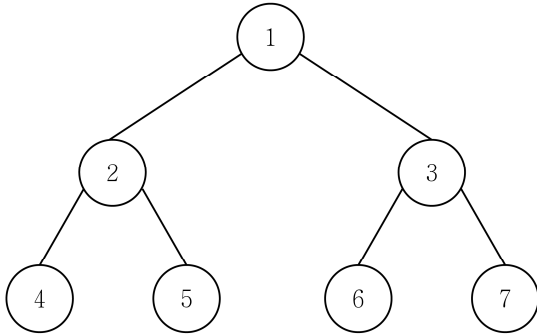


자료구조론

문 1. 세 개의 노드를 가진 무방향 완전 그래프의 신장 트리(spanning tree)는 최대 몇 가지인가?

- ① 1 ② 2
③ 3 ④ 4

문 2. 다음 트리를 중위 순회한 결과로 옳은 것은?



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

문 3. 다음은 입력 개수 n 에 대한 알고리즘 A ~ D의 수행시간 복잡도를 나타낸 것이다. 알고리즘 A ~ D를 수행시간 효율이 좋은 것부터 순서대로 나열한 것은?

알고리즘	수행시간 복잡도
A	$O(n \log n)$
B	$O(2^n)$
C	$O(n^2)$
D	$O(n!)$

- ① A, C, B, D ② A, C, D, B
③ C, A, B, D ④ C, A, D, B

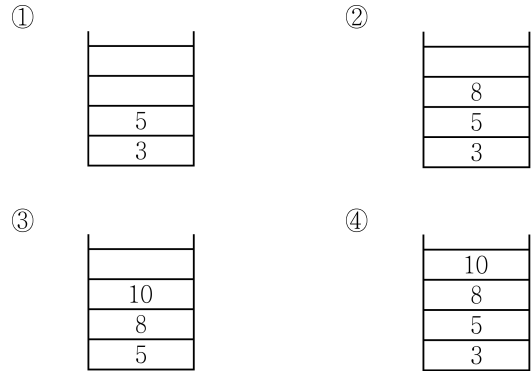
문 4. 이진 트리에 대한 설명으로 옳은 것만을 모두 고르면?

- ㉑. 공백 이진 트리의 높이를 0으로 본다면, 높이가 5인 이진 트리의 최대 노드 수는 15이다.
- ㉒. 9개의 노드를 가지고 있는 이진 트리는 8개의 간선을 가진다.
- ㉓. 루트 노드의 인덱스를 1로 하는 1차원 배열로 완전 이진 트리를 표현할 때, 인덱스 13인 노드의 부모 노드는 인덱스가 7이다.
- ㉔. 7개 노드로 구성할 수 있는 이진 트리의 최대 높이와 최소 높이의 차는 4이다.

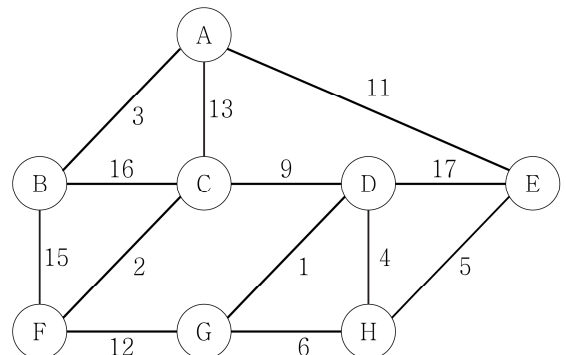
- ① $\neg$, $\perp$
② $\perp$, $\top$
③ $\perp$, $\exists$
④ $\top$, $\exists$

문 5. 크기가 4인 빈 스택에 다음 연산을 차례로 수행한 후의 스택 상태를 바르게 표현한 것은?

push(3) → push(7) → pop() → push(5) → push(8) →
peek() → push(10) → pop()



문 6. 그림은 정점 A ~ H 간의 거리를 나타낸 그래프이다. 다익스트라(Dijkstra) 알고리즘을 이용하여 정점 A로부터 다른 모든 정점까지의 최단 경로를 구하고, 각 최단 경로의 거리를 짧은 것부터 순서대로 나열한 것은?



- ① 3, 11, 13, 15, 16, 20, 21
- ② 3, 11, 13, 15, 16, 21, 22
- ③ 3, 11, 13, 16, 18, 20, 21
- ④ 3, 11, 13, 16, 18, 21, 22

문 7. 다음 데이터를 차례대로 하나씩 입력받아 AVL 트리를 생성할 때, 필요한 회전을 적용 순서대로 바르게 나열한 것은?

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

- ① LL회전, LR회전, LR회전
- ② LL회전, LR회전, RL회전
- ③ RR회전, LL회전, LR회전
- ④ RR회전, LL회전, RL회전

문 8. 다음 C 코드는 원형 연결 리스트에 속한 모든 노드의 값을 더하여 출력한다. (가) ~ (다)에 들어갈 내용을 바르게 연결한 것은?

```
#include <stdio.h>
#include <malloc.h>
typedef struct Node {
    struct Node *a;
    int b;
} Node;

int listSum(Node *p) {
    int sum = 0;
    Node *s = p;
    do {
        sum += (가) ;
        p = (나) ;
    } while ( (다) );
    return sum;
}

int main() {
    int MAX = 5;
    int i, s;
    Node *p[MAX];

    for( i = 0; i < MAX; i++) {
        p[i] = (Node*)malloc(sizeof(Node));
        p[i]->b = i+1;
    }
    for( i = 0; i < MAX-1; i++)
        p[i]->a = p[i+1];
    p[MAX-1]->a = p[0];
    s = listSum(p[0]);
    printf("Sum = %d", s);
}
```

(가)

(나)

(다)

- | | | |
|----------|--------|-----------|
| ① p.b | p.a | p != NULL |
| ② p->a | p->b | p != s |
| ③ p->b | p->a | p != NULL |
| ④ (*p).b | (*p).a | p != s |

문 9. 다음 C 코드를 수행하면 4개의 값이 출력된다. 출력 값 중 나머지 3개와 다른 값을 출력하는 C 코드의 라인 번호는?

라인 번호	C 코드
1	#include <stdio.h>
2	
3	int main() {
4	int array[] = {0,1,2};
5	int *p = array+2;
6	
7	printf("%d\n", *(p-1));
8	printf("%d\n", *p-1);
9	printf("%d\n", *(p--));
10	printf("%d\n", *p);
11	}

- | | |
|-----|------|
| ① 7 | ② 8 |
| ③ 9 | ④ 10 |

문 10. 다음 tree_height() 함수는 이진 트리의 루트 노드를 매개변수로 받아 트리의 높이를 반환한다. (가) ~ (다)에 들어갈 내용을 바르게 연결한 것은?

```
typedef struct node {
    int value;
    struct node *left;
    struct node *right;
} node;

int tree_height (node *ptr) {
    int left_h, right_h;
    if (ptr == NULL)
        return 0;
    else {
        left_h = tree_height (ptr->left);
        right_h = tree_height (ptr->right);
        if ( (가) )
            return (나) ;
        else
            return (다) ;
    }
}
```

(가)

(나)

(다)

- | | | |
|--------------------|------------|-------------|
| ① left_h > right_h | left_h | right_h |
| ② left_h > right_h | left_h + 1 | right_h + 1 |
| ③ left_h < right_h | left_h | right_h |
| ④ left_h < right_h | left_h + 1 | right_h + 1 |

문 11. 10개의 정점을 가진 무방향 그래프가 가질 수 있는 간선의 최대 개수는? (단, 각 정점에서 자기 자신으로의 간선은 허용하지 않고, 두 정점 사이의 간선은 최대 1개이다)

- ① 42
② 43
③ 44
④ 45

문 12. 회문(palindrome)은 영어 단어 radar와 같이 순서를 거꾸로 뒤집어도 원본과 동일한 단어를 의미한다. 다음 C 코드의 isPalindrome() 함수는 입력 문자열이 회문인 경우 1을 반환하고, 그렇지 않을 경우는 0을 반환한다. (가), (나)에 들어갈 내용을 바르게 연결한 것은?

```
#include <stdio.h>
char stack[100];
int top = -1;

void push(char ch) {stack[++top] = ch;}

char pop() {return stack[top--];}

int isEmpty() {return top == -1;}

int isPalindrome(char *str) {
    int i;
    for ( i = 0; str[i]; i++ )
        (가) ;

    for ( i = 0; !isEmpty(); i++ )
        if ( (나) )
            return 0;
    return 1;
}

int main() {
    char input[100];
    scanf("%s", input);
    printf("%s",
        isPalindrome(input) ? "True" : "False");
}
```

(가)

(나)

- ① push(str[i]) str[i] == pop()
② push(str[i]) str[i] != pop()
③ push(str[top-i]) str[i] == pop()
④ push(str[top-i]) str[i] != pop()

문 13. 다음 C 코드는 이진 탐색을 이용하여 정수 데이터를 탐색하는 함수이다. (가), (나)에 들어갈 내용을 바르게 연결한 것은? (단, key는 찾고자 하는 값, a[]는 오름차순으로 정렬된 정수 배열, n은 배열의 크기이다)

```
int search(int key, int a[], int n) {
    int mid;
    int left = 0;
    int right = n-1;
    while ( (가) ) {
        mid = (right + left) / 2;
        if (key == a[mid]) return mid;
        if ( (나) ) right = mid - 1;
        else left = mid + 1;
    }
    return -1;
}
```

(가)

(나)

- ① left < right key < a[mid]
② left < right key > a[mid]
③ left <= right key < a[mid]
④ left <= right key > a[mid]

문 14. 다음 C 코드는 반복구조를 이용하여 n번째 피보나치 수를 구하는 함수이다. (가), (나)에 들어갈 내용을 바르게 연결한 것은? (단, n은 0 이상의 정수이다)

```
int fibonacci(int n){
    int x = 0, y = 1, z = 0;
    if (n < 2) return n;
    while ( (가) ) {
        (나) ;
        x = y;
        y = z;
    }
    return z;
}
```

(가)

(나)

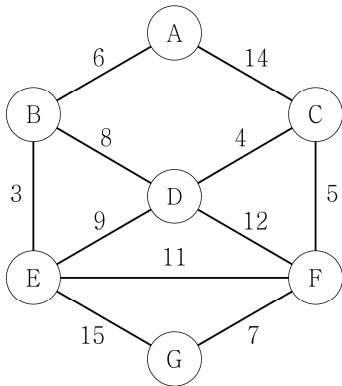
- ① n-- > 1 z = x + y
② n-- > 1 z = x - y
③ n-- >= 1 z = x + y
④ n-- <= 1 z = x - y

문 15. 다음 설명에 해당하는 노드 수 A, B, C의 합은? (단, 트리의 높이는 트리의 최대 레벨을 의미하며, 루트 노드의 레벨을 1로 한다)

- A = 높이가 5인 완전 이진 트리가 가질 수 있는 최소 노드 수
- B = 높이가 6인 포화 이진 트리의 노드 수
- C = 높이가 7인 완전 이진 트리가 가질 수 있는 최대 노드 수

- ① 142
② 206
③ 209
④ 222

문 16. 다음 그래프에 대하여 Sollin 알고리즘으로 최소 비용 신장 트리를 구하고자 한다. 각 노드가 하나의 트리로 구성된 초기 상태에서 한 단계 처리한 이후의 결과로 두 개의 트리로 구성된 포리스트(forest)가 만들어졌을 때, 각 트리의 가중치 합 중 큰 값은?

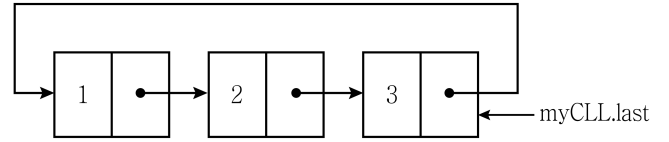


- ① 9
② 12
③ 13
④ 16

문 17. 비어있는 레드 블랙 트리(red black tree)에 차례대로 키(key) 값 1, 3, 2, 4, 5, 6을 넣고 완성한 레드 블랙 트리를 전위 순회한 결과는? (단, 키를 저장하고 있는 노드만 순회하고, 순회 결과에서 키 값 뒤에 B를 붙이면 블랙 노드, R을 붙이면 레드 노드임을 나타낸다)

- ① 2B, 1B, 4R, 3B, 5B, 6R
② 2B, 1R, 4R, 3B, 5B, 6R
③ 4B, 2R, 1B, 3B, 5B, 6R
④ 4B, 2R, 1B, 3R, 5B, 6R

문 18. 그림과 같은 원형 연결 리스트 myCLL을 만들기 위해 다음 파이썬 코드를 작성하였다. (가) ~ (다)에 들어갈 코드를 A ~ D에서 바르게 연결한 것은?



```
class Node:
    def __init__(self, data, next):
        self.data = data
        self.next = next
```

```
class CircularLinkedList:
    def __init__(self):
        self.last = None
```

```
def insert(self, data):
    n = Node(data, None)
    if self.last is None:
        n.next = n
        self.last = n
```

```
    else:
```

- (가)
(나)
(다)

```
myCLL = CircularLinkedList()
myCLL.insert(1)
myCLL.insert(2)
myCLL.insert(3)
```

- A. n.next = self.last
B. n.next = self.last.next
C. self.last = n
D. self.last.next = n

(가) (나) (다)

- ① A C D
② A D C
③ B C D
④ B D C

문 19. 다음 C 코드는 단순 연결 리스트를 이용하여 스택의 pop() 함수를 구현한 것이다. (가), (나)에 들어갈 내용을 바르게 연결한 것은?

```
struct stack {
    int data;
    struct stack* link;
};
struct stack* top = NULL;
int pop() {
    struct stack* temp = top;
    int item;
    if (!temp) {
        printf("Stack is empty.\n");
        exit(1);
    }
    item = temp->data;
    (가) ;
    free(temp);
    (나) ;
}
```

(가)

(나)

- | | |
|---------------------|------------------|
| ① temp = temp->link | return item |
| ② temp = temp->link | return top->data |
| ③ top = temp->link | return item |
| ④ top = temp->link | return top->data |

문 20. 다음 조건에 따라 공백 해시 테이블에 키(key) 값 12, 15, 25, 34, 37, 75, 62, 33, 47, 55, 21, 42, 53을 순서대로 삽입할 때, 충돌 횟수가 가장 많은 키 값은? (단, 해시 함수 또는 선형 조사법을 적용하여 찾아간 버킷에 빈 슬롯이 없을 때마다 충돌이 발생한 것으로 본다)

- 해시 함수: $h(x) = x \bmod 13$
- 충돌 해결책: 선형 조사법(linear probing)
- 해시 테이블의 크기: 13개 버킷(0부터 12까지 인덱스를 가짐)
- 해시 테이블 버킷당 슬롯 수: 1개

- ① 21
② 42
③ 53
④ 62

문 21. 다음 C 코드의 reverseList()는 단순 연결 리스트에 있는 노드의 순서를 역순으로 만드는 함수이다. (가) ~ (다)에 들어갈 내용을 바르게 연결한 것은?

```
struct node {
    int data;
    struct node *link;
} *head;

void reverseList() {
    struct node *prevNode, *curNode;
    prevNode = head;
    curNode = head;
    head = NULL;

    while ( (가) ) {
        curNode = curNode->link;
        prevNode->link = head;
        (나) ;
        (다) ;
    }
}
```

(가)

(나)

(다)

- | | | |
|-------------------|-----------------|--------------------|
| ① curNode != NULL | head = curNode | curNode = prevNode |
| ② curNode != NULL | head = prevNode | prevNode = curNode |
| ③ head != NULL | head = curNode | curNode = prevNode |
| ④ head != NULL | head = prevNode | prevNode = curNode |

문 22. 다음 배열에 대하여, 위치 값이 증가함에 따라 키(key) 값도 이에 정비례하여 증가한다는 가정으로 탐색 위치를 계산하는 보간 탐색을 수행할 때, 키 45를 찾을 때까지 탐색 위치를 계산한 횟수는? (단, 탐색 위치 계산 시 소수점 이하는 버린다)

위치	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
키	2	3	4	7	13	18	22	27	32	42	45	70	72

- ① 2
② 3
③ 4
④ 5

문 23. 다음 C 코드는 쉘 정렬을 구현한 것이다. 코드의 출력 결과는?

```
#include <stdio.h>
void sort(int arr[], int n) {
    int count = 1;
    int i, j, k, h, val;
    for (h = n/2; h > 0; h /= 2) {
        for (i = 0; i < h; i++) {
            for (j = i+h; j < n; j += h) {
                val = arr[j];
                k = j;
                while (k > h-1 && arr[k-h] > val) {
                    arr[k] = arr[k-h];
                    k -= h;
                }
                arr[k] = val;
            }
        }
        if (count == 5)
            for (int t = 0; t < 6; t++)
                printf("%d ", arr[t]);
        count++;
    }
}

int main() {
    int a[6] = {5, 9, 2, 3, 6, 7};
    sort(a, 6);
}
```

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

문 24. 루트 노드의 인덱스를 1로 하는 1차원 배열을 이용하여 최대 힙(max heap)을 구현한 후, 다음 데이터를 차례대로 하나씩 힙에 삽입하였다. 이후 힙 삭제 연산을 1회 수행한 후 배열의 인덱스 6에 저장된 데이터는?

15, 8, 10, 18, 22, 13, 26, 7

- ① 8 ② 10
- ③ 13 ④ 15

문 25. 행 우선 순서로 저장되는 C 언어 2차원 배열 char a[3][5]가 선언되고 초기화되었을 때, a[2][3]과 다른 값을 가지는 것은?

- ① *(a+5*2+3)
- ② *(&a[0][0]+5*2+3)
- ③ (a[0]+5*2)[3]
- ④ (*(a+2))[3]