

## 자료구조론

- 문 1. 다음 해시구조는 개방 주소법(open addressing) 중 선형 검색법(linear probing)을 사용하여 오버플로우를 처리하는 예제이다. 해시함수가 '입력되는 색인키의 첫 번째 문자에 대한 알파벳 순위'라고 가정할 때, 버킷 테이블의 ①, ②에 대한 접근 횟수로 옳게 짝지어진 것은? (예로, 해시함수  $h(\alpha)=0$ ,  $h(\beta)=1$ ,  $h(\text{computer})=2$ ,  $h(\text{data})=3$ ,  $h(\text{email})=4$ ,  $h(\text{father})=5$  등을 의미한다)

| 버킷  | 색인키    | 탐색에 필요한 버킷 접근 횟수 |
|-----|--------|------------------|
| 0   | ascii  | 1                |
| 1   | atoi   | 2                |
| 2   | char   | 1                |
| 3   | define | 1                |
| 4   | equal  | 1                |
| 5   | ceil   | ①                |
| 6   | for    | ②                |
| ... |        |                  |

①

②

① 3

1

② 4

2

③ 1

1

④ 4

1

- 문 2. 다음은 원형연결리스트(circular linked list)의 맨 앞에 새로운 노드를 추가하는 알고리즘의 의사코드이다. ①, ②에 들어갈 문장으로 옳은 것은? (단, 원형연결리스트에서 'last'는 리스트의 맨 마지막 노드를 가리키는 포인터이고 초기치는 NULL이며, 변수 'p'는 새로 삽입되는 노드를 나타낸다. 또한 리스트에 3개 이상의 노드를 추가할 수 있어야 한다)

```

struct node {
    int key;
    struct node *next;
} *last = NULL; //마지막 노드를 가리키는 포인터

```

Function insertFront(struct node \*p)

if (last == NULL) then // 리스트가 빈 경우

last = p;

p→next = p;

else // 리스트에 하나 이상의 노드가 있는 경우

①

②

end if

return;

①

②

① p→next = last→next;

last→next = p;

② last→next = p;

p→next = last→next;

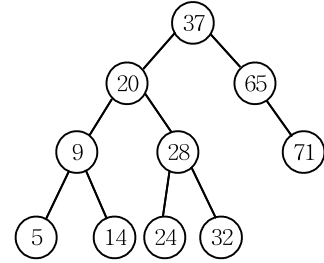
③ p→next = last;

last→next = p;

④ last→next = p;

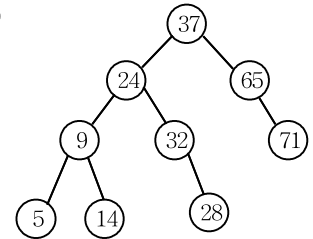
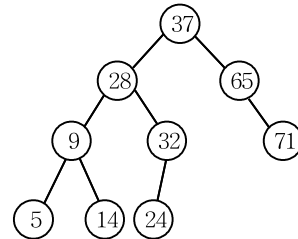
p→next = last;

- 문 3. 다음 이진탐색트리(Binary Search Tree)에서 색인키 '20'을 삭제한 후, 트리를 재구성한 것으로 옳은 것은?



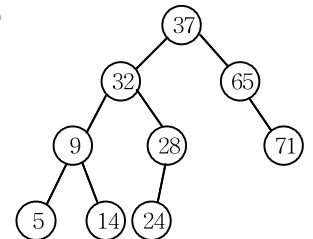
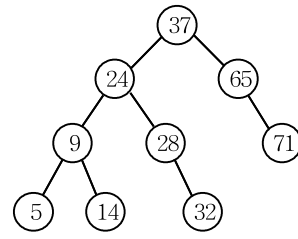
①

②



③

④



- 문 4. calculate() 함수의 기능을 표현하는 수식으로 옳은 것은? (단, n은 양수만을 허용한다)

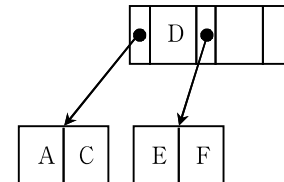
```

double calculate(float x, int n)
{
    double t;
    if (n < 0) exit(1);
    if (n == 0) return 1.0;
    else if (n % 2 == 0) {
        t = calculate(x, n / 2);
        return t * t;
    }
    else
        return x * calculate(x, n - 1);
}

```

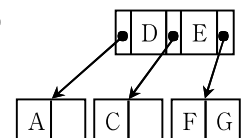
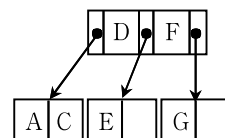
①  $x \log_2 n$ ②  $x^{\log_2 n}$ ③  $xn$ ④  $x^n$ 

- 문 5. 차수가 3인 B-트리에서 G를 삽입한 후 모습은?



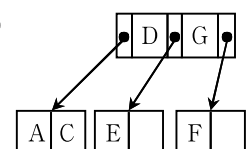
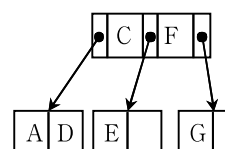
①

②

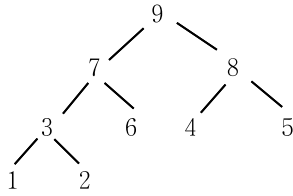


③

④



문 6. 다음 최대힙(Max Heap) 트리를 일차원 배열에 표현할 경우, 배열에서 최댓값을 두 번 삭제한 후 남아있는 배열의 상태로 옳은 것은? (단, 최대힙 배열에서 첫 번째 요소는 비어있다)



최대힙의 배열 표현: 

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

- ① 

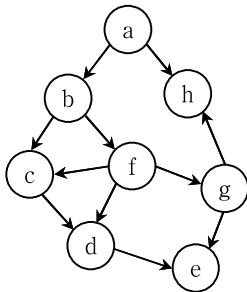
|   |   |   |   |   |   |   |   |  |  |
|---|---|---|---|---|---|---|---|--|--|
| - | 7 | 3 | 6 | 4 | 5 | 1 | 2 |  |  |
|---|---|---|---|---|---|---|---|--|--|
- ② 

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

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

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

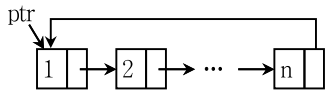
문 7. 다음 방향성 그래프에서 정점 'a'부터 시작하는 너비 우선 탐색(Breadth First Search)을 수행하는 경우, 6번째로 방문될 수 있는 정점은? (단, 정점 'a'는 첫 번째 방문 노드라고 가정한다)



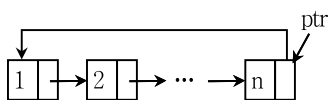
- ① e                      ② f
- ③ g                      ④ h

문 8. 원형연결리스트(circular linked list)의 맨 앞에 새로운 노드를 삽입할 때, ㉠, ㉡에 대한 시간 복잡도 표현은?

㉠ 포인터 ptr이 원형연결리스트의 맨 앞 노드를 가리키는 경우

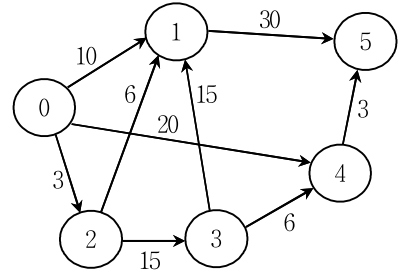


㉡ 포인터 ptr이 원형연결리스트의 맨 뒤 노드를 가리키는 경우



- ㉠                      ㉡
- ①  $O(1)$               ①  $O(1)$
- ②  $O(1)$               ②  $O(n)$
- ③  $O(n)$               ③  $O(1)$
- ④  $O(n)$               ④  $O(n)$

문 9. 다음 그래프는 각 정점들 사이의 거리를 간선에 나타낸 것이다. 정점 0에서 각 정점 1, 2, 3, 4, 5까지의 최단경로를 Dijkstra 최단경로 알고리즘으로 구할 때, 최단경로가 발견된 정점들의 순서로 옳은 것은?



- ① 0, 2, 3, 1, 5, 4
- ② 0, 2, 1, 3, 4, 5
- ③ 0, 2, 4, 1, 3, 5
- ④ 0, 1, 5, 2, 3, 4

문 10. 다음은 이진트리를 후위순회(postorder traversal)와 중위순회(inorder traversal)로 방문한 결과이다. 이 두 가지 순회 결과를 이용하여 이진트리를 구성한 것으로 옳은 것은?

Postorder 방문순서 : ADEBHGJKIFC

Inorder 방문순서 : ABDECGHFJIK

- ①
- ②
- ③
- ④

문 11. 다음 함수를 이용하여 fl(4)를 수행한 결과 값은?

```
int fl(int n)
{
    if (n <= 0)
        return 0;
    else if (n <= 2)
        return n;
    else
        return fl(n-3) + fl(n-2) + fl(n-1);
}
```

- ① 3                      ② 4
- ③ 5                      ④ 6

문 12. 다음 알고리즘의 시간 복잡도를 빅오(O) 표기법으로 표현한 것으로 옳은 것은?

```

Procedure calculate(int n)
{
    int i, k, m, x = 0;
    for (i = 0; i < (n - 5); i++)
        for (k = 0; k < 100; k++)
            for (m = 0; m < 1000; m++)
                x++;
}

```

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

문 13. 다음 배열 데이터에서 첫 번째 원소 '66'을 피벗(pivot)으로 퀵정렬 (Quick Sort)을 수행하는 경우, 순환 함수 quickSort()의 첫 번째 라운드가 완료되었을 때 생성된 배열 상태로 옳은 것은? (단, 퀵정렬 함수는 quickSort(int a[], int left, int right)이고, 매개변수 left와 right는 배열 a에서 정렬할 시작 위치와 끝 위치를 나타낸다)

```
int a[] = { 66, 51, 11, 98, 55, 1, 70, 35, 79, 41 };
```

- ① 11, 51, 35, 41, 55, 1, 66, 70, 79, 98  
 ② 35, 51, 11, 41, 55, 1, 66, 70, 79, 98  
 ③ 55, 79, 70, 98, 66, 1, 11, 35, 51, 41  
 ④ 66, 79, 70, 98, 55, 1, 11, 35, 51, 41

문 14. 행 우선(row major)으로 저장되는 3차원 배열 a[4][5][3]이 있을 때, 이 배열의 첫 번째 원소 a[0][0][0]의 주소를  $\alpha$ 라고 하면, a[2][3][1]의 주소를  $\alpha + i$ 로 표현했을 때  $i$ 의 값은?

- ① 6                                  ② 40  
 ③ 56                                ④ 168

문 15. 다음은 C 언어로 작성된 일차원 배열의 예제 프로그램이다. ㉠에 들어갈 수 있는 문장과 그 출력 결과로 옳게 짝지어진 것은?

```

int main()
{
    int i, j, A[100];
    for (i = 0; i < 100; i++)
        A[i] = i;
    j = i - 1;
    printf("%d", ㉠);
}

```

- |   | ㉠            | 출력  |
|---|--------------|-----|
| ① | $A[j]$       | 100 |
| ② | $\&A[0] + j$ | 99  |
| ③ | $*(A + j)$   | 99  |
| ④ | $*A + j$     | 100 |

문 16. 다항식을 표현하기 위하여 다음의 구조체를 정의하였다. 다항식

$A(x) = \sum_{i=0}^n a_i x^i$ 에 대해 poly.degree =  $n$ , poly.coef[i] =  $a_{n-i}$ 로

표현되며, 구조체 멤버 중 degree는 다항식의 최고차수를 저장하고 배열 coef[MAX\_DEGREE]는 다항식의 최고차수 항부터 최저차수 항까지의 계수를 차례로 저장한다. 이때, 다항식  $A(x)$ 를 저장하기 위한 구조체 변수 poly의 선언 과정에서 다항식 ' $100 \times x^5 + 60 \times x$ '를 초기화하기 위한 ㉠의 내용으로 옳은 것은?

```

#define MAX_DEGREE 6
struct polynomial {
    int degree;
    int coef[MAX_DEGREE];
} poly = ㉠ ;

```

- ① {5, {100, 1, 60}}  
 ② {5, {100, 60, 0, 0, 0, 1}}  
 ③ {5, {100, 0, 0, 0, 60, 0}}  
 ④ {5, {100, 0, 60}}

문 17. 다음 중위 표현식(inorder expression)을 후위 표현식(postorder expression)으로 옳게 변환한 것은?

중위 표현식:  $Y = (1 + ((3 + 4 / 2) * 5)) - 6$

- ① Y 1 3 4 2 5 6 \* / + + =  
 ② Y = 1 3 4 2 / + 5 \* + 6 -  
 ③ = Y 1 3 4 / 2 \* 5 + + 6 -  
 ④ Y 1 3 4 2 / + 5 \* + 6 - =

문 18. 다음은 연결리스트 스택의 삭제 알고리즘이다. ㉠ ~ ㉣에 들어갈 문장으로 옳게 짝지어진 것은? (단, 스택이 비었을 때 top은 NULL 이고, top은 스택의 최상위 노드를 가리키는 포인터이다. 또한 스택이 빈 상태에서 맨 처음 삽입되는 노드의 next는 NULL이다)

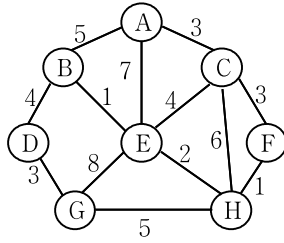
```

typedef struct node{
    int key;
    struct node *next;
} NODE; // 스택의 노드 구조
NODE *top = NULL; // 스택의 최상위 노드를 가리키는 포인터
int pop(void) {
    NODE *temp;
    int ret;
    if ( ㉠ ) {
        printf("\n Stack underflow.");
        return -1;
    }
    ret = top->key;
    ㉡ ;
    ㉢ ;
    free(temp);
    return ret;
}

```

- |   | ㉠                              | ㉡                             | ㉢                                              |
|---|--------------------------------|-------------------------------|------------------------------------------------|
| ① | $top \rightarrow next == NULL$ | $temp = top \rightarrow next$ | $top = top \rightarrow next$                   |
| ② | $top \rightarrow next == NULL$ | $temp = top$                  | $top \rightarrow next = temp \rightarrow next$ |
| ③ | $top == NULL$                  | $temp = top$                  | $top = top \rightarrow next$                   |
| ④ | $top == NULL$                  | $temp = top \rightarrow next$ | $top \rightarrow next = temp \rightarrow next$ |

문 19. 다음 그래프의 정점 A에서 시작하여 Prim 알고리즘을 적용하여 최소신장트리를 구축하고자 할 때, 설명으로 옳지 않은 것은?



- ① 우선순위 큐를 활용하여 구현한다.
- ② 탐욕(greedy) 기법을 이용한 알고리즘이다.
- ③ 마지막으로 선택되는 정점은 G이다.
- ④ 구축된 최소신장트리의 최소비용은 16이다.

문 20. 다음은 이중연결리스트의 맨 앞에 새로운 키 값을 갖는 노드를 삽입하는 알고리즘이다. ㉠ ~ ㉣에 들어갈 문장으로 옳게 짝지어진 것은? (단, head는 이중연결리스트의 맨 앞 노드를 가리키는 포인터이며, 리스트는 초기에 비어있다고 가정한다)

```
struct node { // 이중연결리스트의 노드 구조
    int key;
    struct node *prev; // 이전 노드를 가리키는 포인터
    struct node *next; // 다음 노드를 가리키는 포인터
};
struct node *head = NULL; // 맨 처음 노드를 가리키는 포인터
void insertDoubleList(int ikey)
{
    struct node *temp;
    temp = (struct node *) malloc(sizeof(struct node));
    if (temp == NULL) {
        printf("Memory allocation is failed. \n");
        exit(1);
    }
    temp->key = ikey;
    ㉠ ;
    if (head == NULL) // 리스트에 노드가 없는 경우
        ㉡ ;
    else { // 리스트에 노드가 있는 경우
        ㉢ ;
        head->prev = temp;
    }
    head = temp;
}
```

㉠

㉡

㉢

- ① temp->prev = NULL   temp->prev = head   temp->prev = head
- ② temp->prev = NULL   temp->next = NULL   temp->next = head
- ③ head->next = temp   temp->next = NULL   temp->prev = head
- ④ head->next = temp   temp->prev = head   temp->next = head