

자료구조론(7급)

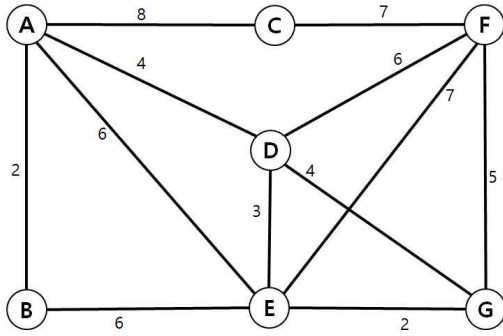
(과목코드 : 080)

2023년 군무원 채용시험

응시번호 :

성명 :

1. 다음 그래프에 대해 크루스칼(Kruskal) 알고리즘으로 최소 비용 신장트리(Minimum Spanning Tree)를 생성할 때 5번째로 포함되는 선분(edge)은?



- ① (A,D) ② (D,G)
③ (D,F) ④ (F,G)

2. 정렬에 대한 설명으로 가장 적절하지 않은 것은?

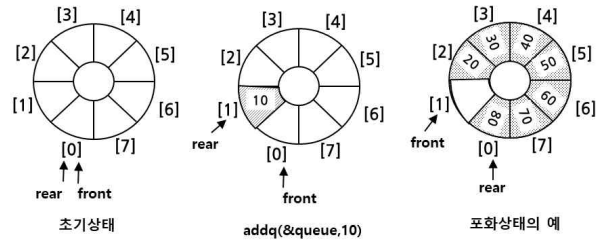
- ① 삽입 정렬(insert sort)과 셸 정렬(shell sort)은 불안정 정렬(unstable sort) 방법이다.
② 퀵 정렬(quick sort)과 병합 정렬(merge sort)은 분할 정복(divide and conquer) 기법을 이용한 정렬 알고리즘이다.
③ 기수 정렬(radix sort)은 항목의 비교를 사용하지 않고 분배를 이용한 정렬 방법이다.
④ 버블 정렬(bubble sort)은 입력자료가 역순으로 정렬되어 있을 때 자리 교환이 가장 많이 발생한다.

3. 다음 알고리즘의 시간 복잡도를 나타낸 것 중 가장 적절한 것은?

```
int f_algorithm(int n)
{
    int total = 0;
    for (int i = n; i > 0; i /= 2)
        total += i;
    return total;
}
```

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

4. C언어로 원형 큐(Circular Queue)구조를 정의하고 데이터를 삽입하는 연산을 구현하였을 때 ㉠, ㉡에 들어갈 코드로 가장 적절한 것은?



```
#define MAX_QSIZE 8
```

```
typedef struct {
```

```
    int data[MAX_QSIZE];
```

```
    int front, rear;
```

```
} QType;
```

```
void addq(QType* q, int x)
```

```
{
```

```
    if ( ㉠ ) {
```

```
        fprintf(stderr, "포화상태 큐\n");
```

```
        exit(1);
```

```
    }
```

```
    ㉡
```

```
    q->data[q->rear] = x;
```

```
}
```

- ① ㉠ $(q->rear) \% MAX_QSIZE == q->front$
 ㉡ $q->rear = (q->rear + 1) \% MAX_QSIZE;$
② ㉠ $q->front == q->rear$
 ㉡ $q->front = (q->front + 1) \% MAX_QSIZE;$
③ ㉠ $(q->rear + 1) \% MAX_QSIZE == q->front$
 ㉡ $q->rear = (q->rear + 1) \% MAX_QSIZE;$
④ ㉠ $(q->front + 1) \% MAX_QSIZE == q->rear$
 ㉡ $q->front = (q->front + 1) \% MAX_QSIZE;$

5. 다음 순서대로 값을 삽입하여 구성된 AVL 트리에 대한 설명으로 가장 적절하지 않은 것은? (단, 높이(height)는 루트(root)노드의 레벨(level)을 1로 했을 때, 그 트리가 가지는 노드의 최대 레벨로 계산한다.)

20, 10, 30, 50, 40, 60

- ① 전위(preorder) 순회를 하면 40 20 10 30 50 60이다.
- ② 높이(height)는 3이다.
- ③ 단말(terminal)노드는 3개이다.
- ④ 루트(root)노드는 30이다.

6. 다음 C언어로 구현한 정렬 알고리즘을 이용하여 내림차순으로 정렬하려고 할 때 ㉠에 알맞은 조건은?

```
#define SWAP(a, b, t) ((t)=(a), (a)=(b), (b)=(t))

void b_sort(int list[], int n)
{
    int i, j, tmp;
    for (i = n - 1; i >= 1; i--) {
        for (j = 0; j < i; j++)
            if (____㉠____)
                SWAP(list[j], list[j + 1], tmp);
    }
}
```

- ① list[j] > list[j + 1]
- ② list[j] < list[j + 1]
- ③ list[i] > list[j + 1]
- ④ list[i] < list[j + 1]

7. 다음 자료구조에 대한 설명 중 가장 적절하지 않은 것은?

- ① 알고리즘의 시간 복잡도를 표시하는 방식 중 빅오 표기법은 함수의 상한을 표시한다.
- ② 시간 복잡도가 $O(n^3)$ 인 알고리즘의 실행시간이 8배 증가했다면 입력의 개수는 2배 증가했을 것이다.
- ③ 어떤 알고리즘의 시간 복잡도가 함수 $f(n) = 2^n + n^{100}$ 라고 하면 $f(n) = O(n^{100})$ 라고 표시할 수 있다.
- ④ 자료구조의 추상 데이터 타입(ADT)은 그 구현 방식을 구체적으로 정의할 필요가 없다.

8. 해시 함수에 대한 다음 설명 중 가장 적절하지 않은 것은?

- ① 제산 함수 $h(k) = k \bmod M$ 에서 M 은 주로 소수(prime number)를 선택하거나 홀수를 선택한다.
- ② 폴딩 함수는 탐색 키가 해시 테이블의 크기보다 더 큰 정수일 경우 사용하며, 탐색 키를 몇 개의 부분으로 나누어 이를 더하거나 비트별 XOR 연산하는 것을 의미한다.
- ③ 중간 제곱 함수는 키를 제공한 다음 중간값의 몇 비트를 취해서 해시 주소를 생성한다.
- ④ 비트 추출 방법은 해시 주소의 집중 현상이 일어날 가능성이 적다.

9. [27, 3, 22, 13, 29, 7]을 다음과 같이 C언어로 구현한 퀵 정렬로 정렬할 때, 첫 번째 출력되는 것으로 가장 적절한 것은?

```
# define MAX_SIZE 6
void qsort(int data[], int left, int right)
{
    int i, j, pivot, temp, m, k;
    if (left < right)
    {
        m = (left + right) / 2;
        pivot = data[m];
        i = left - 1;
        j = right + 1;
        while (1)
        {
            while (data[++i] < pivot);
            while (data[--j] > pivot);
            if (i >= j) break;
            temp = data[i];
            data[i] = data[j];
            data[j] = temp;
        }
        for (k = 0; k < MAX_SIZE; k++)
            printf("%4d", data[k]);
        printf("\n");
        qsort(data, left, i - 1);
        qsort(data, j + 1, right);
    }
}
```

- ① 7 3 13 22 27 29
 ② 3 7 13 22 27 29
 ③ 7 3 13 22 29 27
 ④ 3 7 13 22 29 27

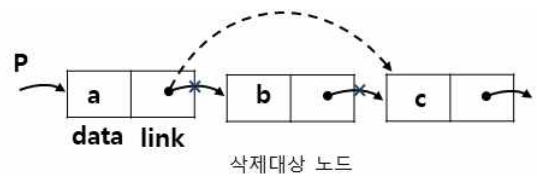
10. 다음 정렬 알고리즘 중 최악의 경우에 대한 시간 복잡도가 가장 낮은 알고리즘은?

- ① 선택 정렬(selection sort)
 ② 버블 정렬(bubble sort)
 ③ 셸 정렬(shell sort)
 ④ 힙 정렬(heap sort)

11. “aabbbcacdb”를 인코딩하기 위하여 허프만 코딩 트리(Huffman Coding Tree)를 만들어 가변길이 코드를 생성하는 것에 대한 설명 중 가장 적절하지 않은 것은?

- ① 빈도수에 대해서 오름차순으로 정리하면 d, c, a, b 순이다.
 ② 생성된 코드의 길이가 가장 짧은 것은 d이다.
 ③ 전체 압축된 코드는 19비트이다.
 ④ 생성된 a의 코드는 2비트이다.

12. 다음과 같은 단순 연결 리스트(singly linked list)에서 노드를 삭제하기 위해서 ㉠에 들어갈 코드로 가장 적절한 것은?



P->link = ㉠

- ① NULL;
 ② P;
 ③ P->link;
 ④ P->link->link;

13. 해시 함수는 $h(x) = x \bmod 7$ 이고, 해시 테이블의 크기는 7이다. 해시 테이블에 충돌이 발생하였을 경우 선형 조사법(linear probing)을 이용하여 해결한다. 키(key) 값이 (11, 3, 7, 4, 10)과 같이 들어왔을 때 생성된 해시 테이블로 가장 적절한 것은?

- ①

[0]	[1]	[2]	[3]	[4]	[5]	[6]
7			3	11	4	10

 ②

[0]	[1]	[2]	[3]	[4]	[5]	[6]
		3	11	4	10	7

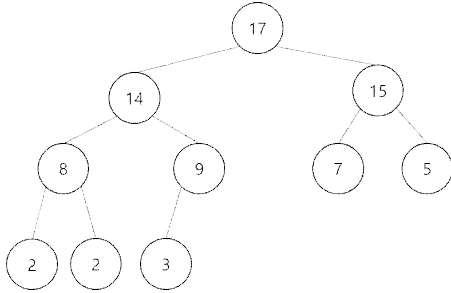
 ③

[0]	[1]	[2]	[3]	[4]	[5]	[6]
10	11		3	4		7

 ④

[0]	[1]	[2]	[3]	[4]	[5]	[6]
7			3	4	10	11

14. 그림과 같은 최대 힙(max heap)에 10을 삽입하고, 4를 삽입하고, 삭제를 두 번 수행하였다. 최대 힙을 배열 a에 저장한 경우 a[5]에 해당하는 값은? (단, 데이터는 a[1]부터 저장된다.)



- ① 3
② 4
③ 7
④ 9

15. 다음 숫자들에 대해 가장 낮은 자리부터 시작하는 기수정렬을 수행하려고 한다. 첫 번째 패스를 마친 후의 정렬 결과로 옳은 것은?

17	20	11	34	41	43	19	32
----	----	----	----	----	----	----	----

- ① 17 11 19 20 34 32 41 43
② 19 11 17 20 32 34 43 41
③ 20 11 41 32 43 34 17 19
④ 20 41 11 32 43 34 17 19

16. 다음 숫자들에 대하여 셸 정렬을 수행하려고 한다. 간격 7로 정렬한 후에, 간격 3으로 정렬한 상태로 옳은 것은?

15	24	26	7	20	21	12	4	8	1	9	23	6	25	28
----	----	----	---	----	----	----	---	---	---	---	----	---	----	----

- ① 4 8 1 7 9 6 12 15 23 21 20 24 26 25 28
② 4 8 1 7 20 6 12 15 24 26 9 23 21 25 28
③ 1 4 6 7 8 9 12 15 20 21 23 24 25 26 28
④ 9 12 4 7 1 15 23 6 8 20 21 24 26 25 28

17. 다음 인접 행렬과 같이 주어진 무방향 그래프에서 Dijkstra 알고리즘을 이용하여 정점 A로부터의 최단 거리를 찾고자 한다. 정점을 선택해나가는 과정에서 세 번째 정점을 선택한 후의 거리 배열로 적절한 것은? (단, 아래의 인접 행렬에서 0은 간선이 없음을 의미하며, 정점 A를 첫 번째로 선택한 정점으로 한다.)

	A	B	C	D	E	F	G
A	0	11	6	5	0	0	0
B	11	0	0	3	7	0	0
C	6	0	0	0	1	0	0
D	5	3	0	0	8	4	10
E	0	7	1	8	0	13	0
F	0	0	0	4	13	0	9
G	0	0	0	10	0	9	0

①

A	B	C	D	E	F	G
0	8	6	5	7	9	15

②

A	B	C	D	E	F	G
0	8	6	5	13	9	15

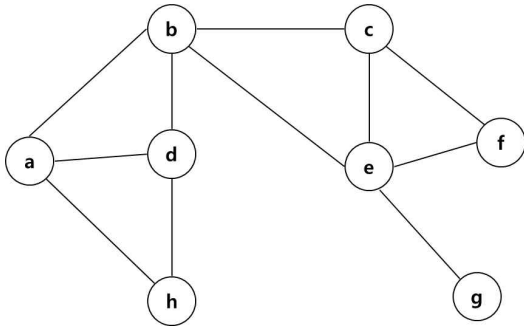
③

A	B	C	D	E	F	G
0	11	6	5	13	9	15

④

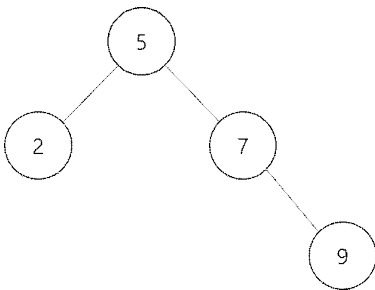
A	B	C	D	E	F	G
0	11	6	5	∞	∞	∞

18. 다음 그래프를 너비 우선 탐색(BFS : Breath First Search)하였을 때 결과로 가장 적절한 것은?



- ① a, b, c, d, e, f, g, h
- ② a, b, d, h, c, e, f, g
- ③ a, b, c, e, f, g, d, h
- ④ b, c, f, e, d, a, h, g

19. 다음과 같은 AVL 트리에서 8을 삽입할 경우의 회전은 무엇인가?



- ① LL
- ② RR
- ③ LR
- ④ RL

20. 해싱 시 충돌(collision) 및 오버플로우와 관련된 다음 설명 중 가장 적절하지 않은 것은?

- ① 선형 조사법(linear probing)은 오버플로우가 발생한 경우 다음 빈 버킷을 조사하는 방법이다.
- ② 이차 조사법(quadratic probing)은 군집화(clustering) 및 결합(coalescing) 문제를 완전히 해결할 수 있다.
- ③ 이중 해싱법(double hashing)은 오버플로우가 발생한 경우 기존의 해시 함수와 다른 해시 함수를 사용한다.
- ④ 체이닝(chaining) 방법을 사용하면 오버플로우가 일어나지 않는다.

21. 다음의 정렬 방법 중 최악의 경우 $O(n^2)$ 의 시간 복잡도를 가지는 정렬 방법이 아닌 것은?

- ① 선택 정렬
- ② 삽입 정렬
- ③ 합병 정렬
- ④ 퀵 정렬

22. 2-3-4 트리에 대한 설명 중 가장 적절하지 않은 것은?

- ① 2-3-4 트리는 하나의 노드가 4개의 자식까지 가질 수 있다.
- ② 2-3-4 트리에서 4-노드의 부모는 4-노드가 될 수 있다.
- ③ 2-3-4 트리는 2-3 트리와 달리 루트에서 단말 노드로 한 번만 이동하면서 삽입이 가능하다는 장점이 있다.
- ④ 2-3-4 트리에 삽입할 경우, 삽입 노드를 찾는 루트에서 단말 노드의 순회 과정 중 4-노드를 만나면 미리 분할한다.

23. 다음은 n 개의 정점으로 이루어진 그래프 G 의 모든 정점 쌍 사이의 최단 경로를 구하기 위한 Floyd 알고리즘을 수도코드(pseudo code)로 나타낸 것이다. 그래프 G 는 인접행렬 A 로 구현되어 있다고 가정할 때, 빈 칸 ㉠ 및 ㉡에 해당하는 것은?

```
Floyd(G):
  for k ← 0 to n-1
    for i ← 0 to n-1
      for j ← 0 to n-1
        A[i][j] = min( ㉠, ㉡ );
  }
```

- ① ㉠ $A[k][j]$ ㉡ $A[k][i] + A[j][k]$
 ② ㉠ $A[k][j]$ ㉡ $A[k][i]$
 ③ ㉠ $A[i][j]$ ㉡ $A[i][k] + A[k][j]$
 ④ ㉠ $A[i][j]$ ㉡ $A[j][k]$
24. 다음은 연결 리스트를 이용하여 구현한 스택인 연결된 스택(linked stack)의 pop 연산을 구현한 C언어 코드이다. top을 연결된 스택의 최상단 노드를 가리키는 전역 포인터 변수라고 할 때, 빈 칸에 알맞은 코드로 적절한 것은? (단, 연결된 스택에 하나 이상의 원소가 있다고 가정한다.)

```
typedef struct Node {
  int data;
  struct Node *link;
} Node;

Node* pop() {
  Node* tmp = top;
  int data = tmp->data;
  
  free(tmp);
  return data;
}
```

- ① $top = top->link;$
 ② $top->link = top;$
 ③ $tmp = tmp->link;$
 ④ $tmp->link = tmp;$

25. 배열 표현법은 이진 트리를 배열로 표현하기 위한 방법으로, 노드 i 의 부모 노드 인덱스는 $i/2$ 이고 왼쪽 및 오른쪽 자식의 인덱스가 각각 $2*i$, $2*i+1$ 인 관계가 성립한다. 다음과 같이 배열 표현법으로 표현된 이진 트리가 있다고 할 때, 이를 후위 순회(postorder traversal)한 결과로 옳은 것은? (단, 빈 칸은 해당 인덱스에 저장된 노드가 없음을 의미한다.)

node		A	B	C	D	E		F	G		H	I			J	
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

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