

2021학년도 2학기 자료구조 기말고사

2021. 12. 9

점수

학번:

성명:

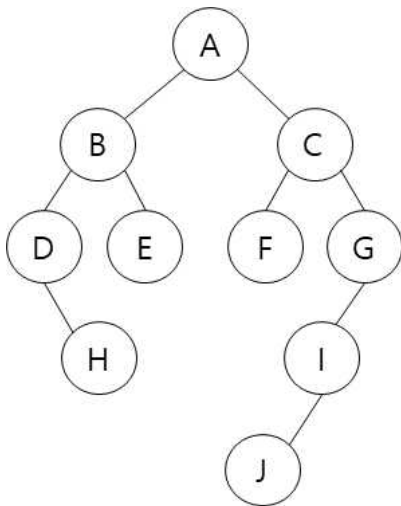
※ (1 - 20) 아래 문제를 읽고 물음에 답하시오. [100점 각 5점]

1. 다음 자료 구조 중에서 비선형 구조로만 묶은 것은?

- | | |
|------------------------|---------------|
| ㄱ. 스택(stack) | ㄴ. 트리(tree) |
| ㄷ. 연결 리스트(linked list) | ㄹ. 그래프(graph) |

- ① ㄱ, ㄴ ② ㄱ, ㄷ
③ ㄴ, ㄷ ④ ㄴ, ㄹ

2. 다음과 같은 이진트리(binary tree)에서 전위순회(preorder traversal)와 중위순회(inorder traversal)를 했을 때, 두 순회 결과에서 노드 값의 방문 순서가 일치하는 횟수는? (전위순회의 k번째 노드 값과 중위순회의 k번째 노드 값이 같을 때, 일치하는 횟수를 1회로 한다)



- ① 0회 ② 1회
③ 2회 ④ 3회

3. N개의 노드를 가진 완전 이진 트리(complete binary tree)에 대한 설명으로 옳지 않은 것은? (단, 루트(root)노드의 높이는 1로 한다)

- ① 리프(leaf) 노드의 개수는 리프노드가 아닌 노드의 개수와 같거나 하나가 더 많다.
② 높이가 K라면 노드 개수 N의 범위는 $2^{K-1} \leq N \leq 2^K - 1$ 이다.
③ Full binary tree는 완전 이진 트리의 일종이다.
④ 완전 이진 트리를 최악으로 구성할 경우 높이는 $\lceil \log_2(N+1) \rceil$ 이다.

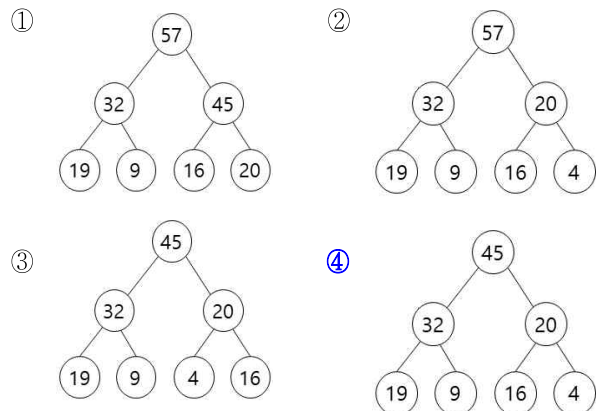
4. 최소힙(min heap)으로 우선순위 큐를 구현하려 한다. 우선순위를 나타내는 8개의 데이터가 큐에 다음과 같은 순서대로 삽입되었다. 1개의 데이터가 큐에서 삭제된 후, 재 정렬된 힙에서 가장 마지막 원소는 무엇인가? (단, 숫자가 작을수록 우선순위가 높다고 가정한다)

24, 17, 29, 22, 31, 27, 18, 21

- ① 21 ② 27
③ 29 ④ 31

5. 다음과 같은 키 값을 가지는 원소들을 공백 힙(empty heap)에 차례대로 삽입하여 최대 힙(max heap)을 생성한 다음, 삭제 연산을 한 번 수행한 후의 결과는?

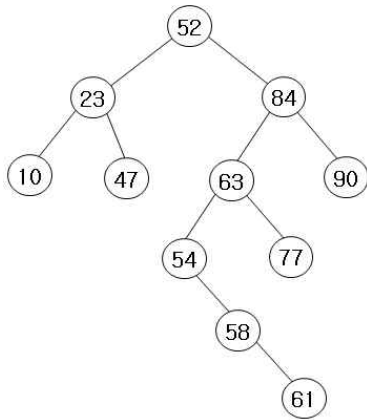
4, 16, 57, 9, 32, 45, 20, 19



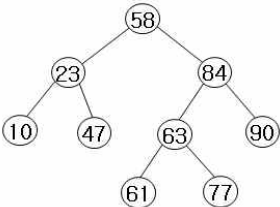
6. N 개의 키를 가진 최대힙에서 루트노드를 삭제하는데 소요되는 수행시간은?

- ① $O(1)$ ② $O(\log N)$
 ③ $O(N)$ ④ $O(N \log N)$

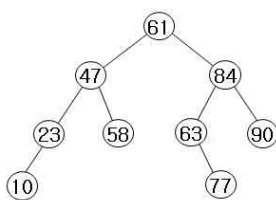
7. 다음 이진 탐색트리(binary search tree)에서 루트노드가 2번 삭제된 후의 상태는?



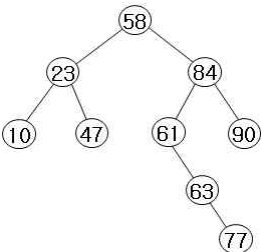
①



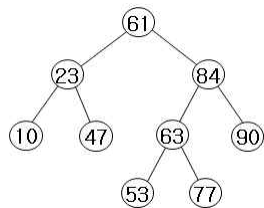
②



③



④



8. 다음과 같은 9개의 키 값을 순서대로 입력하여 생성한 이진 탐색 트리(binary search tree)의 높이는?

17, 16, 22, 15, 30, 24, 20, 11, 14

- ① 3 ② 4
 ③ 5 ④ 6

9. 다음은 이진 탐색 트리(binary search tree)에서 최소 키 값을 가지는 노드에 대한 포인터를 반환하는 메소드를 파이썬으로 구현한 프로그램의 일부이다. (가)와 (나)부분에 해당되는 문장이 바르게 나열된 것은?

```
class TreeNode:
    def __init__(self, key, value, left=None, right=None):
        self.key = key
        self.value = value
        self.left = left
        self.right = right
```

```
class BST:
    def __init__(self):
        self.root = None

    def min(self):
        if self.root == None:
            return None
        return self.minimum(self.root)
```

```
def minimum(self, n):
    if n.left == None:
        ____ (가) ____
    else :
        ____ (나) ____
```

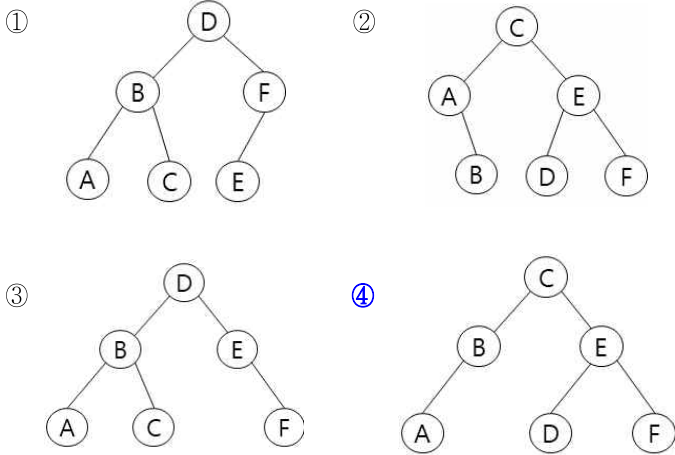
(가)

(나)

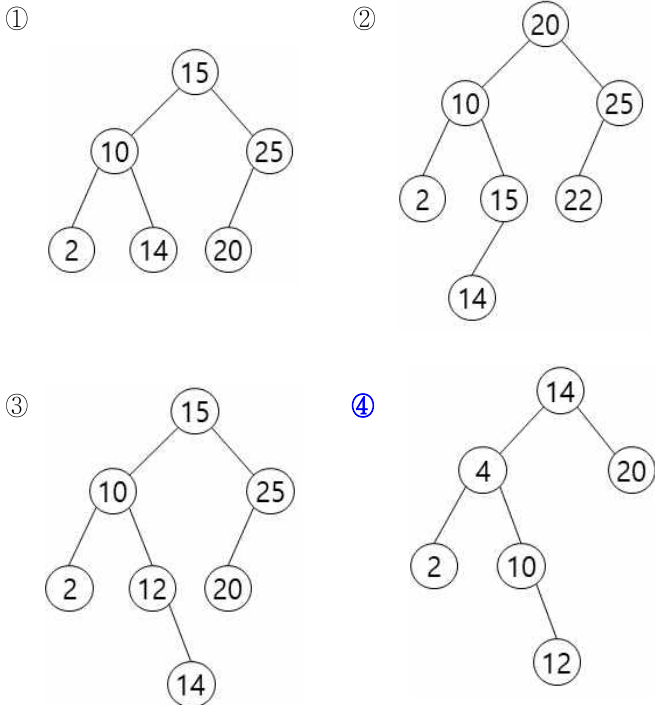
- | | |
|--------------------------------|-----------------------------|
| ① return n | return n.left |
| ② return n.right | return n.left |
| ③ return n | return self.minimum(n.left) |
| ④ return self.minimum(n.right) | return self.minimum(n.left) |

10. 다음과 같은 순서로 입력되는 키 값에 대해 올바르게 AVL-트리를 구성한 것은?

A F B E C D



11. 다음 중 AVL 트리가 아닌 것은?



12. 다음 키 값을 이용하여 AVL 트리를 생성할 때, 이에 대한 설명으로 옳지 않은 것은?

12, 11, 15, 5, 3, 7, 6, 1, 13, 8, 4

- ① AVL트리에서 7을 검색하기 위해서는 4번의 비교가 필요하다.
- ② AVL트리의 루트 값은 11이다.
- ③ 7이 삽입될 때, AVL 트리의 균형이 깨져서 재구성이 발생한다.
- ④ 6은 리프노드이다.

13. 다음 인접 행렬로 표현되는 그래프에 대한 설명으로 옳지 않은 것은?

	1	2	3	4
1	0	0	0	1
2	1	0	1	1
3	1	0	0	0
4	0	1	0	0

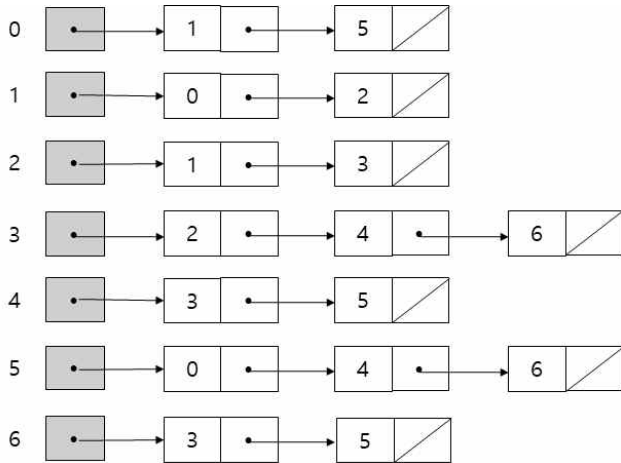
- ① 방향 그래프(directed graph)이다.
- ② 간선(edge)의 개수는 6이다.
- ③ 강력 연결 요소(strongly connected component)는 2개이다.
- ④ 방향 사이클이 존재한다.

14. 다음의 방향성 그래프(directed graph)를 인접행렬로 표현한 것이다. 이에 대한 위상 정렬(topological sorting)의 결과로 옳은 것은? (단, 행(row)은 출발 노드로 열(column)은 도착 노드로 매핑된다)

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

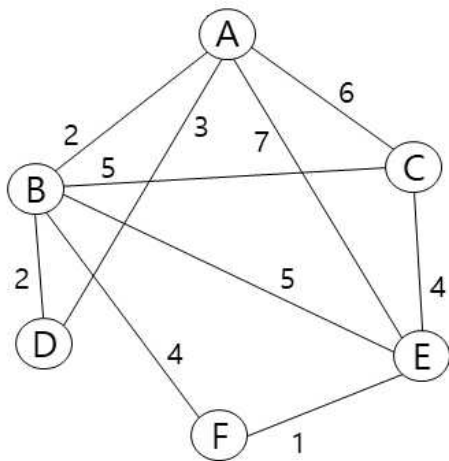
- ① 0, 4, 3, 5, 1, 6, 2
- ② 0, 1, 3, 5, 2, 4, 6
- ③ 6, 2, 1, 5, 3, 4, 0
- ④ 0, 1, 2, 6, 4, 3, 5

15. 다음과 같은 인접리스트로 표현된 그래프에 대해 노드 2를 시작 정점으로 너비우선탐색(breadth first search)을 수행했을 때 노드들의 방문 순서를 올바르게 나열한 것은?



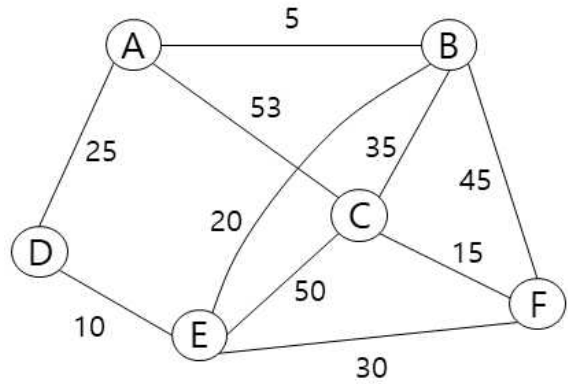
- ① 2 → 1 → 0 → 5 → 4 → 3 → 6
 ② 2 → 1 → 3 → 0 → 4 → 5 → 6
 ③ 2 → 1 → 0 → 5 → 4 → 6 → 3
 ④ 2 → 1 → 3 → 0 → 6 → 4 → 5

16. 다음 비용 그래프에서 최소 비용 신장 트리를 구하고자 한다. Prim 알고리즘을 이용하여 최소 비용 신장 트리를 구할 때, 4번째로 연결되는 간선(edge)으로 옳은 것은? (단, 시작 노드는 A이다)



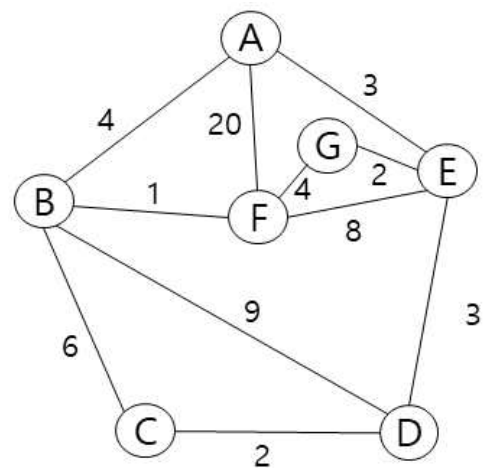
- ① (A, D) ② (B, F)
 ③ (E, F) ④ (F, C)

17. 다음 그래프를 이용하여 Kruskal 알고리즘을 적용하였을 때, 생성된 최소 비용 신장 트리의 비용은?



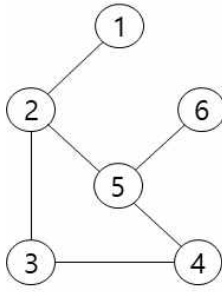
- ① 75 ② 80
 ③ 85 ④ 95

18. Dijkstra 알고리즘을 이용하여 다음 그래프의 정점 A에서 각 정점까지의 최단경로를 구할 때, 최단경로 발견 순서가 정점 G 바로 다음인 정점은? (단, 간선 옆의 수는 각 정점들 사이의 거리를 나타낸다)



- ① 정점 C ② 정점 D
 ③ 정점 E ④ 정점 F

19. 다음 그래프에는 몇 개의 서로 다른 신장트리가 존재하는가?



- ① 4 ② 5
③ 6 ④ 7

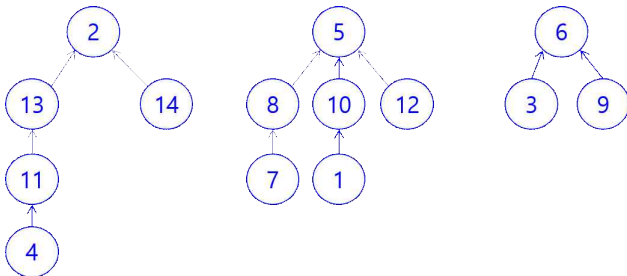
20. 정점을 6개 가진 무방향 완전그래프에 있는 간선의 수는?

- ① 6 ② 12
③ 15 ④ 30

21. 서로 공통된 원소를 갖지 않는 서로소 집합(disjoint set)에 대해서 다음 질문에 답하십시오. (단, 기본연산으로 두 트리 중 하나를 다른 트리의 서브트리로 만들어 주는 Union 연산과 루트에 도달할 때 까지 원소를 포함하는 집합을 탐색하는 Find 연산이 있음) (30점)

(1) 서로소 집합의 배열 표현을 트리(그림)로 표현하십시오. (단, 집합의 트리(그림) 표현 시 각 노드들은 자식에서부터 부모로 가는 링크로 연결됨) (10점)

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
parent	10	-1	6	11	-1	-1	8	5	6	5	13	5	2	2



(2) 서로소 집합의 기본 연산인 Union과 Find는 구현하기 쉬운 반면 성능이 좋지 않으며, 트리의 높이가 커지는 변질 트리(degenerate tree)가 만들어질 수 있다는 단점이 있다. 가중 병합(weighted union)과 경로압축(path compression, 또는 collapsing find)을 사용하여 이를 개선하기 위한 방안을 논하십시오.(12점)

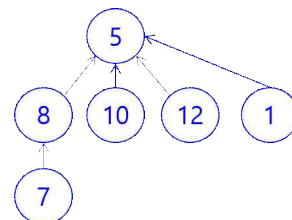
(정답)

1. 가중병합: 그래프의 모든 정점의 부모의 인덱스를 저장하는 **parent 배열(1점)**에 (1)과 같이 자신이 부모임을 나타내는 -1에서 확장하여, -N과 같이 표현하여 **음수는 자신이 부모임을 의미(2점)**하고 **N은 전체 정점의 개수를 의미(2점)**하도록 하여 **N이 큰 (즉, weight가 큰) 트리의 부모에 작은 트리를 자식으로 병합(2점)**하여 트리의 높이를 줄일 수 있다.

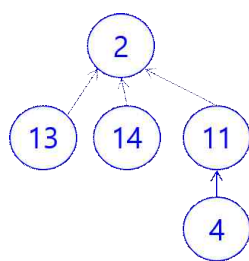
2. 경로압축: **find 연산을 수행하면서(1점) 루트까지 올라가는 경로 상의(2점) 각 노드의 부모노드를 루트로 갱신(2점)**하여 (이후의 find 연산을 더 빠르게 수행할 수 있도록) 트리의 높이를 줄일 수 있다.

(3) (1)의 표현된 트리에서 아래의 연산을 수행한 결과를 도시하십시오. (단, collapsingFind는 경로압축을 이용한 Find 연산이고, weightedUnion은 가중병합을 이용한 Union 연산임) (8점)

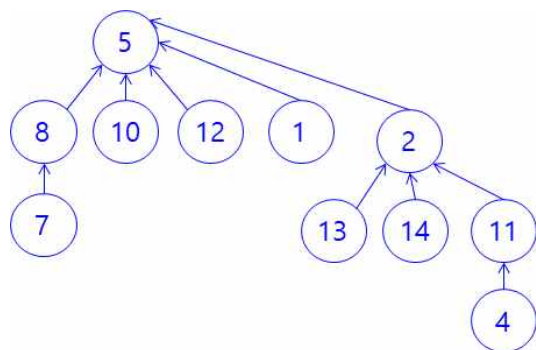
1) collapsingFind(1)



2) collapsingFind(11)



3) weightedUnion (collapsingFind(1), collapsingFind(11))



4) collapsingFind(9)

