

<b>자료구조론</b>	<b>국가 전산 7급</b>	<b>2023년 9월 23일</b>
--------------	-----------------	---------------------

♣ 필기합격인원/합격선(52명/79점) - 선발예정인원 38명 ♣

1. 다음 C 코드의 시간복잡도(time complexity)는? [2023년 국가 7급]

```
void func(int n)
{
    int i, j, k = 0;
    for(i = 0; i < (n + 200); i++)
        for(j = 0; j < 300; j++)
            printf("k = %d\n", k++);
}
```

- ①  $O(n)$
- ②  $O(n^2)$
- ③  $O(\log_2 n)$
- ④  $O(n \log_2 n)$

♣ 시간복잡도

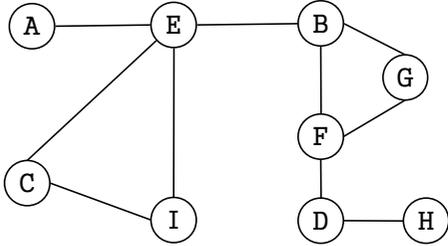
```
void func(int n)
{
    int i, j, k = 0;
    for(i = 0; i < (n + 200); i++)           //O(n)
        for(j = 0; j < 300; j++)           //O(1) - 상수이므로
            printf("k = %d\n", k++);
}
```

• 시간복잡도 =  $O(n) \times O(1) = O(n)$

<b>빅오 (O)</b>	$n \geq n_0$ 를 만족하는 모든 $n$ 에 대하여 $f(n) \leq c \cdot g(n)$ 인 조건을 만족하는 두 양의 상수 $c$ 와 $n_0$ 가 존재하기만 하면 $f(n) = O(g(n))$ 이다.	최대시간 예측 (상한)
---------------	--	-----------------

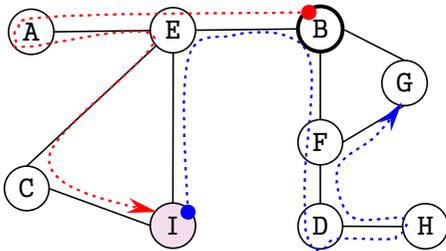
정답 : ①

2. 다음 그래프(graph)에 대하여 깊이우선탐색(DFS, depth first search)을 하려고 한다. 시작정점(vertex)이 B일 때, 정점 I는 몇 번째에 방문하는가? (단, B가 첫 번째 방문 정점이고, 탐색 시 인접한 정점들은 알파벳 오름차순으로 방문한다) [2023년 국가 7급]



- ① 4                      ② 5                      ③ 8                      ④ 9

☞ 깊이우선탐색



• 깊이우선탐색(B) = B-E-A-C-I-F-D-H-G

정답 : ②

3. 이중 연결리스트(doubly linked list)에 대한 설명으로 옳지 않은 것은? [2023년 국가 7급]

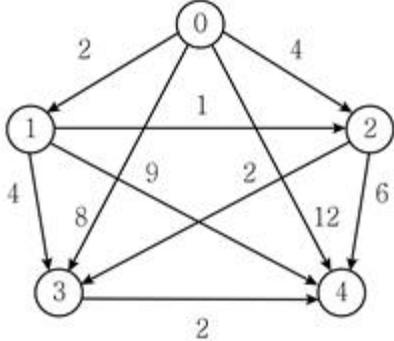
- ① 이전 노드와 이후 노드에 대한 링크(link)를 가지고 있어 양방향 탐색이 가능하다.  
 ② 단순 연결리스트(singly linked list)에 비해 기억장소가 추가로 소요된다.  
 ③ 이진트리(binary tree)에 활용이 가능하다.  
 ④ 다항식(polynomial), 희소행렬(sparse matrix)을 표현할 수 없다.

☞ 이중 연결리스트

- 다항식(polynomial), 희소행렬(sparse matrix)을 표현할 수 없다.(×)  
 → 모든 자료구조는 배열 또는 연결리스트로 표현할 수 있다.  
 → 단일 연결리스트로 표현할 수 있는 것은 당연 이중 연결리스트로 표현할 수 있다.

정답 : ④

4. 다음은 정점 간의 거리를 표시한 방향그래프(directed graph)이다. 시작정점이 0일 때, 정점 3과 정점 4까지 최단경로의 거리를 바르게 연결한 것은? [2023년 국가 7급]



- |      |      |
|------|------|
| 정점 3 | 정점 4 |
| ① 5  | 7    |
| ② 5  | 8    |
| ③ 8  | 11   |
| ④ 8  | 12   |

☞ 시작정점이 0일 때, 정점 3과 정점 4까지 최단경로

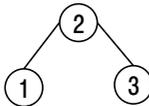
- (0, 3) : 0 → 1 → 2 → 3 = 2 + 1 + 2 = 5
- (0, 4) : 0 → 1 → 2 → 3 → 4 = 2 + 1 + 2 + 2 = 7

정답 : ①

5. 어떤 트리 3개에 대한 모든 노드 개수의 합이 100이면 간선(edge) 개수의 합은? [2023년 국가 7급]

- ① 94                      ② 95                      ③ 96                      ④ 97

☞ 간선 개수의 합



트리에서 노드 수가 3개이면 간선 수는 2개이다. 하나 차이이다.

- 트리가 3개인 경우 간선 개수의 합 = 노드 개수의 합 - 3 = 100 - 3 = 97

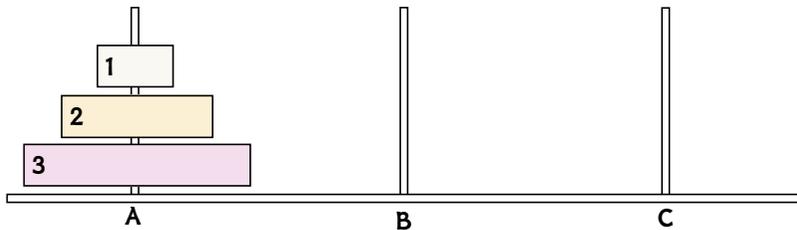
정답 : ④

6. 다음 하노이탑 문제를 해결하는 파이썬(Python) 코드의 시간복잡도는? (단,  $n$ 은 원판의 개수이고 first, second, third는 기둥 번호이다) [2023년 국가 7급]

```
def hanoiT(n, first, second, third):  
    if n == 1:  
        print("원판1 : %s 에서 %s"%(first, third))  
    else:  
        hanoiT(n - 1, first, third, second)  
        print("원판%d : %s 에서 %s"%(n, first, third))  
        hanoiT(n - 1, second, first, third)
```

- ①  $O(n)$                       ②  $O(n^2)$                       ③  $O(2^n)$                       ④  $O(\log_2 n)$

☞ 하노이탑 - 하노이탑은 퍼즐의 일종이다.(위키백과 참조)



3개의 기둥과 이 기둥에 꽂을 수 있는 다양한 크기의 원판들이 있고, 퍼즐을 시작하기 전에는 한 기둥에 원판들이 작은 것이 위에 있도록 순서대로 쌓여 있다.

게임의 목적은 다음 두 가지 조건을 만족시키면서, 한 기둥에 꽂힌 원판들을 그 순서 그대로 다른 기둥으로 옮겨서 다시 쌓는 것이다.

- ① 한 번에 하나의 원판만 옮길 수 있다.
- ② 큰 원판이 작은 원판 위에 있어서는 안 된다.

하노이탑 문제는 재귀호출을 이용하여 풀 수 있는 가장 유명한 예제 중의 하나이다. 프로그래밍 수업에서 알고리즘 예제로 많이 사용한다.

원판이  $n$ 개 일 때,  $2^n - 1$ 번의 이동으로 원판을 모두 옮길 수 있다. 시간복잡도 :  $O(2^n)$   
 $2^n - 1$ 는 메르센 수라고 부른다.

참고로, 64개의 원판을 옮기는 데  $2^n - 1 \approx 18446744073709551615$ 번을 움직여야 하고, 한번 옮기는 시간을 1초로 하면, 64개의 원판을 옮기는 데 5849억 4241만 7355년 걸린다.

7. 어떤 이진트리의 전위순회(preorder traversal)와 중위순회(inorder traversal)한 순서가 다음과 같을 때, 이 트리의 단말노드만을 모두 나열한 것은? [2023년 국가 7급]

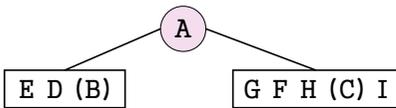
- 전위순회: A, B, D, E, C, F, G, H, I
- 중위순회: E, D, B, A, G, F, H, C, I

- ① A, D, E                      ② B, C, D, F
- ③ D, E, F, I                 ④ E, G, H, I

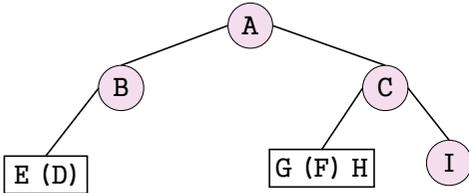
♣ 이진트리 순회

- 전위순회(중, 좌, 우): (A), B, D, E, C, F, G, H, I ← A가 근노드이다.
- 중위순회(좌, 중, 우): E, D, B, (A), G, F, H, C, I → A를 기준으로 좌우측으로 배치

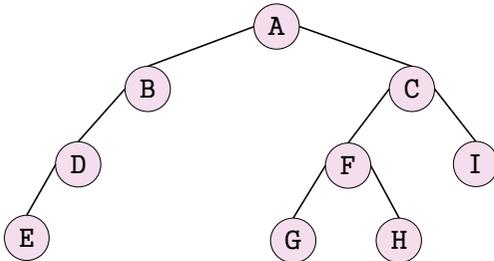
↓  
 ↓ 전위순회로부터 A가 근노드라는 것을 알 수 있다.  
 ↓ A를 중심으로 좌우측에 배치된다.



↓ 전위순회로부터 각 서브트리의 근노드는 B와 C임을 알 수 있다.  
 ↓ 각 서브트리에서 B와 C를 중심으로 좌우측에 배치된다.

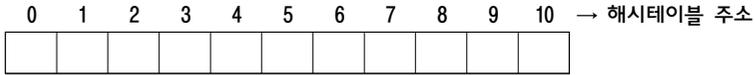


↓ 전위순회로부터 각 서브트리의 근노드는 D와 F임을 알 수 있다.  
 ↓ 각 서브트리에서 D와 F를 중심으로 좌우측에 배치된다.



• 단말노드 : E, G, H, I

8. 다음 비어 있는 크기가 11인 해시테이블(hash table)에 입력키가 4, 6, 15, 26, 17 순서로 저장 되었다. 입력키 26이 저장된 해시테이블의 주소는? (단, 해시함수는  $h(k) = k \bmod 11$ 이고, 충돌 해결은 선형조사법(linear probing)을 사용한다) [2023년 국가 7급]

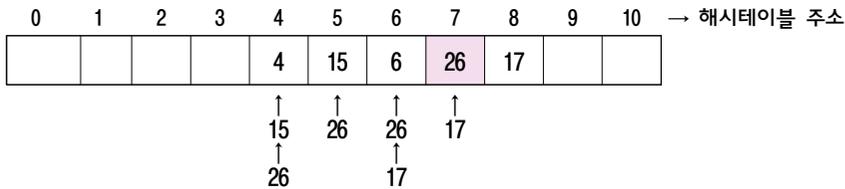


- ① 1
- ② 4
- ③ 6
- ④ 7

☞ 선형조사법

- 오버플로(overflow)가 발생한 곳에서 가장 가까운 빈 버킷을 찾아 저장한다. (색인증가를 1)
- 선형개방주소법(linear open addressing)이라고도 한다.

// 입력키 : 4, 6, 15, 26, 17



$h(4) = 4 \bmod 11 = 4$	키 4는 버킷주소 4에 저장
$h(6) = 6 \bmod 11 = 6$	키 6은 버킷주소 6에 저장
$h(15) = 15 \bmod 11 = 4$	키 15는 1차 충돌 발생, 색인증가를 1 키 15는 버킷주소 5에 저장
$h(26) = 26 \bmod 11 = 4$	키 26은 1차 충돌 발생, 색인증가를 1 키 26은 2차 충돌 발생, 색인증가를 1 키 26은 3차 충돌 발생, 색인증가를 1, 26은 버킷주소 7에 저장
$h(17) = 17 \bmod 11 = 6$	키 17은 1차 충돌 발생, 색인증가를 1 키 17은 2차 충돌 발생, 색인증가를 1, 17은 버킷주소 8에 저장

정답 : ④

9. 다음은 유클리드 알고리즘(euclidean algorithm)을 사용하여 두 정수의 최대공약수(greatest common divisor)를 구하는 파이썬 프로그램이다. gcd(12, 21)를 실행한 후 count 값과 (가)에 들어갈 내용을 바르게 연결한 것은? [2023년 국가 7급]

```
def gcd(m, n):
    global count
    count = count + 1
    if m < n:
        temp = m
        m = n
        n = temp
    if m % n == 0:
        return n
    else:
        return gcd( (가) )
count = 0
gcd(12, 21)
```

- |       |          |
|-------|----------|
| count | (가)      |
| ① 2   | m, m % n |
| ② 2   | n, n % m |
| ③ 3   | m, n % m |
| ④ 3   | n, m % n |

☞ 최대공약수

```
def gcd(m, n):
    global count
    count = count + 1
    if m < n:
        temp = m
        m = n
        n = temp
    if m % n == 0:
        return n
    else:
        return gcd(n, m % n)
count = 0
gcd(12, 21)
print(count) #출력 : 3
```

**유클리드 호제법에 의한 gcd구하기**

- gcd(12, 21) = ?  
 ↓ count = 1, 12와 21을 교환
- gcd(21, 12) : m % n = 21 % 12 = 9  
 ↓ 재귀호출
- gcd(12, 21%12) = gcd(12, 9)  
 ↓ count = 2 재귀호출
- gcd(9, 12%9) = gcd(9, 3)  
 ↓ count = 3 재귀호출
- gcd(3, 9%3) = gcd(3, 0)
- 나머지가 0이 될 때까지 재귀호출(count = 3)
- 나머지가 0이 되었을 때, 좌측 값이 gcd이다.

10. 배열(array)로 표현한 이진트리 중에서 최소힙(minimum heap)에 해당하지 않는 것은? [2023년 국가 7급]

- ① 

1	3	13	7	9	10	20	17	25
---	---	----	---	---	----	----	----	----
- ② 

1	3	13	9	7	20	17	25	10
---	---	----	---	---	----	----	----	----
- ③ 

1	3	17	7	13	25	20	9	10
---	---	----	---	----	----	----	---	----
- ④ 

1	9	3	17	10	13	7	20	25
---	---	---	----	----	----	---	----	----

☞ 최소힙

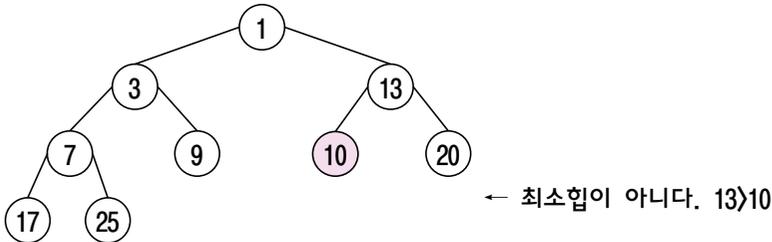
- 최소힙 : 최소트리아면서 완전이진트리이다.
- 최소트리 : 부모노드 값  $\leq$  자식노드 값(단, 자식이 있을 때)

// 이런 문제는 배열구조를 트리로 그려보면 답을 쉽게 찾을 수 있다.

- ① 

1	3	13	7	9	10	20	17	25
---	---	----	---	---	----	----	----	----

↓ 트리로 그리면



- ② 

1	3	13	9	7	20	17	25	10
---	---	----	---	---	----	----	----	----

↓ 트리로 그리면

