

자료구조론

문 1. 다음 함수를 이용하여 recursive(4)를 수행한 결과 값은?

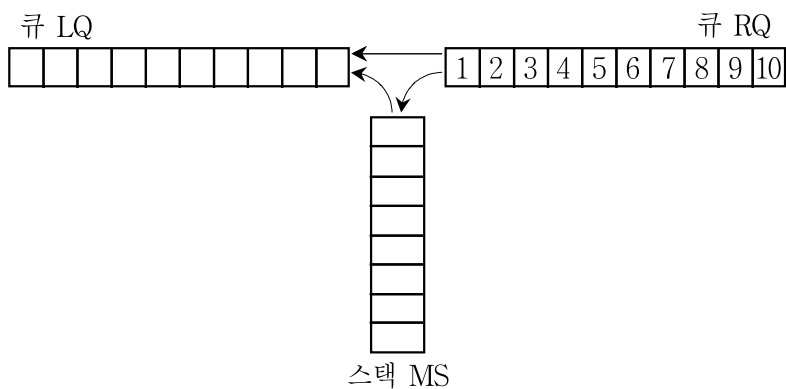
```
int recursive( int n ) {
    if(n<1) return 2;
    else return( 22 * recursive(n-1) + 1);
}
```

- ① 149 ② 265
③ 373 ④ 597

문 2. if 문장에서 포인터 변수 pt1과 pt2에 대한 설명으로 옳지 않은 것은?

- ① if (&pt1==&pt2)의 경우에 서로 다른 포인터 변수가 동일한 주소에 있을 수 있다.
② if (pt1==pt2)는 같은 대상을 가리키는지를 검증한다.
③ if (*pt1==*pt2)의 경우에 가리키는 대상이 다를지라도 변수 값은 같을 수 있다.
④ if (pt1==pt2)는 가리키는 대상의 주소가 동일한 것인지를 검증한다.

문 3. 스택 수(stack number)란 다음 장치의 왼쪽 큐 LQ에 들어 갈 수 있는 숫자열을 말한다. 오른쪽 큐 RQ로부터 데이터가 하나씩 중간에 있는 스택 MS를 거치거나 바로 LQ로 입력될 수 있다. 즉, 데이터가 LQ에 입력될 때 RQ에서 직접 입력되거나 MS에서 한 데이터를 삭제해서 LQ에 추가할 수 있다. 다음 장치에서 생성할 수 있는 스택 수는?



- ① 1 8 5 6 7 4 9 3 10 2 ② 3 4 6 7 8 5 9 2 1 10
③ 3 4 8 5 6 7 2 9 1 10 ④ 9 4 7 5 6 3 8 1 2 10

문 4. 합병정렬(merge sort)에 대한 설명으로 옳지 않은 것은?

- ① 점근분석(asymptotic analysis)할 때, 퀵정렬(quick sort)의 최악수행시간 보다 빠르다.
② 추가적인 메모리 사용없이 간단히 구현할 수 있다.
③ 크기 n인 입력에 대한 최악수행시간을 T(n)이라 할 때, T(n) = 2 × T(n/2) + cn으로 나타낼 수 있다. (단, c ≥ 1 상수)
④ 분할정복(divide and conquer) 기법을 사용한다.

문 5. 방향그래프의 행(row)은 출발 노드로 열(column)은 도착 노드로 매핑하여 다음과 같이 인접행렬로 표현될 때, 위상 정렬(topological sort)이 가능한 것은?

①		a	b	c	d	e	f
a			1	1			
b				1	1		
c					1	1	
d				1			1
e					1		1
f							

②		a	b	c	d	e	f
a			1	1			
b				1	1		
c					1	1	
d							1
e		1		1			1
f							

③		a	b	c	d	e	f
a			1	1			
b				1	1		
c					1	1	
d							1
e					1		1
f							

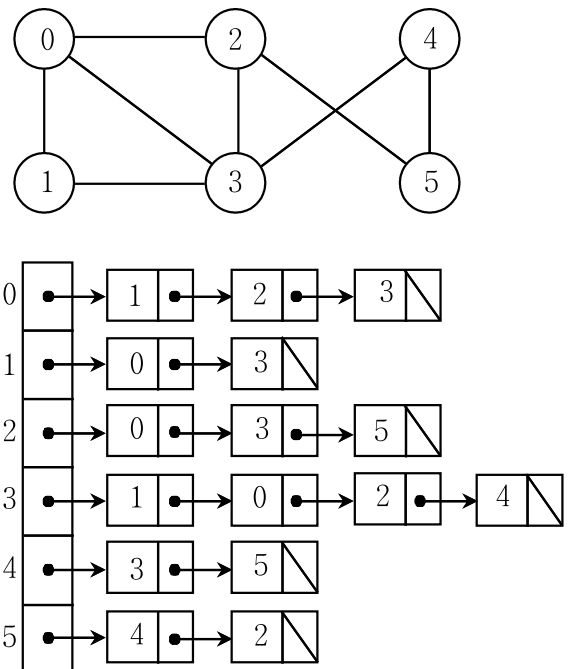
④		a	b	c	d	e	f
a			1	1			
b				1	1		
c					1	1	
d							1
e					1		1
f	1						

문 6. 다음 데이터를 이용하여 AVL 트리를 생성할 때, 이에 대한 설명으로 옳지 않은 것은?

12, 11, 10, 5, 3, 7, 6, 1, 13, 2, 4

- ① AVL트리에서 7을 검색하기 위해서는 4번의 비교가 필요하다.
② AVL트리의 루트 값은 5이다.
③ 4가 삽입될 때, AVL 트리의 균형이 깨져서 재구성이 발생한다.
④ 6은 리프노드이다.

문 7. 다음 연결리스트(linked list)로 표현된 그래프를 위한 두 연산(A, B)에 대해 ‘최적으로 구현’한다고 할 때, 연산 A, B 각각에 대한 최악의 시간복잡도는? (단, 그래프의 노드의 개수를 n이라고 하고, 각 노드 쌍 사이에는 최대 1개의 간선(edge)이 존재할 수 있다)



연산 (A): 노드 i에서 노드 j로의 간선을 삽입한다. (단, 삽입 연산은 항상 간선이 존재하지 않는 노드 쌍에 대해서 수행된다)

연산 (B): 노드 i에서 노드 j로의 간선을 찾아 제거한다.

- | | |
|------------|----------|
| 연산 (A) | 연산 (B) |
| ① O(1) | O(n) |
| ② O(log n) | O(1) |
| ③ O(1) | O(log n) |
| ④ O(n) | O(n) |

문 8. 다음 행렬은 일곱 개의 정점(vertex)을 갖는 그래프의 간선(edge)들 간의 거리를 나타낸다. Dijkstra 알고리즘을 적용하여 정점 A로부터 나머지 정점들까지 최단경로를 찾을 때, 이에 해당하는 정점의 순서는? (단, ∞ 는 두 정점 간에 해당 간선이 없다)

	A	B	C	D	E	F	G
A	0	4	9	∞	∞	∞	∞
B	4	0	4	10	∞	∞	∞
C	9	4	0	∞	8	∞	1
D	∞	10	∞	0	2	5	∞
E	∞	∞	8	2	0	∞	6
F	∞	∞	∞	5	∞	0	3
G	∞	∞	1	∞	6	3	0

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

문 9. 이진탐색트리의 노드 삽입에 대한 다음 C 프로그램의 ㉠ ~ ㉣에 들어갈 내용은? (단, T는 이진탐색트리, X는 이진탐색트리에 삽입할 값이다)

```

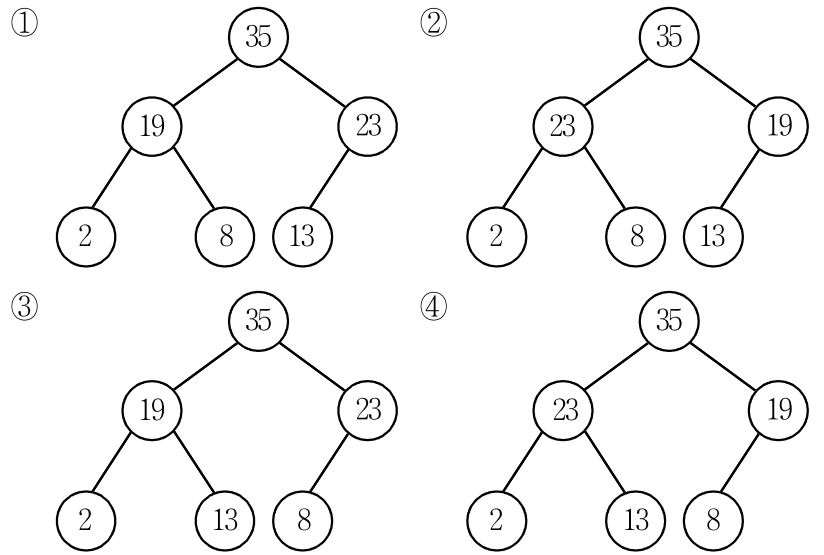
struct TreeNode {
    int Element;
    struct TreeNode *Left;
    struct TreeNode *Right;
};
typedef struct TreeNode *SearchTree;

SearchTree Insert( int X, SearchTree T )
{
    if ( T == NULL ) {
        T = (SearchTree)malloc(sizeof(struct TreeNode));
        T->Element = X;
        T->Left = T->Right = NULL;
    } else if ( X < T->Element )
        ㉠ _____
    else if ( X > T->Element )
        ㉣ _____
    return T;
}
    
```

- ㉠ $T \rightarrow \text{Left} = \text{Insert}(X, T);$ $T \rightarrow \text{Right} = \text{Insert}(X, T);$
- ㉡ $T \rightarrow \text{Left} = \text{Insert}(X, T \rightarrow \text{Left});$ $T \rightarrow \text{Right} = \text{Insert}(X, T \rightarrow \text{Right});$
- ㉢ $T \rightarrow \text{Left} = \text{Insert}(X, T \rightarrow \text{Element});$ $T \rightarrow \text{Right} = \text{Insert}(X, T \rightarrow \text{Element});$
- ㉣ $T \rightarrow \text{Left} = \text{Insert}(X, T \rightarrow \text{Right});$ $T \rightarrow \text{Right} = \text{Insert}(X, T \rightarrow \text{Left});$

문 10. 다음 데이터들을 공백힙(empty heap)에 차례대로 삽입하여 최대힙(max heap)을 생성하였다. 생성된 최대힙에 삭제 연산을 두 번 수행한 후의 결과는?

2, 13, 55, 8, 35, 47, 23, 19



문 11. 다음 연산식을 후위(postfix) 연산식으로 변환했을 때, 4번째 연산자는?

$1 + (2 - 3 * 4 / (5 + 6)) - 7$

- ① +
- ② -
- ③ *
- ④ /

문 12. 다음 선형연결리스트(linear linked list)에서 LEE를 찾기 위한 비교 횟수는? (단, HEADER는 3 이다)

Index	Name	Pointer
1	KIM	6
2	PARK	1
3	CHOI	4
4	BAEK	2
5	LEE	7
6	MIN	5
7	ANN	8
8	JUNG	-

- ① 4
- ② 5
- ③ 6
- ④ 7

문 13. 정렬(sort)에 대한 설명으로 옳지 않은 것은?

- ① 퀵정렬(quick sort)은 스택을 이용하여 수행한다.
- ② 셸정렬(shell sort)은 최적의 경우에 선택정렬(selection sort)보다 빠르다.
- ③ 힙정렬(heap sort)은 배열을 이용한 포화 이진트리(full binary tree)를 사용한다.
- ④ 10개 데이터의 버블정렬(bubble sort)은 최대 45번의 비교 연산을 수행한다.

문 14. 해시함수(hash function)로 $h(k)=k \bmod 7$ 을 사용하고, 선형탐색(linear probing) 기법을 이용하여 충돌을 해결하고자 한다. 키가 3, 10, 5, 9, 12 순서대로 입력되었을 때, 생성된 해시테이블은?

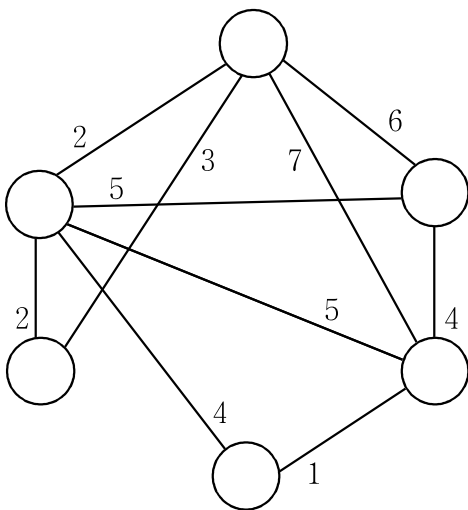
- | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|
| ① | [0] | [1] | [2] | [3] | [4] | [5] | [6] |
| | 12 | 9 | | 3 | 10 | 5 | |
| ② | [0] | [1] | [2] | [3] | [4] | [5] | [6] |
| | 12 | | | 3 | 10 | 5 | 9 |
| ③ | [0] | [1] | [2] | [3] | [4] | [5] | [6] |
| | | 9 | 10 | 3 | 12 | 5 | |
| ④ | [0] | [1] | [2] | [3] | [4] | [5] | [6] |
| | | | 9 | 3 | 10 | 5 | 12 |

문 15. 비어있는 이진 탐색 트리(binary search tree)에 다음 키 값들을 차례로 삽입하여 동적으로 이진 탐색 트리를 구성하였을 때, 5를 탐색하는데 필요한 비교 횟수는?

23, 18, 2, 16, 4, 10, 5, 39, 26, 15

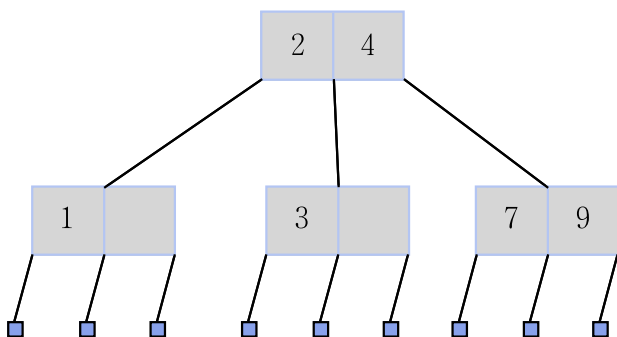
- ① 3 ② 5
③ 7 ④ 9

문 16. 다음 그래프를 이용하여 Kruskal 알고리즘을 적용하였을 때, 생성된 최소 비용 신장 트리의 비용은?



- ① 11 ② 12
③ 13 ④ 14

문 17. 다음과 같은 4개의 노드를 가진 2-3 트리에 키 값 6을 추가하였을 때, 트리의 전체 노드의 수는?



- ① 4개 ② 5개
③ 6개 ④ 7개

문 18. 다항식(polynomial) 표현을 위해 구조체 poly를 정의하였다. 이 구조체를 이용하여 $A(x) = -3x^{999} - 2x^{888} + 1$ 과 $B(x) = x^{10} + x^9 + x^5$ 을 저장하기 위한 최소 메모리 용량[byte]은? (단, 정수(int)를 표현하는 데 2 byte를 사용하고, 실수(float)를 표현하는 데 4 byte를 사용한다)

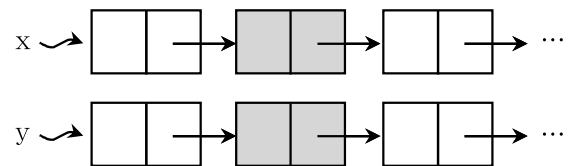
```
typedef struct {
    int exp;
    float coef;
} poly;
```

- ① 22 ② 24
③ 36 ④ 38

문 19. 다음 중 컴퓨터 프로그래밍을 통하여 문제를 해결하는데 최적의 시간복잡도가 가장 큰 것은?

- ① 임의의 정수 100,000개를 선택정렬(selection sort)로 정렬하는 문제
- ② 정렬된 100,000개의 정수에서 임의의 정수를 이진탐색하는 문제
- ③ 반복 호출로 피보나치수열 $F_0, F_1, F_2, \dots$ 에서 F_{100000} 값을 구하는 문제
- ④ 재귀호출로 하노이 탑(tower of Hanoi)의 100,000개 원판 이동 문제

문 20. 구조체 list를 이용한 두 개의 단순연결리스트 x, y의 후위노드 (음영부분)들을 서로 스왑(swap)하기 위한 코드는? (단, 후위 노드들의 next는 NULL이 아니다)



```
struct list {  int data;  struct list *next;  } *x, *y, *tmp;
```

- ① tmp = x->next;
x->next->next = y->next->next;
y->next->next = tmp;
tmp = y->next;
x->next = y->next;
y->next = tmp;
- ② tmp = x->next->next;
x->next->next = y->next->next;
y->next->next = tmp;
tmp = x->next;
x->next = y->next;
y->next = tmp;
- ③ tmp = x->next;
x->next = y->next;
y->next = tmp;
tmp = x->next;
x->next->next = y->next->next;
y->next->next = tmp;
- ④ tmp = x->next->next;
y->next->next = tmp;
x->next->next = y->next->next;
tmp = x->next;
y->next = tmp;
x->next = y->next;