

자료구조론

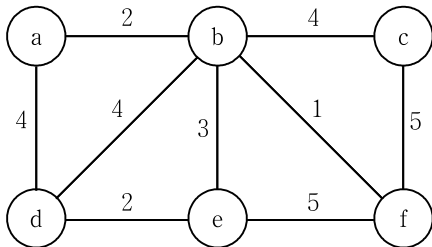
문 1. 최악 시간 복잡도(worst-case time complexity)가 $O(n \log n)$ 인 정렬 방식만을 모은 것은? (단, n 은 데이터의 개수이다)

- ① 합병(merge)정렬, 힙(heap)정렬
- ② 삽입(insertion)정렬, 버블(bubble)정렬
- ③ 선택(selection)정렬, 퀵(quick)정렬
- ④ 퀵(quick)정렬, 힙(heap)정렬

문 2. 그래프는 정점(vertex)의 집합 V 와 간선(edge)의 집합 E 로 이루어진다. 정점이 a, b, c 세 개 존재하고 간선이 a 와 b 사이에 하나, b 와 c 사이에 하나 존재한다고 할 때, 정점의 집합 V 는 $V = \{a, b, c\}$ 와 같은 형태로 표현하고, 간선의 집합 E 는 $E = \{(a, b), (b, c)\}$ 와 같이 나타낸다고 하자. 정점의 집합과 간선의 집합을 이용한 그래프 표현으로 옳은 것은?

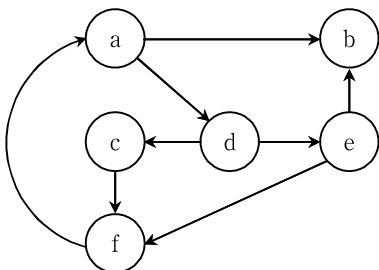
- ① $V = \{a, b, c, d, e, f\}$, $E = \{(a, b), (b, c), (c, a)\}$
- ② $V = \{a, a, c\}$, $E = \{(a, c), (a, a), (b, b)\}$
- ③ $V = \{a, c\}$, $E = \{(a, b), (a, c), (b, c)\}$
- ④ $V = \{a, b, c, d\}$, $E = \{(b, a), (d, b), (e, a)\}$

문 3. 다음 가중 그래프(weighted graph)에서 최소 비용 신장 트리(minimum spanning tree)를 얻기 위해 노드 a 에서 시작하여 Prim 알고리즘을 적용할 때 신장 트리의 최소 비용과 세 번째로 선택된 간선을 순서대로 나열한 것은?



- ① 12, (b, e)
- ② 13, (b, e)
- ③ 12, (d, e)
- ④ 13, (d, e)

문 4. 다음 방향 그래프는 강력 연결 요소(strongly connected component)들이 존재한다. 각 강력 연결 요소에 있는 정점들의 개수를 구한 후, 큰 수부터 나열한 것은?



- ① 4, 1, 1
- ② 4, 2
- ③ 5, 1
- ④ 6

문 5. 다음은 단순 연결 리스트를 역순으로 만드는 알고리즘의 주요 부분이다. 알고리즘의 ㉠ 부분에 들어갈 내용은? (단, start는 리스트의 시작 노드를 나타내는 포인터이다)

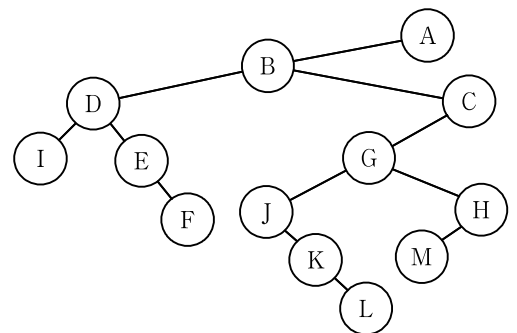
```
p = start;
q = NULL;
while (p != NULL)
{
    ㉠
    p = p->link;
    q->link = r;
}
start = q;
```

- ① $q = q \rightarrow \text{link}; r = q;$
- ② $r = q; q = p;$
- ③ $r = q; q = p \rightarrow \text{link};$
- ④ $q = q \rightarrow \text{link};$

문 6. 이진 트리로 구성하는 것이 불가능한 것은? (단, 루트(root) 노드의 단계(level)는 1이라고 가정한다)

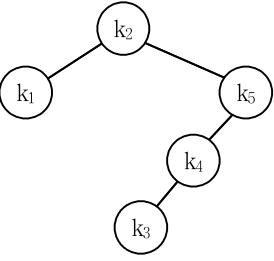
- ① 노드 개수가 23개이고 높이가 5인 이진 트리
- ② 높이가 5이고 노드 개수가 10개이며 단말 노드 개수가 6개인 이진 트리
- ③ 노드 개수가 20개이고 간선이 19개인 이진 트리
- ④ 높이가 6이고 노드 개수가 32개인 완전 이진 트리

문 7. 다음은 어떤 일반 트리를 이진 트리로 변환한 후의 모습이다. 이에 대한 설명으로 옳은 것은? (단, 일반 트리를 이진 트리로 변환할 때, 이진 트리의 왼쪽 노드는 일반 트리의 자식(child) 중에 하나를 가리키기 위해 사용되며, 이진 트리의 오른쪽 노드는 일반 트리의 형제(sibling)들을 연결하기 위해 사용된다)



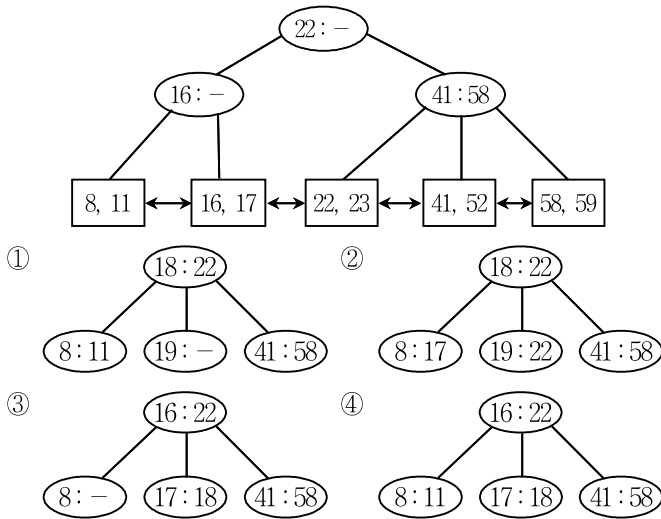
- ① 일반 트리에서 B의 자식 노드 개수는 2개이다.
- ② 일반 트리에서 C의 부모 노드는 B이다.
- ③ 일반 트리에서 노드 D에서 J까지 경로 길이는 4이다.
- ④ 일반 트리에서 단말 노드 개수는 7개이다.

문 8. 탐색할 키 k_i 와 탐색할 확률 p_i 가 (가)와 같이 주어질 때 이진 탐색 트리(binary search tree)를 (나)와 같이 구현하였다. 이진 탐색 트리의 총비용을 $\sum_{i=1}^n p_i \cdot \text{level}(k_i)$ 이라고 할 때, 이진 탐색 트리 (나)의 총비용은? (단, 실패한 비용은 없다고 가정하며, 루트(root) 노드의 단계(level)는 1이다)

(가)						(나)					
k_i	1	2	3	4	5						
p_i	0.2	0.45	0.05	0.1	0.2						

- ① 1.65
② 1.75
③ 1.85
④ 1.95

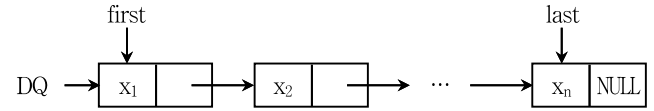
문 9. 차수가 3인 다음 B^+ -트리에 키 값 1, 18, 19를 순차적으로 삽입할 때, 최종적인 B^+ -트리의 인덱스 노드(비단말 노드)들의 모습은? (단, 데이터 노드(단말 노드)는 최대 2개의 원소를 포함할 수 있으며, 데이터 노드의 분할 시 중간 위치의 값은 오른쪽 노드에 포함되도록 한다)



문 10. 레드-블랙(red-black) 트리는 모든 노드의 색상이 레드 또는 블랙인 이진 탐색 트리이다. 키 값 7, 3, 6, 2, 1, 8, 4를 차례대로 삽입하여 레드-블랙 트리를 만들었을 때, 레드 링크로 연결된 노드 쌍이 아닌 것은? (단, 레드 링크는 레드 자식 노드를 가리키는 링크이다)

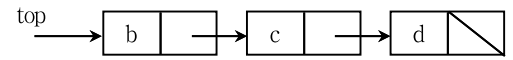
- ① (1, 2)
② (2, 6)
③ (3, 4)
④ (7, 8)

문 11. 데크(deque: double-ended queue)는 삽입과 삭제가 양끝에서 임의로 수행되는 자료구조이다. 다음 그림과 같이 단순 연결 리스트(singly linked list)로 데크를 구현한다고 할 때 $O(1)$ 시간 내에 수행할 수 없는 연산은? (단, first와 last는 각각 데크의 첫 번째 원소와 마지막 원소를 가리키며, 연산이 수행된 후에도 데크의 원형이 유지되어야 한다)

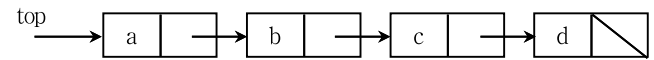


- ① insertFirst 연산: 데크의 첫 번째 원소로 삽입
② insertLast 연산: 데크의 마지막 원소로 삽입
③ deleteFirst 연산: 데크의 첫 번째 원소를 삭제
④ deleteLast 연산: 데크의 마지막 원소를 삭제

문 12. 다음은 연결 리스트를 이용하여 스택을 표현한 것이다. 이에 대한 설명으로 옳지 않은 것은? (단, push는 스택에 자료를 삽입하는 연산이고, pop은 스택에서 자료를 삭제하는 연산이다)



(ㄱ) push 연산 수행 전 스택



(ㄴ) push 연산 수행 후 스택

- ① 스택에 가장 최근에 입력된 자료는 top이 지시한다.
② 스택에 입력된 자료 중 d가 가장 오래된 자료이다.
③ (ㄴ)에서 자료 c를 가져오려면 pop 연산이 2회 필요하다.
④ (ㄱ)에서 자료의 입력된 순서는 d, c, b이다.

문 13. 다음은 x^n 을 구하는 알고리즘이다. 이 알고리즘의 시간 복잡도는? (단, $n > 0$ 이며, even(m)은 m이 짝수일 때 참을 반환하고 그렇지 않을 때 거짓을 반환하는 함수이다)

```
int PowersRec(int x, int n)
{
    int Pow;
    if (n == 1)
        Pow = x;
    else {
        if (even(n))
            Pow = PowersRec(x*x, n/2);
        else
            Pow = x * PowersRec(x*x, (n-1)/2);
    }
    return (Pow);
}
```

- ① $\Theta(1)$
② $\Theta(\log n)$
③ $\Theta(n)$
④ $\Theta(n \log n)$

- 문 14. 다음은 배열 A에 저장된 n개의 정수를 오름차순으로 정렬하는 삽입 정렬(insertion sort) 알고리즘이다. ㉠과 ㉡에 순서대로 들어갈 내용으로 옳은 것은?

```
void sort(int A[ ], int n)
{
    int i, j, key;
    for (i = 1; i < n; i++) {
        key = A[i];
        for (j = i - 1; ㉠ ; j--)
            ㉡
        A[j+1] = key;
    }
}
```

㉠

㉡

- ① $j \geq 0 \ \&\& \ key > A[j]$ $A[j+1] = A[j];$
 ② $j > 0 \ \&\& \ key \geq A[j]$ $A[j-1] = A[j];$
 ③ $j > 0 \ \&\& \ key < A[j]$ $A[j] = A[j+1];$
 ④ $j \geq 0 \ \&\& \ key < A[j]$ $A[j+1] = A[j];$

- 문 15. 순환(recursive) 함수 f()에 대한 첫 번째 호출이 f(7)일 때, f(7)을 포함하여 함수 f()가 호출되는 총 횟수는?

```
int f(int n)
{
    if (n == 0 || n == 1)
        return n;
    return f(n-1) + f(n-2);
}
```

- ① 25
 ② 35
 ③ 41
 ④ 51

- 문 16. 다음 수식에 대한 후위표기식(postfix notation)을 구한 후, 스택을 이용하여 수식을 계산하고자 한다. 수식 계산 과정에서 스택에 일곱 번째로 push되는 값은?

$$(A-B)*C-(D+E)/F$$

- ① D
 ② E
 ③ D+E
 ④ (A-B)*C

- 문 17. 행 우선(row major) 배열 A[3:6][2:7][8:12]에서 A[4][5][10]은 배열 A의 몇 번째 원소인가? (단, 첫 번째 원소는 A[3][2][8]이고, 마지막 원소는 A[6][7][12]이다)

- ① 45
 ② 46
 ③ 47
 ④ 48

- 문 18. X, Y, Z는 $m \times n$ 희소행렬에서 원소의 값이 0이 아닌 l개의 각 원소를 표현하기 위해 <행, 열, 값> 3원소 쌍을 사용하는 2차원 배열이며, 변환규칙은 (가)와 같다. 행렬 P와 Q가 (나)와 같이 X, Y로 각각 표현되었을 때, P와 Q를 곱한 행렬 R을 Z로 표현할 경우, 이에 대한 설명으로 옳지 않은 것은? (단, 행렬의 행 번호와 열 번호는 0부터 시작한다)

(가)	○ 행 우선 배열이며, 크기는 $(l+1) \times 3$ 이다.
	○ 첫 번째 행의 3개 열에는 희소행렬에 대한 행의 크기(m), 열의 크기(n), 0이 아닌 원소의 수(l)를 순서대로 저장한다.
(나)	○ 나머지 l개의 각 행에는 0이 아닌 l개의 원소들에 대한 정보가 저장되며, 해당 행의 각 열에는 원소들의 행 번호, 열 번호, 저장된 값이 각각 저장된다.
	○ 원소들에 대한 정보를 저장할 때는 원소들의 행 번호와 열 번호가 차례대로 오름차순이 되도록 저장한다.

X:	열	0	1	2
	행			
	0	3	4	3
	1	0	3	4
	2	1	1	7
Y:	열	0	1	2
	행			
	0	4	2	3
	1	0	1	1
	2	3	0	3
	3	3	1	2

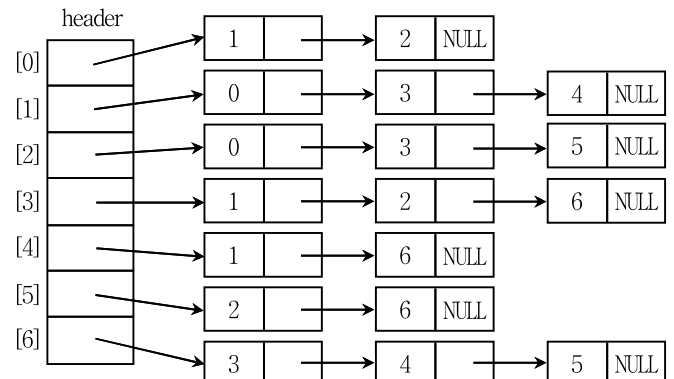
- ① $Z[0][1] = 2$ 이다. ② $Z[0][2] = 4$ 이다.
 ③ $Z[1][0] = 0$ 이다. ④ $Z[3][2] = 2$ 이다.

- 문 19. 다음 다섯 개의 숫자를 갖는 최소 힙(min-heap)을 만들려고 한다. 구성할 수 있는 서로 다른 최소 힙의 경우의 수는?

$$1, 2, 3, 4, 5$$

- ① 1개 ② 6개
 ③ 8개 ④ 10개

- 문 20. 다음 인접 리스트(adjacency list)로 표현된 그래프에서 깊이 우선 탐색(DFS: Depth First Search)으로 순회하고자 한다. 정점(vertex) 0부터 시작한 순회 순서는?



- ① 0, 1, 2, 3, 4, 5, 6
 ② 0, 1, 3, 2, 4, 6, 5
 ③ 0, 1, 3, 2, 5, 6, 4
 ④ 0, 1, 3, 2, 5, 4, 6