☑ 만든이 : 지안에듀 조현준

- 성균관대학교 정보공학 전공
- CISA, CISSP, 정보보안기사

☑ 저서

- TopSpot 정보보호론 이론편/문제편
- 알기쉬운 정보보안기사 필기편/실기편(알기사)
- TopSpot 자료구조론 이론편/쪽집게 기출문제

☑ 무료 동영상 안내

- 지안에듀 홈페이지 이용(www.zianedu.com)
  - [빅데이터 3.2] 조현준 정보보호론 연도별 기출문제풀이강의(전산 전직렬)
- 유튜브 자투리10분 기출20문항정리
  - 유튜브에서 조현준 정보보호론 검색

2019년 국가직 7급 기출문제

1. ○△×

m원 탐색 트리(m-way search tree)에 대한 설명으로 옳지 않은 것은?

① 공백(empty)이 아닌 m원 탐색 트리의 루트(root) 노드는 최대 m개의 서브-트리(sub-tree)를 가질 수 있다.
② 공백이 아닌 m원 탐색 트리의 루트 노드에 저장된 키(key)들은 정렬되어 있어야 한다.
③ 공백이 아닌 m원 탐색 트리의 각 노드가 가지는 서브-트리 역시 m원 탐색 트리이다.
④ 차수(degree)가 m이고 높이(height)가 h인 다원 탐색 트리가 가질 수 있는 최대 노드 수는 $(m^h - 1)$이다. (단, 루트만 있는 트리의 높이는 1로 한다.)

정답 ④

높이가 h일 때 m원 검색트리가 가질 수 있는 키 값의 최대 수는 $m^h - 1$ 이고, 노드의 최대수는 $\frac{m^h-1}{m-1}$ 이다.

2. ○△×

다음은 C 언어로 구현한 원형 큐(circular queue) 삽입 알고리즘이다. ㉠, ㉡에 들어갈 내용으로 옳은 것은?

```
#define MAX_QUEUE_SIZE 10
int queue[MAX_QUEUE_SIZE];
int front = rear = -1;

void addq(int front, int *rear, int item) {
    ______㉠______
    if (______㉡______) {
        printf("Queue is full!!₩n");
        return -1;
    }
    queue[*rear] = item;
}
```

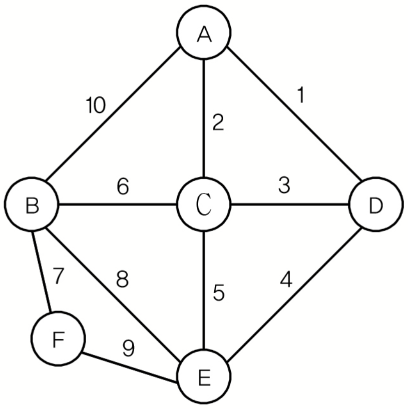
① ㉠ *rear = (*rear+1) % MAX_QUEUE_SIZE;
  ㉡ front == *rear
② ㉠ *rear = (front+1) % MAX_QUEUE_SIZE;
  ㉡ front == *rear + 1
③ ㉠ front = (front+1) % MAX_QUEUE_SIZE;
  ㉡ front == *rear
④ ㉠ front = (*rear+1) % MAX_QUEUE_SIZE;
  ㉡ front == *rear + 1

정답 ①

원형 큐의 삽입은 rear의 위치를 증가시킨 다음 front = rear의 조건으로, Full인지를 검사한다. front ≠ rear인 경우, 값(item)을 큐에 삽입한다.

3. ○△×

다음 그래프를 Kruskal 알고리즘을 이용하여 최소 비용 신장 트리로 만들 때 선택되지 않는 간선은? (단, 알고리즘은 노드 A에서 시작한다)



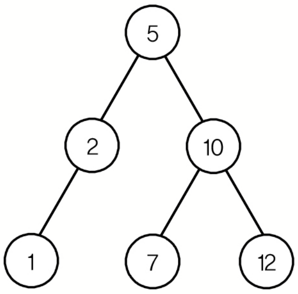
① (A, C) ② (B, C)
③ (B, F) ④ (C, D)

정답 ④

Kruskal 알고리즘은 한 번에 하나씩 T에 간선을 추가해 가면서 최소 비용 신장트리 T를 구축한다. 이 알고리즘은 T에 포함될 간선을 비용의 크기순으로 선택한다. 간선이 선택되는 순서는 (A, D) → (A, C) → (D, E) → (B, C) → (B, F)이다. (C, D)는 세 번째 선택 시 고려사항이 되는데, 사이클을 형성하므로 선택되지 않는다.

4. ○△×

다음은 이진 트리(binary tree)이다. 매개변수 ptr로 함수 func를 호출하였을 때, 결과 값은? (단, ptr은 루트 노드에 대한 포인터이다.)




```
struct node {
    int data;
    struct node *left;
    struct node *right;
};
int func(struct node *ptr) {
    if (ptr == NULL)
        return 0;
    else if (ptr->left == NULL
                && ptr->right == NULL)
        return 1;
    else
        return (ptr->data + func(ptr->left)
                    + func(ptr->right));
}
```

① 10 ② 17
③ 20 ④ 37

정답 ③

알고리즘을 살펴보면 단말노드인 경우 1을 더하고, 단말노드가 아닌 경우 원소값을 더하고, 좌우 자식 노드를 방문한다. 함수 호출 시 결과값은 5+2+1+10+1+1 = 20이 된다.

5. ○△×

$n \times n$ 하삼각 행렬은 총 $n(n+1)/2$개의 원소를 갖는다. 원소 a_{ij} (1 ≦ i ≦ j ≦ n)가 최소 공간을 사용하도록 1차원 배열 $b[k]$에 행우선순서(row-major order)로 저장했을 때, 하삼각 행렬의 원소 a_{ij}가 저장될 배열 $b[k]$의 색인 k를 계산하는 식으로 옳은 것은? (단, i는 행 색인 값, j는 열 색인 값으로 하고, 1차원 배열 b에는 하삼각 행렬의 0 값은 저장되지 않으며, b의 색인 k의 값은 0부터 시작한다.)

정방 행렬(square matrix) 중에서 대각선보다 위의 모든 원소가 0인 경우를 특별히 하삼각 행렬(lower triangular matrix)이라 하며, 다음은 4×4 하삼각 행렬의 예이다. (단, a_{ij}는 0이 아닌 어떤 실수 값을 의미하고, 0은 반드시 0이어야 함을 의미한다)

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

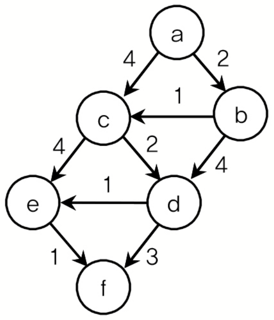
① $(i-1) \times n + (j-1)$
② $j \times (j-1)/2 + (i-1)$
③ $i \times (i-1)/2 + (j-1)$
④ $(j-1) \times n + (i-1)$

정답 ③

행우선순서로 저장된 하삼각 행렬에서 배열 $b[k]$의 색인 k를 계산하는 식을 구하기 위해서는 (1) 1 + 2 + 3 + … (i−1) $= i \times (i-1)/2$와 (2) 열 번호를 구한 뒤 더하면 된다. 임의의 원소 하나를 대입하면 검증가능하다.

6. ○△×

다음과 같은 방향 그래프에 대하여 Dijkstra의 알고리즘을 적용하여 최단경로를 구하고자 한다. 노드 a에서 시작하여 노드 f로 가는 최단 경로를 찾아 가는 과정에서 노드 a에서부터 다른 노드까지 최단 경로를 차례로 알게 된다. 이 과정에서 알게 되는 최단 경로의 노드 순서로 옳은 것은?



① a − b − c − d − e − f
② a − b − d − f
③ a − c − d − f
④ a − c − d − e − f

정답 ①

Dijkstra 알고리즘은 시작 정점에서 최소 distance 값을 가지는 정점을 선택해 나가는 방법이다. 노드 a를 시작점으로 하여 노드 b(2), 노드 c(4) 중에서 작은 값을 갖는 b를 선택한다. 이후에 b를 경유지로 추가하고 작업을 반복하면 c가 선택된다. 이때 경로비용은 3(a → b → c)이다. 동일 과정을 반복 적용하면 노드는 a − b − c − d − e − f 순서로 선택된다.

7. ○△×

다음과 같이 배열에 저장된 데이터에 대하여 내림차순으로 히프 정렬(heap sort)을 수행할 때, 첫 번째 데이터를 출력하고 히프를 재구성 한 후에 배열의 여섯 번째에 있는 값은? (단, 배열의 첫 번째 색인 값은 1이다)

[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
21	37	22	17	11	30	5	10	9

① 5　　② 9
③ 10　　④ 11

정답 ②

내림차순 히프 정렬을 수행하기 위해서는 최대 히프를 먼저 구성해야 한다. 최대 히프 구성 후 배열은 다음과 같다.

[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
37	21	30	17	11	22	5	10	9

첫 번째 데이터를 출력하고 히프를 재구성하기 위해서는 마지막 원소 9를 첫 번째 원소로 이동 후 재구성하면 된다. 재구성 후 배열은 다음과 같다.

[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
30	21	22	17	11	9	5	10	–

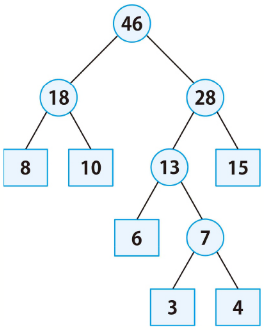
8. ○△×

6개 외부 노드의 가중치가 각각 q1 = 3, q2 = 4, q3 = 6, q4 = 8, q5 = 10, q6 = 15인 허프만 트리의 설명으로 옳은 것은? (단, 허프만 트리는 $\sum_{1 \le i \le n} q_i d_i$가 최소가 되는 트리로서, q_i는 외부 노드의 가중치이고 d_i는 루트 노드에서부터 외부 노드까지의 거리이며, n은 외부 노드의 수이다)

① 외부 노드에서 루트까지의 거리와 외부 노드의 가중치 곱의 총합($\sum_{1 \le i \le n} q_i d_i$)은 112이다.
② 루트까지의 거리가 가장 긴 외부 노드에서 루트까지의 거리는 5이다.
③ 루트까지의 거리가 가장 긴 외부 노드는 q_1, q_2, q_3이다.
④ 루트까지의 거리가 가장 짧은 외부 노드는 q_6이고 그 다음으로 짧은 외부 노드는 q_4와 q_5이다.

정답 ①

허프만 트리를 구성하면 다음과 같다.



② 루트까지의 거리가 가장 긴 외부 노드에서 루트까지의 거리는 4이다.
③ 루트까지의 거리가 가장 긴 외부 노드는 q_1, q_2이다.
④ 루트까지의 거리가 가장 짧은 외부 노드는 q_4, q_5, q_6이다.

9. ○△×

다음과 같이 배열에 저장된 입력 자료들을 퀵 정렬(quick sort)로 오름차순 정렬하고자 한다. 정렬과정에서 단계별 정렬 순서로 나타날 수 없는 것은? (단, 피봇(pivot)은 마지막 레코드로 선택한다.)

15	3	1	35	20	4	18	26

① 15, 3, 1, 18, 20, 4, 26, 35
② 15, 3, 1, 4, 20, 18, 26, 35
③ 1, 3, 4, 18, 20, 15, 26, 35
④ 1, 3, 4, 15, 20, 18, 26, 35

정답 ②

퀵 정렬을 단계적으로 적용하면 다음과 같다.

초기	15	3	1	35	20	4	18	**26**
1단계	15	3	1	18	20	4	**26**	35
2단계	15	3	1	18	20	**4**	26	35
3단계	1	**3**	4	18	20	15	26	35
4단계	1	3	4	18	20	**15**	26	35
5단계	1	3	4	15	20	**18**	26	35
5단계	1	3	4	15	18	20	26	35

위의 과정에서 굵은 글씨는 피봇이고, 단계의 일부 중간 과정은 생략함.

10. ○△×

다음과 같이 중위 표기법(infix notation)으로 되어 있는 수식을 후위 표기법(postfix notation)으로 변환하기 위해 스택을 이용한다. 변환 과정에서 스택에 토큰이 가장 많이 쌓여 있을 때의 스택 내 연산자 토큰 수는?

a*(b−c*d)/e

① 3개 ② 4개
③ 5개 ④ 6개

정답 ②

토큰	스택	출력
a		a
*	*	a
(	* (	a
b	* (	ab
−	* (−	ab
c	* (−	abc
*	* (− *	abc
d	* (− *	abcd
)	*	abcd*−
/	/	abcd*−*
e	/	abcd*−*e
eos		abcd*−*e/

스택에 토큰이 가장 많이 쌓여 있을 때는 「* (− *」로 4개이다.

11. ○△×

그래프에 대한 설명으로 옳은 것만을 모두 고르면?

ㄱ. 무방향 그래프(undirected graph)를 인접행렬(adjacency matrix)로 표현하면 주대각선을 중심으로 상위 행렬과 하위 행렬이 대칭이 된다.
ㄴ. 사이클이 없고 n − 1개의 간선(edge)을 갖는 연결 그래프(connected graph)를 트리(tree)라고 한다.
ㄷ. n개의 정점(vertex)을 가진 방향 그래프의 최대 간선 수는 $\frac{n(n+1)}{2}$ 개이다.
ㄹ. 경로를 구성하는 간선의 수를 길이라고 할 때, 방향 그래프에서 사이클(cycle)의 길이는 항상 3 이상이다.

① ㄱ, ㄴ ② ㄱ, ㄷ
③ ㄴ, ㄷ ④ ㄷ, ㄹ

정답 ①

ㄷ. n개의 정점으로 구성된 방향 그래프에서 최대 간선 수는 $n(n-1)$이다.
ㄹ. 경로를 구성하는 간선의 수를 길이라고 할 때, 방향 그래프에서 사이클(cycle)의 길이는 항상 2이상이다.

12. ○△×

일반 리스트(general list) $L = (e_1, e_2, \dots, e_n)$에 대해, 함수 head(L)은 첫 번째 원소인 e_1을 반환하고, 함수 tail(L)은 리스트 $(e_2, e_3, \dots, e_n)$를 반환한다고 할 때, 다음과 같은 리스트 A에 대한 함수의 수행 결과 중 문자 'c'를 반환하는 것은?

A = ((a, b), c, d)

① head(head(A)) ② head(tail(A))
③ head(tail(head(A))) ④ head(tail(tail(A)))

정답 ②

① head(head(A)) = a
② head(tail(A)) = c
③ head(tail(head(A))) = b
④ head(tail(tail(A))) = d

13. ○△×

다음은 단순 연결 리스트(singly linked list)에서 특정 값(value)을 가진 노드의 개수를 반환하는 C 함수이다. ㉠, ㉡에 들어갈 내용으로 옳은 것은?

```
struct node {
    int data;
    struct node *link;
};
int countNode(struct node *ptr, int value)
{
    int count = 0;
    while (______㉠______) {
        if ( ptr->data == value ) count++;
        ______㉡______
    }
    return count;
}
```

	㉠	㉡
①	ptr == NULL	ptr->link = ptr;
②	ptr == NULL	ptr = ptr->link;
③	ptr != NULL	ptr->link = ptr;
④	ptr != NULL	ptr = ptr->link;

정답 ④

단순 연결 리스트에서 특정 값을 가진 노드의 개수를 구하기 위해서는 마지막 노드(ptr != NULL)가 아닐때까지 전진(ptr = ptr->link)하면서 구하면 된다.

14. ○△×

다음은 Fibonacci 수열을 계산하는 C 함수이다. 함수 fibonacci(4)를 호출하였을 때 화면에 출력되는 숫자의 순서로 옳은 것은?

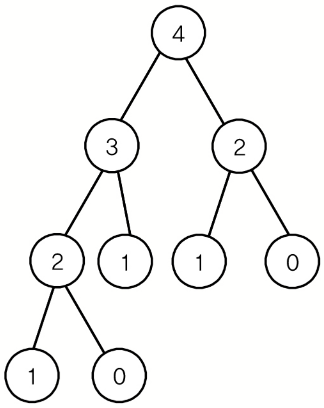
```
int fibonacci( int n )
{
    printf("%d\n", n);

    if ( n == 0 )
        return 0;
    else if ( n == 1 )
        return 1;
    else
        return ( fibonacci( n - 1 )
                    + fibonacci( n - 2 ) );
}
```

① 4, 3, 2, 2, 1, 1, 0, 1, 0
② 4, 3, 2, 1, 0, 2, 1, 1, 0
③ 4, 3, 2, 1, 0, 1, 2, 1, 0
④ 4, 3, 2, 1, 0, 2, 1, 0, 1

정답 ③

보기는 피보나치 함수로 피보나치 트리를 이용하면 쉽다. 좌측 서브트리를 먼저 탐색하고, 우측 서브트리를 탐색한다.



15. ○△×

다음은 이진 탐색을 수행하는 C 함수이다. 이 함수를 사용하여 0부터 20까지의 정수가 차례대로 저장되어 있는 배열 b에 대한 이진 탐색을 수행할 때, key 값과 a[middle] 값을 비교하는 횟수가 다른 것은? (단, 배열의 색인은 0부터 시작한다)

```
int binarySearch(int *a, int key, int low,
                                  int high)
{
    int middle;
    while (low <= high) {
        middle = (low + high) / 2;
        if (key == a[middle])
            return middle;
        else if (key > a[middle])
            low = middle + 1;
        else
            high = middle - 1;
    }
    return -1;
}
```

① binarySearch(b, 4, 1, 20)
② binarySearch(b, 7, 1, 20)
③ binarySearch(b, 9, 1, 20)
④ binarySearch(b, 14, 1, 20)

정답 ②

이진검색에서 비교횟수를 구하면 다음과 같다.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	3	4	5	2	4	3	4	5	1	4	3	4	5	2	4	5	3	4	5

16. ○△×

시간 복잡도 함수에 대한 점근 표기법으로 옳은 것만을 모두 고르면?

ㄱ. $n^{2n} + 6\cdot 2^n = O(n^{2n})$
ㄴ. $n^2/\log n = \Theta(n^2)$
ㄷ. $n^3 2^n + 6n^2 3^n = O(n^2 2^n)$
ㄹ. $6n^3 + \log n = O(n^3)$

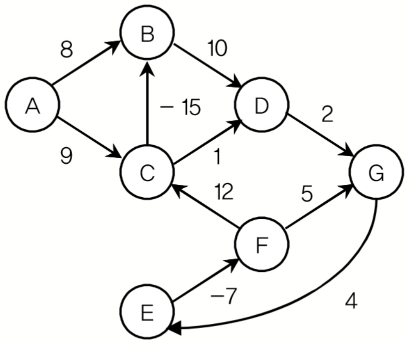
① ㄱ, ㄷ　　② ㄱ, ㄹ
③ ㄴ, ㄷ　　④ ㄴ, ㄹ

정답 ②

ㄴ. $n^2/\log n = \Theta(n^2/\log n)$
ㄷ. $n^3 2^n + 6n^2 3^n = O(n^2 3^n)$

17. ○△×

다음 그래프에는 음의 가중치가 존재한다. 이 그래프에서 노드 A에서 노드 F로의 최단 경로를 구하고자 할 때, 최소 비용은?



① 3　　② 5
③ 7　　④ 9

정답 ①

A에서 F까지 최단 경로를 구하면 다음과 같다.
A → C → B → D → G → E → F
이때 비용은 9 - 15 + 10 + 2 + 4 - 7 = 3이다.

18. ○△×

다음은 단순 연결 리스트(singly linked list) L에서 리스트의 원소를 역순으로 바꾸는 함수이다. 함수에서 ㉠ 부분에 들어갈 프로그램 코드(code)로 옳은 것은?

```
typedef struct node{
    struct node *link;
    char data;
} listNode;
typedef struct{
    listNode *head;
} linkedList;

void reverse(linkedList *L)
{
    listNode *p, *q, *r;
    p = L->head;
    q = r = NULL;
    while ( p != NULL)  {
        r = q;
        q = p;
        p = p->link;
        _____㉠_____
    }
    L->head = q;
}
```

① q = q->link; ② q->link = r;
③ p->link = q; ④ r->link = q;

정답 ②

보기 알고리즘은 p, q, r 이라는 세 개의 포인터 변수를 이용하여, q는 p를 잇고, r은 q를 잇는 리스트를 역순 구조로 바꾸는 연결리스트의 응용이다. 코드「q->link = r;」는 q의 링크 방향을 바꾸기 위해 필요하다.

19. ○△×

주어진 정수 배열 {26, 13, 77, 61, 35, 11, 8, 48, 15, 19}에 대한 합병 정렬(merge sort)의 순환 알고리즘(recursive algorithm)을 다음과 같이 구현하였다. 이 프로그램의 수행 과정에서 형성되는 배열의 순서로 옳지 않은 것은?

```
#include <stdio.h>
#define MAX 10
void merge( int list[], int temp[], int left,
            int middle, int right)
{
    int i=left, j=middle+1, k=left, t;
    for( ; i <= middle && j <= right ;  ) {
        if( list[i] <= list[j] )
        temp[k] = list[i++];
        else
        temp[k] = list[j++];
        k += 1;
    }
    if( i > middle ) {
        for( t = j ; t <= right ;)
        temp[k++] = list[t++];
    } else {
        for( t = i ; t <= middle ;)
        temp[k++] = list[t++];
    }
    for( t = left ; t <= right ; t++ )
        list[t] = temp[t];
}

void mergeSort( int list[], int temp[], int left,
                int right)
{
    int middle = 0;
    if( left < right ) {
        middle = (left+right)/2;
        mergeSort(list, temp, left, middle);
        mergeSort(list, temp, middle+1, right);

        merge(list, temp, left, middle, right);
    }
}

int main(){
    int source[10]={26, 13, 77, 61, 35, 11, 8,
                    48, 15, 19};
    int temp[10] = {0x0,};

    mergeSort(source, temp, 0, 9);
}
```

① 13, 26, 35, 61, 77, 11, 8, 48, 15, 19
② 13, 26, 77, 61, 35, 11, 8, 48, 15, 19
③ 13, 26, 35, 61, 77, 8, 11, 15, 19, 48
④ 13, 26, 35, 61, 77, 8, 11, 15, 48, 19

부록 최신기출문제

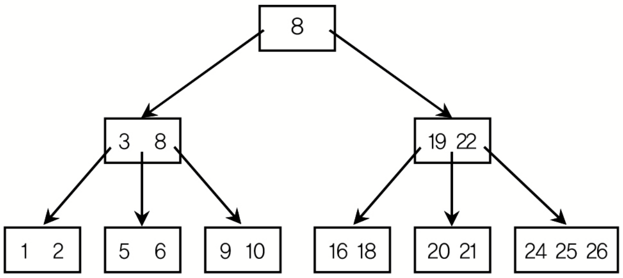
정답 ④

순환 합병정렬로 순서대로 하면 다음과 같다.

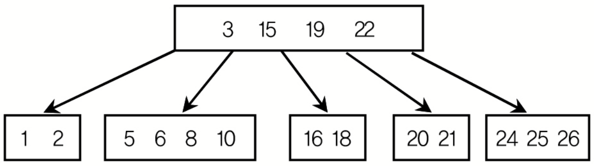
초기	26, 13, 77, 61, 35, 11, 8, 48, 15, 19
1단계	13, 26, 77, 61, 35, 11, 8, 48, 15, 19
2단계	13, 26, 77, 61, 35, 11, 8, 48, 15, 19
3단계	13, 26, 77, 35, 61, 11, 8, 48, 15, 19
4단계	13, 26, 35, 61, 77, 11, 8, 48, 15, 19
5단계	13, 26, 35, 61, 77, 8, 11, 48, 15, 19
6단계	13, 26, 35, 61, 77, 8, 11, 48, 15, 19
7단계	13, 26, 35, 61, 77, 8, 11, 48, 15, 19
8단계	13, 26, 35, 61, 77, 8, 11, 15, 19, 48
9단계	8, 11, 13, 15, 19, 26, 35, 48, 61, 77

20. ○△×

다음은 노드의 차수가 5인 B-트리의 현재 상태를 표현한 것이다. 현 상태에서 값 9를 삭제하였을 때, 결과 트리에 존재하는 노드 수를 종류별로 옳게 나타낸 것은? (단, n-노드란 (n - 1)개의 키를 저장한 노드를 의미한다)



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

정답 ④

9 삭제 후 B-트리의 결과는 다음과 같다.