

자료구조론(7급)

(과목코드 : 080)

2024년 군무원 채용시험

응시번호 :

성명 :

1. 다음은 정수를 매개변수로 받아 오른쪽과 같이 작은 자릿수부터 차례로 출력하는 재귀 알고리즘이다. 빈칸에 들어갈 코드로 가장 적절한 것은?

<pre>void print_digits(int n) { if (n < 10) printf("%d\n", n); else { } }</pre>	<실행 예시> n = 25689일 때 9 8 6 5 2
---	---

- ① print_digits(n/10);
printf("%d\n", n%10);
- ② print_digits(n%10);
printf("%d\n", n/10);
- ③ printf("%d\n", n/10);
print_digits(n%10);
- ④ printf("%d\n", n%10);
print_digits(n/10);

2. 최소 힙에 대한 설명으로 가장 적절한 것은?

- ① 모든 노드는 자식이 없거나 두 개의 자식을 가진다.
- ② 가장 큰 키(key)는 레벨(또는 깊이)이 가장 큰 노드에 저장되어 있다.
- ③ 부모 노드의 키 값이 자식 노드의 키 값보다 같거나 작아야 한다.
- ④ 삭제 연산은 가장 큰 키를 힙에서 제거한다.

3. 다음은 배열을 이용한 정수 리스트 ADT에서 위치 pos에 요소 item을 삽입하는 add 함수이다. 빈칸에 들어갈 코드로 가장 적절한 것은? (단, 배열이 만원(full)인지, pos 위치가 유효한지 등 예외 상황에 대한 검사는 add 함수 호출 전에 이미 수행됐다고 가정하라.)

```
typedef struct {
    int V[100]; // 요소를 저장하는 배열
    int n;      // 현재 요소의 개수
} List;

void add(List *L, int pos, int item)
{
    int i;
    for (i = L->n-1; i >= pos; i = i-1)
        

    L->V[pos] = item;
    L->n = L->n+1;
}
```

- ① L->V[i+1] = L->V[i];
- ② L->V[i] = L->V[i+1];
- ③ L->V[i-1] = L->V[i];
- ④ L->V[i] = L->V[i-1];

4. 다음 중 해당 연산을 수행하기 위해 필요한 보조자료구조의 성격이 가장 다른 것은?

- ① 트리의 후위순회
- ② 트리의 중위순회
- ③ 트리의 레벨순회
- ④ 그래프의 깊이우선탐색

5. 헤드 노드(데이터를 가지고 있지 않은 특별한 노드)를 가지는 이중 연결 리스트에서 노드 p 다음에 새로운 노드 new_node를 삽입하는 코드로 가장 적절하지 않은 것은? (단, 코드에서 prev는 이전 노드에 대한 링크, next는 다음 노드에 대한 링크이다.)

- ① new_node->prev = p;
new_node->next = p->next;
p->next->prev = new_node;
p->next = new_node;
- ② p->next->prev = new_node;
p->next = new_node;
new_node->prev = p;
new_node->next = p->next;
- ③ new_node->next = p->next;
new_node->prev = p;
p->next = new_node;
new_node->next->prev = new_node;
- ④ p->next->prev = new_node;
new_node->next = p->next;
p->next = new_node;
new_node->prev = p;

6. 다음은 자식이 하나인 노드가 없는 어떤 이진 트리를 전위순회한 결과이다. 이 트리를 후위 순회한 결과로 가장 적절하지 않은 것은?

전위순회: A B C D E F G

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

7. 다음 중 정렬 알고리즘에 대한 설명으로 가장 적절하지 않은 것은?

- ① 합병정렬은 최악의 경우와 최선의 경우의 수행시간 복잡도가 동일하다.
- ② 퀵정렬은 주어진 입력을 둘로 나누어 재귀적으로 해결하는 동적 프로그래밍 기법을 사용한다.
- ③ 삽입정렬의 최선의 경우 수행시간 복잡도는 $O(n)$ 이다.
- ④ ힵ정렬은 ힵ을 생성한 후 ힵ에서 최대 (혹은 최소) 원소를 하나씩 뽑아내는 방식으로 정렬하는 알고리즘이다.

8. 아래의 알고리즘을 이용하여 배열 $A = [6, 4, 7, 2, 9, 1, 5, 3]$ 를 오름차순으로 정렬하려고 한다. 다음 중 이 알고리즘에 의해 나타날 수 있는 배열의 중간 상태로 가장 적절하지 않은 것은?

```
sort(A, left, right)
if (left < right)
    mid = (left+right)/2
    sort(A, left, mid)
    sort(A, mid+1, right)
정렬된 두 배열 A[left..mid]와 A[mid+1..right]를
하나의 정렬된 배열 A[left..right]로 만든다.
```

- ① [4, 6, 2, 7, 9, 1, 5, 3]
- ② [4, 6, 2, 7, 1, 9, 3, 5]
- ③ [2, 4, 6, 7, 1, 9, 3, 5]
- ④ [2, 4, 6, 7, 1, 3, 5, 9]

9. 다음 중 각 탐색 자료구조에서 탐색 연산과 삽입 연산의 시간 복잡도로 가장 적절한 것은?

- ① 무순배열 - 탐색 $O(n)$, 삽입 $O(1)$
- ② 순서배열 - 탐색 $O(\log n)$, 삽입 $O(\log n)$
- ③ 이진탐색트리 - 탐색 $O(\log n)$, 삽입 $O(\log n)$
- ④ AVL트리 - 탐색 $O(1)$, 삽입 $O(\log n)$

10. 다음과 같이 엑셀에 저장된 학생 데이터(행)를
분반 기준으로 정렬하되, 같은 분반의 데이터는
학번순으로 정렬하려고 한다. 이와 같이 정렬
하기 위한 방법으로 가장 적절하지 않은 것은?

학생	분반	학번	이름
학생1	1	12345	홍길동
학생2	2	23456	고길동
...			

- ① 합병정렬을 사용하여 전체 학생을 분반 기준으로
먼저 정렬한 후, 같은 분반의 학생들끼리
따로 삽입정렬을 사용하여 학번순으로 정렬한다.
- ② 힙정렬을 사용하여 전체 학생을 분반 기준으로
먼저 정렬한 후, 같은 분반의 학생들끼리
따로 퀵정렬을 사용하여 학번순으로 정렬한다.
- ③ 퀵정렬을 사용하여 전체 학생을 학번 기준으로
먼저 정렬한 후, 전체 학생을 다시 삽입정렬을
사용하여 분반순으로 정렬한다.
- ④ 선택정렬을 사용하여 전체 학생을 학번 기준으로
먼저 정렬한 후, 전체 학생을 다시 힙정렬을
사용하여 분반순으로 정렬한다.

11. 다음은 오름차순으로 정렬된 배열 A[l..r]에서
키 k의 위치를 찾는 이진탐색 알고리즘의 의사
코드이다. 빈칸 ㉠ 및 ㉡에 들어갈 코드로 가장
적절한 것은? (단, 배열의 모든 원소는 서로
다르고, 키가 존재하지 않으면 -1을 반환한다.)

```

Alg bin_search( k, A, l, r )
if ( ㉠ ) return -1
mid ← (l+r)/2
if (k == A[mid]) return (생략)
else if (k < A[mid]) return (생략)
else return ㉡

```

- ① ㉠ $l \geq r$ ㉡ bin_search(k, A, l, mid - 1)
- ② ㉠ $l \geq r$ ㉡ bin_search(k, A, mid+1, r)
- ③ ㉠ $l > r$ ㉡ bin_search(k, A, l, mid - 1)
- ④ ㉠ $l > r$ ㉡ bin_search(k, A, mid+1, r)

12. 다음 중 이진탐색트리에 대한 설명으로 가장
적절한 것은?

- ① 탐색 연산은 탐색할 키와 방문한 노드의 키를
비교하여 같지 않은 경우, 대소 관계에 따라
자식 노드 중 하나를 따라서 내려간 후 같은
작업을 반복하므로 재귀로 구현하는 것은
불가능하다.
- ② 탐색 키가 이진탐색트리에 없는 경우, 탐색
연산은 항상 트리 높이에 비례하는 시간이
필요하다.
- ③ 삽입 연산에 의해 높이가 증가하는 노드가
반드시 존재한다. (단, 삽입할 키를 저장하기
위해 새로 생성되는 노드는 제외한다.)
- ④ 삭제할 키를 저장하는 노드가 두 개의 자식을
가지는 경우, 중위순회 선행자 또는 계승자를
찾는 과정이 필요하다.

13. 다음 중 높이가 5인 AVL 트리의 최소 노드
개수와 최대 노드 개수로 가장 적절한 것은?
(단, 루트 노드 하나만 있는 트리의 높이는 0으로
정의한다.)

- ① 최소 20개, 최대 63개
- ② 최소 20개, 최대 64개
- ③ 최소 21개, 최대 63개
- ④ 최소 21개, 최대 64개

14. 다음 중 해싱(hashing)에 대한 설명으로 가장
적절하지 않은 것은?

- ① 키들의 상대적 순위보다는 키 값 자체에 의해
저장위치가 결정된다.
- ② 해시 테이블의 크기를 초과하는 개수의 키를
저장하는 것은 불가능하다.
- ③ 적재율(저장된 항목의 개수/해시 테이블의
크기)이 1에 가까울수록 성능이 저하된다.
- ④ O(1) 시간 탐색이 보장되지 않는다.

15. 다음 의사코드가 나타내는 그래프 알고리즘에 대한 설명으로 가장 적절하지 않은 것은?

```
func(v)// v는 그래프의 한 정점
    정점 v를 방문되었다고 표시
    for (all u ∈ {v에 인접한 정점})
        if (u가 아직 방문되지 않았으면)
            func(u)
```

- ① 보조자료구조로 스택이 사용된다.
- ② 그래프를 인접행렬로 표현하는 경우, 위 알고리즘의 수행시간은 $O(V+E)$ 이다. (단, V는 정점 개수, E는 간선 개수를 의미한다.)
- ③ 위 알고리즘을 활용하여 무방향 그래프의 연결 요소(또는 연결 성분)의 개수를 구할 수 있다.
- ④ 위 알고리즘을 활용하여 방향 비싸이클 그래프의 위상정렬을 수행할 수 있다.

16. 다음 C 언어 함수를 사용하여 func(5)를 수행했을 때, 반환되는 변수 cnt의 값은?

```
int func(int n)
{
    int cnt = 0, i, j;
    for (i = n; i > 0; i--)
        for (j = 0; j < i; j++)
            cnt++;
    return cnt;
}
```

- ① 10
- ② 15
- ③ 20
- ④ 21

17. 다음 중 입력 데이터의 크기가 충분히 크다고 할 때, 실행시간이 가장 적게 걸리는 알고리즘으로 적절한 것은?

- ① 난수로 이루어진 리스트를 사용한 퀵 정렬 알고리즘
- ② 정렬되지 않은 리스트를 사용한 순차 탐색 알고리즘
- ③ 정렬된 리스트를 사용한 이진 탐색 알고리즘
- ④ 재귀 함수를 사용한 하노이의 탑 알고리즘

18. 다음 중 시간복잡도의 크기 순서로 바르게 나열된 것은?

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

19. 다음 전위 표기식(prefix notation)을 중위 표기식(infix notation)으로 나타낸 것 중 가장 적절한 것은?

```
전위 표기식 : + a / * b - c d e
```

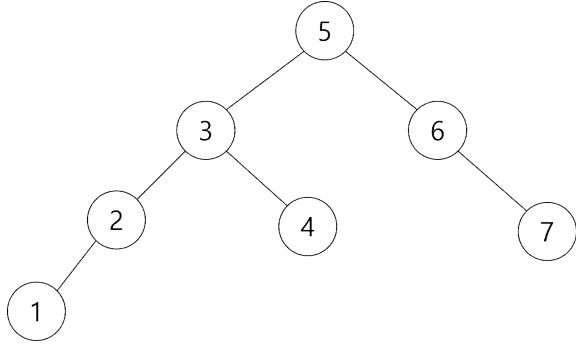
- ① $a + b * c - d / e$
- ② $(a + b) * c - d / e$
- ③ $a + b * (c - d) / e$
- ④ $(a + b) * (c - d) / e$

20. 다음과 같은 C 언어 함수 xyz(186, 720)을 수행했을 때, 반환되는 매개변수 b의 값은?

```
int xyz(int a, int b)
{
    int r = a%b;
    while (r != 0) {
        a = b;
        b = r;
        r = a%b;
    }
    return b;
}
```

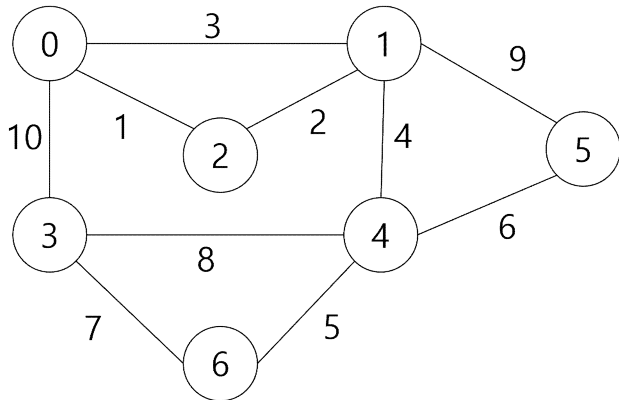
- ① 3
- ② 6
- ③ 18
- ④ 24

21. 공백 트리에 차례대로 키 값을 추가하여 AVL 트리를 만든다고 할 때, 다음과 같은 AVL 트리가 생성되지 않는 키 값의 순서는?



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

22. 다음과 같은 그래프에 대해 최소 비용 신장 트리를 구한다고 할 때, 트리를 구성하는 데 사용되지 않은 간선은?



- ① 간선 (0, 1)
- ② 간선 (1, 2)
- ③ 간선 (3, 6)
- ④ 간선 (4, 5)

23. 다음 중 최대 힙으로 적절하지 않은 것은?

- ①

[1]	[2]	[3]	[4]	[5]	[6]	[7]
7	3	6	1	2	5	4
- ②

[1]	[2]	[3]	[4]	[5]	[6]	[7]
7	6	3	4	5	2	1
- ③

[1]	[2]	[3]	[4]	[5]	[6]	[7]
7	6	4	1	2	3	5
- ④

[1]	[2]	[3]	[4]	[5]	[6]	[7]
7	6	5	4	3	2	1

24. A와 B가 스택이고 a, b, c, d, e가 객체라고 할 때, 다음과 같은 일련의 삽입(push)과 삭제(pop) 연산을 수행한 후 스택 A와 B의 탑(top)에 있는 원소로 옳게 짝지어진 것은?

```

push(A, a)
push(B, b)
push(A, c)
push(B, d)
push(B, e)
push(B, pop(A))
pop(B)
push(A, pop(B))
pop(B)
  
```

- ① A: a, B: e
- ② A: c, B: e
- ③ A: e, B: b
- ④ A: e, B: d

25. 다음과 같은 2개의 해시 함수가 있다고 하자.

$$h_1(k) = k \bmod 7$$

$$h_2(k) = 5 - (k \bmod 5)$$

이중 해싱법(double hashing)으로 다음과 같은 키 값을 해시 테이블에 차례대로 삽입한다고 할 때, 최종적으로 생성되는 해시 테이블의 모습으로 가장 적절한 것은?

키 값 : 2, 9, 3, 10, 5

- ①

[0]	[1]	[2]	[3]	[4]	[5]	[6]
		2	9	3	10	5
- ②

[0]	[1]	[2]	[3]	[4]	[5]	[6]
		2	3	9	10	5
- ③

[0]	[1]	[2]	[3]	[4]	[5]	[6]
	10	2	9		3	5
- ④

[0]	[1]	[2]	[3]	[4]	[5]	[6]
	10	2	9	3	5	