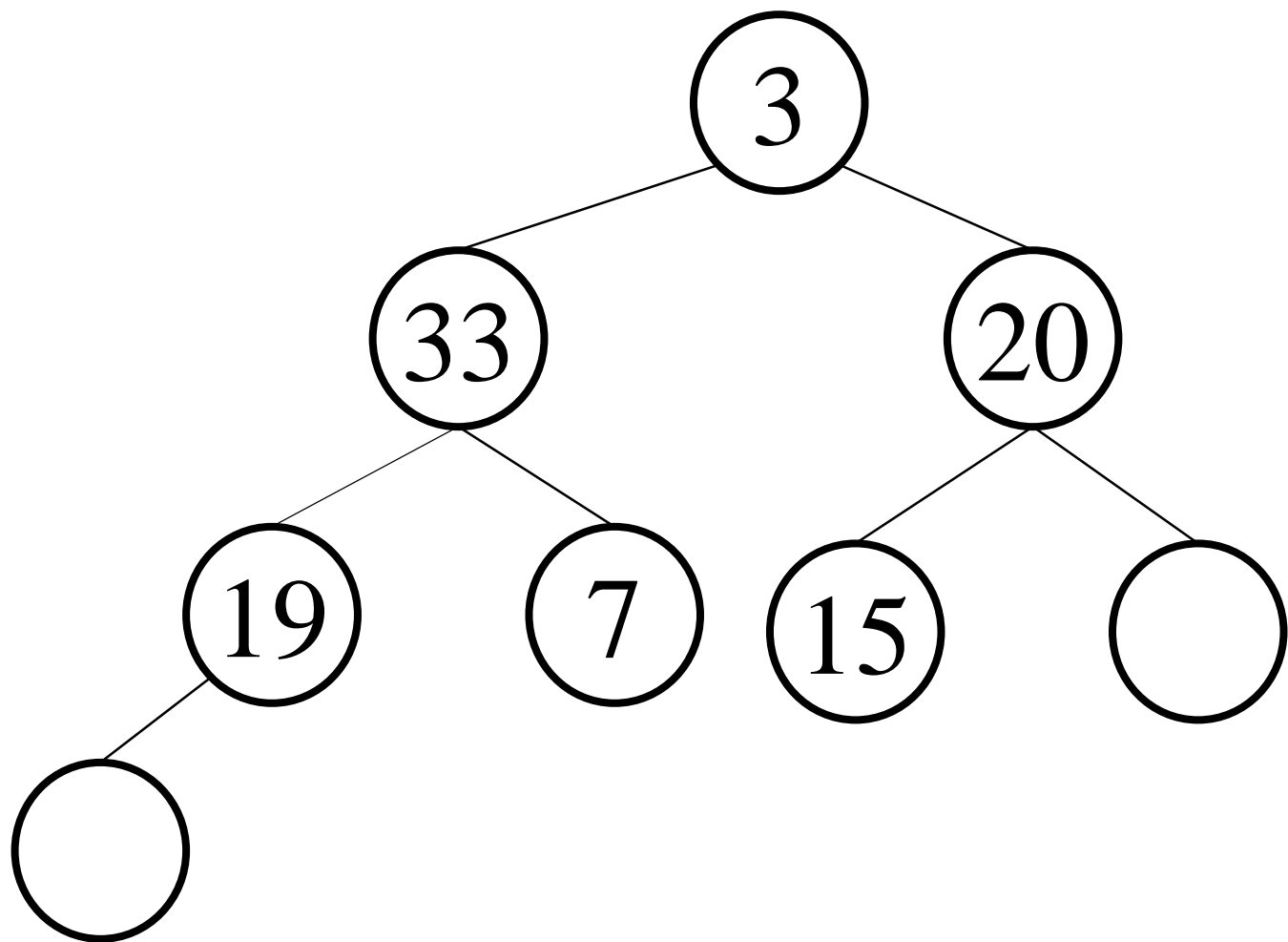
56



삭제된 원소

56

45



삭제된 원소

56

45

## 문제 24

◆ 트리를 표현할 때 가장 적합한 자료구조는? 답 : 다

가. Stack      나. Queue      다. Linked List      라. Circular Queue

다 = 트리는 자식노드의 주솟값을 가져야하므로 연결리스트가 가장 적절하다.

## 문제 25

◆ 정점이 8개인 무방향 완전 그래프의 간선의 수는 몇 개 인가? 답 : 28

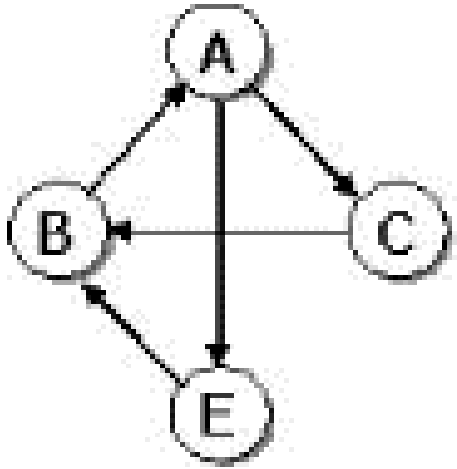
- 완전 그래프란 간선을 최대한으로 가진 그래프를 말한다.

n개의 정점을 가진 무방향 그래프의 최대 간선 수는  $n(n-1)/2$ 이다.

그러므로 정점이 8개인 무방향 완전그래프의 간선의 수는  $8(8-1)/2 = 28$ , 28개이다.

## 문제 26

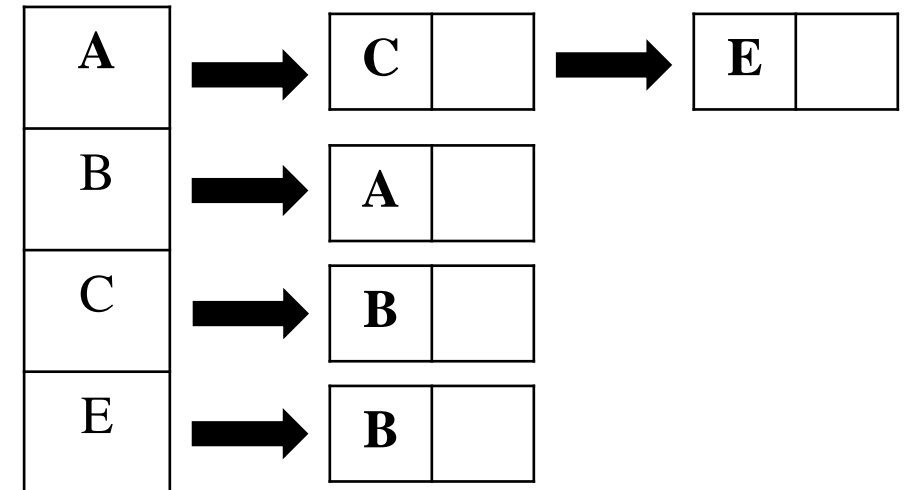
- ◆ 다음 그래프를 인접 행렬과 인접 리스트로 표현하여라.



인접 행렬

	0	1	2	3
A	0	0	1	1
B	1	0	0	0
C	0	1	0	0
E	0	1	0	0

인접 리스트

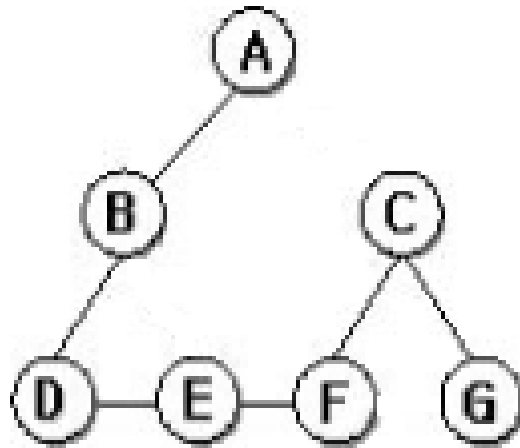
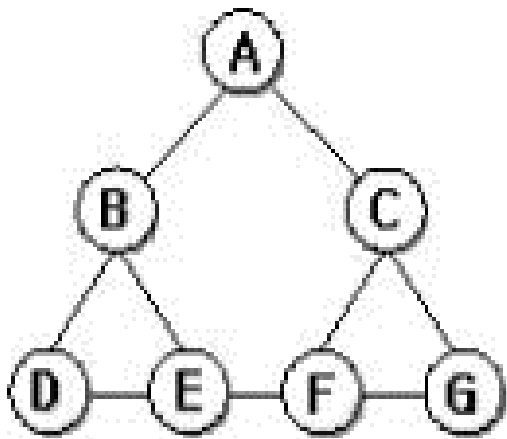


## 문제 27

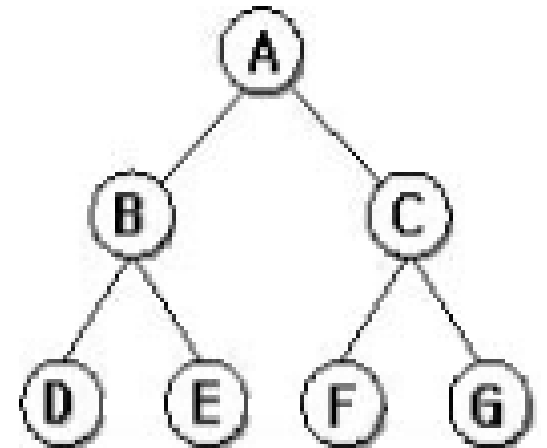
- ◆ 다음 그래프의 깊이 우선 신장트리와 너비 우선 신장 트리를 구하여라.

깊이 우선 탐색 : 최대한 깊이 내려간 뒤, 더 이상 갈 곳이 없는 경우 옆으로 이동

너비 우선 탐색 : 최대한 넓게 이동한 후 더 이상 갈 곳이 없는 경우 아래로 이동



깊이 우선 신장트리



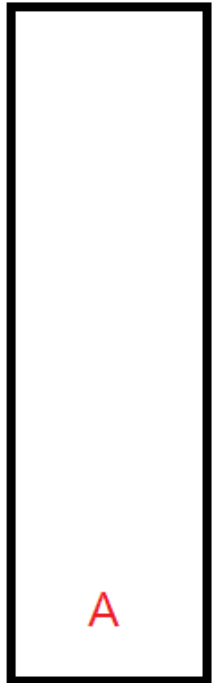
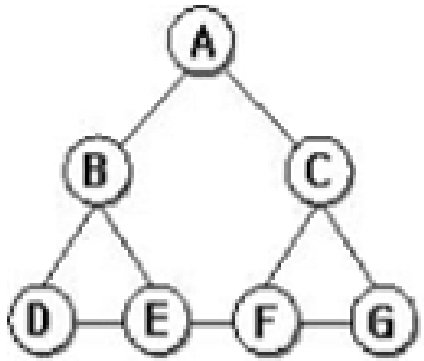
너비 우선 신장트리

## 문제27

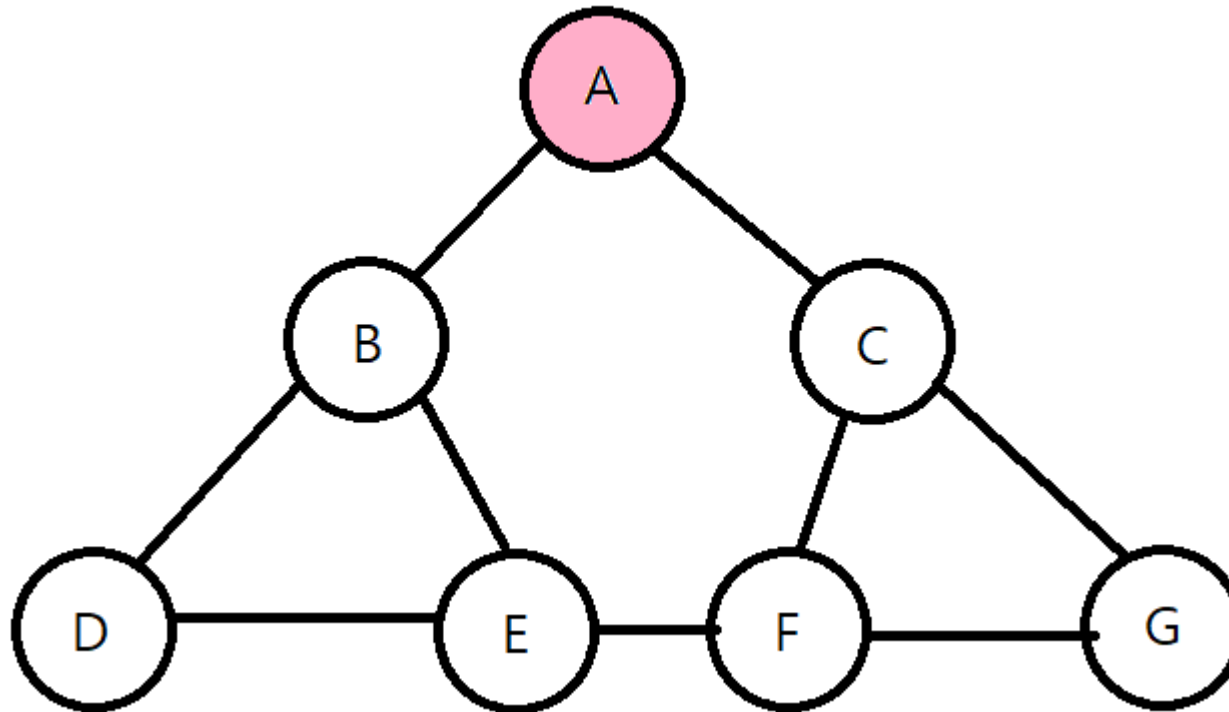
Q. 다음 그래프의 깊이 우선 신장트리와 너비 우선 신장 트리를 구하여라.

\*답

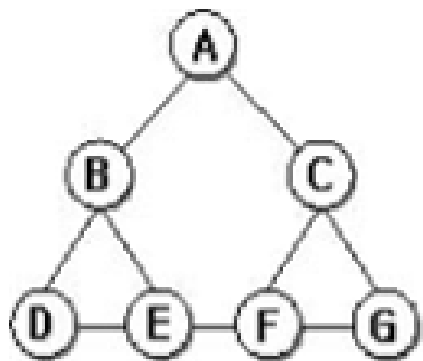
깊이 우선 신장 트리 : A



POP/A방문 표시



## 문제27

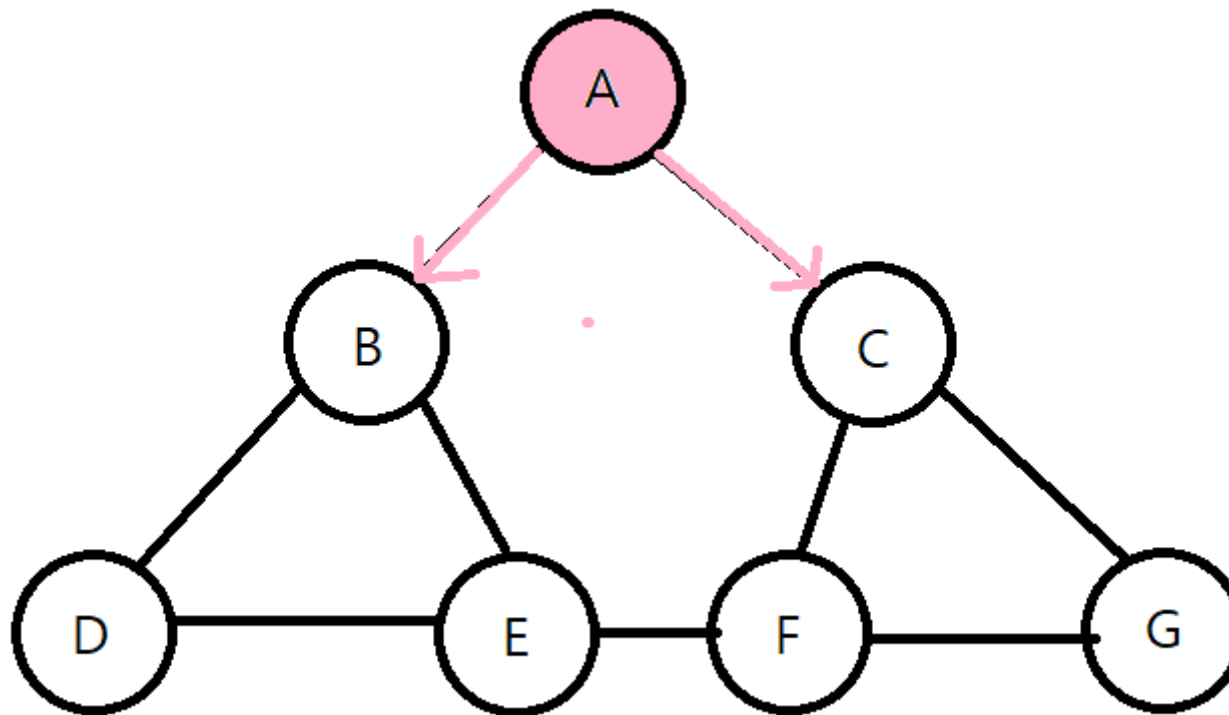


\*답

깊이 우선 신장 트리 : A

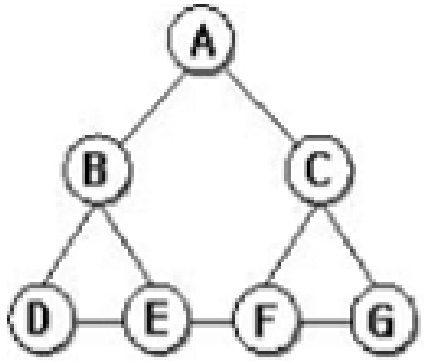


PUSHC, B





## 문제27

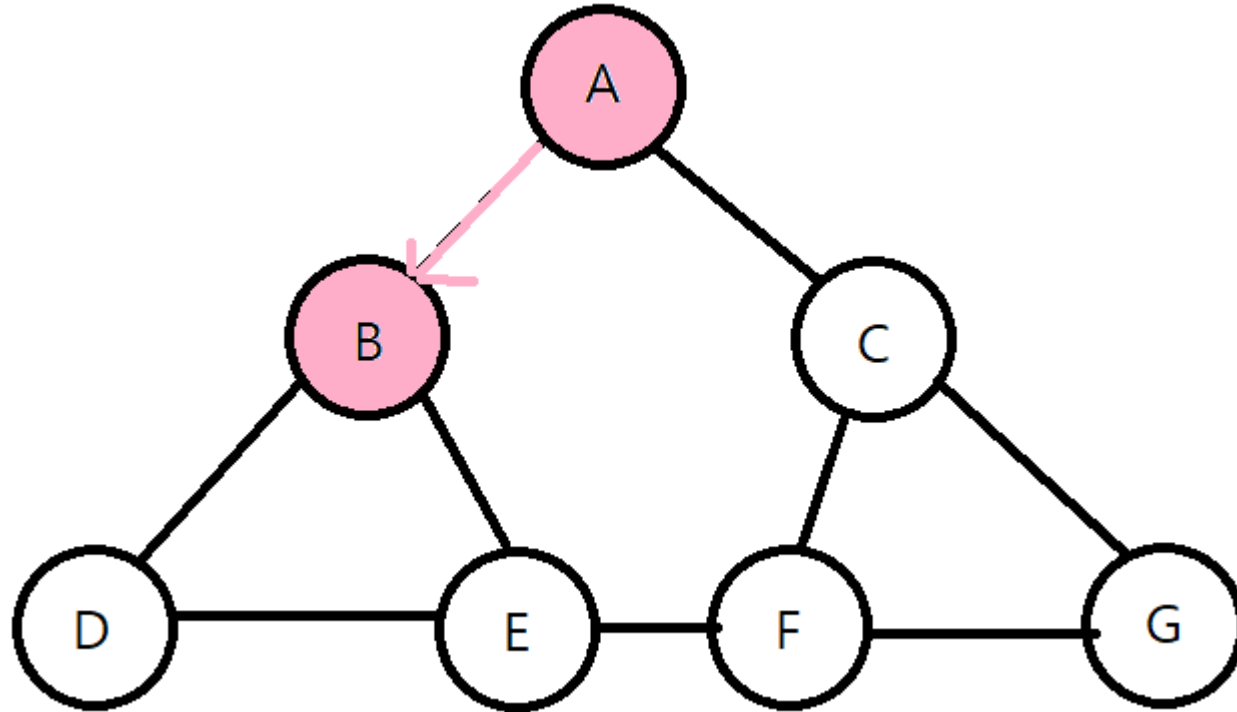


\*답

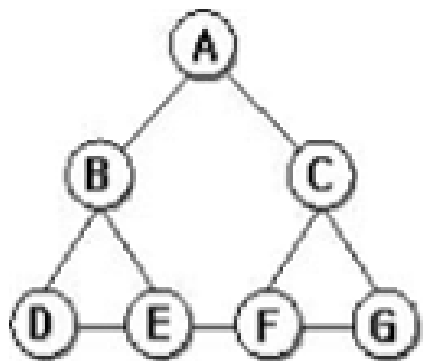
깊이 우선 신장 트리 : A-B



POP/B방문표시

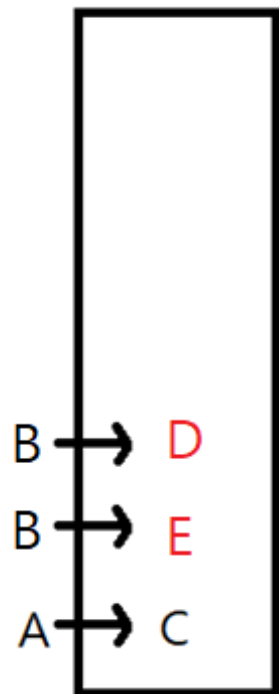


## 문제27

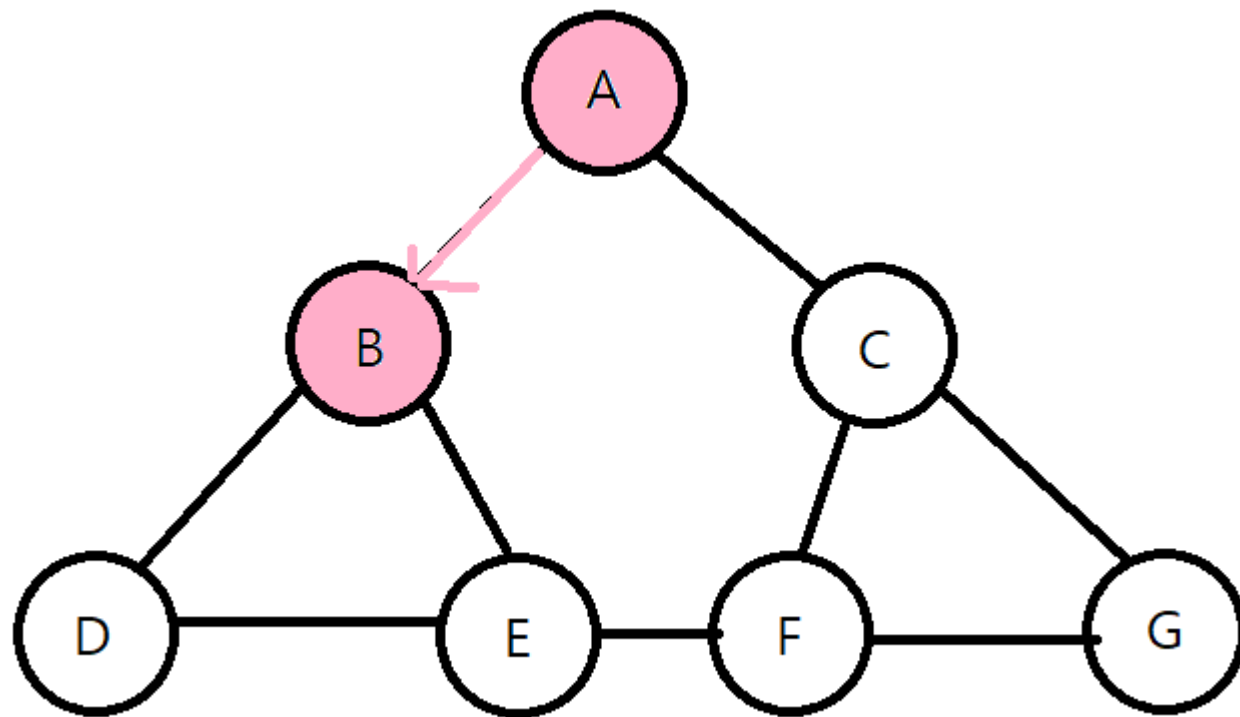


\*답

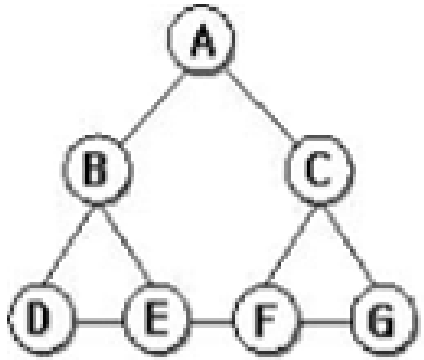
깊이 우선 신장 트리 : A-B



PUSH D,E

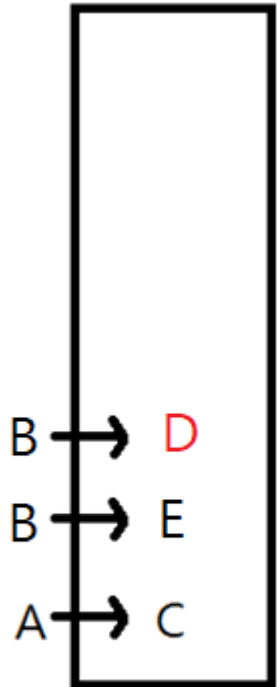


## 문제27

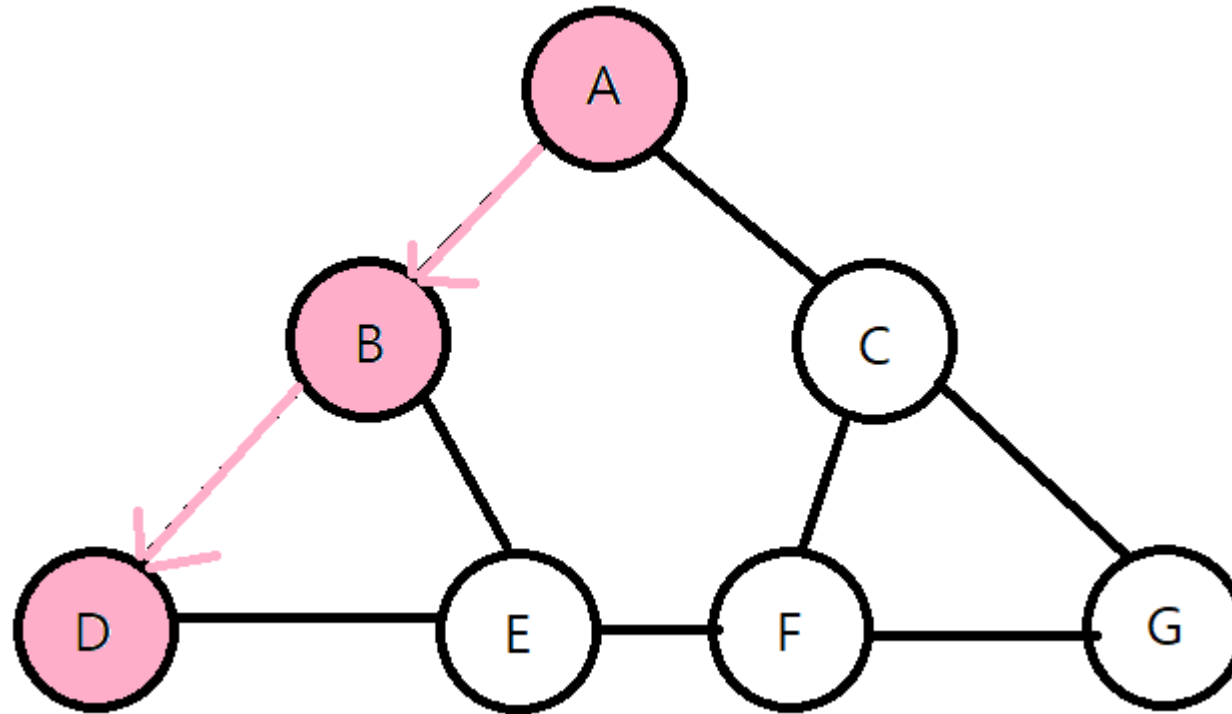


\*답

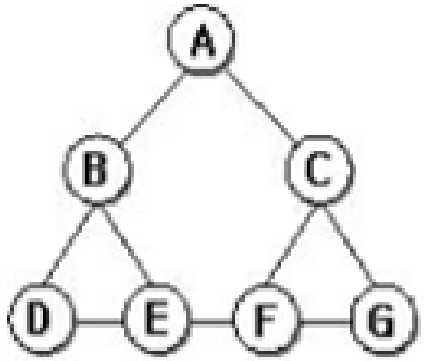
깊이 우선 신장 트리 : A-B-D



POP/D방문표시

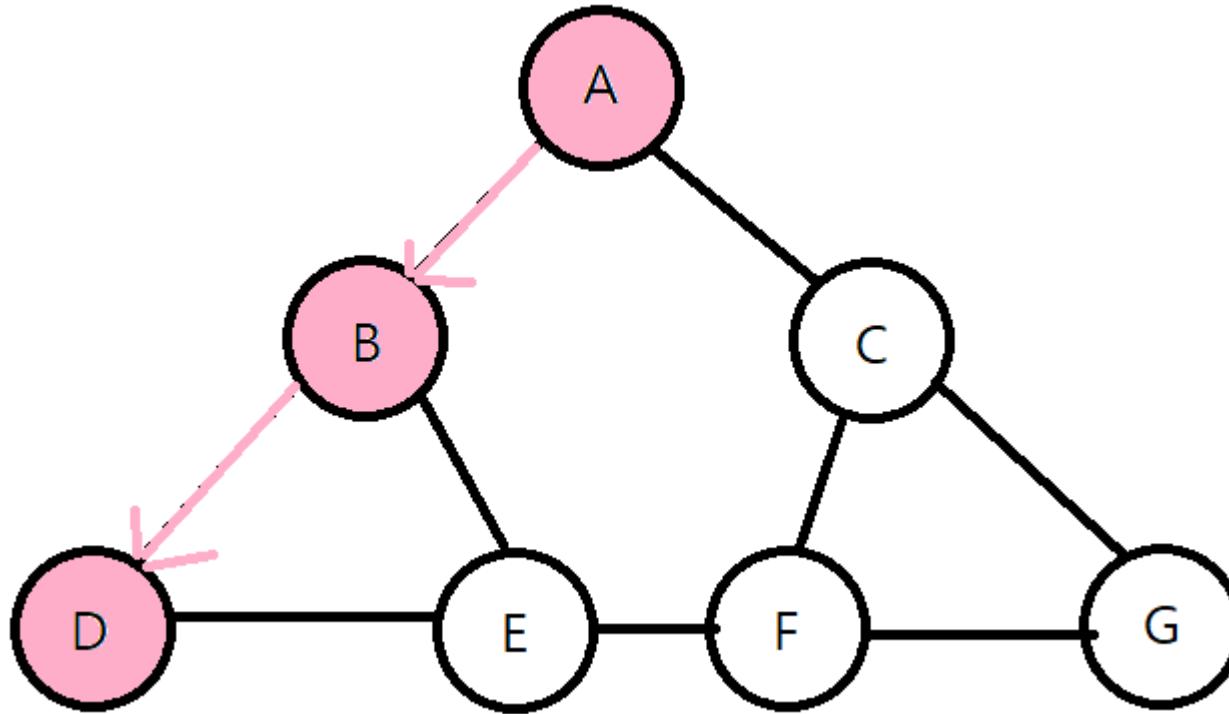
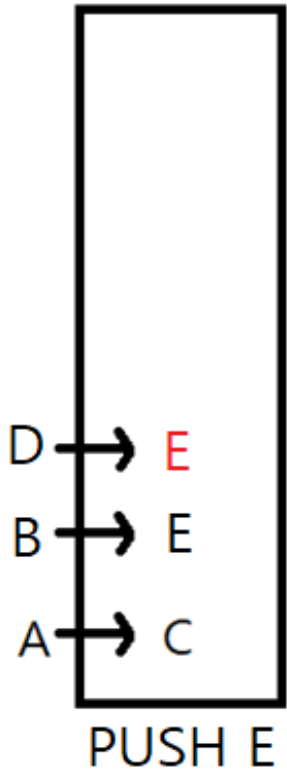


## 문제27

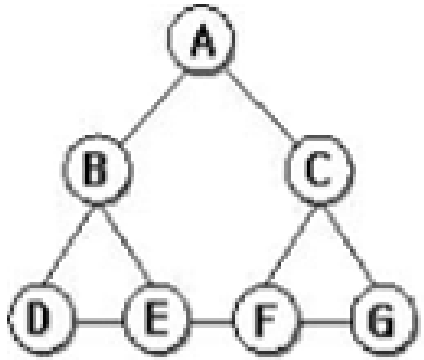


\*답

깊이 우선 신장 트리 : A-B-D

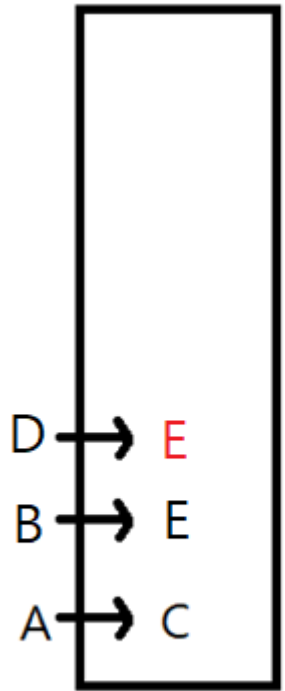


## 문제27

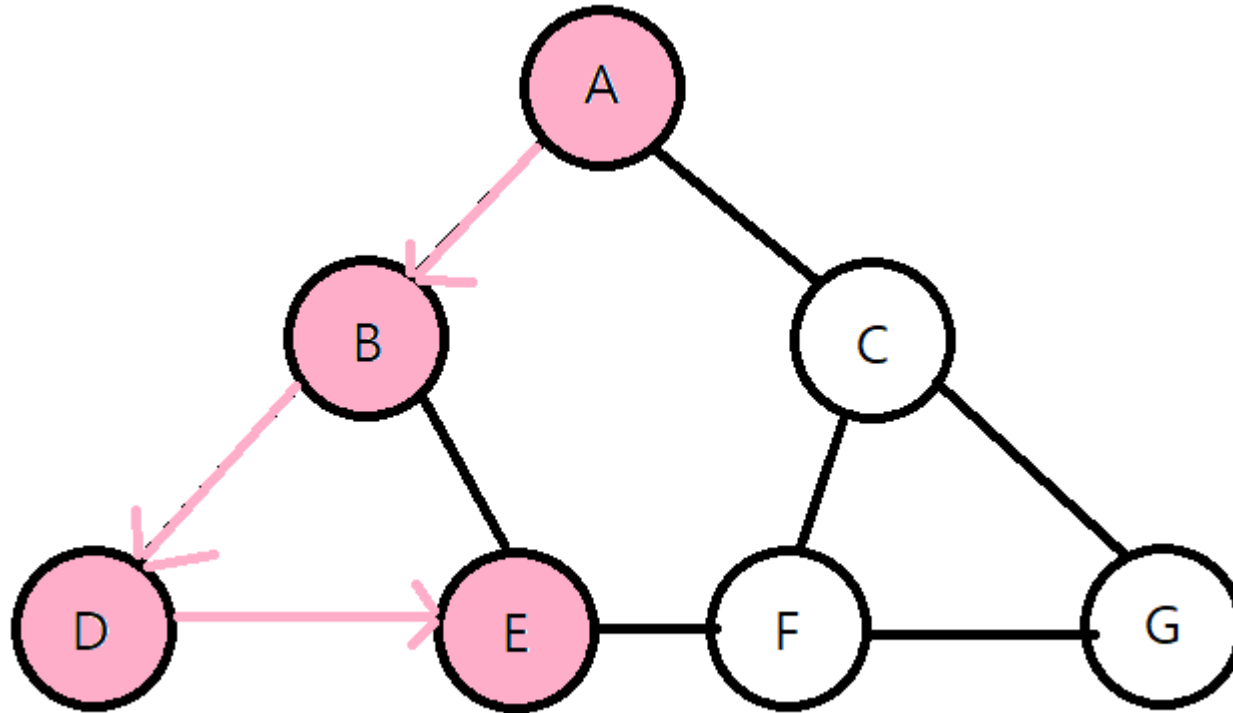


\*답

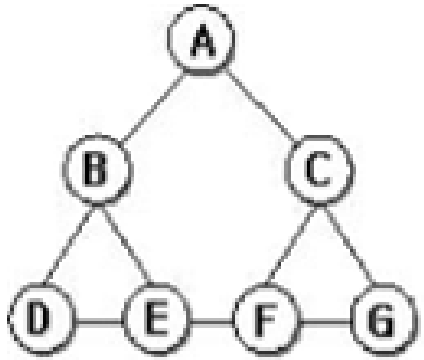
깊이 우선 신장 트리 : A-B-D-E



POP/E방문표시



## 문제27

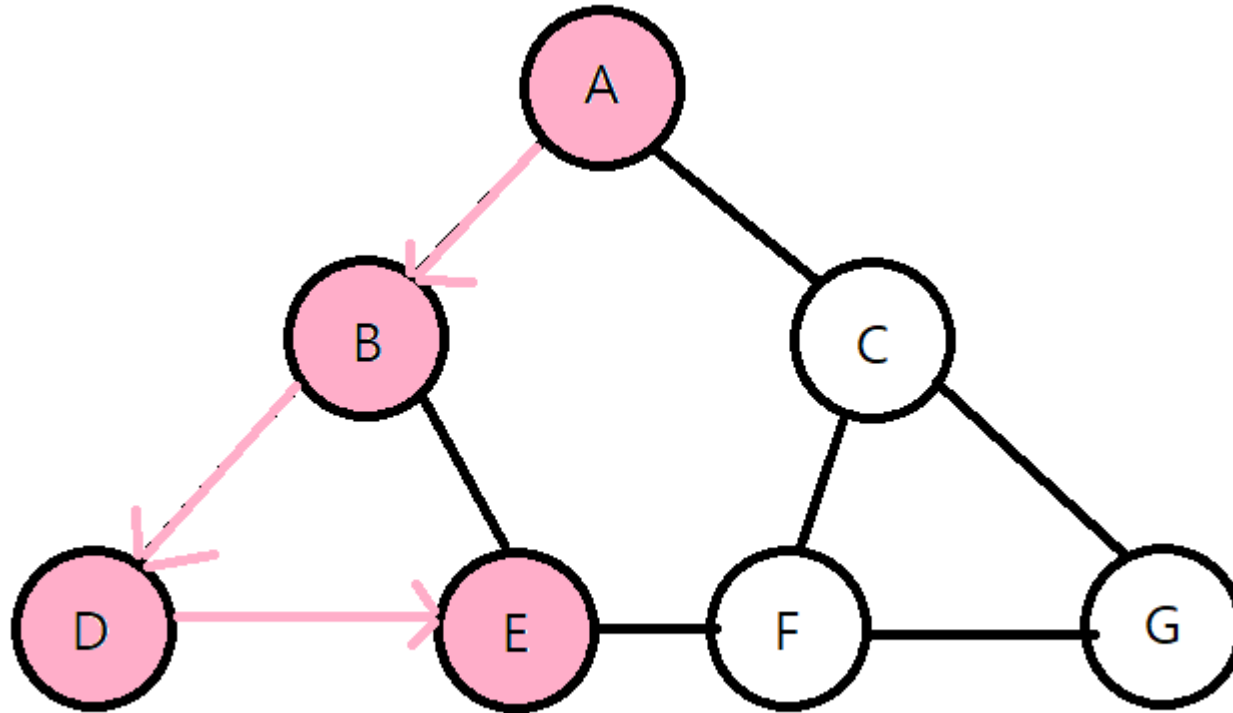


\*답

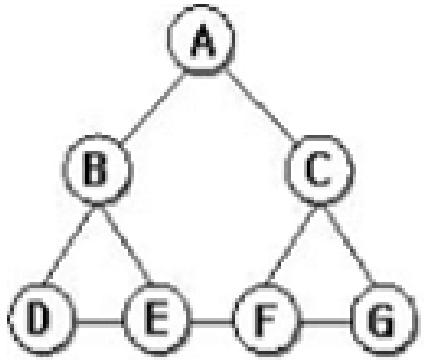
깊이 우선 신장 트리 : A-B-D-E



PUSH F



## 문제27

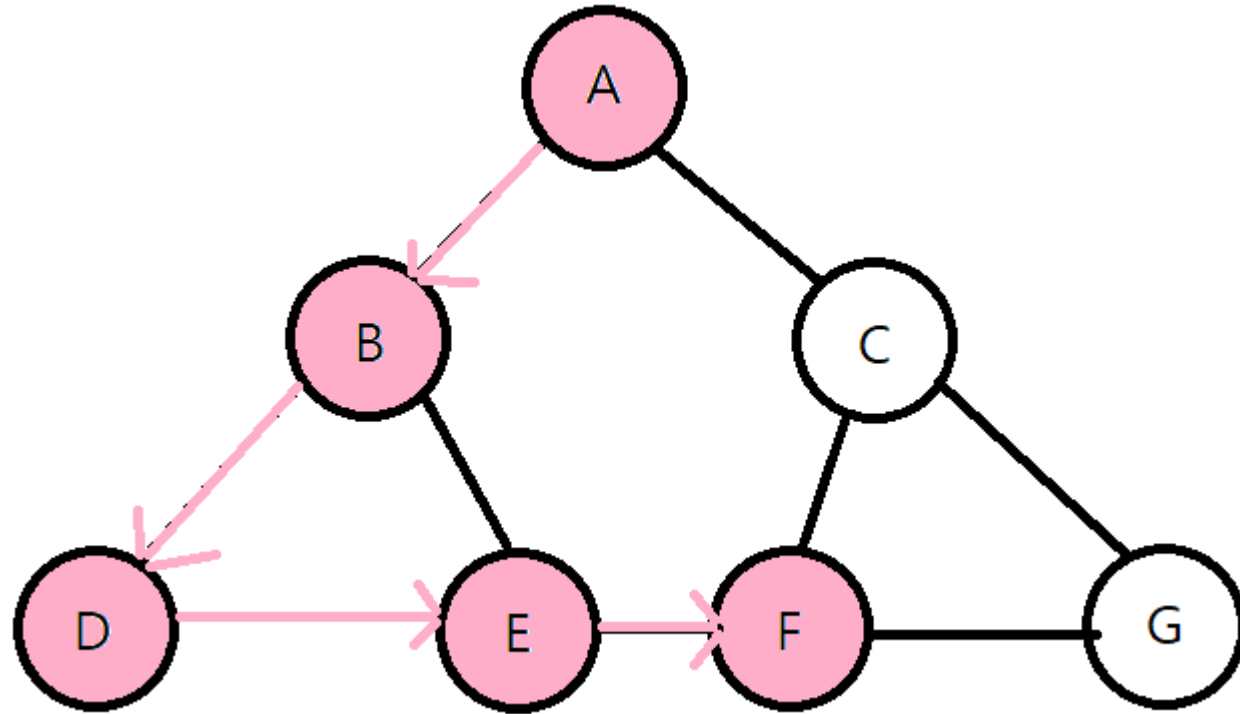


\*답

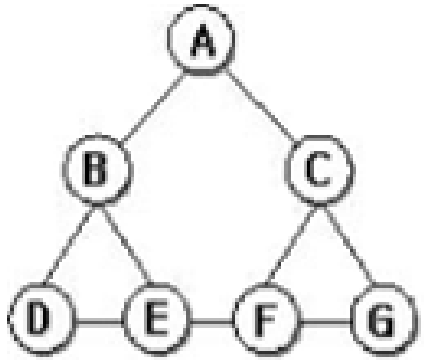
깊이 우선 신장 트리 : A-B-D-E-F



POP/F방문표시

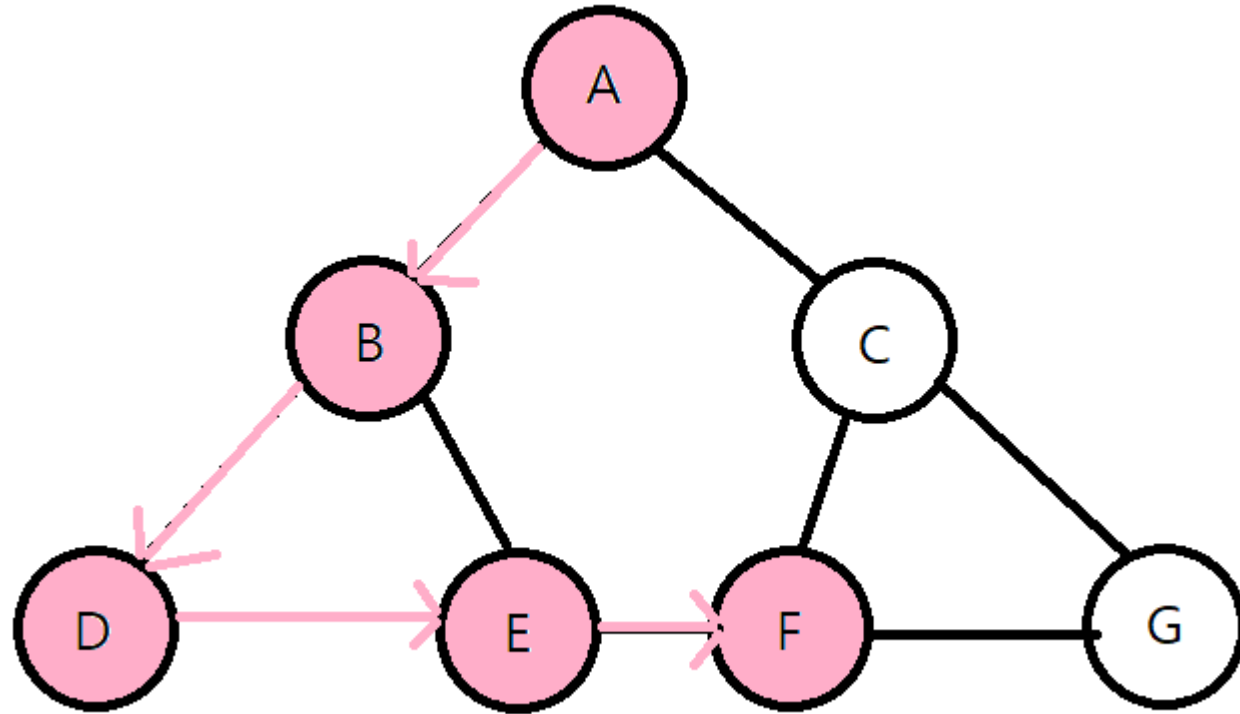
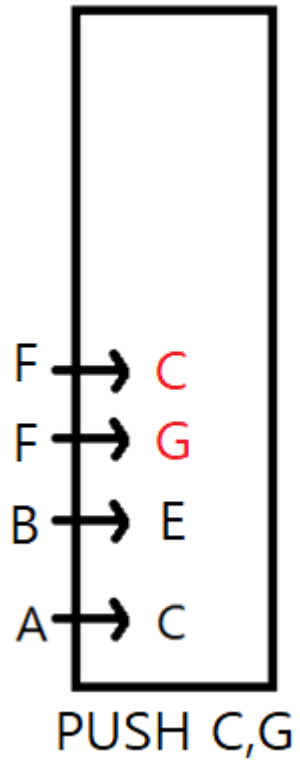


## 문제27



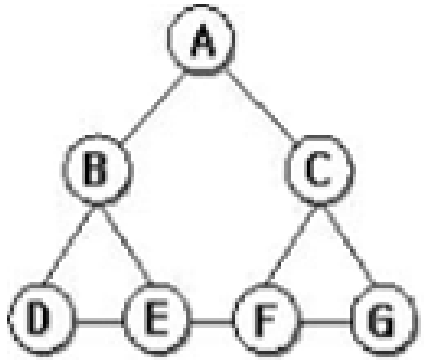
\*답

깊이 우선 신장 트리 : A-B-D-E-F



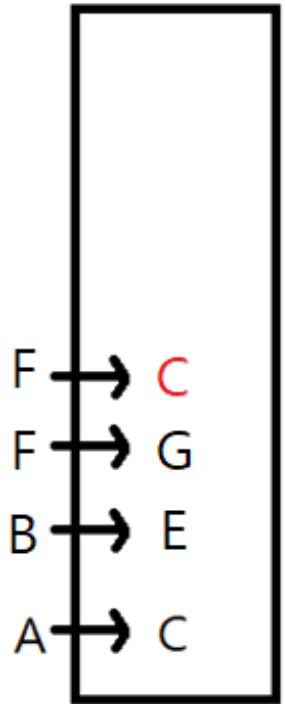


## 문제27

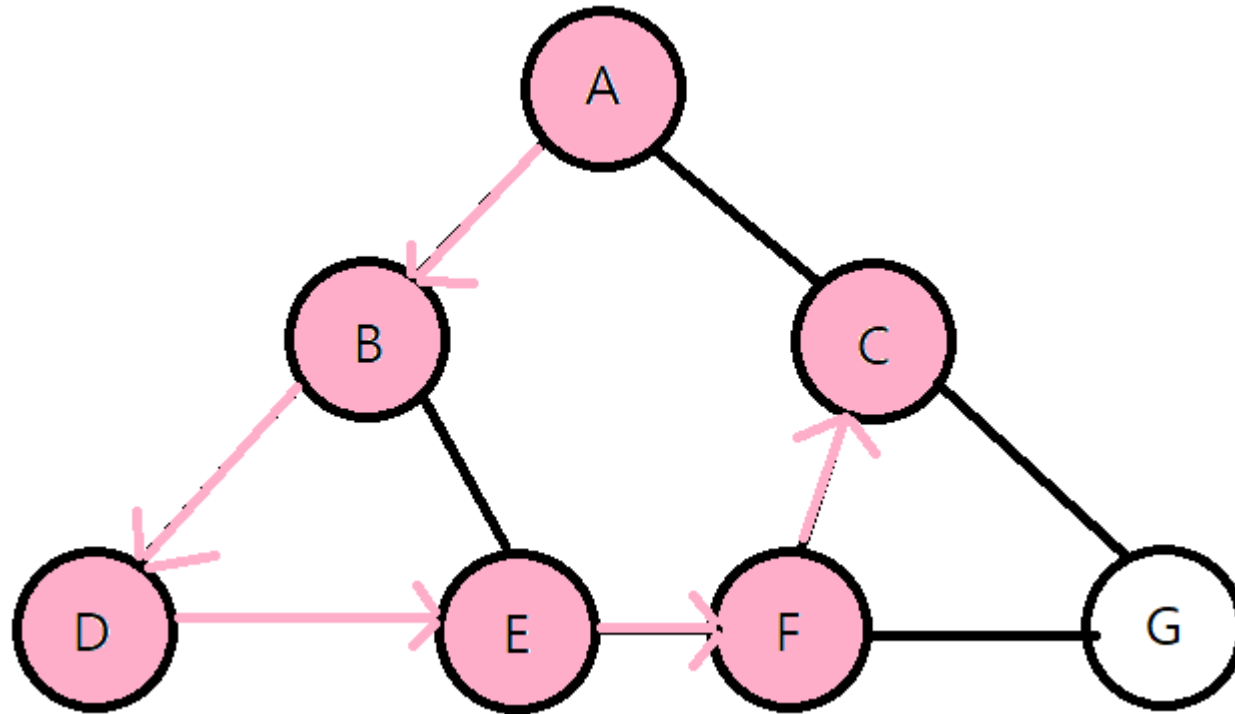


\*답

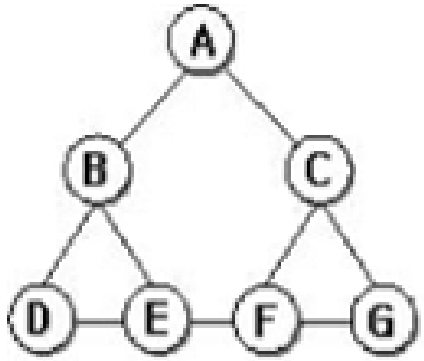
깊이 우선 신장 트리 : A-B-D-E-F-C



POP/C방문표시

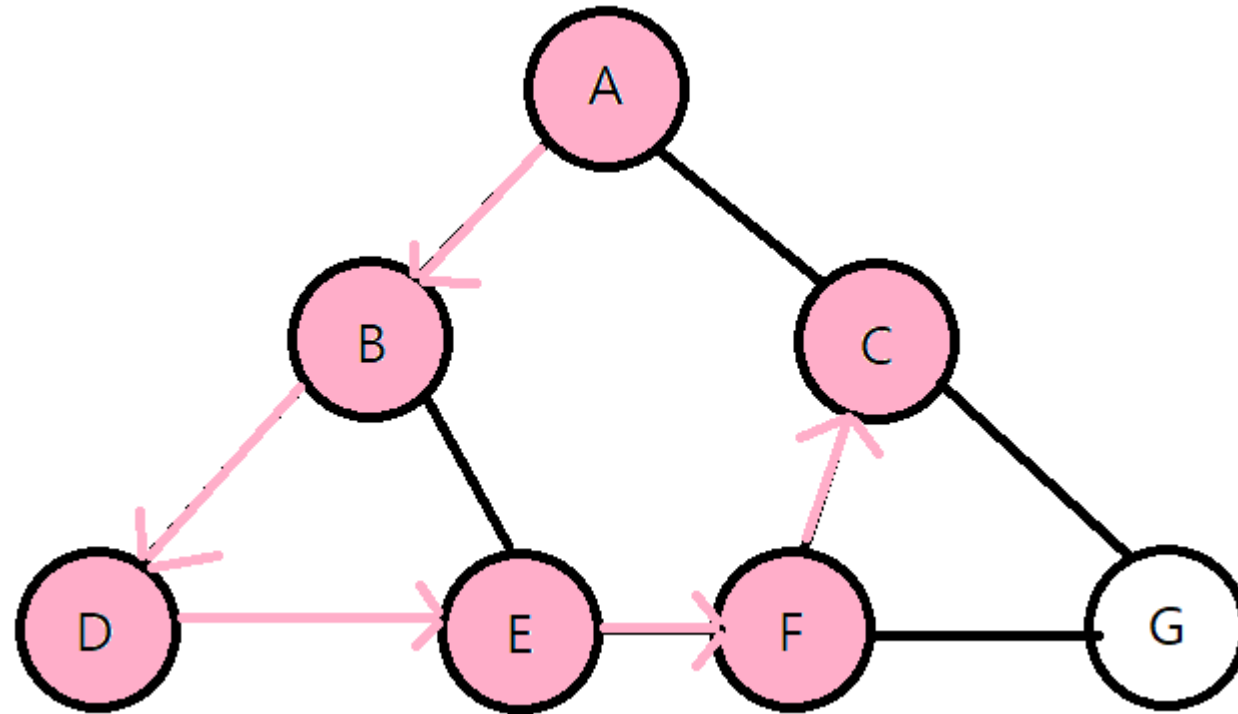
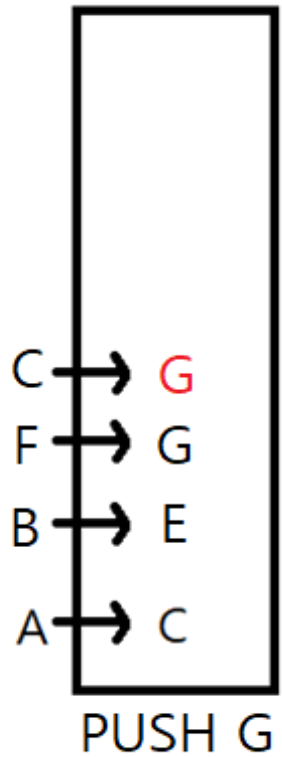


## 문제27

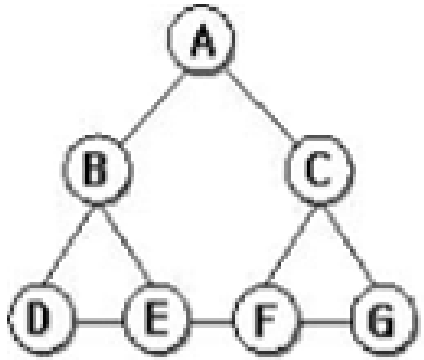


\*답

깊이 우선 신장 트리 : A-B-D-E-F-C

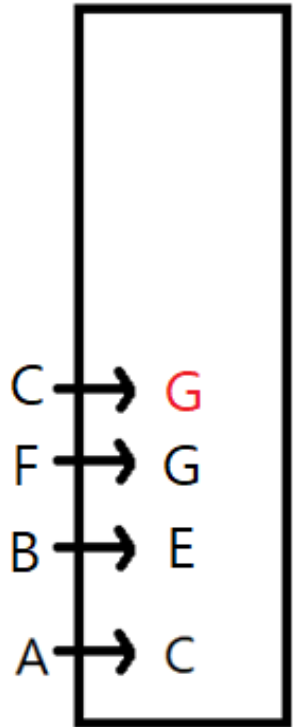


## 문제27

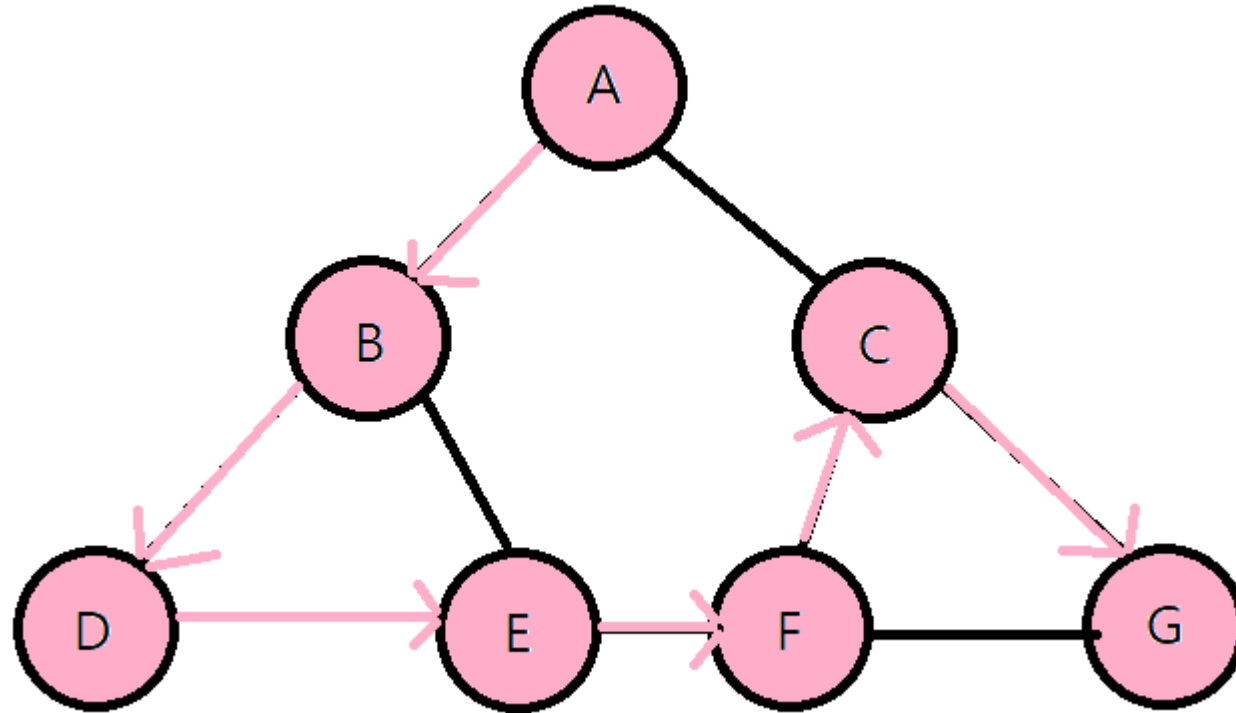


\*답

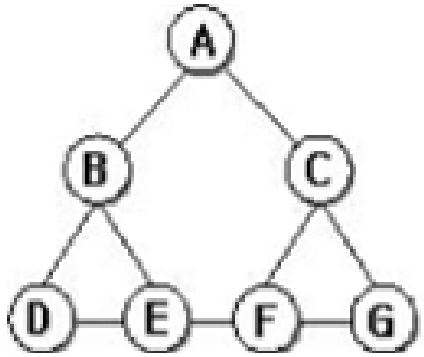
깊이 우선 신장 트리 : A-B-D-E-F-C-G



POP/G방문표시

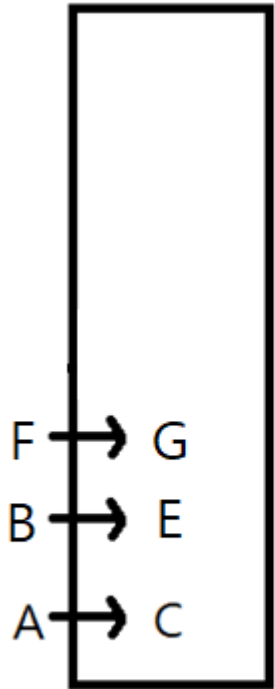


## 문제27

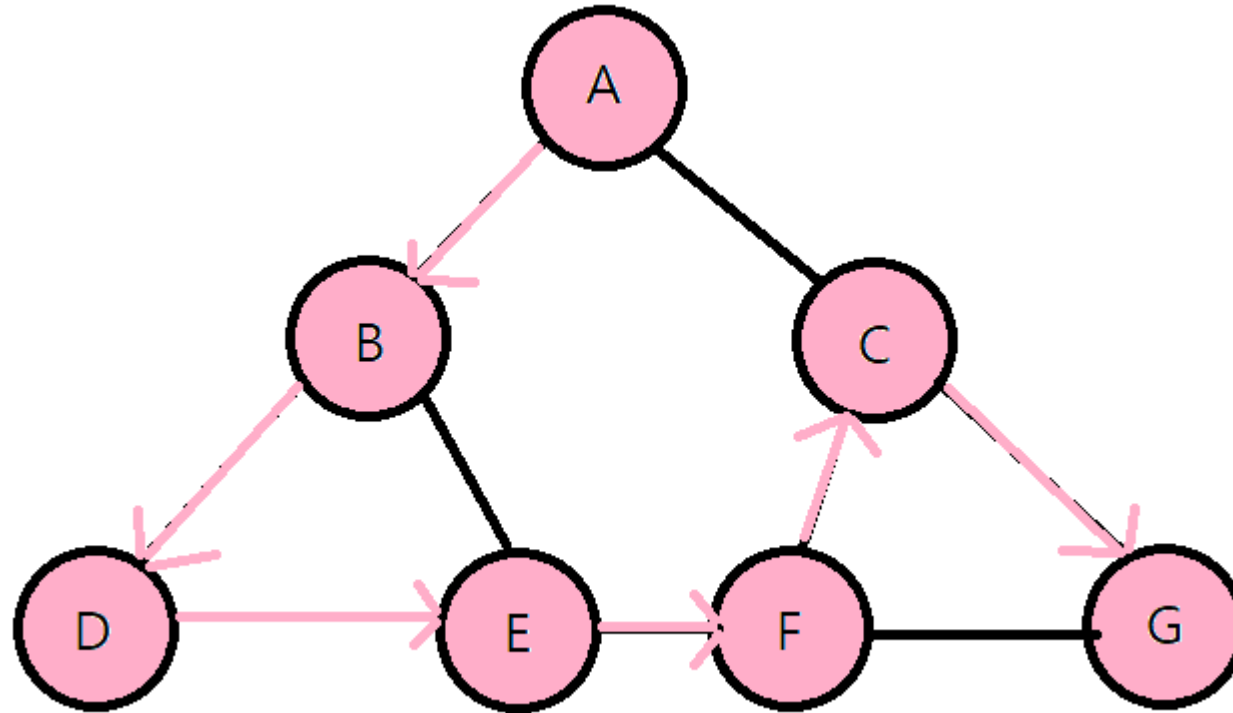


\*답

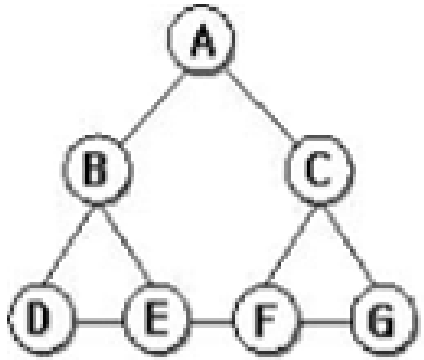
깊이 우선 신장 트리 : A-B-D-E-F-C-G



POP/G,E,C방문안함

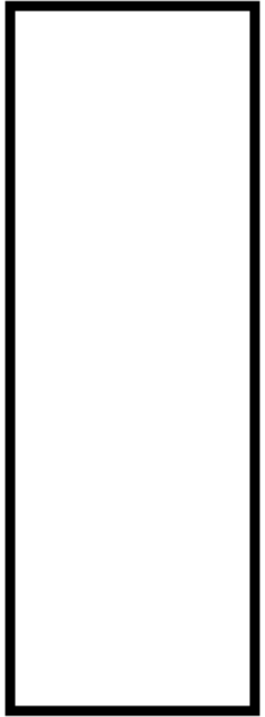


## 문제27

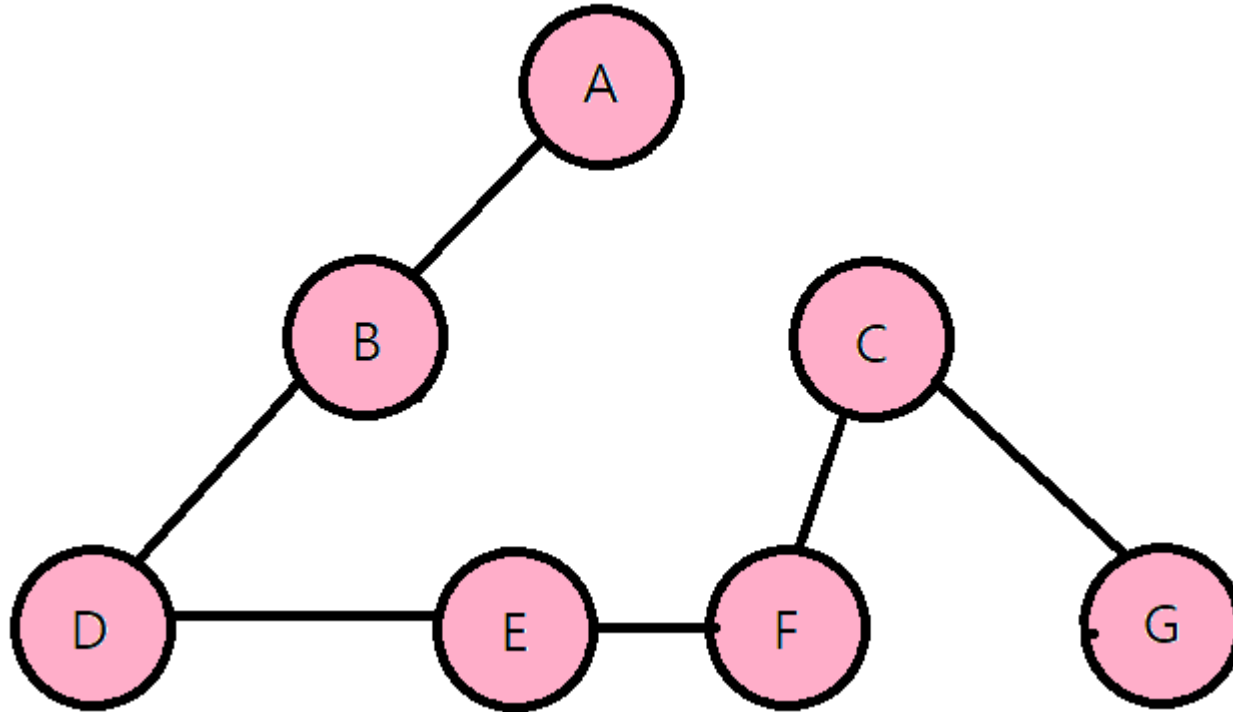


\*답

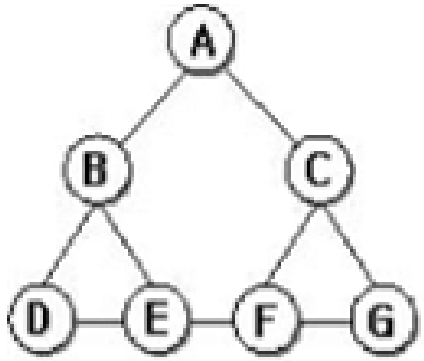
깊이 우선 신장 트리 : A-B-D-E-F-C-G



끝

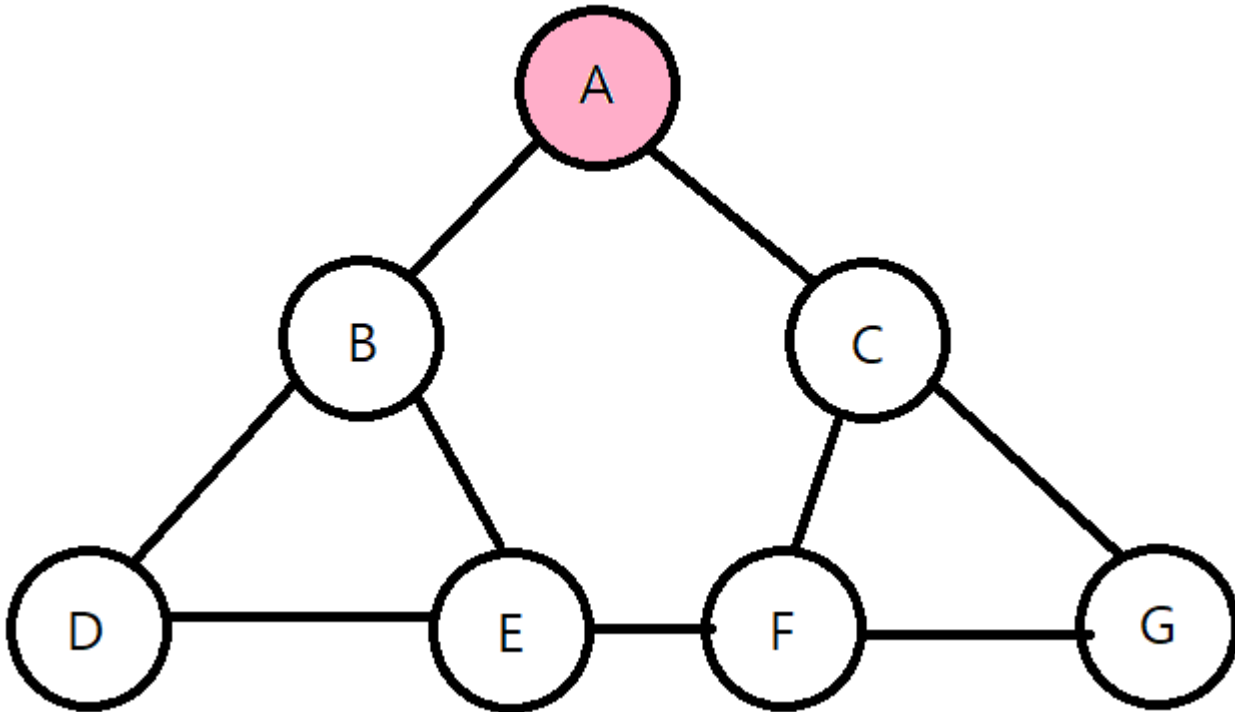


## 문제27



\*답

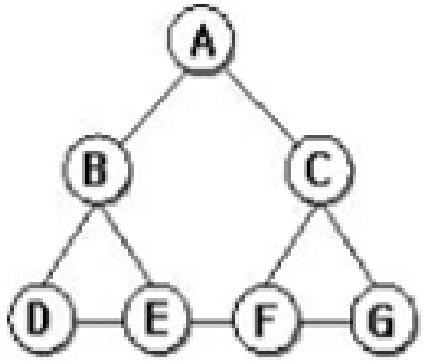
너비 우선 신장 트리 :A



QUEUE

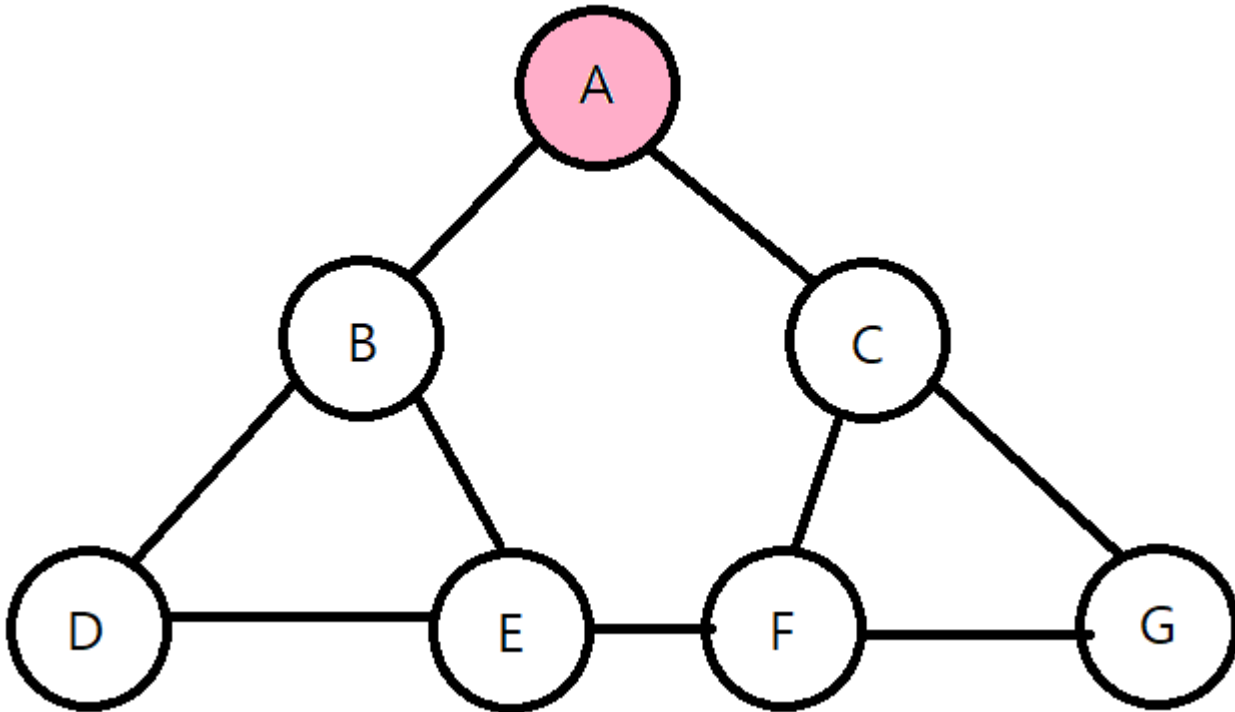


## 문제27



\*답

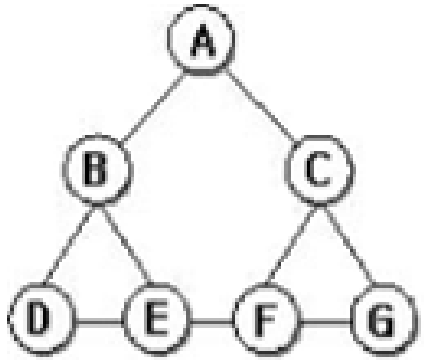
너비 우선 신장 트리 :A



QUEUE

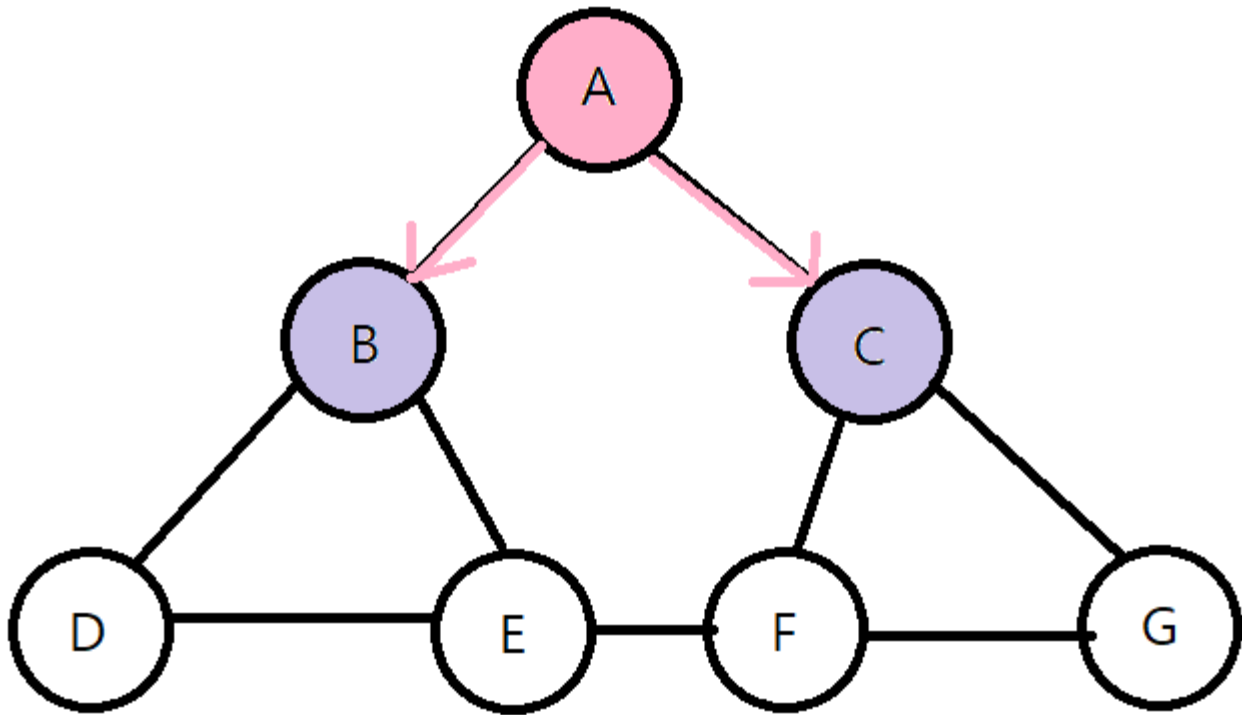


## 문제27



\*답

너비 우선 신장 트리 : A



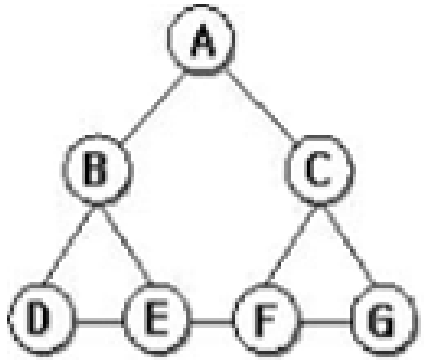
QUEUE



B,C 삽입 dequeue

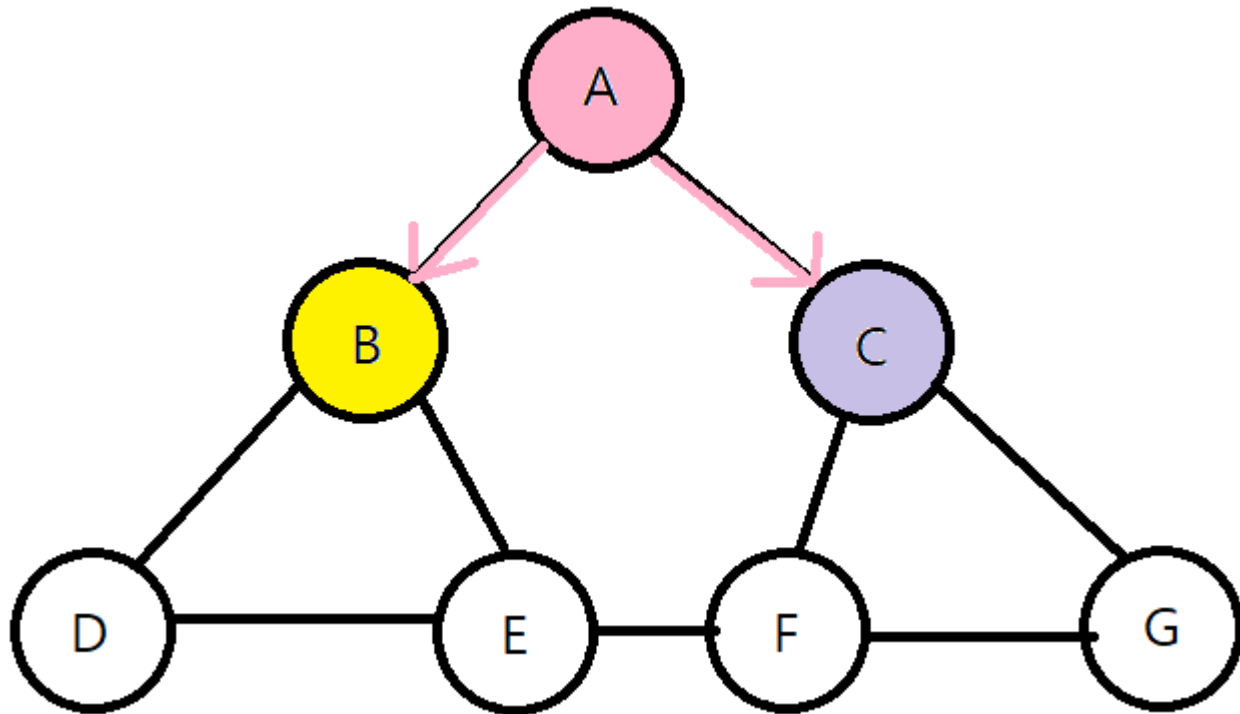


## 문제27



\*답

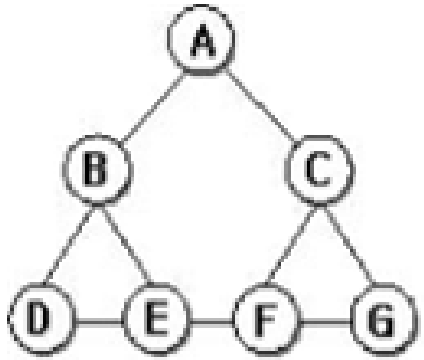
너비 우선 신장 트리 :A



QUEUE

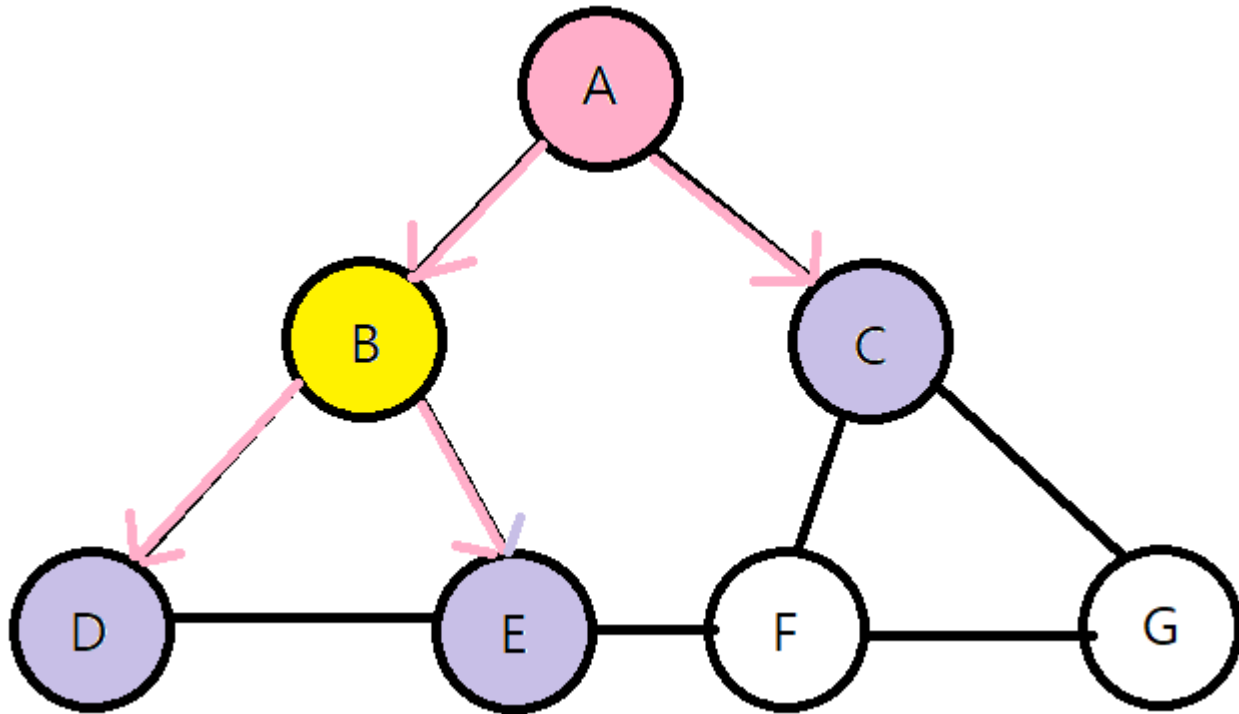
A B C

## 문제27



\*답

너비 우선 신장 트리 : A-B

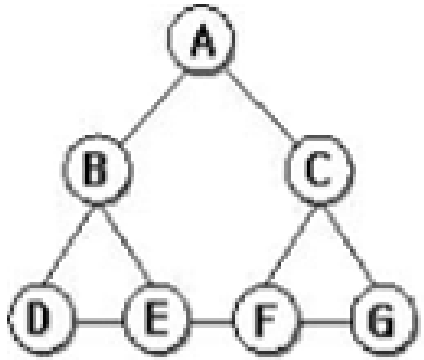


QUEUE

A B C D E

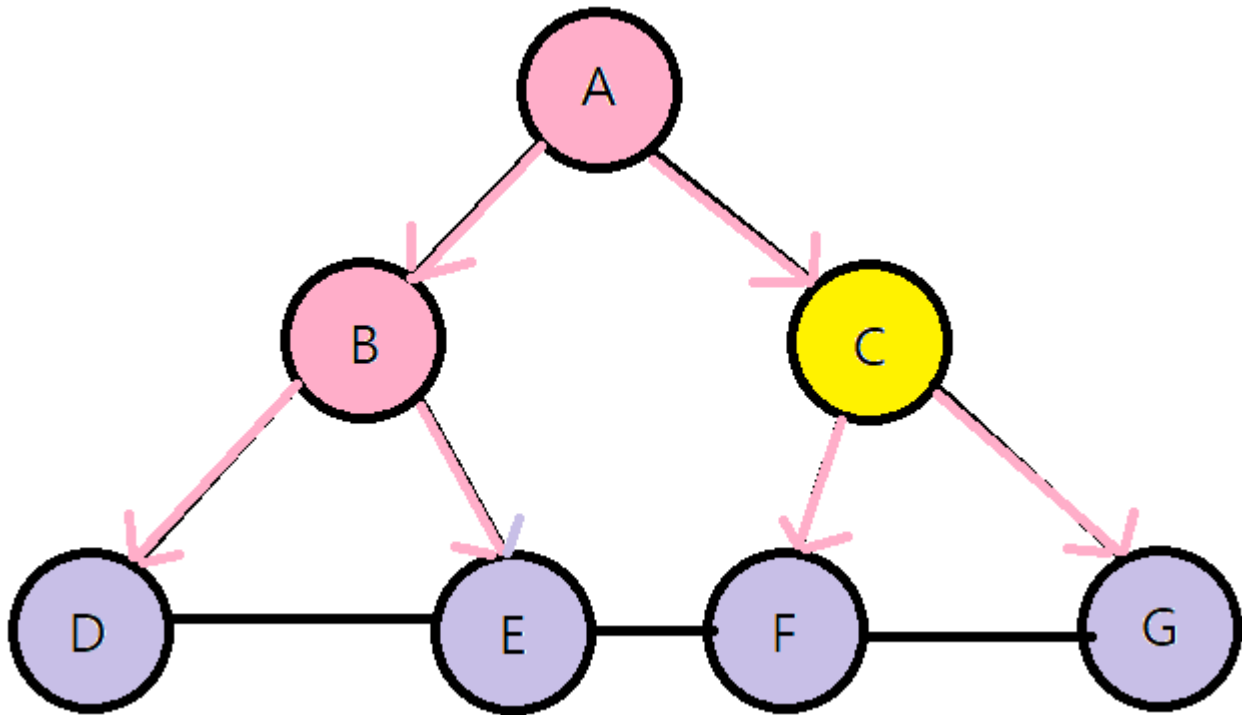
B이웃D,E삽입 dequeue

## 문제27



\*답

너비 우선 신장 트리 : A-B-C

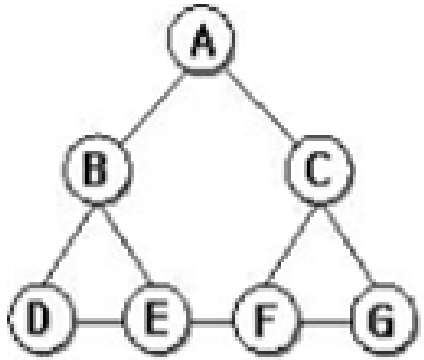


QUEUE

A B C D E F G

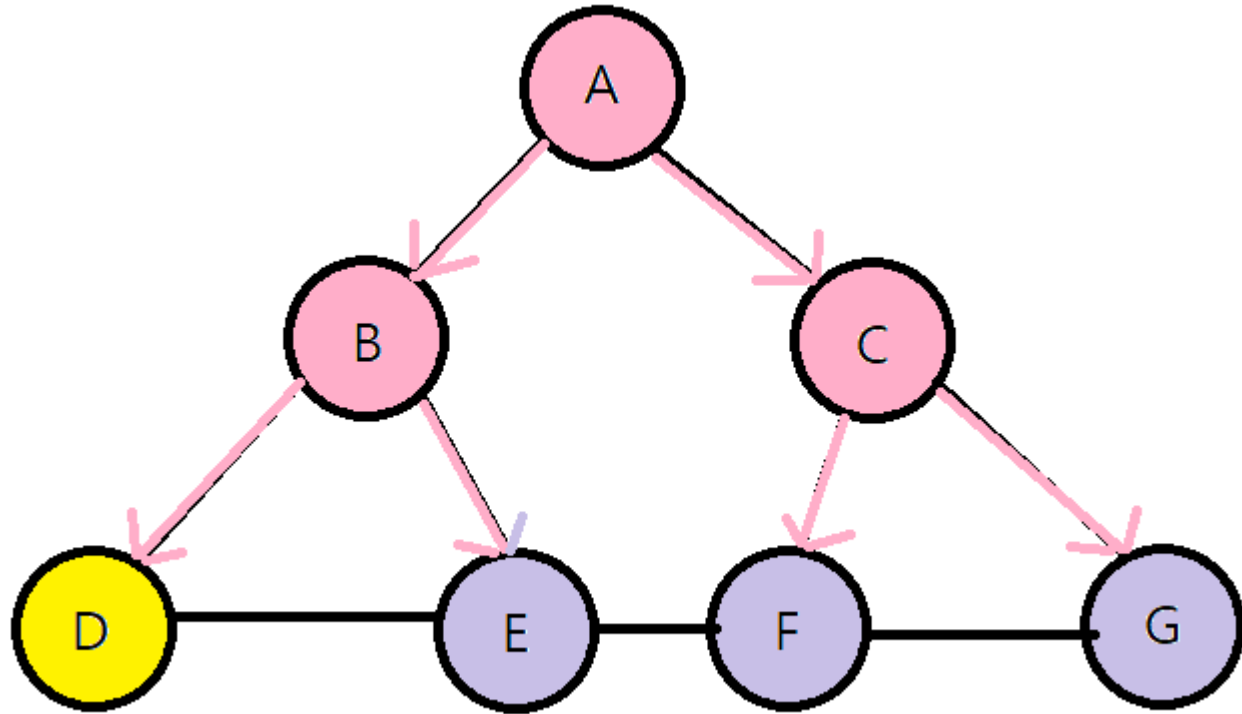
C이웃F,G삽입 dqueue

## 문제27



\*답

너비 우선 신장 트리 : A-B-C-D

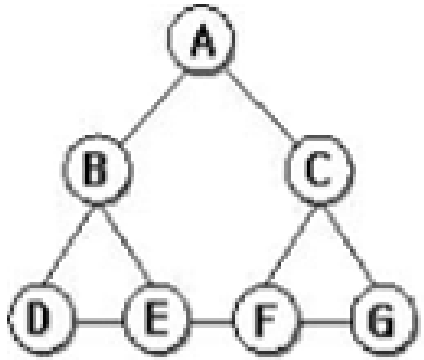


QUEUE

A B C D E F G

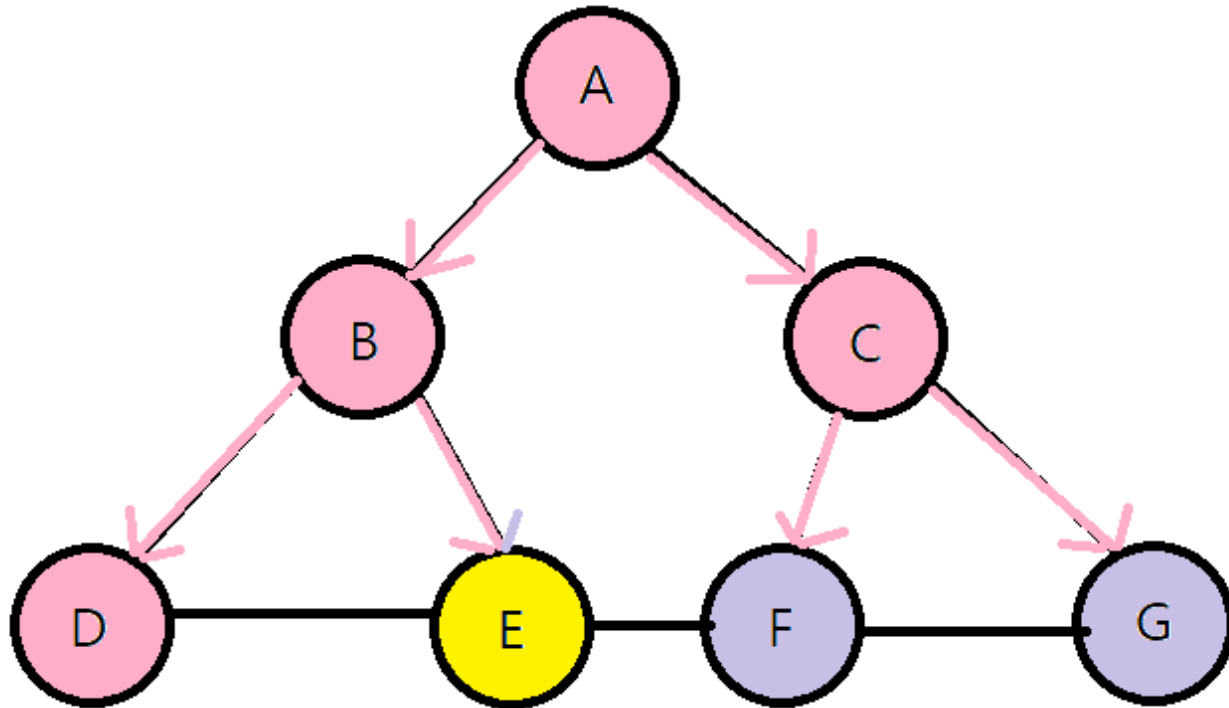
dqueue

## 문제27



\*답

너비 우선 신장 트리 : A-B-C-D-E

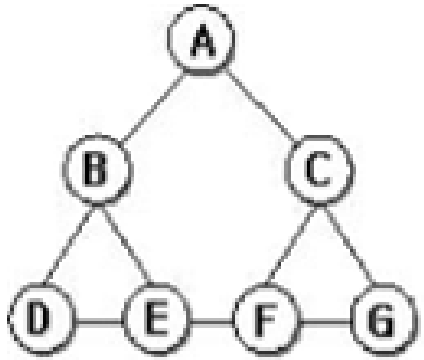


QUEUE

A B C D E F G

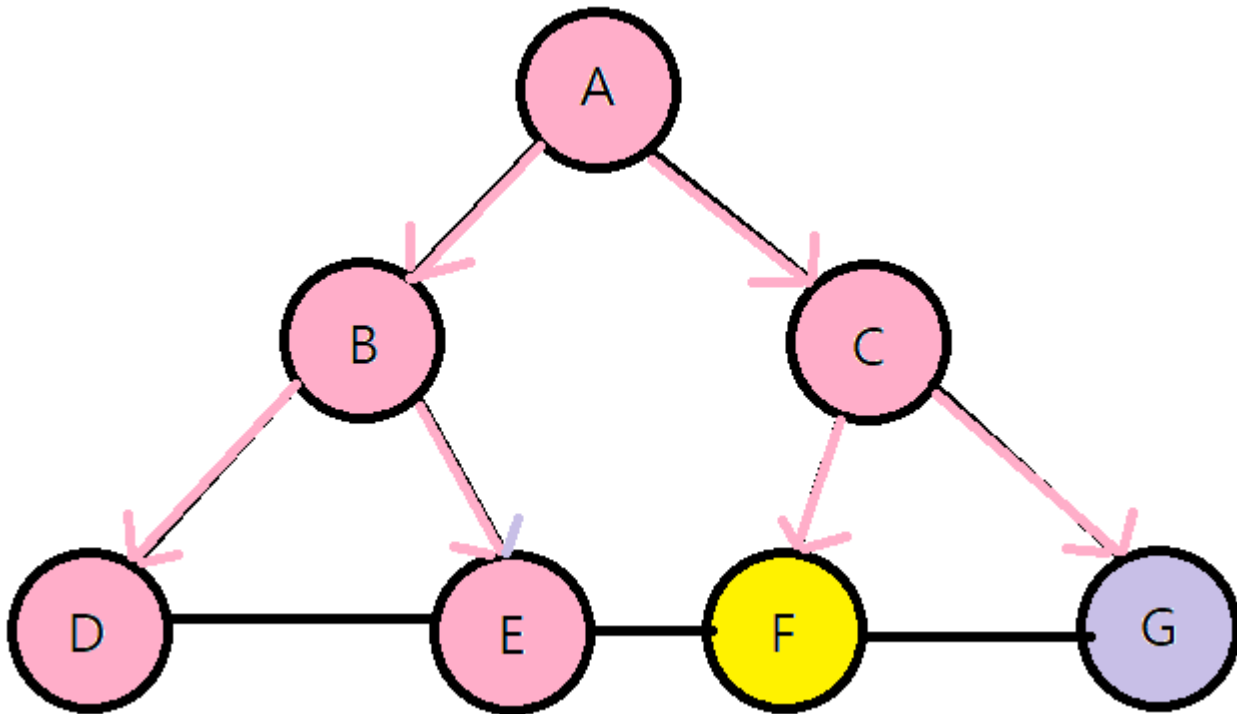
dqueue

## 문제27



\*답

너비 우선 신장 트리 : A-B-C-D-E-F

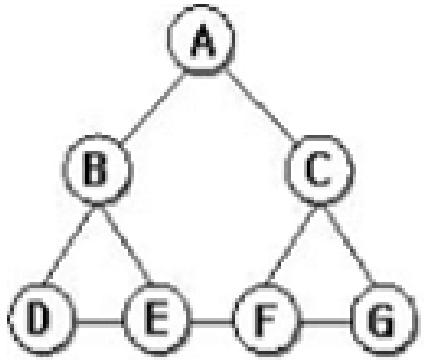


QUEUE



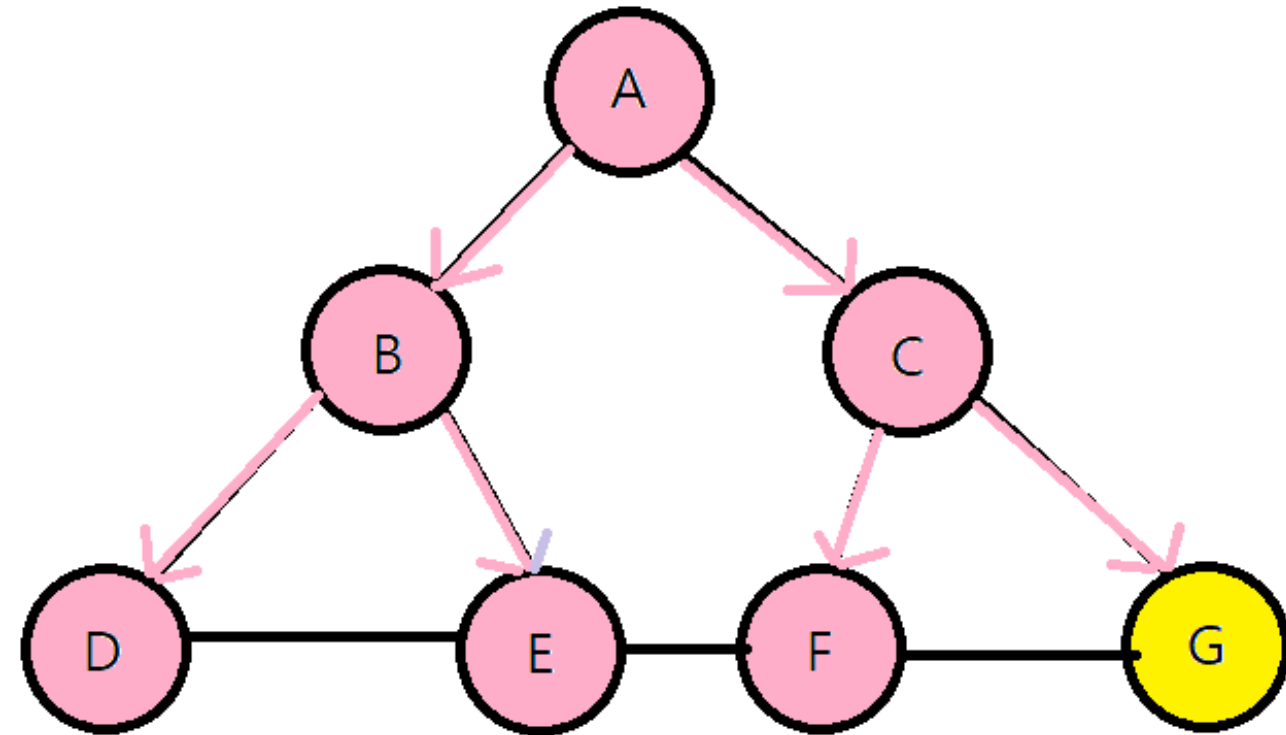
dqueue

## 문제27



\*답

너비 우선 신장 트리 : A-B-C-D-E-F-G

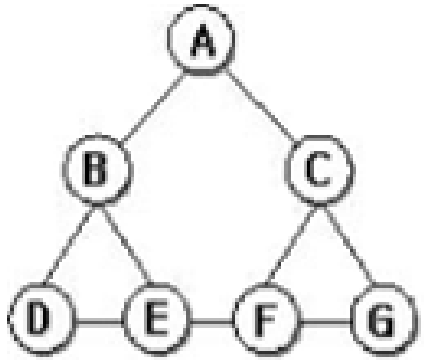


QUEUE



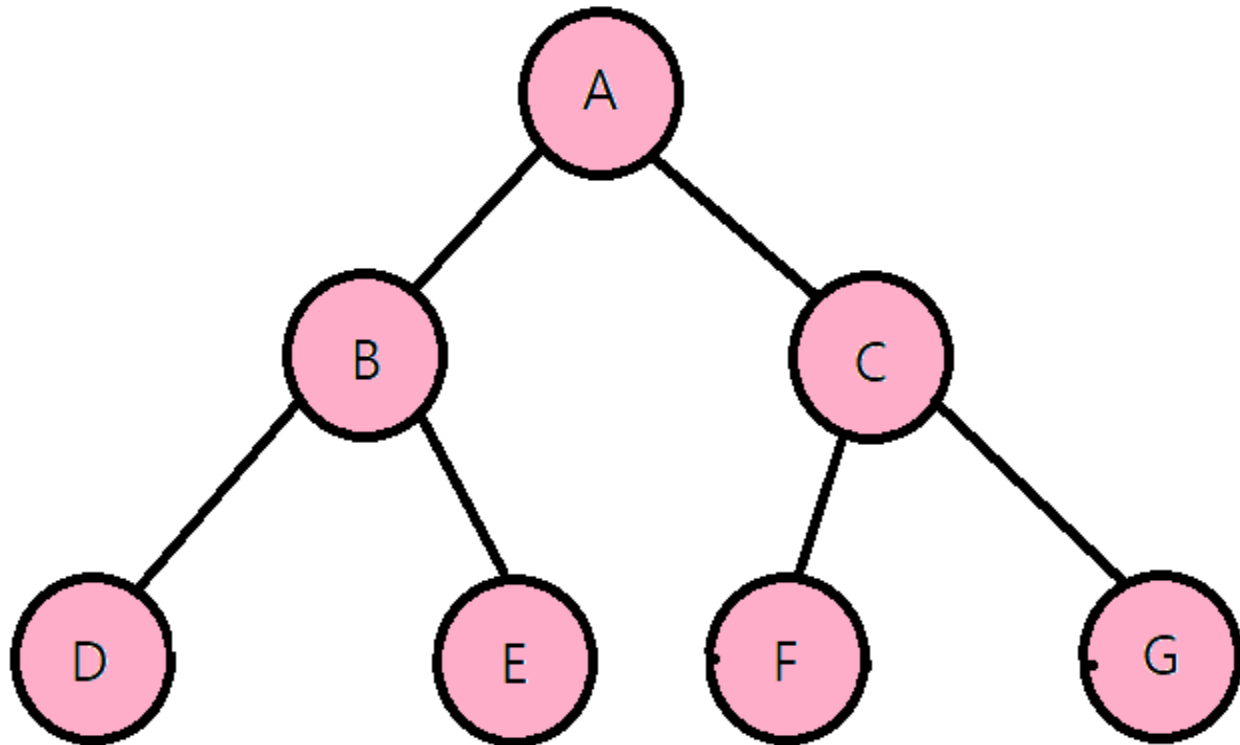
dqueue

## 문제27



\*답

너비 우선 신장 트리 : A-B-C-D-E-F-G



QUEUE



끝



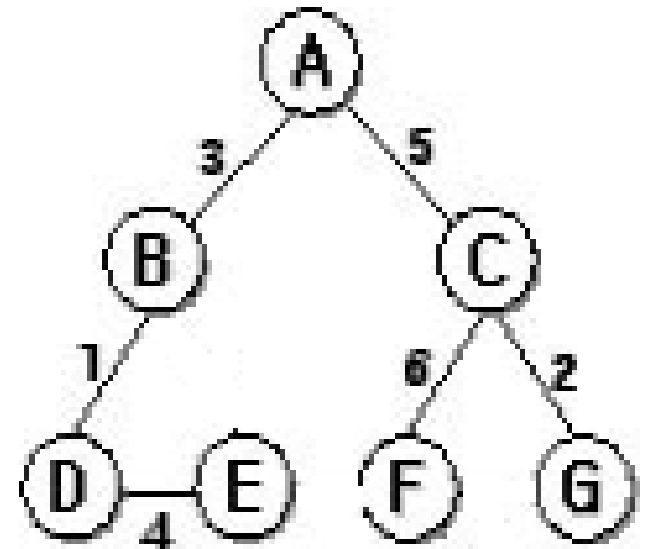
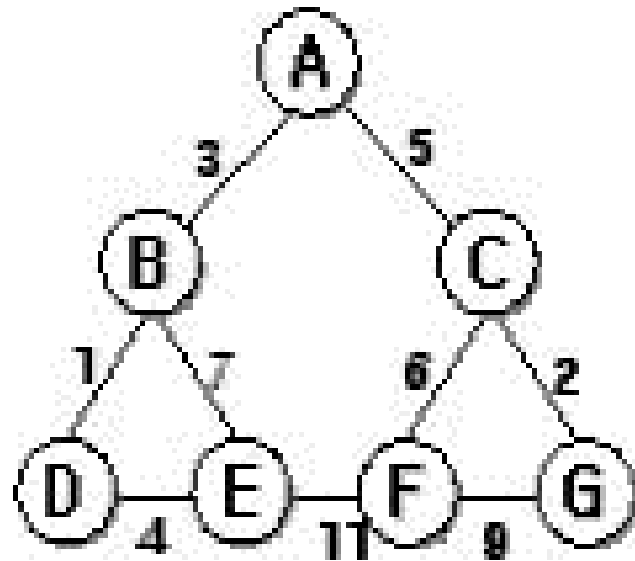
## 문제 28

◆ 다음 가중치 그래프의 최소 비용 신장 트리를 구하여라.

크루스컬 알고리즘을 사용하여, 가중치가 가장 적은 간선을 선택

1. B->D를 연결 (가중치 1)
2. C->G를 연결 (가중치 2)
3. A->B를 연결 (가중치 3)
4. D->E를 연결 (가중치 4)
5. A->C를 연결 (가중치 5)
6. C->F를 연결 (가중치 6)

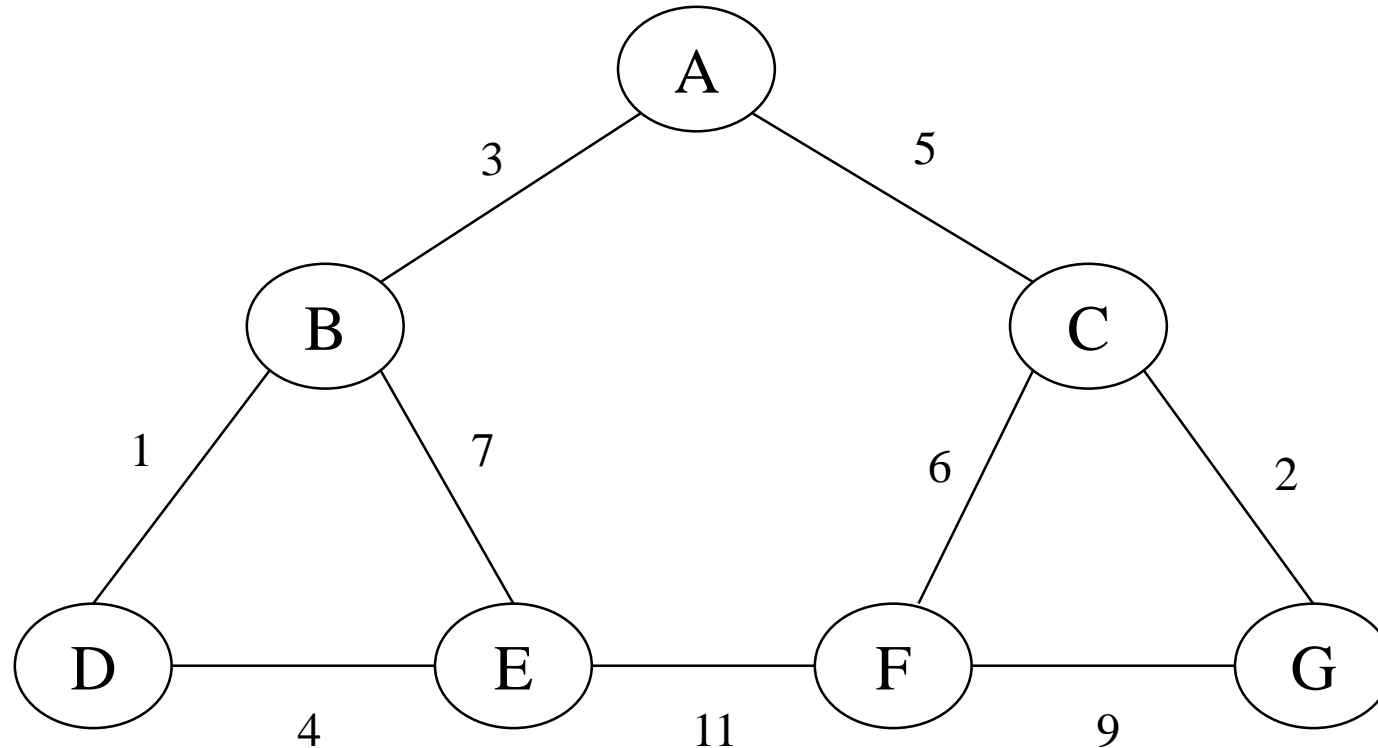
모든 정점들이 연결되었고 순환되지 않는다.



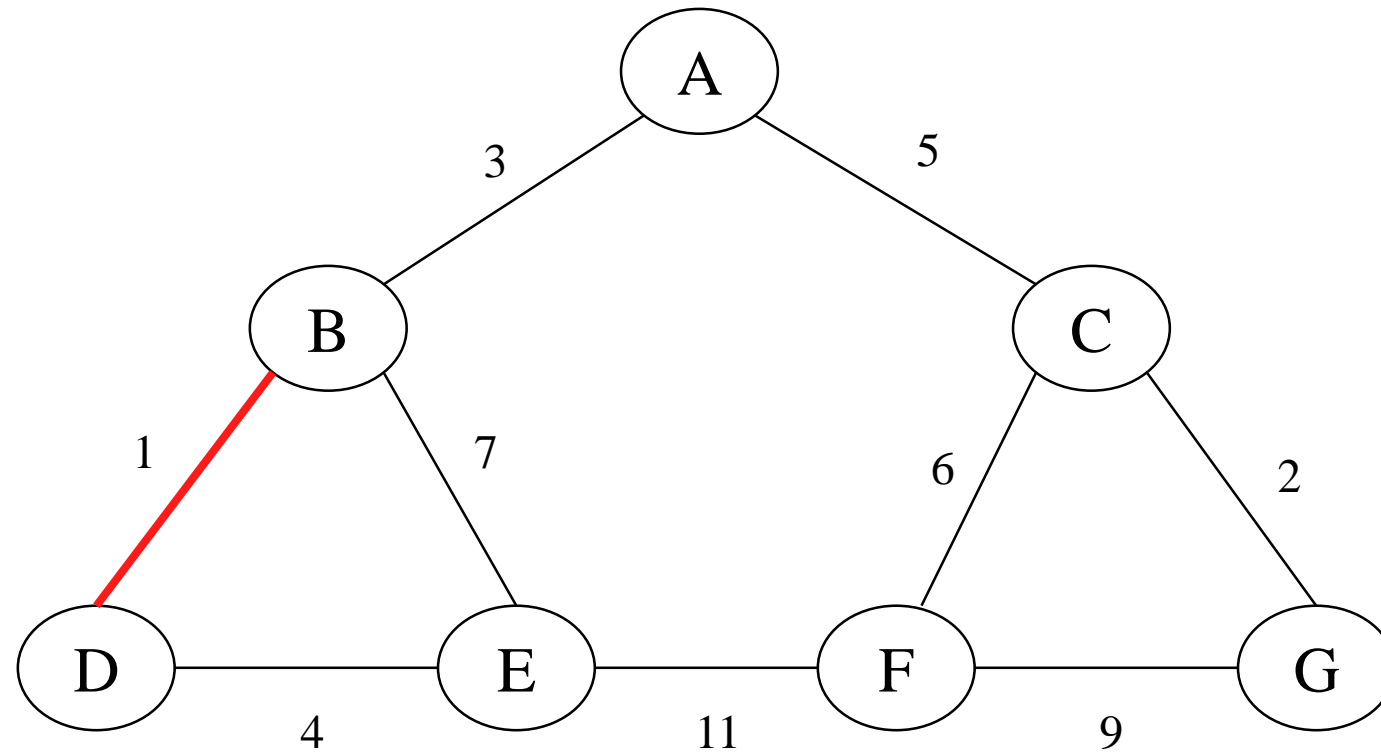
## 문제 28

Q. 다음 가중치 그래프의 최소 비용 신장 트리를 구하여라.

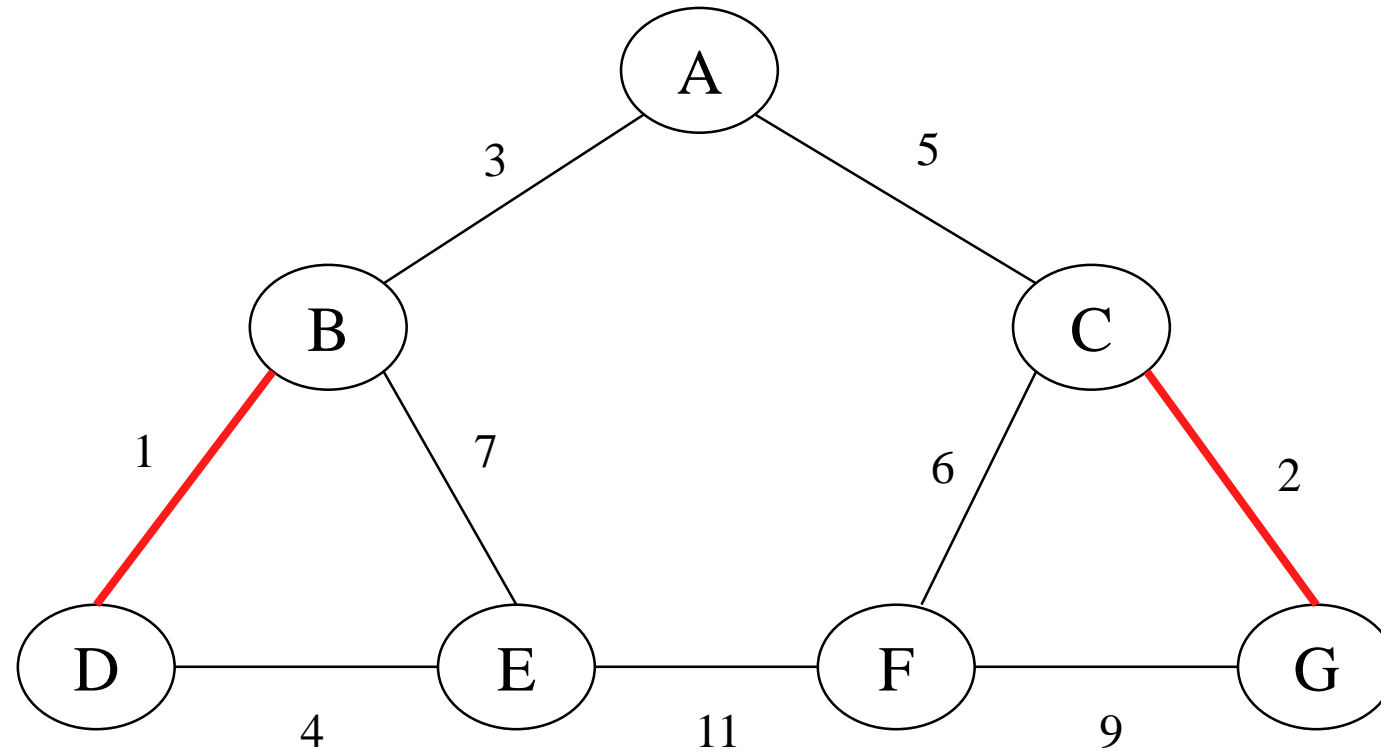
Kruskal's 알고리즘을 이용하여 간선을 가중치가 작은 순으로 정렬해보면



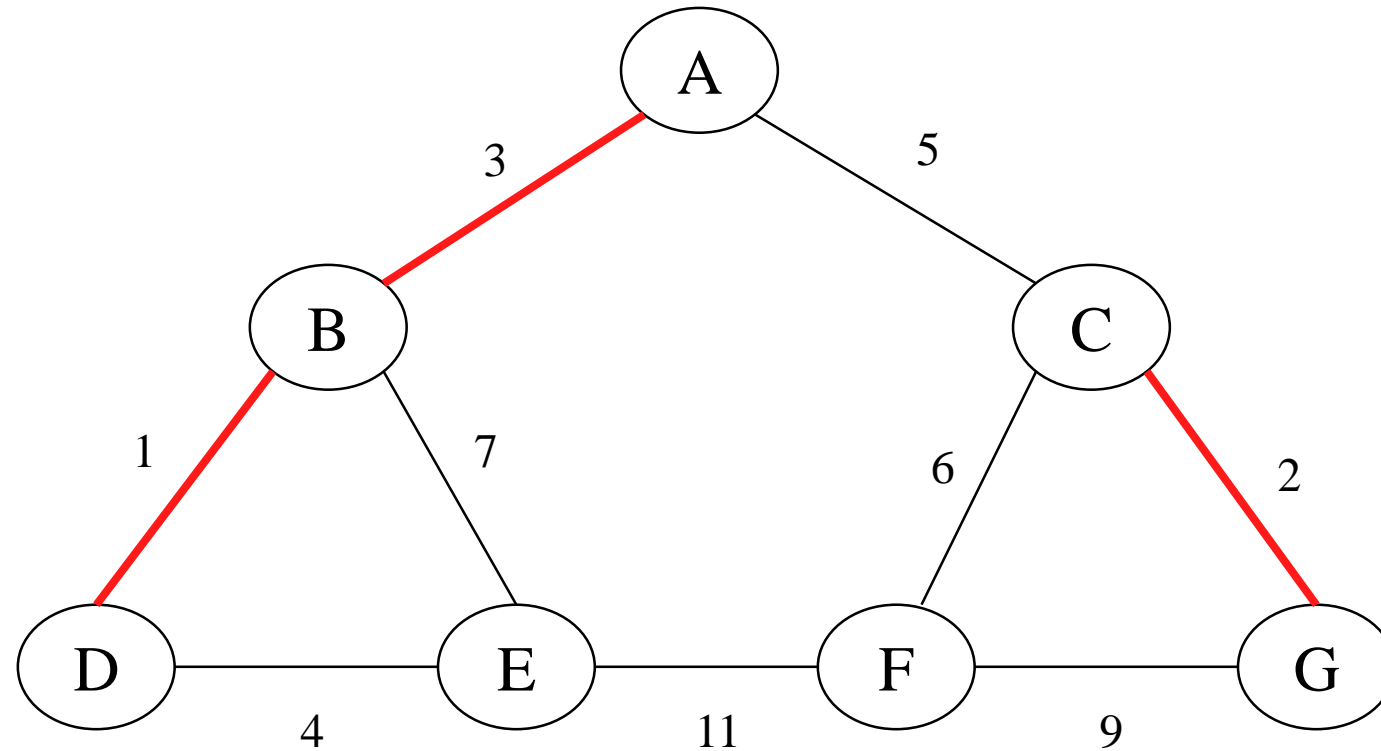
## 문제 28



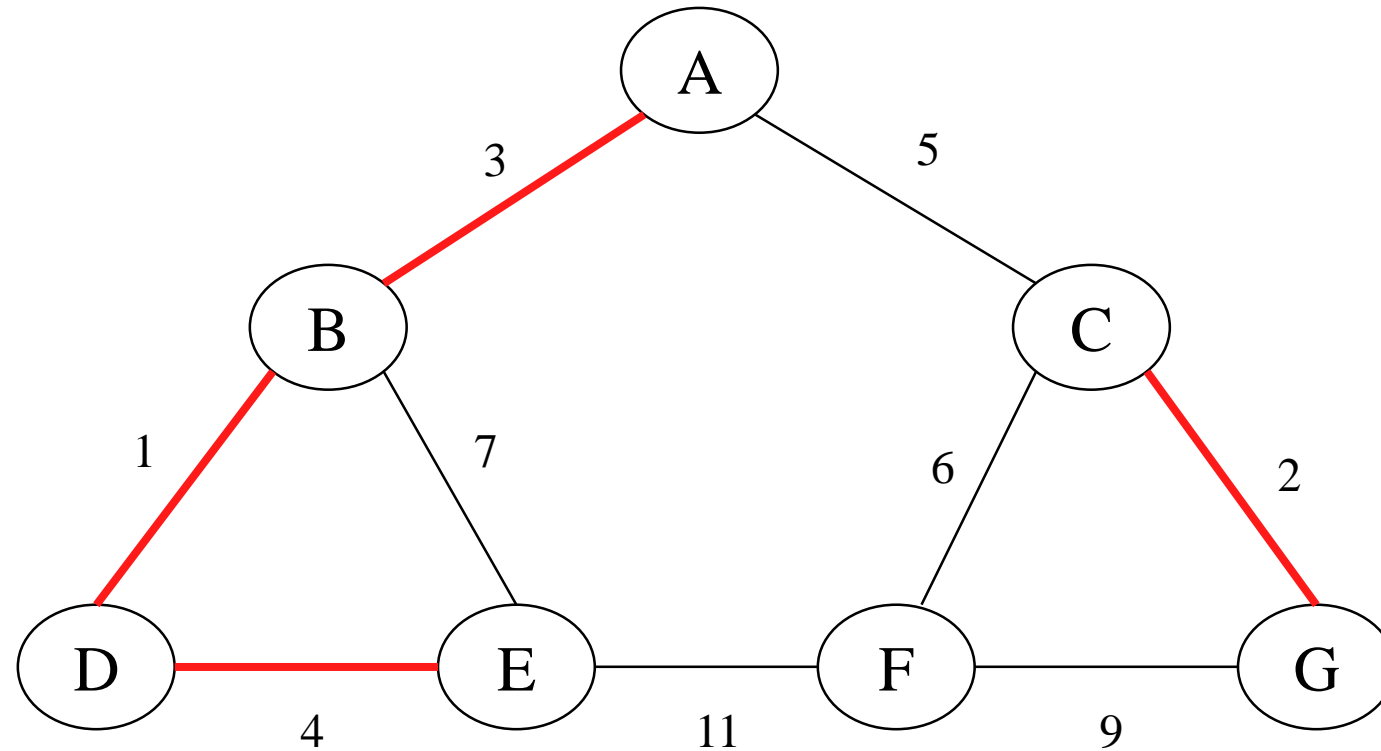
## 문제 28



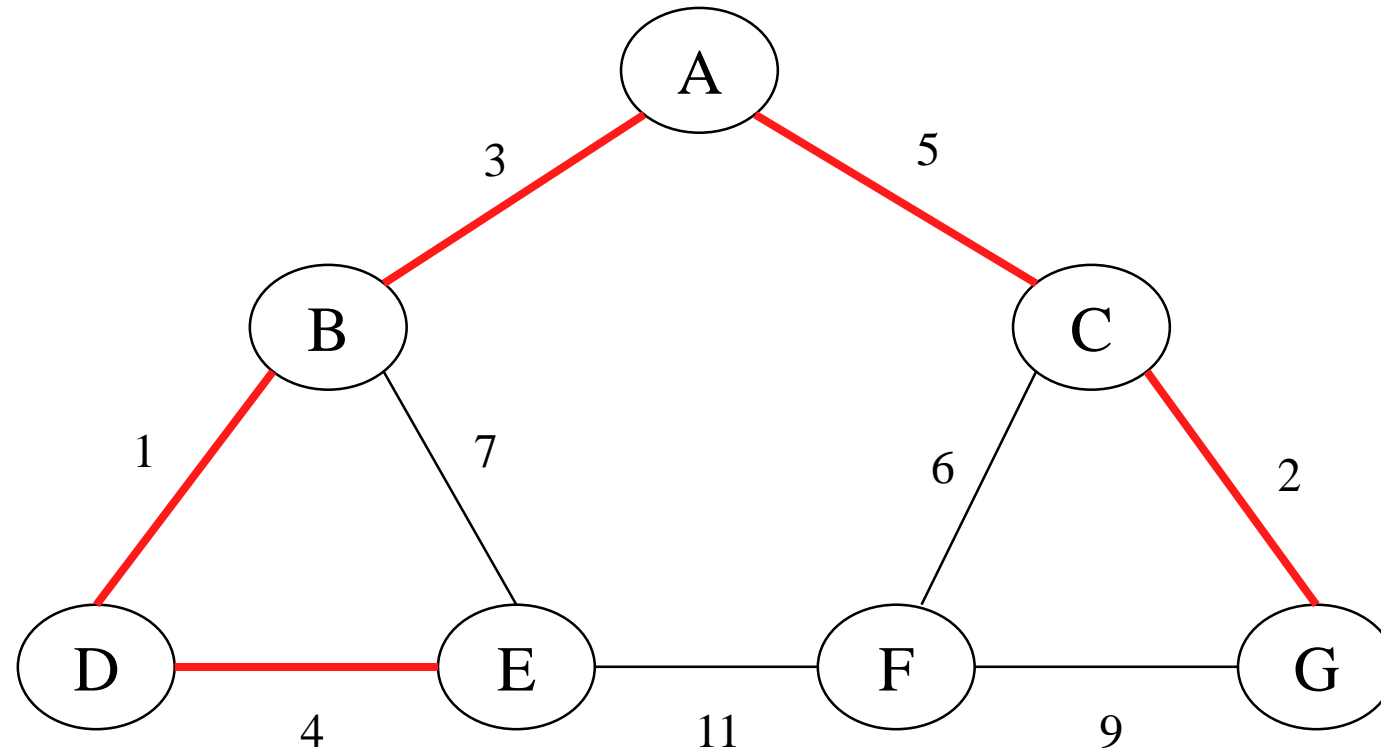
## 문제 28



## 문제 28

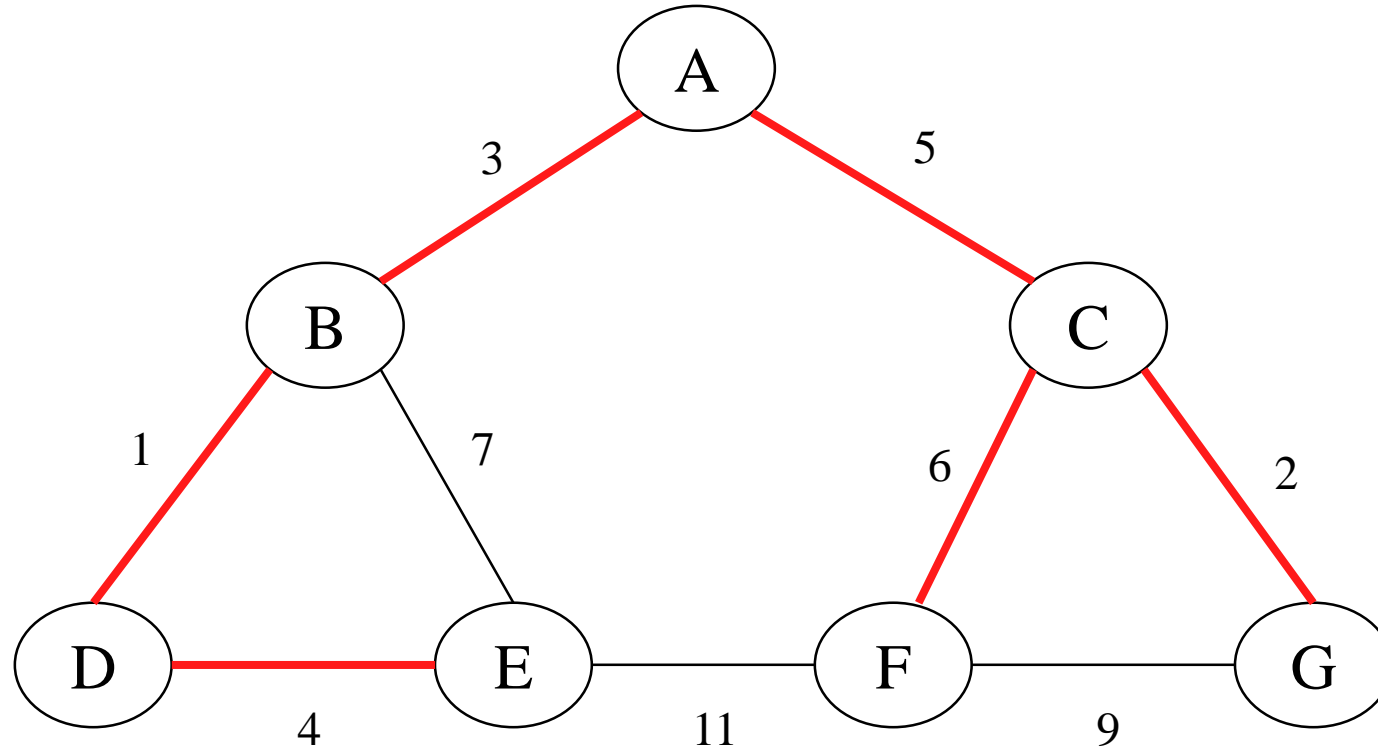


## 문제 28



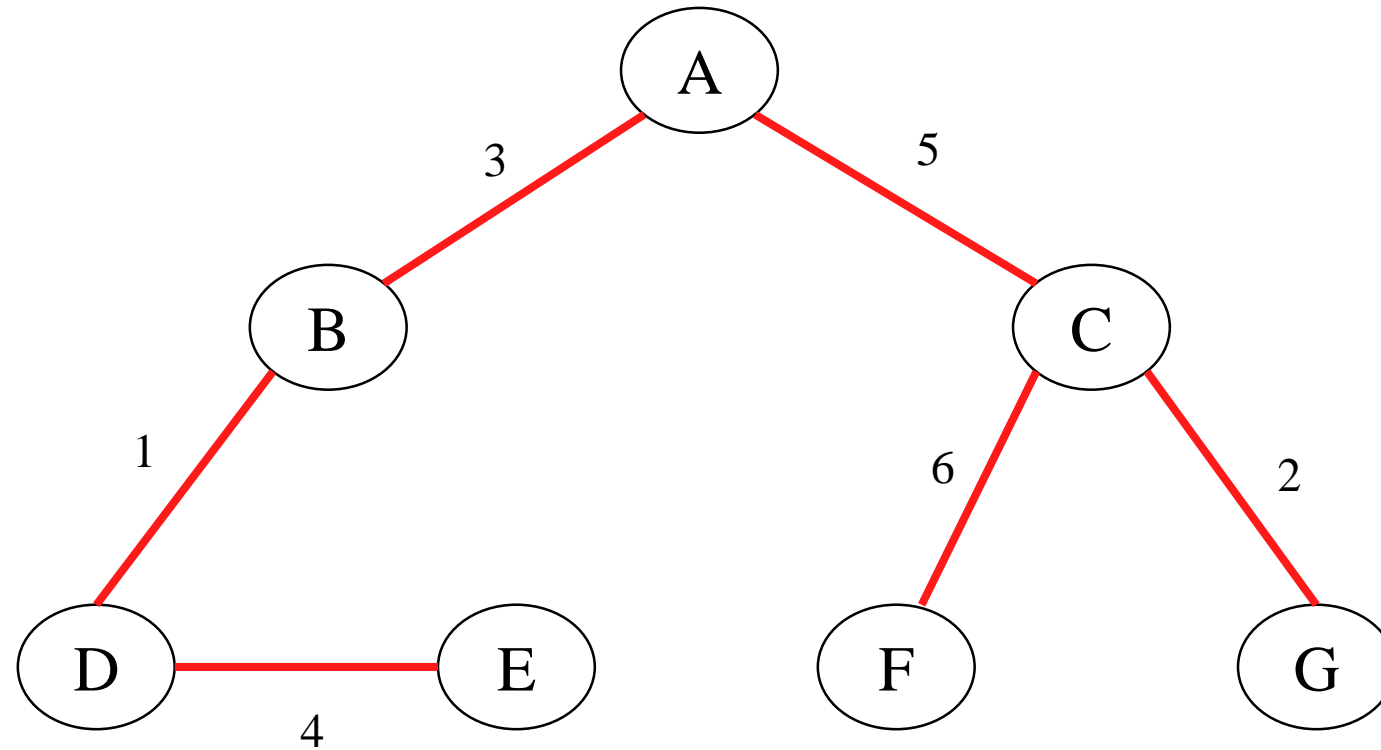
## 문제 28

사이클을 형성하지 않고 모든 정점이 연결되었으므로 종료





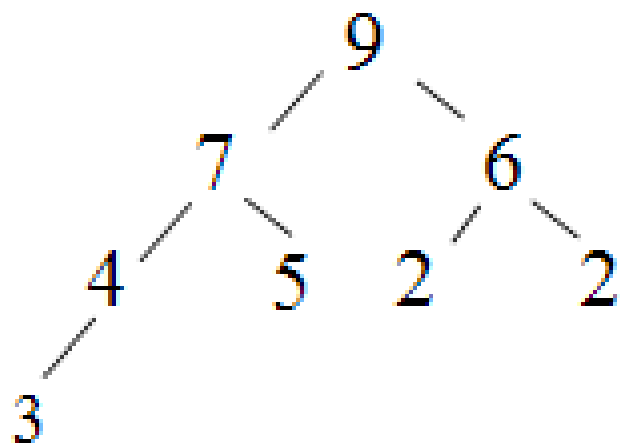
## 문제 28



## 문제 29

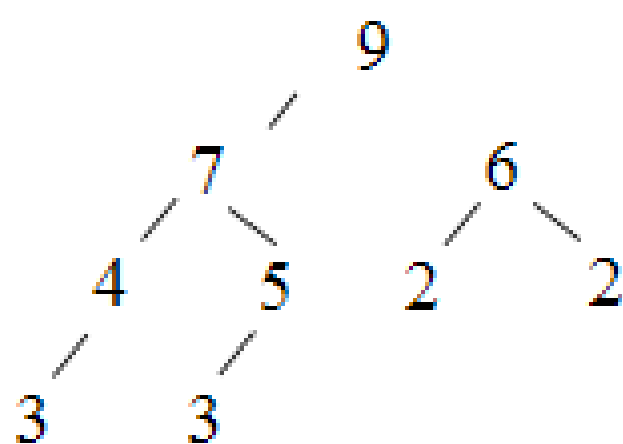
- 다음과 같은 key 값이 있다고 하자.
- Key: 3, 7, 2, 9, 5, 6, 2, 4
- 위의 key 값을 max heap에 순서대로 삽입하였을 때의 결과 heap을 그리시오.
- 또한 최대값을 삭제하는 deleteMax()를 실행한 다음의 구조도 그리시오.

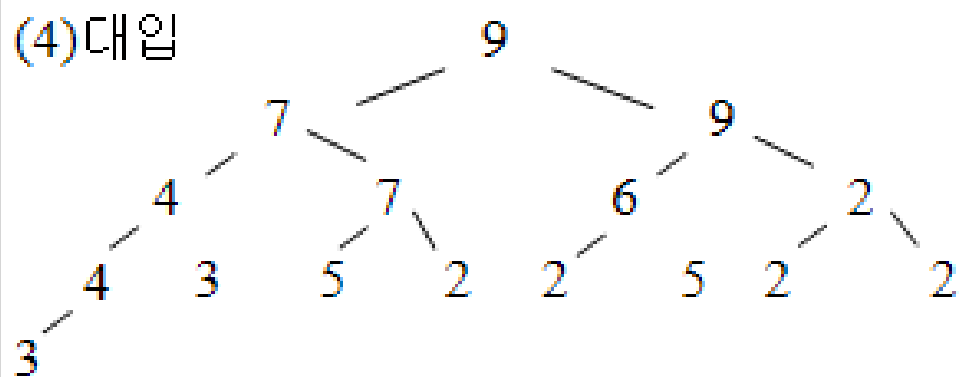
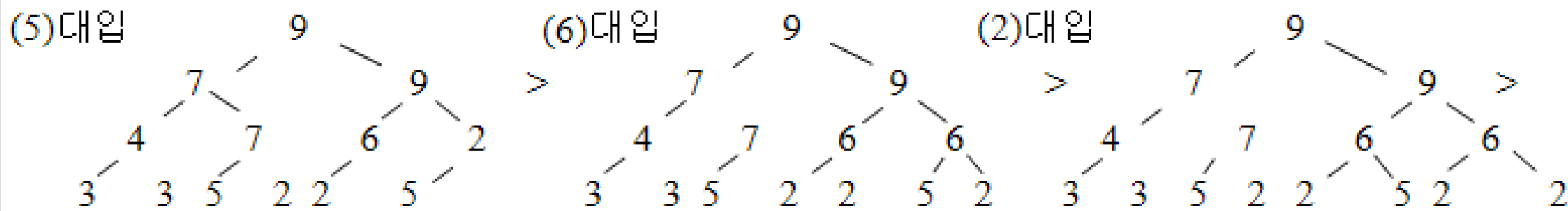
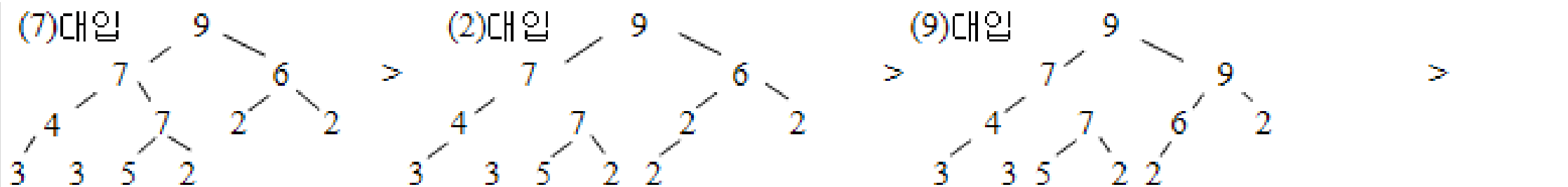
key 값을 max heap에 차례대로 삽입한 경우



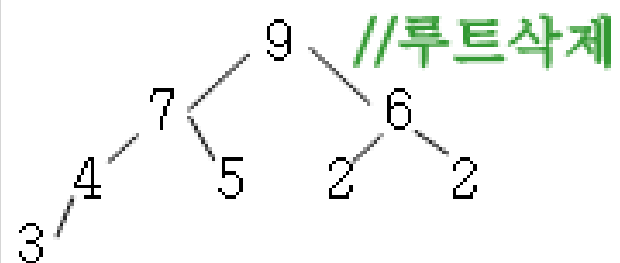
>

(3)대입

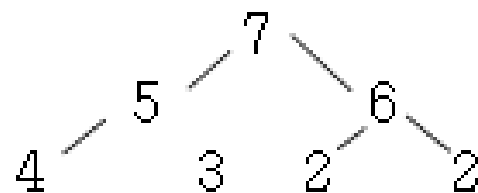




- ◆ 최대 값을 삭제하는 deleteMax()를 실행한 다음의 구조



맥스힙 실행  
->

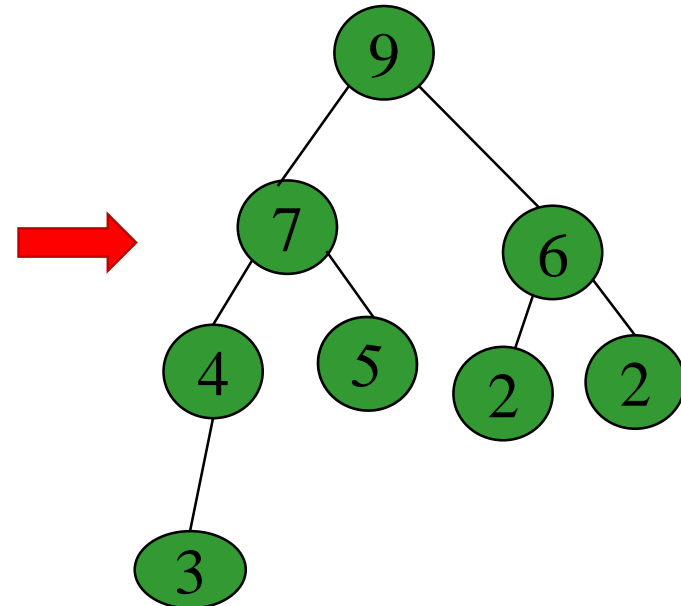
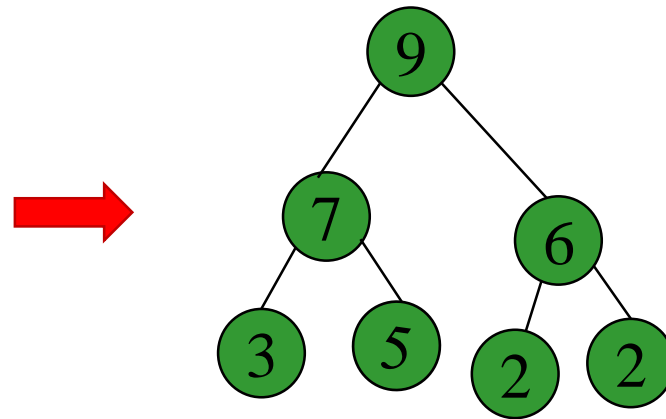
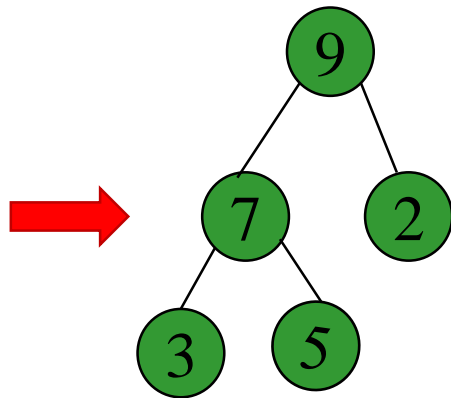
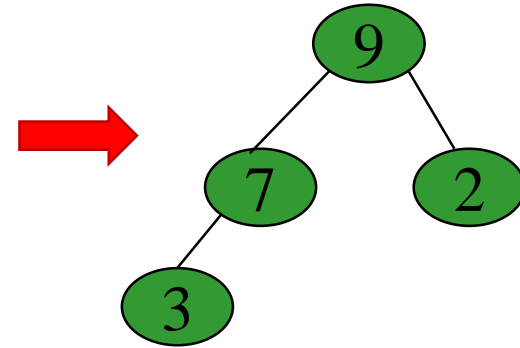
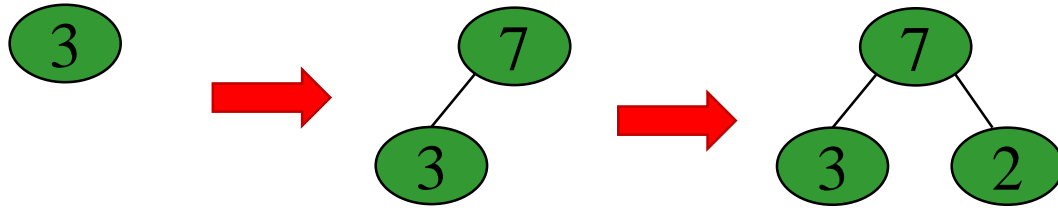


//3을 루트로 올림

정답!

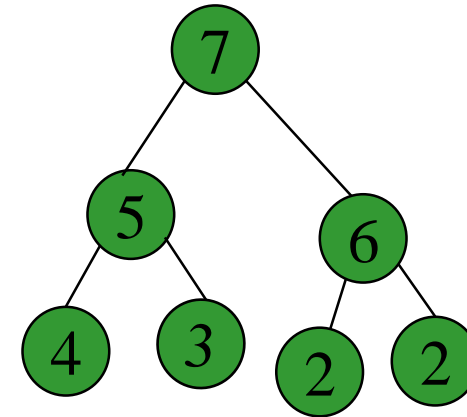
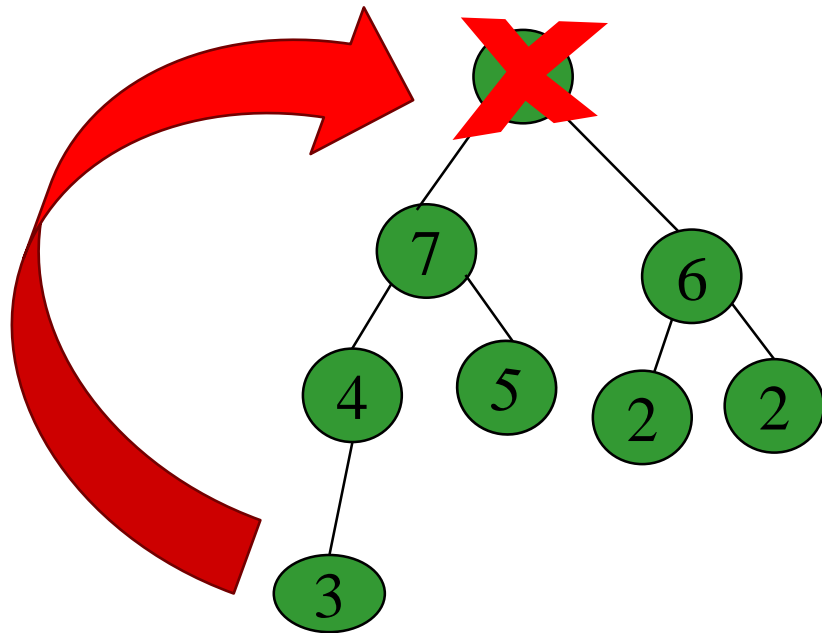
1. Max Heap 에 삽입

\* Key : 3, 7, 2, 9, 5, 6, 2, 4



정답!

## 2. Delete Max()



## 문제 30

- ♦ 30.Binary search tree (이진 탐색 트리) 를 구성하는 방법에 대해서 논하고 이의 worst case와 average case search cost 에 대해서 논하시오.

이진 탐색 트리를 구성하는 방법:

각 노드의 왼쪽 서브트리에는 해당 노드의 값보다 작은 값을 지닌 노드들로 이루어져 있다.

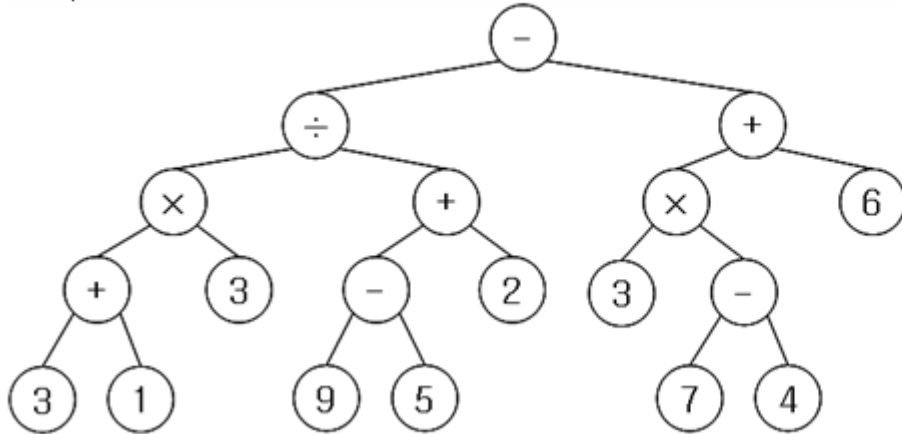
각 노드의 오른쪽 서브트리에는 해당 노드의 값보다 큰 값을 지닌 노드들로 이루어져 있다.

중복된 노드는 없으며 왼쪽 서브트리, 오른쪽 서브트리 또한 이진 탐색 트리이다.

- ♦ worst case : 경사이진트리 시간복잡도 :  $O(n)$
- ♦ average case search cost :  $O(\log N)$

# 문제 31

31 다음의 트리를 postorder와 preorder로 각각 순행하였을 때의 결과 식을 쓰시오.



◆ 답

- preorder 결과 :  $-/x+313+-952+x3-746$  (부모노드>왼쪽자식노드>오른쪽자식노드)
- postorder 결과 :  $31+3x95-2+/-74-3x6+-$  (왼쪽자식노드>오른쪽자식노드>부모노드)



## 문제 32

32. 그래프 자료구조를 저장하는데 크게 Linked list를 이용하는 방법과 Adjacency matrix를 이용하는 방법이 있다. 각각의 장단점을 설명하시오.

### -LinkedList

장점 : 자료의 삽입과 삭제가 용이하다.

단점 : 포인터의 사용으로 인해 저장공간의 낭비가 있다.

탐색시간이 많이 소요된다.

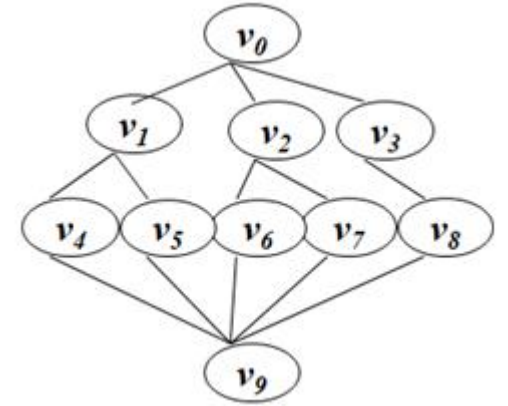
### -Adjacency matrix

장점 : 간단한 그래프를 표현하기에 적합하며 탐색도 비교적 쉽게 가능하다.

단점 : 희소행렬의 경우 공간의 낭비

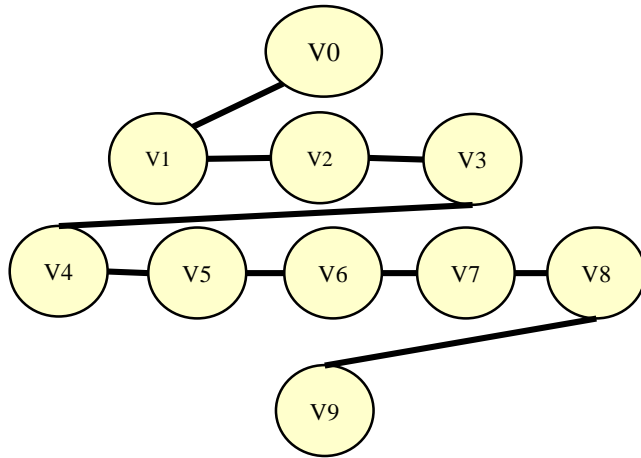
## 문제 33

33. 다음의 그래프가 주어졌을 때, depth-first search와 breadth-first search를 통해 생성되는 spanning tree를 그리시오. 단, 노드  $v_0$ 를 시작 노드로 한다.

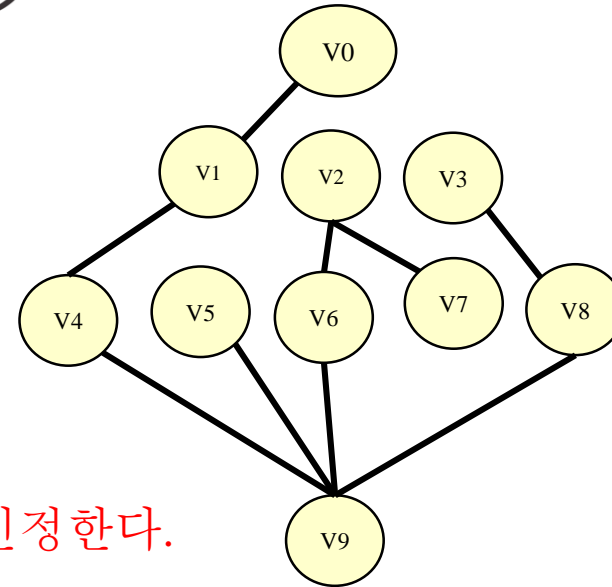


◆ 답

BFS



DFS



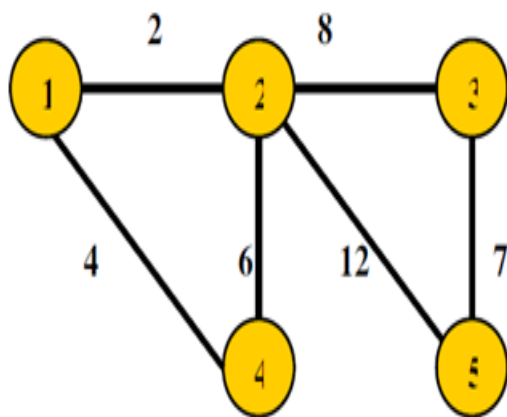
큐와 스택의 상태를 보여주어야 답으로 인정한다.

# 문제 34

34. 다음의 그래프에 대해 Kruskal's 알고리즘과 Prim's 알고리즘을 이용하여 minimum cost spanning tree(최소비용 간선 트리)를 생성하고자 한다. 각 알고리즘의 수행 과정 및 결과를 보여라

< 최소비용 간선 트리 >

1. 모든 정점(노드)이 연결되어 있어야 한다.
2. 자기 자신으로 돌아오는 사이클이 없어야 한다.
3. 가중치 합이 최소가 된다.
- 4 종류 : (Kruskal 알고리즘, Prim 알고리즘 )



<Kruskal 알고리즘>

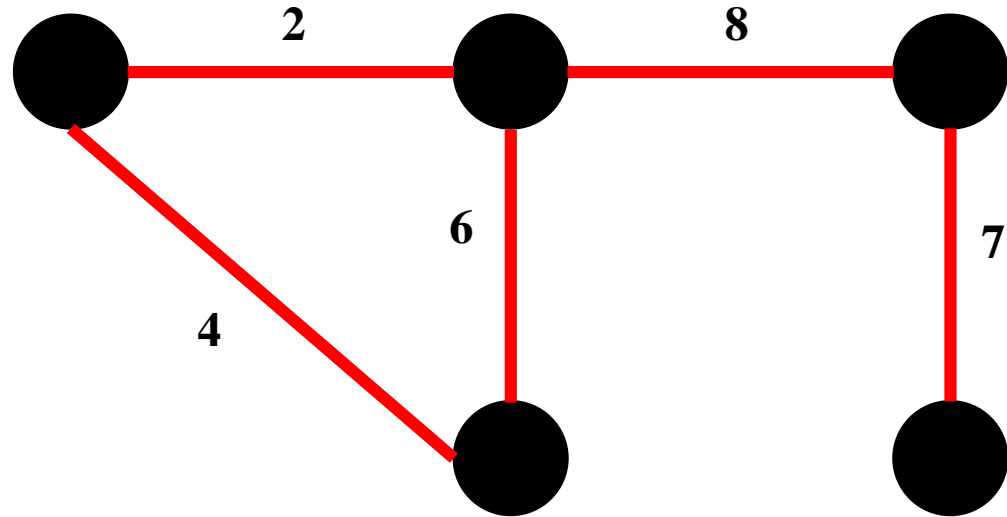
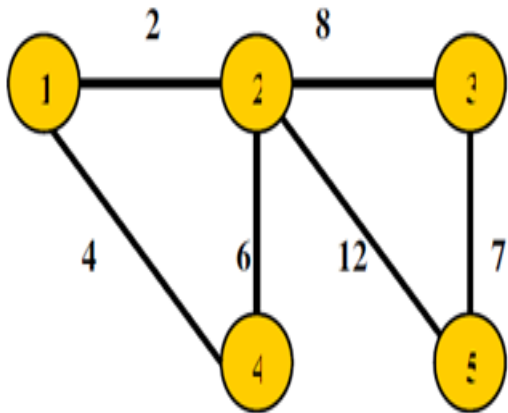
1. 여러 트리가 합쳐져 나중에는 하나의 트리가 되는 것
2. 간선이 작은 순으로 정점이 연결된다.
3. (정점 -1) 개의 간선을 가진다.

<Prim 알고리즘>

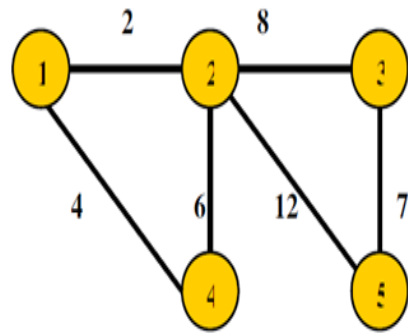
1. 단일 트리의 형태이다.
2. 단일 정점에서 이웃한 정점과의 가중치를 비교해 나가면서 트리가 커지는 형식을 가진다.

# Kruskal 알고리즘

1. 가장 작은 가중치를 가진 간선부터 시작한다.
2. 그 다음 작은 간선을 이어준다.
3. 사이클이 생기지 않도록 한다.



# Prim 알고리즘



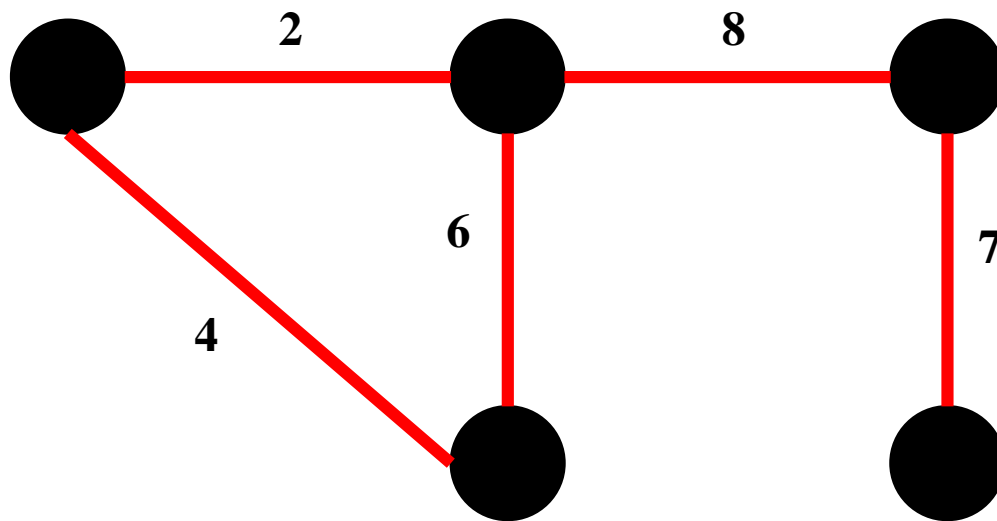
1. 처음은 하나의 정점을 선택한다.

2. 이웃한 정점과 가중치를 비교한다.

3. 비교한 간선 중 가장 작은 간선을 선택하여 이어 준다.

4. 연결된 정점들을 하나의 트리로 본다.

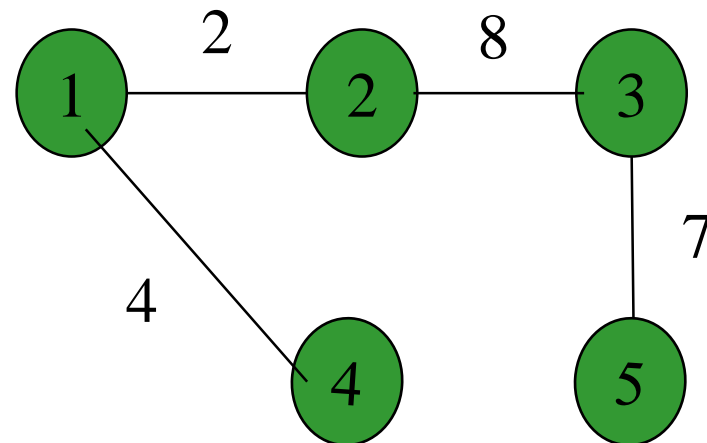
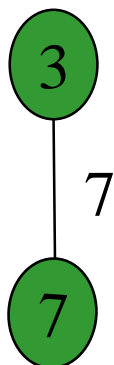
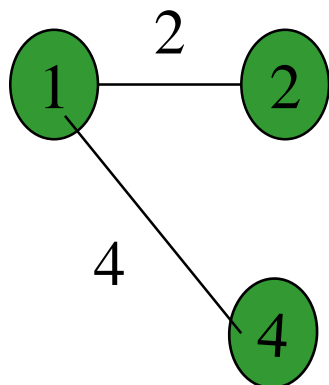
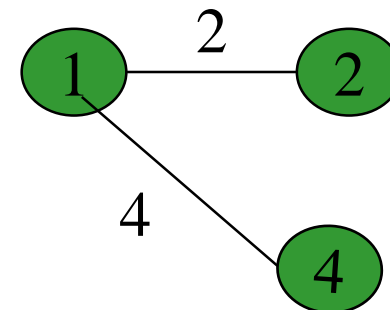
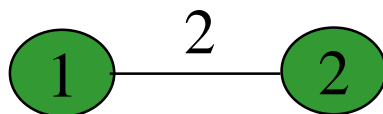
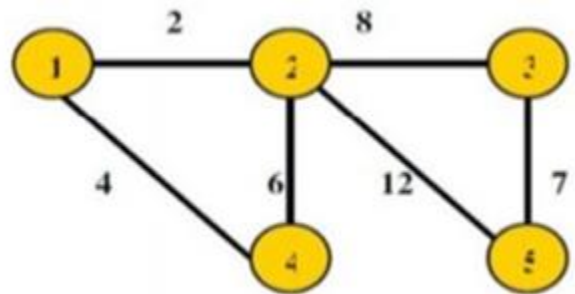
5. 트리의 인접한 간선 중 가중치가 가장 작은 간선을 선택하여 이어 나간다.



정답!

**Kruskal's Algorithm** : 2 4 7 8 : Total weight of tree : 21

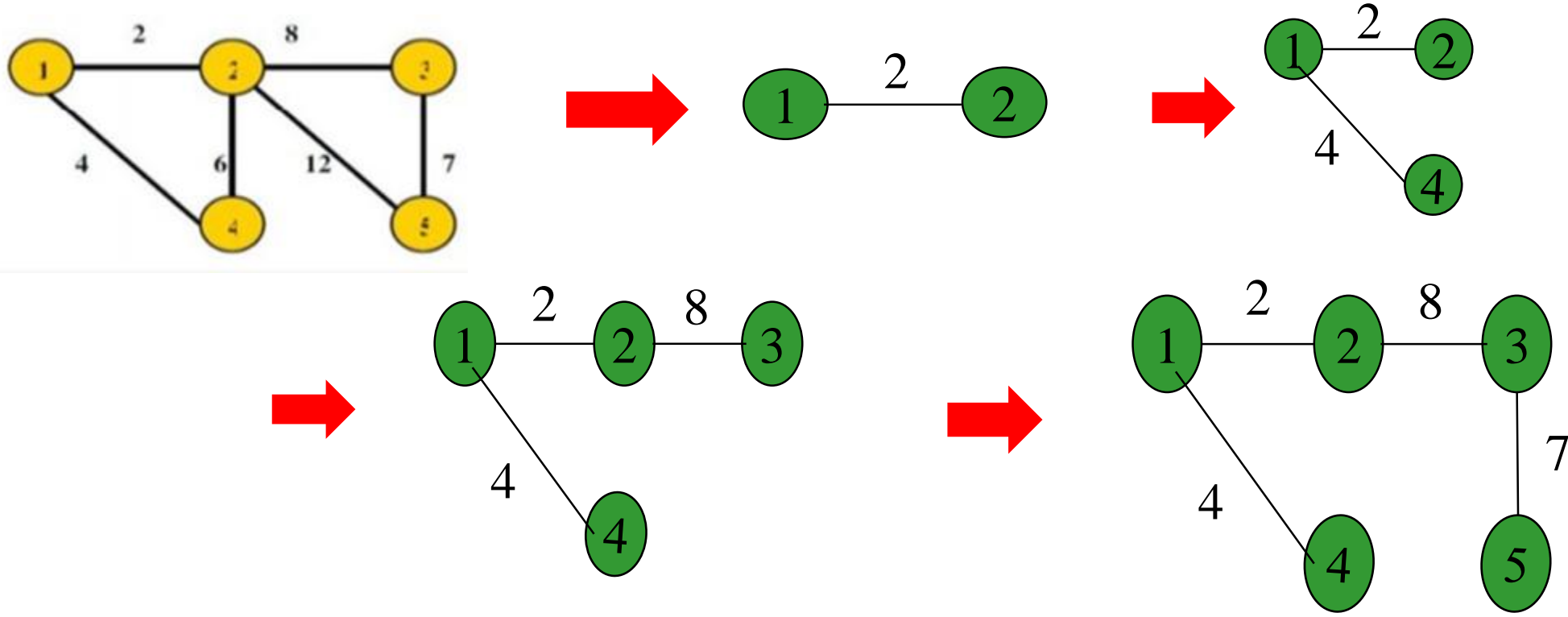
가장 적은 비용이 드는 간선들을 사이클 만들어지는 간선 제외  
하고 선택했을 경우



정답!

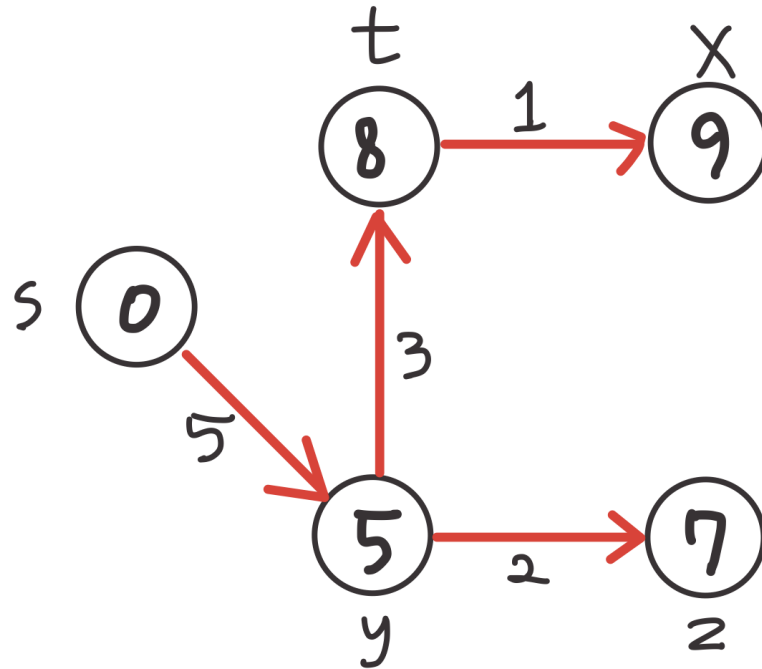
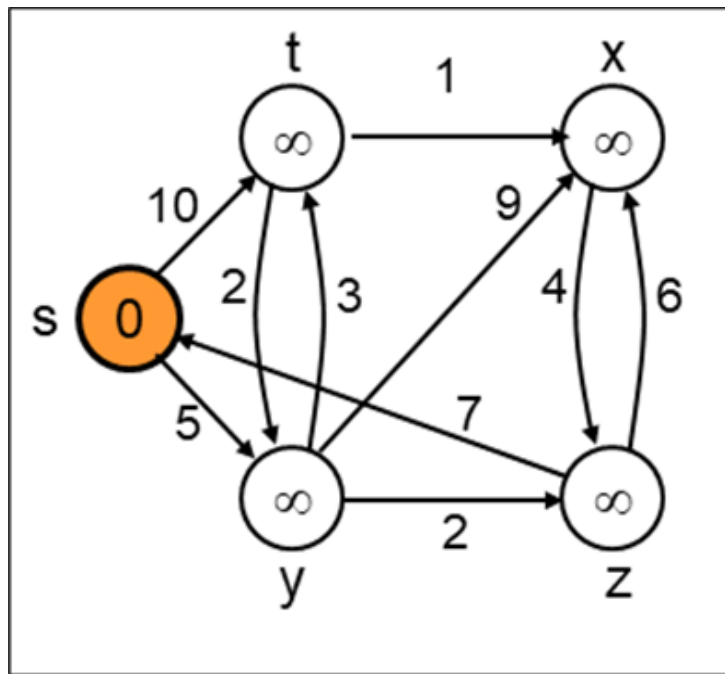
Prim's Algorithm : 2 4 8 7 : Total weight of tree : 21

정점을 선택하고 그것과 연결된 가장 적은 비용의 간선들을 사이클이 만들어지는 간선을 제외하고 선택했을 경우



# 문제 35

35. 다익스트라의 알고리즘을 이용하여 최단경로를 구하시오. 출발점은 정점 0.



Q: S T Y X Z

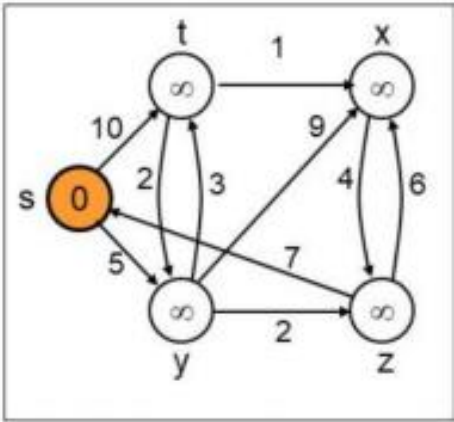
0	$\infty$	$\infty$	$\infty$	$\infty$
	10	5	$\infty$	$\infty$
	8		9	7
	8		13	9

S: {s, y, z, t, x}

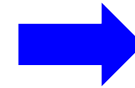
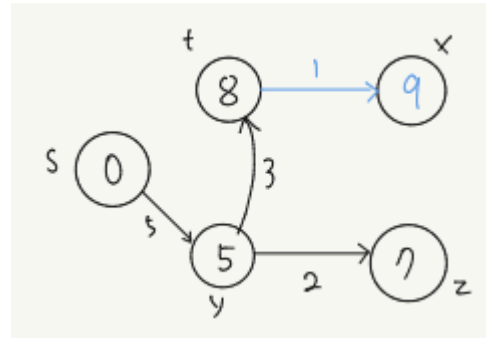
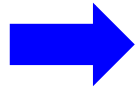
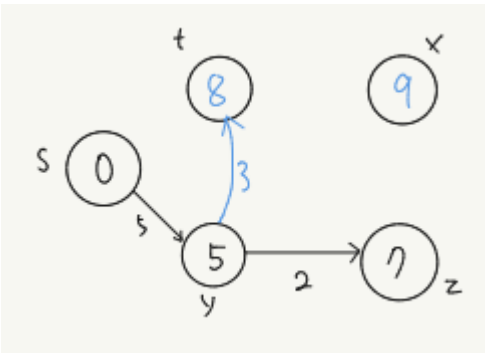
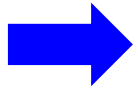
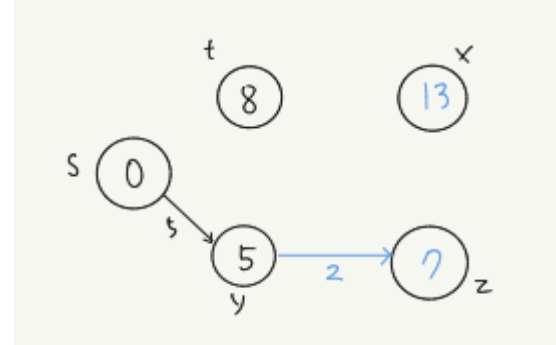
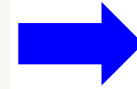
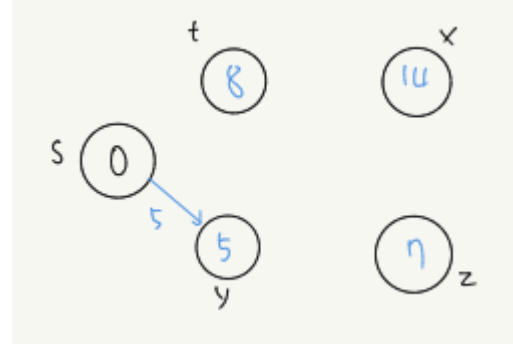
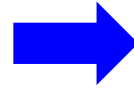
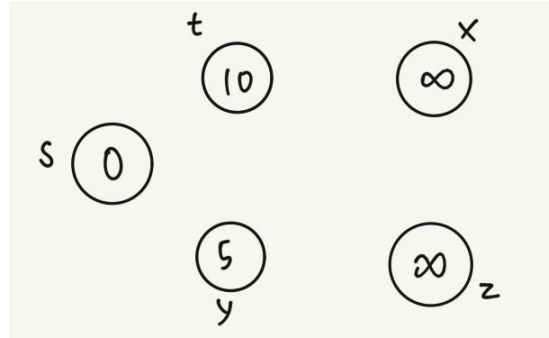
y	t	z	x
5	8	7	9



정답!



=



**S**{s,y,z,t,x}  
**Dist**=0+5+2+3+1=11