

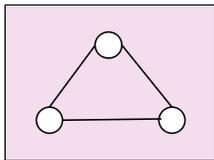
자료구조론	국가 전산 7급	2022년 10월 15일
--------------	-----------------	----------------------

♣ 필기합격인원/합격선(58명/77점) - 선발예정인원 48명 ♣

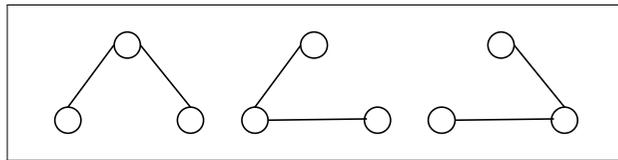
1. 세 개의 노드를 가진 무방향 완전그래프의 신장트리(spanning tree)는 최대 몇 가지인가?
[2022년 국가 7급]

- ① 1 ② 2 ③ 3 ④ 4

☞ 세 개의 노드를 가진 무방향 완전그래프의 신장트리



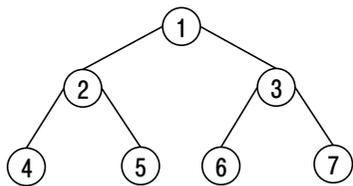
무방향 완전그래프



신장트리 : 3가지

정답 : ③

2. 다음 트리를 중위순회한 결과로 옳은 것은? [2022년 국가 7급]



- 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

☞ 트리 순회

- 전위순회(중간,좌,우) : 1, 2, 4, 5, 3, 6, 7
- 중위순회(좌,중간,우) : 4, 2, 5, 1, 6, 3, 7
- 후위순회(좌,우,중간) : 4, 5, 2, 6, 7, 3, 1

정답 : ③

3. 다음은 입력 개수 n 에 대한 알고리즘 A~D의 수행시간 복잡도를 나타낸 것이다. 알고리즘 A~D를 수행시간 효율이 좋은 것부터 순서대로 나열한 것은? [2022년 국가 7급]

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

♣ 알고리즘 복잡도

• $O(n \log n) < O(n^2) < O(2^n) < O(n!)$

$\log_2 n$	n	$n \log_2 n$	n^2	n^3	2^n	$n!$
0	1	1	1	1	2	1
1	2	2	4	8	4	2
2	4	8	16	64	16	24
3	8	24	64	512	256	40,320
4	16	64	256	4096	65,536	2,004,189,814

[함수 값 변화]

• n 이 충분히 클 때, 함수 값 크기는 다음 순서이다.

$$\log_2 n < n < n \log_2 n < n^2 < n^3 < 2^n < n!$$

◆ \sqrt{n} 의 점근 표기

• $\sqrt{2} = 1.41421\dots$ (무리수 : 순환하지 않은 무한소수)

• $\log_2 n$ 과 \sqrt{n}

n	1	2	4	8	16	32	64	128	256
$\log_2 n$	0	1	2	3	4	5	6	7	8
\sqrt{n}	1	$\sqrt{2}$	2	$2\sqrt{2}$	4	$4\sqrt{2}$	8	$8\sqrt{2}$	16

• $\log_2 n \leq \sqrt{n}$ 이다. 해서, $\sqrt{n} = \Omega(\log_2 n)$ 이다.

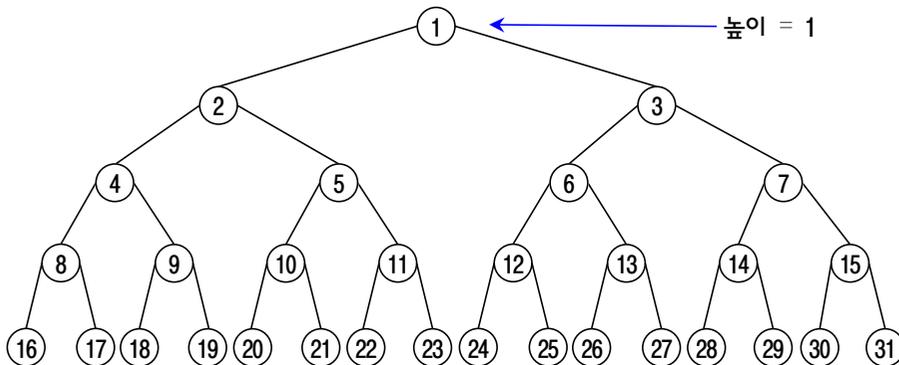
4. 이진트리에 대한 설명으로 옳은 것만을 모두 고르면? [2022년 국가 7급]

- ㄱ. 공백 이진트리의 높이를 0으로 본다면, 높이가 5인 이진트리의 최대 노드 수는 15이다.
 ㄴ. 9개의 노드를 가지고 있는 이진트리는 8개의 간선을 가진다.
 ㄷ. 루트노드의 인덱스를 1로 하는 1차원 배열로 완전이진트리를 표현할 때, 인덱스 13인 노드의 부모노드는 인덱스가 7이다.
 ㄹ. 7개 노드로 구성할 수 있는 이진트리의 최대 높이와 최소 높이의 차는 4이다.

- ① ㄱ, ㄴ ② ㄴ, ㄷ ③ ㄴ, ㄹ ④ ㄷ, ㄹ

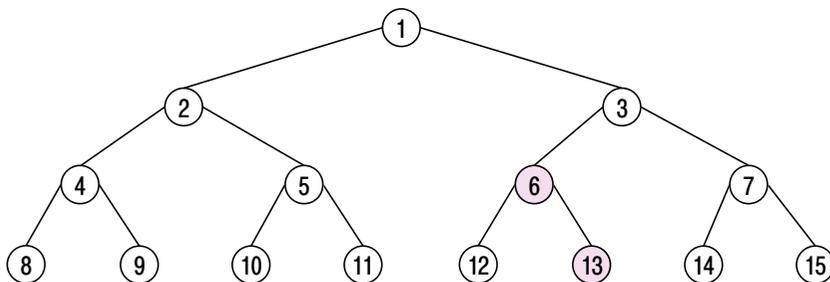
☞ 이진트리

- ㄱ. 공백 이진트리의 높이를 0으로 본다면, 높이가 5인 이진트리의 최대 노드 수는 15이다.(×)
 → 공백 이진트리의 높이를 0으로 본다면, 높이가 5인 이진트리의 최대 노드 수는 31이다.



• 최대 노드 수 = $2^k - 1 = 2^5 - 1 = 31$

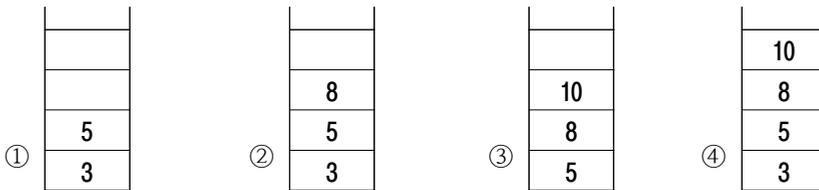
- ㄷ. 루트노드의 인덱스를 1로 하는 1차원 배열로 완전이진트리를 표현할 때, 인덱스 13인 노드의 부모노드는 인덱스가 7이다.(×) → 인덱스 13인 노드의 부모노드는 인덱스가 6이다.



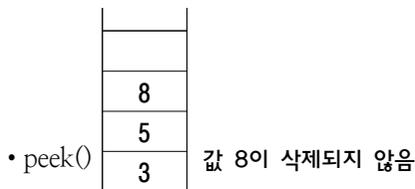
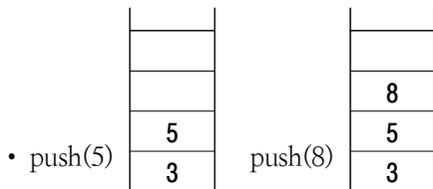
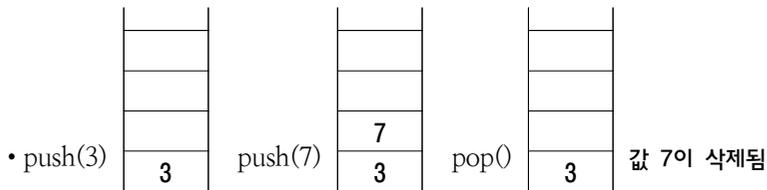
• 인덱스 13인 노드의 부모노드는 인덱스가 6이다.

5. 크기가 4인 빈 스택에 다음 연산을 차례로 수행한 후의 스택 상태를 바르게 표현한 것은?
 [2022년 국가 7급]

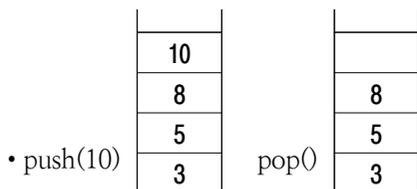
 push(3) → push(7) → pop() → push(5) → push(8) → peek() → push(10) → pop()



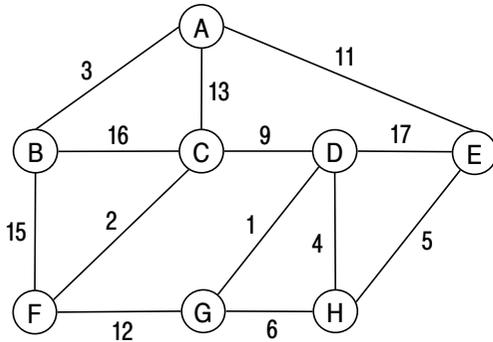
☞ 스택



문제에서 peek()의 기능을 제시하지 않았지만 영어단어 의미 그대로 적용해야 함 (옛보다)

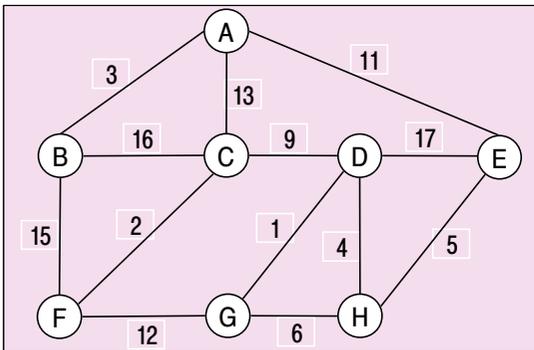


6. 그림은 정점 A~H 간의 거리를 나타낸 그래프이다. 다익스트라(Dijkstra) 알고리즘을 이용하여 정점 A로부터 다른 모든 정점까지의 최단경로를 구하고, 각 최단경로의 거리를 짧은 것부터 순서대로 나열한 것은? [2022년 국가 7급]

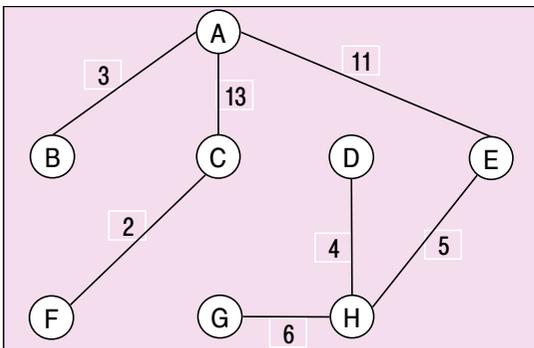


- ① 3, 11, 13, 15, 16, 20, 21 ② 3, 11, 13, 15, 16, 21, 22
 ③ 3, 11, 13, 16, 18, 20, 21 ④ 3, 11, 13, 16, 18, 21, 22

☞ 최단경로 - 다익스트라 알고리즘



↓ 정점 A에서 다른 모든 정점까지의 최단경로



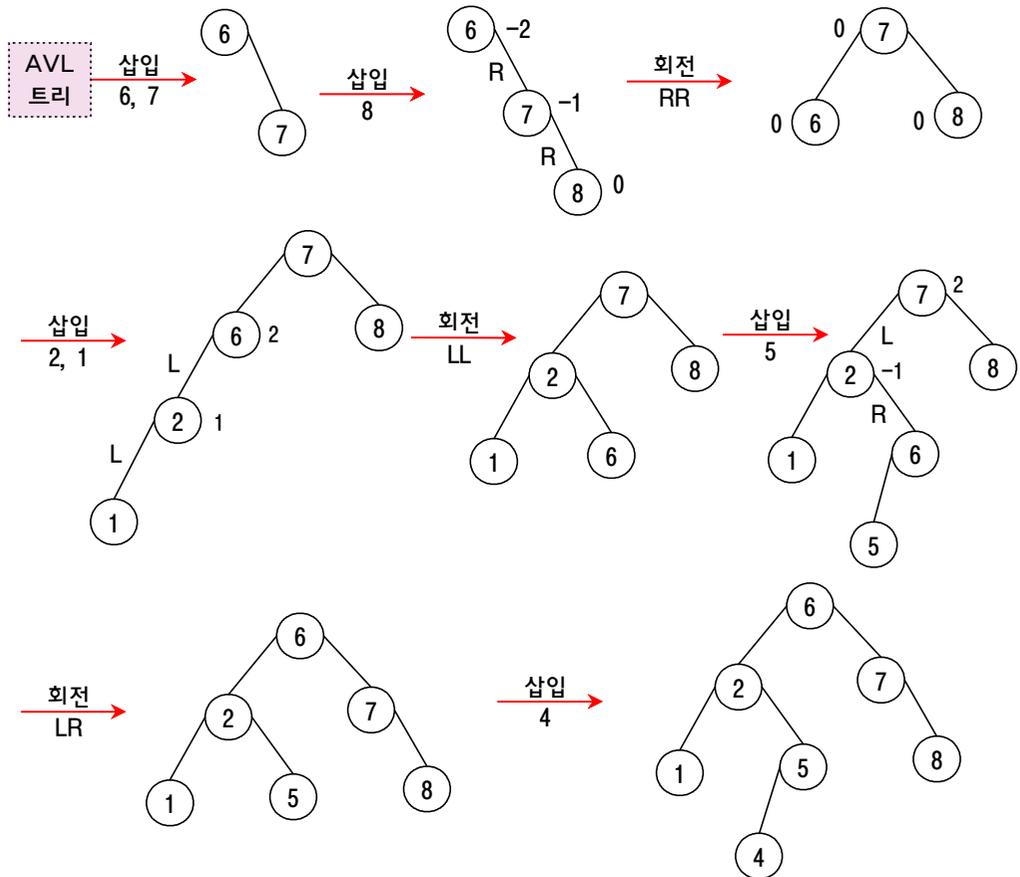
- A→B : 3
 - A→C : 13
 - A→E→H→D : 11 + 5 + 4 = 20
 - A→E : 11
 - A→C→F : 13 + 2 = 15
 - A→E→H→G : 11 + 5 + 6 = 22
 - A→E→H : 11 + 5 = 16
- //최단경로 순 : 3 11 13 15 16 20 22

7. 다음 데이터를 차례대로 하나씩 입력받아 AVL 트리를 생성할 때, 필요한 회전을 적용 순서대로 바르게 나열한 것은? [2022년 국가 7급]

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

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

☞ AVL 트리



• 회전 : RR회전, LL회전, LR회전

정답 : ③

8. 다음 C 코드는 원형 연결리스트에 속한 모든 노드의 값들을 더하여 출력한다. (가)~(다)에 들어갈 내용을 바르게 연결한 것은? [2022년 국가 7급]

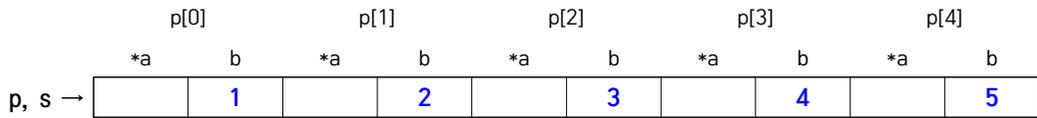
```
#include <stdio.h>
#include <malloc.h>
typedef struct Node
{
    struct Node *a;
    int b;
} Node;
int listSum(Node *p) {
    int sum = 0;
    Node *s = p;
    do {
        sum += (가) ;
        p = (나) ;
    } while ( (다) );
    return sum;
}
int main() {
    int i, s, MAX = 5;
    Node *p[MAX];
    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 |

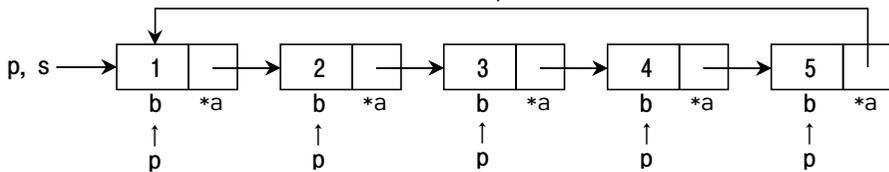
☞ 원형 연결리스트

```
typedef struct Node
{
    struct Node *a;
    int b;
} Node;
Node *p[5];          //포인터 배열
```

↓
다음과 같은 포인터 배열에 값을 저장한 것이다.



↓ 이를 원형연결리스트 그림으로 다시 그리면



↓ 원형 연결리스트에 속한 모든 노드의 값들을 더함

```
int listSum(Node *p) {
    int sum = 0;
    Node *s = p;
    do {
        sum += (가) (*p).b ; //포인터 p가 가리키는 노드의 값을 sum에 누적
        p = (나) (*p).a ; //추적, 포인터 p는 다음 노드를 가리킴
    } while ( (다) p != s ); //포인터 p와 s가 가리키는 노드가 다르면 참
    return sum;
}
```

```
// 구조체 포인터
(*p).a ≡ p->a;
(*p).b ≡ p->b;
```


10. 다음 tree_height() 함수는 이진트리의 루트노드를 매개변수로 받아 트리의 높이를 반환한다. (가)~(다)에 들어갈 내용을 바르게 연결한 것은? [2022년 국가 7급]

```

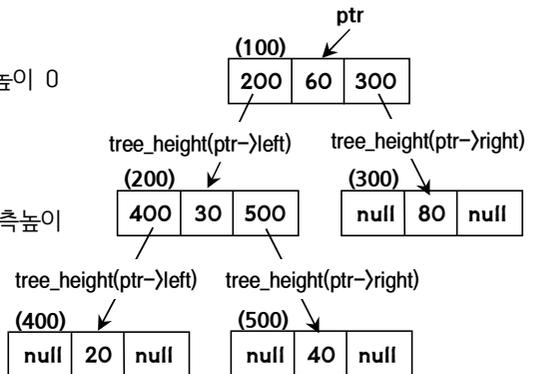
typedef struct node {
    int value;
    struct node *left;
    struct node *right;
} node;
int tree_height (node *ptr) {
    int left_h, right_h;
    if (ptr == NULL) return 0;
    else {
        left_h = tree_height(ptr->left);
        right_h = tree_height(ptr->right);
        if ( (가) )
            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 |

♣ 트리의 높이

```

int tree_height (node *ptr) {
    int left_h, right_h;
    if(ptr == NULL) return 0; //공백트리 : 높이 0
    else {
        left_h = tree_height(ptr->left);
        right_h = tree_height(ptr->right);
        if ( (가) left_h > right_h ) //좌측높이 > 우측높이
            return (나) left_h + 1 ; //좌측높이
        else
            return (다) right_h + 1 ; //우측높이
    }
}
    
```



정답 : ②