

자료구조론(7급)

(과목코드 : 080)

2022년 군무원 채용시험

응시번호 :

성명 :

1. 다음은 어떤 이진트리에 대한 전위(pre-order) 및 중위(in-order) 순회 시 방문순서이다. 이 이진트리에 대한 설명으로 가장 옳지 않은 것은?

전위순회: b a c f d e g

중위순회: a c b d e f g

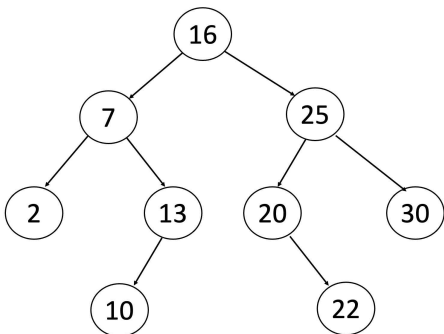
- ① 루트노드는 b이다.
- ② 단말노드 중에는 c와 e가 있다.
- ③ 후위(post-order) 순회 시 마지막 바로 전에 방문노드는 g이다.
- ④ 후위(post-order) 순회 시 처음 방문 노드는 c이다.

2. 다음 중위 표기식을 전위 표기로 나타낸 것으로 가장 옳은 것은?

$(a+b)*c-(d-e)*6$

- ① $ab+c*de-6*-$
- ② $-*+abc*-de6$
- ③ $-+ab*c-de*6$
- ④ $-+ab**--de6$

3. 다음 이진탐색트리에서 16을 삭제 후 다시 삽입하였다. 연산이 끝난 후 왼쪽과 오른쪽의 양쪽 자식이 모두 내부노드인 노드들을 모두 나열한 리스트로 가장 옳은 것은?



- ① 7, 13, 20, 25 ② 7, 13, 25
- ③ 7, 13, 20 ④ 7, 16, 25

4. QD는 덱(deque)을 이중연결리스트(doubly linked list)로, QS는 덱(deque)을 단일연결리스트(singly linked list)로 각각 구현하였다. 이때 덱(deque)의 후방(rear)에서 한 요소를 삭제하는 연산의 시간 복잡도로 가장 옳은 것은?

- ① QD: $O(1)$, QS: $O(\log \quad)$
- ② QD: $O(1)$, QS: $O(1)$
- ③ QD: $O(1)$, QS: $O(N)$
- ④ QD: $O(\log(N))$, QS: $O(N)$

5. 퀵 정렬(quick sort)에 대한 설명 중 가장 옳지 않은 것은?

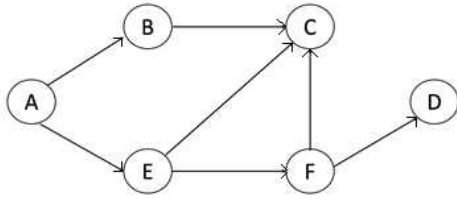
- ① 평균적인 시간복잡도는 $O(N\log(N))$ 이다.
- ② 최악의 시간복잡도는 $O(N)$ 이다.
- ③ 배열의 마지막 원소를 피벗으로 선택하는 경우가 최악의 선택방법이라고 할 수 있다.
- ④ 배열에서 랜덤하게 피벗을 선택하는 것이 안전한 선택이라고 할 수 있다.

6. 스택에서 아래와 같은 연산들을 순서대로 실행했을 때, 출력되는 결과로 가장 옳은 것은?

- 1. push(7)
- 2. push(9)
- 3. pop()
- 4. push(3)
- 5. push(6)
- 6. pop()
- 7. pop()
- 8. push(2)
- 9. pop()
- 10. pop()

- ① 7, 9, 3, 6, 2
- ② 9, 7, 6, 3, 2
- ③ 2, 6, 3, 9, 7
- ④ 9, 6, 3, 2, 7

7. 다음 그래프에서 위상정렬(topological sort)을 한 결과로 가장 옳은 것은?



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

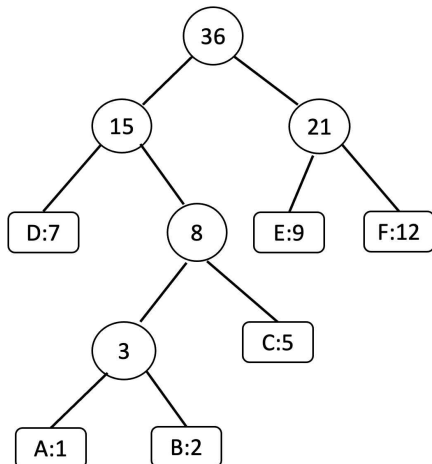
8. 다음 코드의 출력으로 가장 옳은 것은?

```

1. int A(int a, int b) {
2.     if (b == 1) return a;
3.     else return a + A(a, b-1);
4. }
5. main () {
6.     A(3, 4);
7. }
  
```

- ① 12
- ② 9
- ③ 6
- ④ 10

9. 허프만(Huffman) 코드는 손실 없는 데이터 압축 코딩 기법이다. 다음 허프만 트리에서 문자 'B'의 허프만 코드로 가장 옳은 것은?
(단, 각 단말 노드에서 문자열의 숫자는 해당 문자의 빈도수이다.)



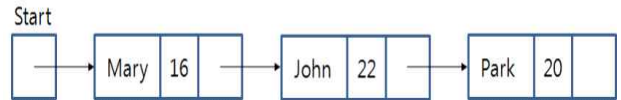
- ① 0111
- ② 0101
- ③ 011
- ④ 01

10. 변수 Start가 다음과 같이 선언되어 있을 때,

```

struct node { char name[10]; int age;
              struct node *ptr;}
struct node *Start;
  
```

아래의 그림을 참조하여 *((*Start).name+1)의 값으로 가장 옳은 것은?



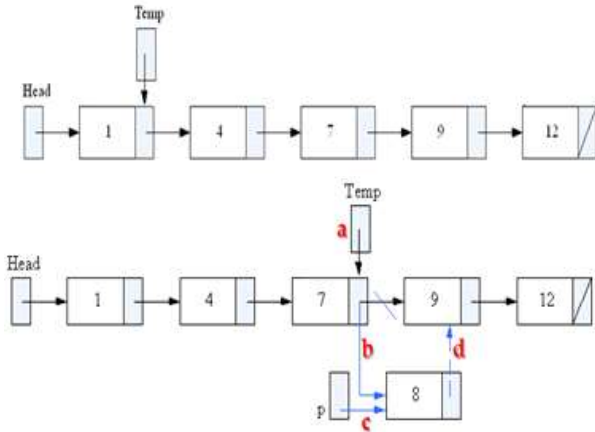
- ① "Mary"
- ② 16
- ③ "John"
- ④ 'a'

11. A, B, C에 들어갈 단어로 가장 옳은 것은?

리스트의 값들을 오름차순 또는 내림차순으로 저장하는데, 배열 또는 포인터를 이용한 연결 리스트를 이용한다고 가정한다.
만약 리스트에 검색이 많이 일어난다면 A를 사용하는 것이 유리하다.
만약 리스트에 삽입, 삭제 연산이 많이 일어난다면 B를 사용하는 것이 유리하다.
삽입, 삭제가 많은 경우 A를 사용하면 C 연산이 많이 필요하게 되어 성능에 부정적 영향을 끼친다.

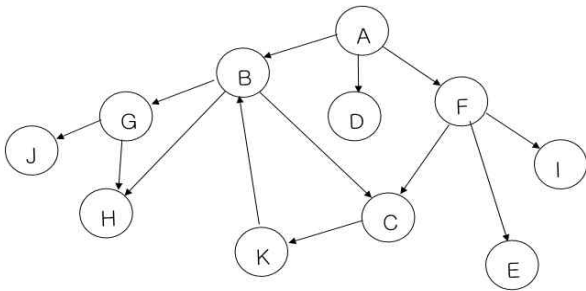
- ① A: 배열, B: 연결 리스트, C: 데이터 이동
- ② A: 배열, B: 연결 리스트, C: 메모리 할당
- ③ A: 연결 리스트, B: 배열, C: 데이터 이동
- ④ A: 연결 리스트, B: 배열, C: 메모리 할당

12. 다음 연결 리스트에서 7과 9사이에 새로운 값 8을 삽입할 경우, 아래 a, b, c, d 연산들의 순서로 가장 옳은 것은?



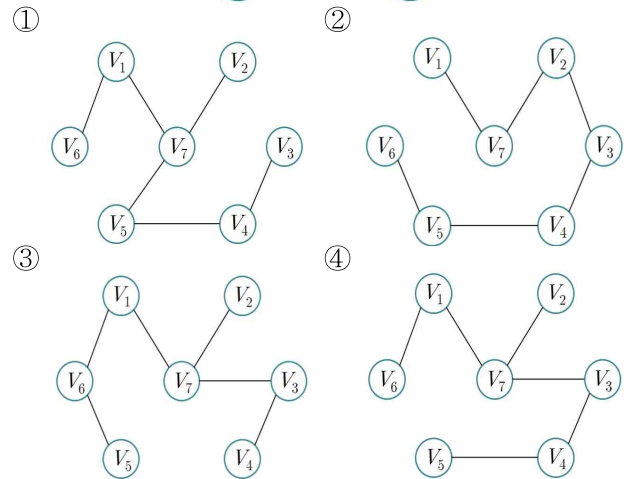
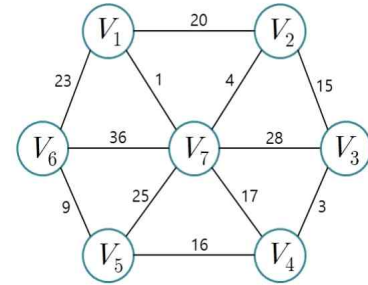
- ① $a \rightarrow b \rightarrow c \rightarrow d$
 ② $a \rightarrow c \rightarrow b \rightarrow d$
 ③ $c \rightarrow a \rightarrow d \rightarrow b$
 ④ $c \rightarrow a \rightarrow b \rightarrow d$

13. 아래와 같은 그래프를 스택을 사용하여 깊이 우선 탐색하려고 한다. 탐색 순서로 가장 옳은 것은?



- ① $A \rightarrow B \rightarrow D \rightarrow F \rightarrow G \rightarrow C \rightarrow E \rightarrow I \rightarrow J \rightarrow H \rightarrow K$
 ② $A \rightarrow B \rightarrow D \rightarrow F \rightarrow G \rightarrow K \rightarrow C \rightarrow E \rightarrow I \rightarrow J \rightarrow H$
 ③ $A \rightarrow B \rightarrow G \rightarrow J \rightarrow H \rightarrow C \rightarrow K \rightarrow D \rightarrow F \rightarrow E \rightarrow I$
 ④ $A \rightarrow B \rightarrow G \rightarrow J \rightarrow H \rightarrow K \rightarrow D \rightarrow F \rightarrow C \rightarrow E \rightarrow I$

14. 다음과 같은 그래프가 주어졌을 때, 최소비용 신장 트리(Minimum Cost Spanning Tree)를 구한 것으로 가장 옳은 것은?



15. 다음의 정렬 알고리즘 중 안정정렬 알고리즘으로 가장 옳지 않은 것은?

- ① 버블(bubble)정렬 ② 선택(selection)정렬
 ③ 삽입(insertion)정렬 ④ 합병(merge)정렬

16. 아래와 같은 합병정렬(merge sort)에서는 매 단계에서 합병함수(merge)를 사용하게 된다. 길이 2인 두 개의 리스트를 합병함수를 사용하여 합병할 경우, 필요한 최소의 비교 횟수로 옳은 것은?

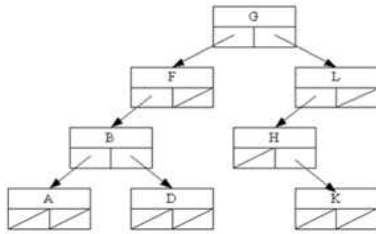
```
void MergeSort(int A[], int First, int Last)
{
    if (First < Last)
    {
        int Middle = (First + Last) / 2;
        MergeSort(A, First, Middle);
        MergeSort(A, Middle+1, Last);
        Merge(A, First, Middle, Last);
    }
}
```

- ① 1 ② $N/2$
 ③ $N-1$ ④ N

17. node의 개수가 N 인 완전 이진트리의 높이를 구하는 식으로 가장 옳은 것은? (단, 루트 노드의 높이는 0으로 가정한다.)

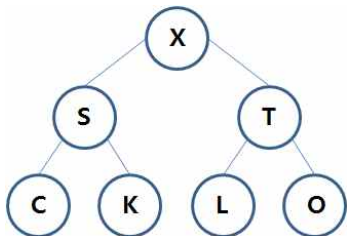
- ① $\lfloor \log N \rfloor$
- ② N^2
- ③ $N-1$
- ④ $\min(\text{left subtree의 높이}, \text{right subtree의 높이}) + 1$

18. 아래 이진 탐색 트리(binary search tree)에서 root node인 G-node가 삭제될 때, 새로운 root node가 될 것으로 가장 옳은 것은?



- ① L
- ② K
- ③ H
- ④ D

19. 아래 그림의 heap은 부모 node의 값이 자식 node들보다 크거나 같은 max heap이다. 이와 같은 모양의 heap을 만들어 낼 수 있는 삽입 순서로 가장 옳은 것은?



- ① C, T, S, X, K, L, O
- ② S, L, O, X, T, C, K
- ③ C, K, O, L, T, S, X
- ④ S, T, L, C, K, X, O

20. 아래의 배열을 통해 key가 70인 레코드를 보간 탐색으로 찾기 위한 비교 횟수로 가장 옳은 것은? (단, 탐색위치 계산은 반올림을 사용한다.)

index	1	2	3	4	5	6	7	8
key	10	22	31	42	55	59	70	83
index	9	10	11	12	13	14	15	16
key	92	105	111	127	149	149	152	163

- ① 1
- ② 2
- ③ 3
- ④ 4

21. 해쉬 함수가 $h(x) = x \bmod 11$ 과 같이 주어졌다고 가정하자. 해쉬 테이블의 크기는 11이다.

H_Table[0 : 10] (단, 인덱스는 0부터 10까지)

만약 해쉬 테이블에서 충돌이 일어날 경우 제공 탐색법을 이용하여 해결한다. 입력 레코드들의 key 값이 2, 46, 12, 3, 59, 87, 23, 84와 같이 들어왔을 때, H_Table[6]의 값으로 옳은 것은?

- ① 12
- ② 23
- ③ 84
- ④ 46

22. 아래의 알고리즘 mystery1과 mystery2의 시간 복잡도를 빅오 표기법으로 나타낸 것으로 가장 옳은 것은?

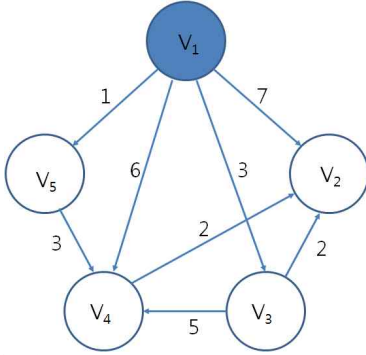
```

void mystery1 (int n) {
    int a[n][n], b[n][n], i, j;
    b[0][0]=a[n-1][n-1];
}

void mystery2 (int n) {
    int a[n][n], b[n][n], i, j;
    for (i = 1; i <= n; i++){
        for (j = 1; j <= 1000; j++){
            b[i][j]=a[i][j]; }
    }
}
    
```

- ① $O(1)$, $O(n)$
- ② $O(1)$, $O(n^2)$
- ③ $O(n)$, $O(n^2)$
- ④ $O(1)$, $O(1)$

23. 아래의 그래프에서 V_1 을 출발점으로 정했을 때, 나머지 node들(V_2, V_3, V_4, V_5)로 가는 최단 경로의 값들로 가장 옳은 것은?



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

24. 다음 문자열 K의 각 문자를 최소 힙(min heap)의 자료구조에 차례로 삽입하였을 때, 단말노드(leaf node)들에 저장된 문자들을 왼쪽부터 차례로 나열한 것으로 옳은 것은?
(왼쪽 → 오른쪽)

삽입할 문자열 (K): g f b c a e d

- ① g c f e ② g f e d
- ③ d g f e ④ f g e d

25. 그래프 알고리즘인 Prim의 알고리즘과 Dijkstra의 알고리즘을 비교한 것으로 가장 옳지 않은 것은?

- ① Dijkstra의 알고리즘은 최단경로를, Prim의 알고리즘은 최소 신장 트리(minimum spanning tree)를 찾는다.
- ② 일반적으로 Dijkstra의 알고리즘은 방향성(directed), 무방향성(undirected) 그래프에 모두 적용가능하나, Prim의 알고리즘은 무방향성 그래프에 적용가능하다.
- ③ Prim의 알고리즘은 음수 엣지 비용(negative edge cost)을 처리할 수 있지만, Dijkstra의 알고리즘은 실패할 수도 있다.
- ④ 여러 도시들을 연결하는 도로 건설 비용의 총합을 최소화하기 위한 응용에는 Dijkstra 알고리즘이 더 적절하다.