

자료구조론	국가 전산 7급	2018년 8월 18일
--------------	-----------------	---------------------

☞ 합격선/최종합격인원(74.16점/31명) - 채용예정인원 29명 / 필기합격 42명 ☞

1. 다음 C 언어 함수에 의해 구현된 정렬 방식은? [2018년 국가 7급]

```
-----  
void whatsort(int a[ ], int size)  
{  
    int i, j, temp;  
    for (i = (size-1); i > 0; i--)  
    {  
        for (j = 1; j <= i; j++)  
        {  
            if (a[j-1] > a[j])  
            {  
                temp = a[j-1];  
                a[j-1] = a[j];  
                a[j] = temp;  
            }  
        }  
    }  
}
```

- ① 삽입정렬(insertion sort)
- ② 선택정렬(selection sort)
- ③ 힙정렬(heap sort)
- ④ 버블정렬(bubble sort)

☞ 버블정렬

```
// 프로그램 분석  
if (a[j-1] > a[j]) //인접한 배열요소를 비교하여 맞교환하는 원리(버블정렬)  
{  
    temp = a[j-1];  
    a[j-1] = a[j];  
    a[j] = temp;  
}
```

정답 : ④

2. 다음 중위표현식(infix expression)을 후위표현식(postfix expression)으로 변환한 후, 스택을 이용하여 후위표현식을 계산하고자 한다. 후위표현식의 계산 과정에서 스택에 여덟 번째로 삽입(push)되는 값은? [2018년 국가 7급]

$$(4 + 2) / 3 + (6 / 2 - 1)$$

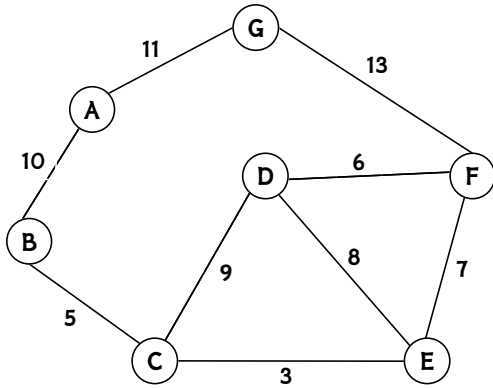
- ① 1 ② 2
 ③ 3 ④ 4

☞ 후위표현식의 계산 과정

- $(4 + 2) / 3 + (6 / 2 - 1) \rightarrow$ 중위표기법
 ↓ 연산우선순위로 괄호로 묶는다.
 $(((4 + 2) / 3) + ((6 / 2) - 1))$
 ↓ 연산자를 해당 괄호 뒤로 이동시킨다.
 $(((4 2) + 3) / ((6 2) / 1) -) +$
 ↓ 괄호를 없앤다.
 $4 2 + 3 / 6 2 / 1 - + \rightarrow$ 후위표기법
- 먼저, 스택에 여덟 번째로 삽입(push)되는 값 = $6 / 2 = 3$
 → 이유는, 후위표기법에서 앞에서 여덟 번째 위치한 것은 연산자 /이므로
 → 읽은 자료가 피연산자이면 그냥 스택에 넣는다.(push)
 → 연산자이면 스택에서 피연산자를 꺼내어(pop). 연산하고 연산 결과만을 넣는다.
- 후위표기식의 연산 원리는 수식을 왼쪽에서 오른쪽으로 읽어가면서 연산자를 만났을 때 연산한다. 연산 대상은 연산자를 만나기 직전에 읽은 피연산자들이다.
- 후위표기식의 연산 과정을 Stack 그림으로 나타내면 다음과 같다.

						2		1		
	2		3		6	6	3	3		
4	4	6	6	2	2	2	2	2	2	4
push(4)	push(2)	pop(2)	push(3)	pop(3)	push(6)	push(2)	pop(2)	push(1)	pop(1)	pop(2)
		pop(4)		pop(6)			pop(6)		pop(3)	pop(2)
		+ 연산		/ 연산			/ 연산		- 연산	+ 연산
		push(6)		push(2)			push(3)		push(2)	push(4)

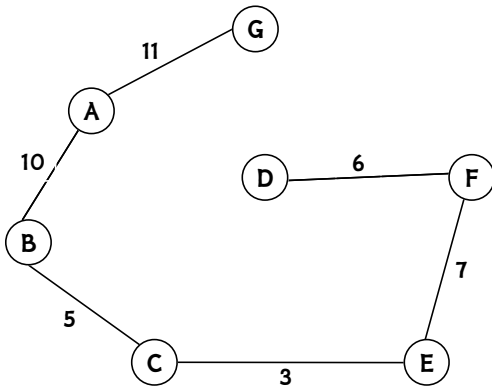
3. Prim 알고리즘을 사용하여 다음 가중 그래프(weighted graph)의 최소비용신장트리(minimum cost spanning tree)를 구성할 때, 최소비용과 마지막으로 선택되는 간선은? (단, 시작 정점은 A이다) [2018년 국가 7급]



- ① 42, (A, G)
- ② 43, (A, G)
- ③ 44, (F, G)
- ④ 45, (F, G)

☞ Prim 알고리즘 - 최소비용신장트리

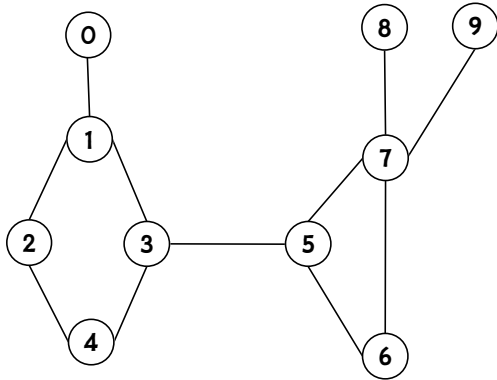
• 시작 정점이 A이므로



- 선택되는 간선 순서 : (A, B) → (B, C) → (C, E) → (E, F) → (D, F) → **(A, G)**
- 최소비용 = 10 + 5 + 3 + 7 + 6 + 11 = 42

정답 : ①

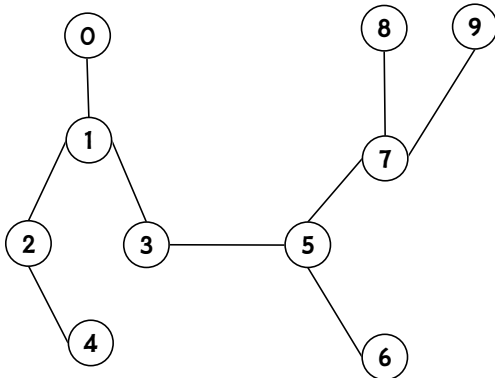
4. 시작 정점이 6일 때, 다음 그래프에 대한 깊이우선탐색(DFS: Depth First Search)의 방문 순서는? (단, 인접한 정점들은 오름차순으로 방문한다) [2018년 국가 7급]



- ① 6, 5, 3, 1, 0, 2, 4, 7, 8, 9
- ② 6, 5, 7, 3, 8, 9, 1, 4, 0, 2
- ③ 6, 5, 3, 4, 2, 1, 0, 7, 8, 9
- ④ 6, 5, 7, 3, 1, 4, 0, 2, 8, 9

♣ 깊이우선탐색(DFS) - 인접한 정점들은 오름차순으로 방문

• 시작 정점이 6



• DFS(6) = 6, 5, 3, 1, 0, 2, 4, 7, 8, 9

• 만약, 너비우선탐색(BFS)

BFS(6) = 6, 5, 7, 3, 8, 9, 1, 4, 0, 2

정답 : ①

5. C 언어 함수로 구현된 <정렬 알고리즘>과 크기가 5인 배열 list를 이용하여 sort(list, 5)를 수행 하였다. <설명>에서 옳은 것만을 모두 고르면? (단, sort(list, 5) 수행 전에 list 배열에 4, 9, 8, 6, 3이 순서대로 저장되어 있다) [2018년 국가 7급]

<정렬 알고리즘>

```
#define SWAP(x, y, t) ((t)=(x), (x)=(y), (y)=(t))
void sort(int list[ ], int n)
{
    int i, j, least, temp;
    for (i = 0; i < n - 1; i++)
    {
        least = i;
        for (j = i + 1; j < n; j++)
            if (list[j] < list[least]) least = j;
        SWAP(list[i], list[least], temp);
    }
}
```

<설 명>

- ㄱ. sort(list, 5)의 수행이 완료될 때까지 SWAP은 4회 수행된다.
- ㄴ. SWAP이 두 번째 수행된 후 배열 list에는 3, 4, 8, 6, 9가 순서대로 저장되어 있다.
- ㄷ. 정렬 알고리즘의 시간복잡도를 빅세타(Θ) 표기법으로 표현한 것은 $\Theta(n \log n)$ 이다.

- ① ㄱ
- ② ㄱ, ㄴ
- ③ ㄱ, ㄷ
- ④ ㄴ, ㄷ

☞ 정렬 알고리즘(오름차순 선택정렬)

- 4, 9, 8, 6, 3 → 초기 상태 배열
- 3, 9, 8, 6, 4 → SWAP이 첫 번째 수행된 후 배열
- 3, 4, 8, 6, 9 → SWAP이 두 번째 수행된 후 배열
- 3, 4, 6, 8, 9 → SWAP이 세 번째 수행된 후 배열
- 3, 4, 6, 8, 9 → SWAP이 네 번째 수행된 후 배열(정렬 완료)

• 정렬 알고리즘 시간복잡도 : $\Theta(n^2)$