



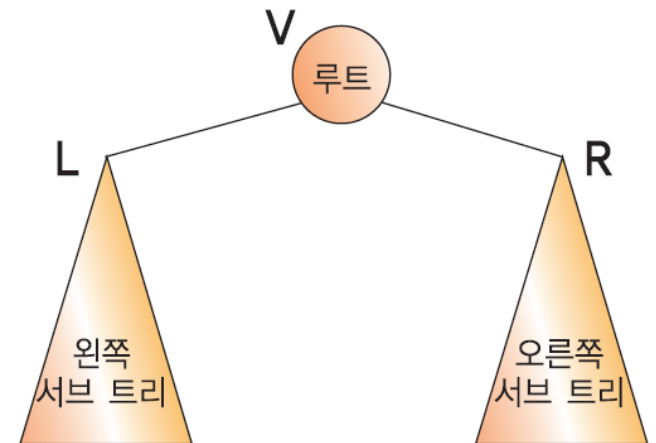
CSE2010 자료구조론

Week 6: Tree 3

ICT융합학부 한진영

이진 트리 순회(traversal)

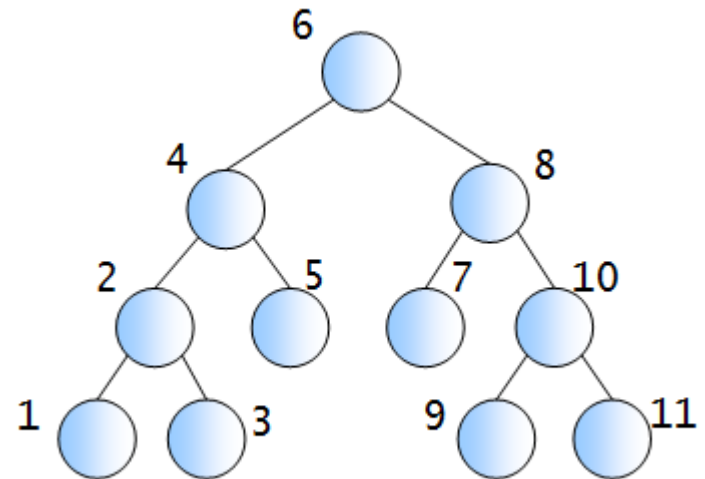
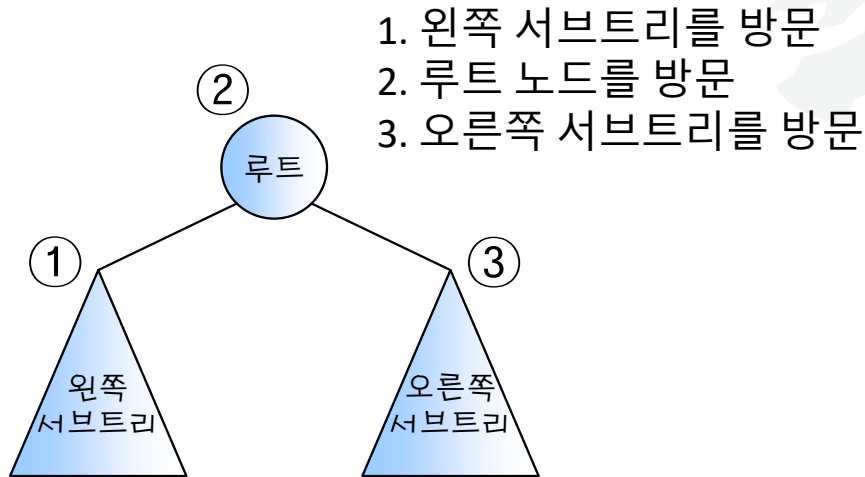
- 트리 순회 : 트리의 모든 노드를 한번씩 방문하는 것
 - 트리의 구조를 분석하거나, 각 노드에 저장되어 있는 정보를 체계적으로 읽어올 때 사용
- 3가지의 기본적인 순회방법
 - 전위순회(preorder traversal): VLR
 - 자손 노드보다 루트 노드를 먼저 방문
 - 중위순회(inorder traversal): LVR
 - 왼쪽 자손, 루트, 오른쪽 자손 순으로 방문
 - 후위순회(postorder traversal): LRV
 - 루트 노드보다 자손을 먼저 방문



중위 순회

■ 알고리즘

- 먼저 트리의 왼쪽으로 NULL 노드를 만날 때까지 계속 내려간 다음, NULL 노드를 만나면 그 NULL 노드의 부모 노드를 방문하고 다시 오른쪽 부트리에 대해 같은 방법으로 수행함
- 만약 오른쪽 부트리에 더 이상 방문할 노드가 없다면 트리의 바로 윗 레벨의 방문되지 않은 마지막 노드에 대해서 계속적으로 운행



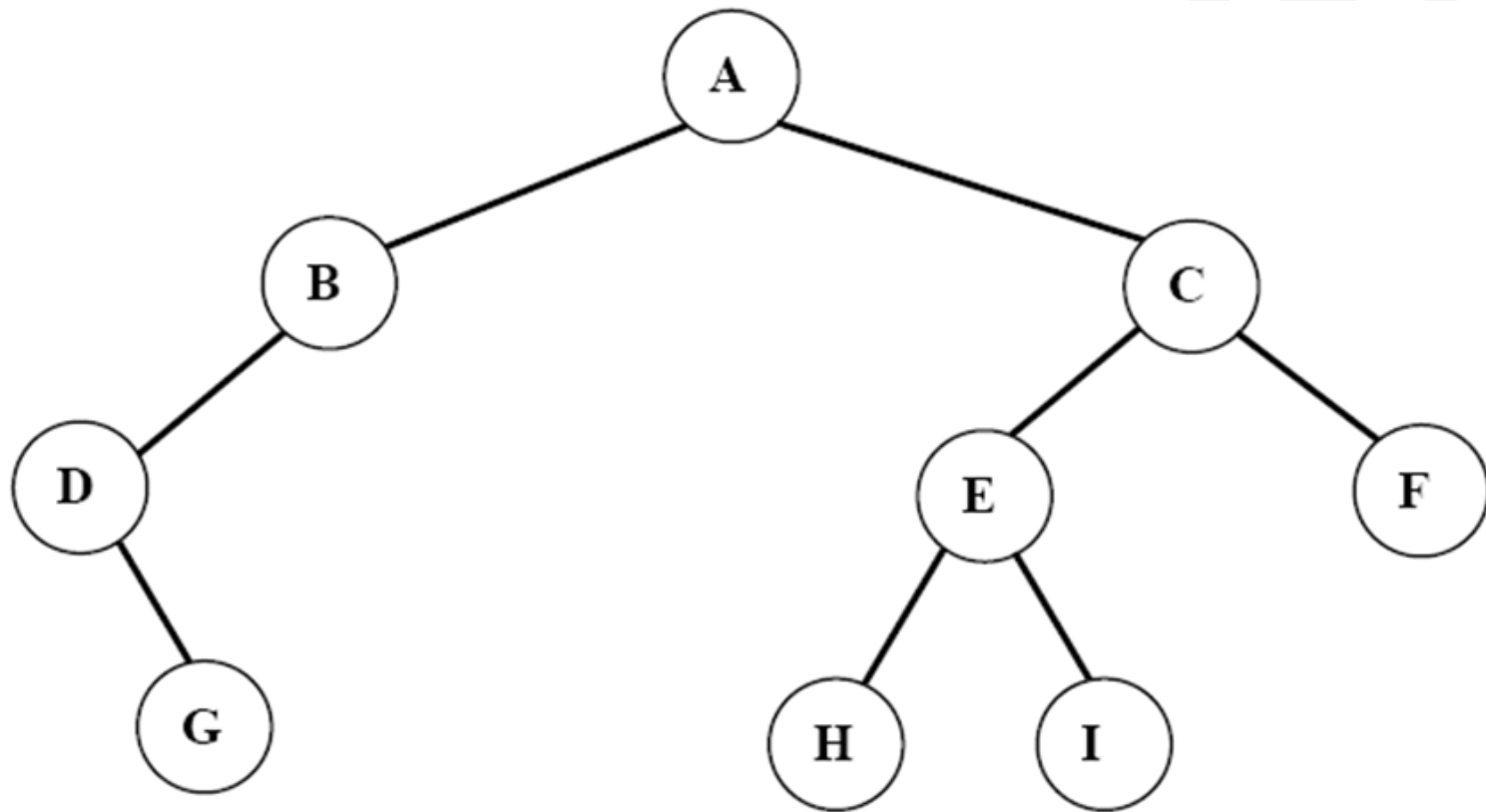
중위 순회 알고리즘

- 순환 호출 활용

```
inorder(x)
```

```
if  $x \neq \text{NULL}$   
    then inorder(LEFT( $x$ ));  
        print DATA( $x$ );  
        inorder(RIGHT( $x$ ));
```

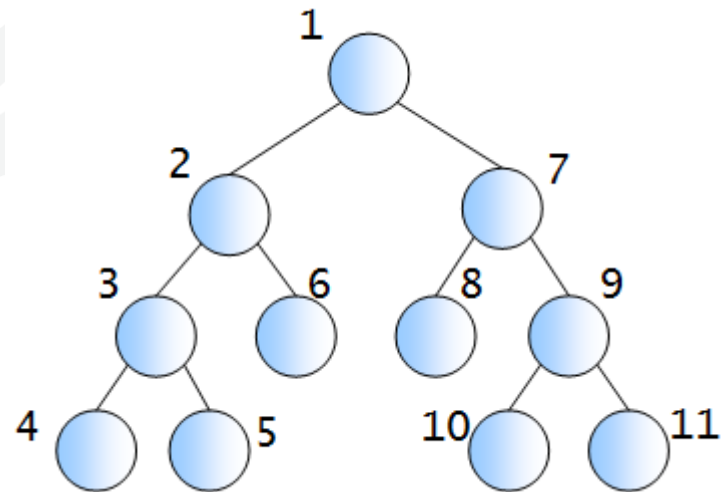
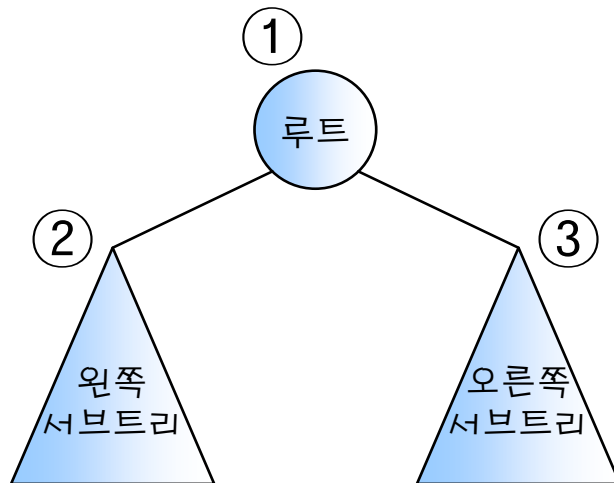
중위 순회 예



전위 순회

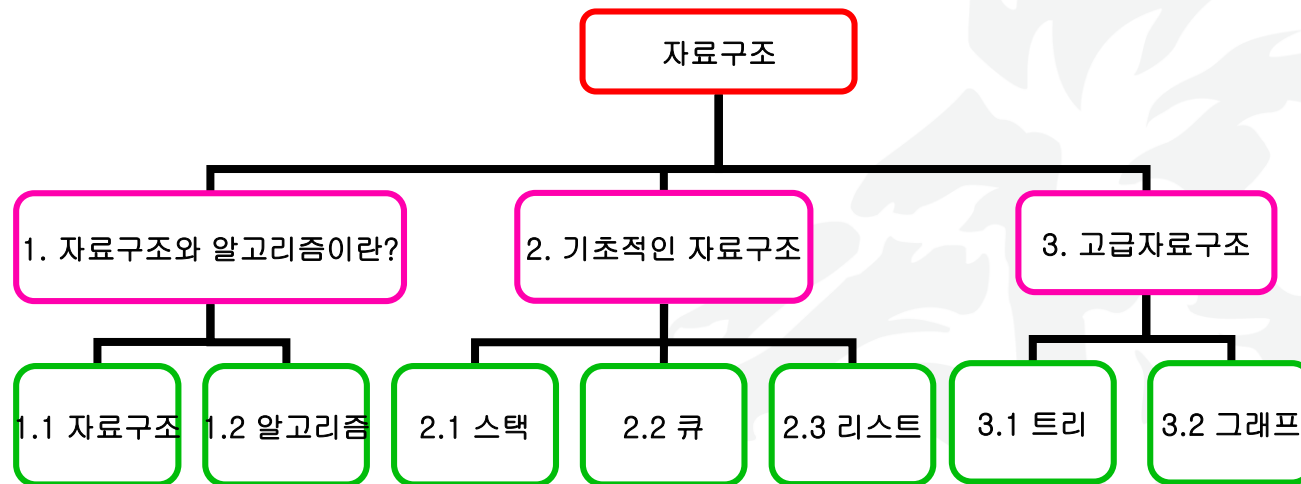
■ 기본 알고리즘

- 1. 루트 노드를 방문
- 2. 왼쪽 서브트리를 방문
- 3. 오른쪽 서브트리를 방문



전위 순회 응용

■ 구조화 된 문서 출력



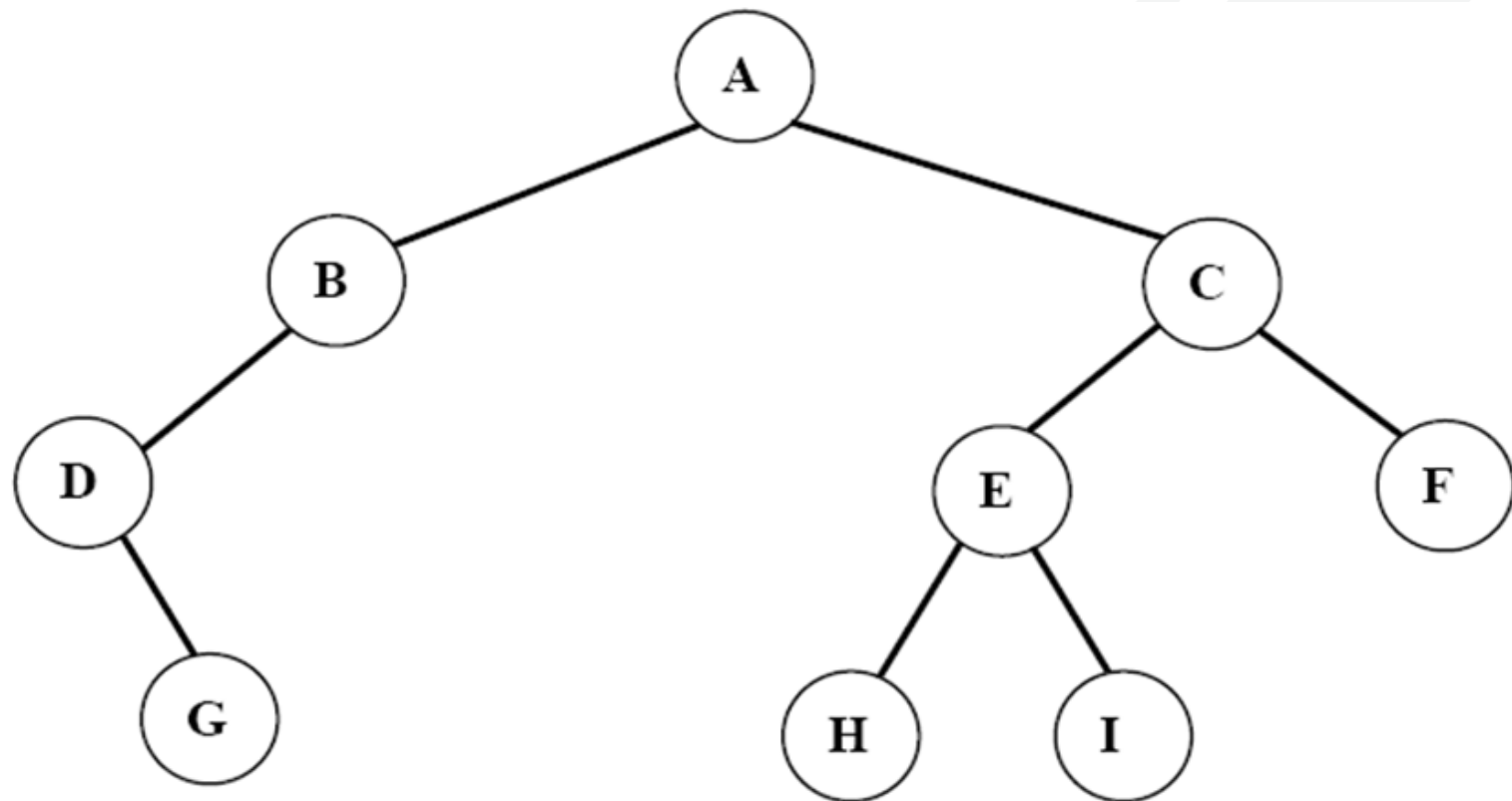
전위 순회 알고리즘

- 순환 호출 활용

preorder(x)

```
if x≠NULL
    then print DATA(x);
        preorder(LEFT(x));
        preorder(RIGHT(x));
```

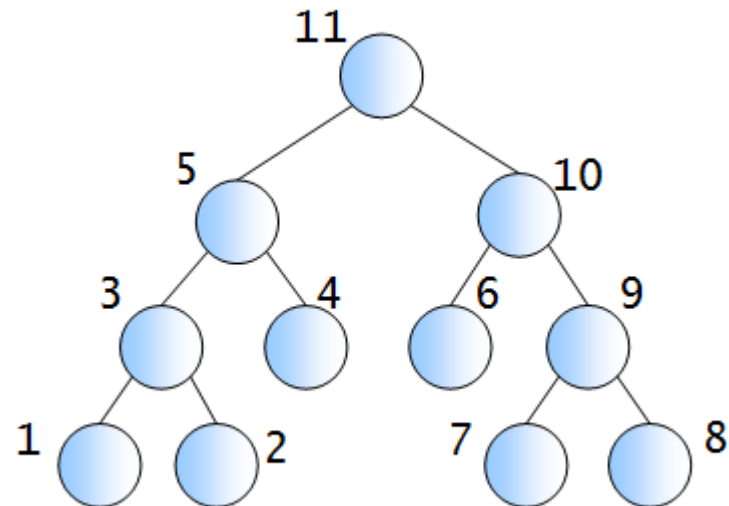
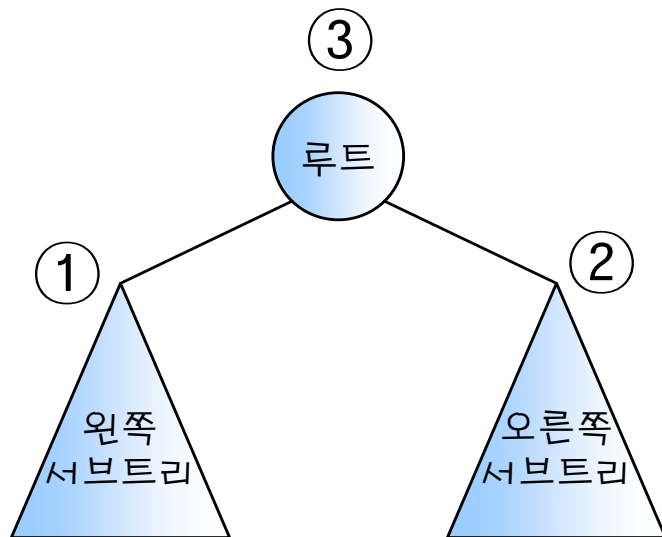

전위 순회 예



후위 순회

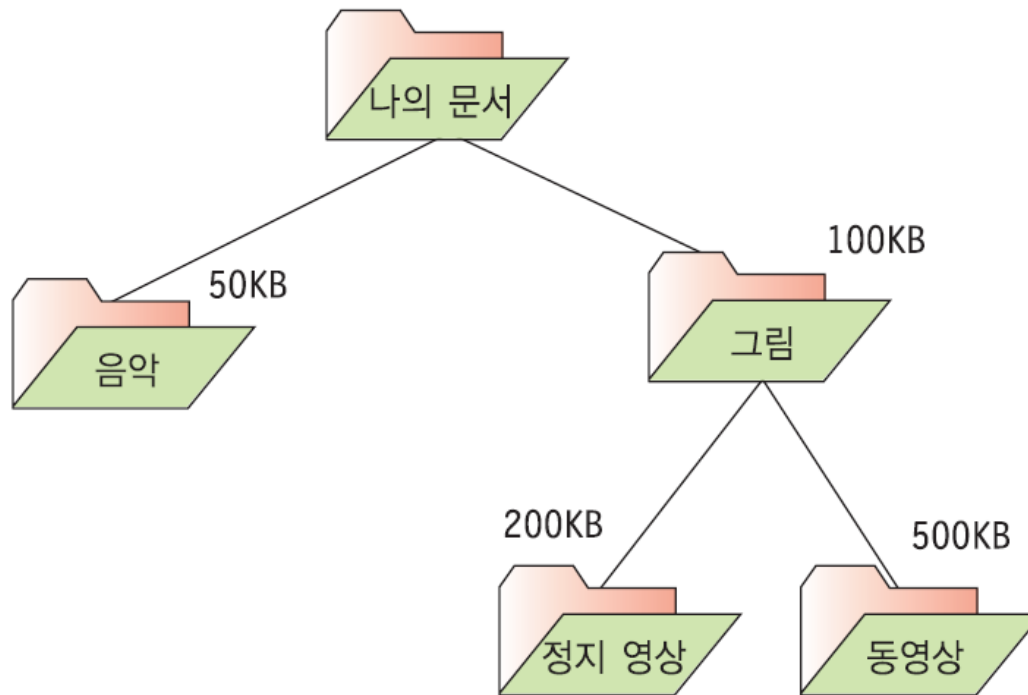
■ 알고리즘

- 1. 왼쪽 서브트리를 방문
- 2. 오른쪽 서브트리를 방문
- 3. 루트 노드를 방문



후위 순회 응용

■ 디렉토리 용량 계산



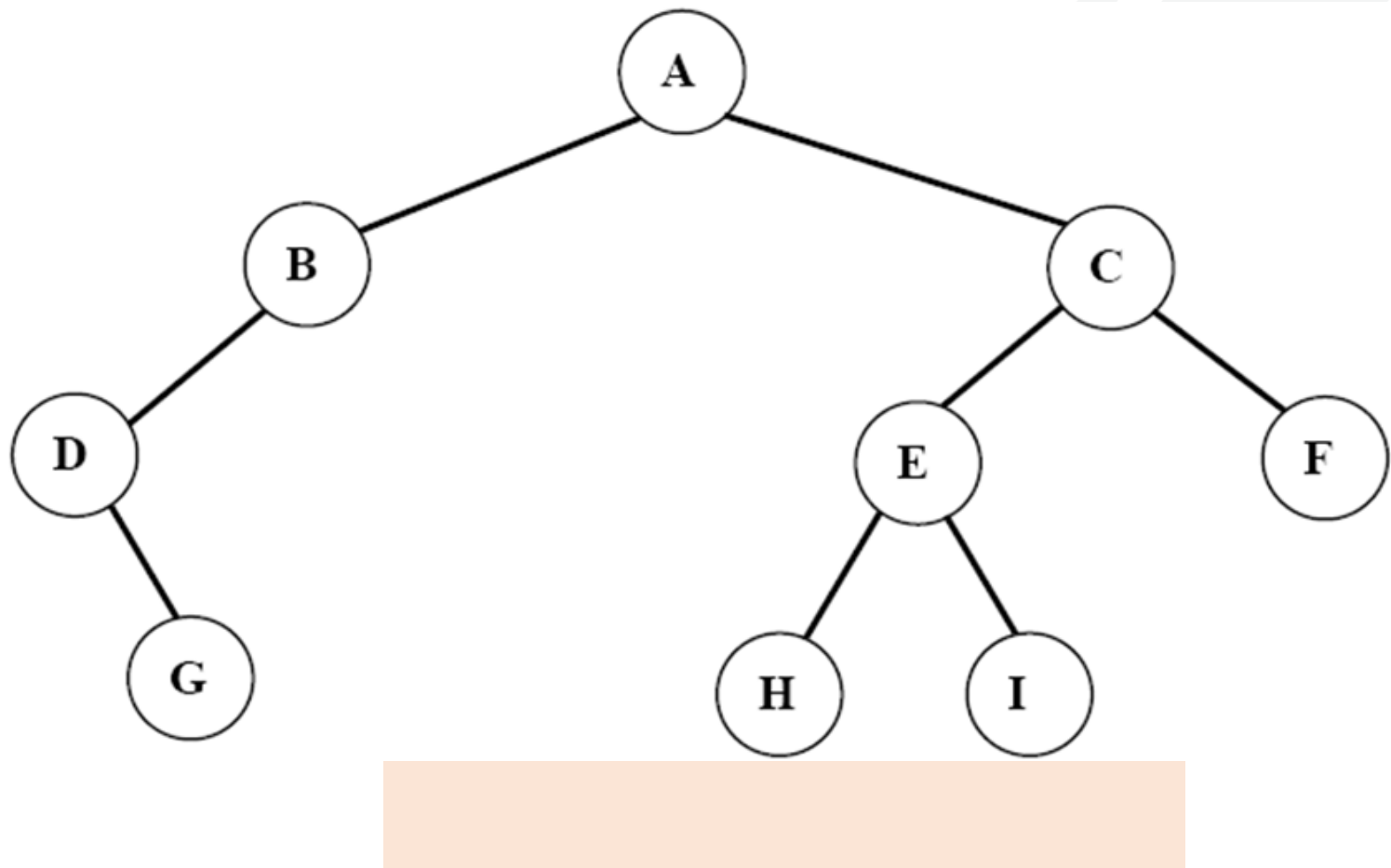
후위 순회 알고리즘

- 순환 호출 활용

postorder(x)

```
if x≠NULL
    then postorder(LEFT(x));
        postorder(RIGHT(x));
        print DATA(x);
```

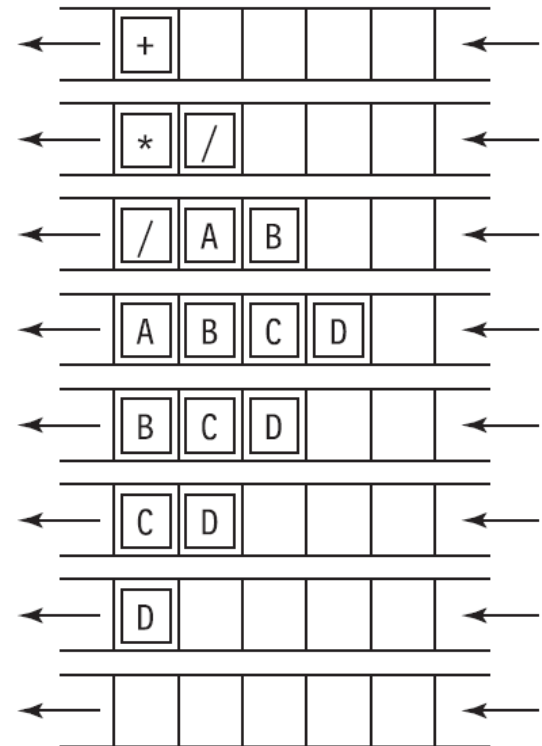
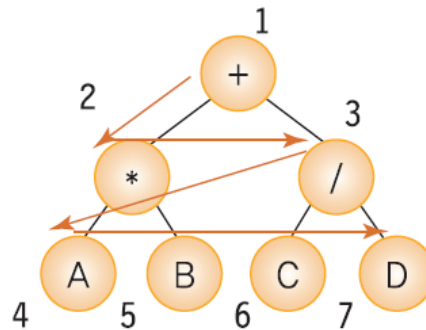
후위 순회 예



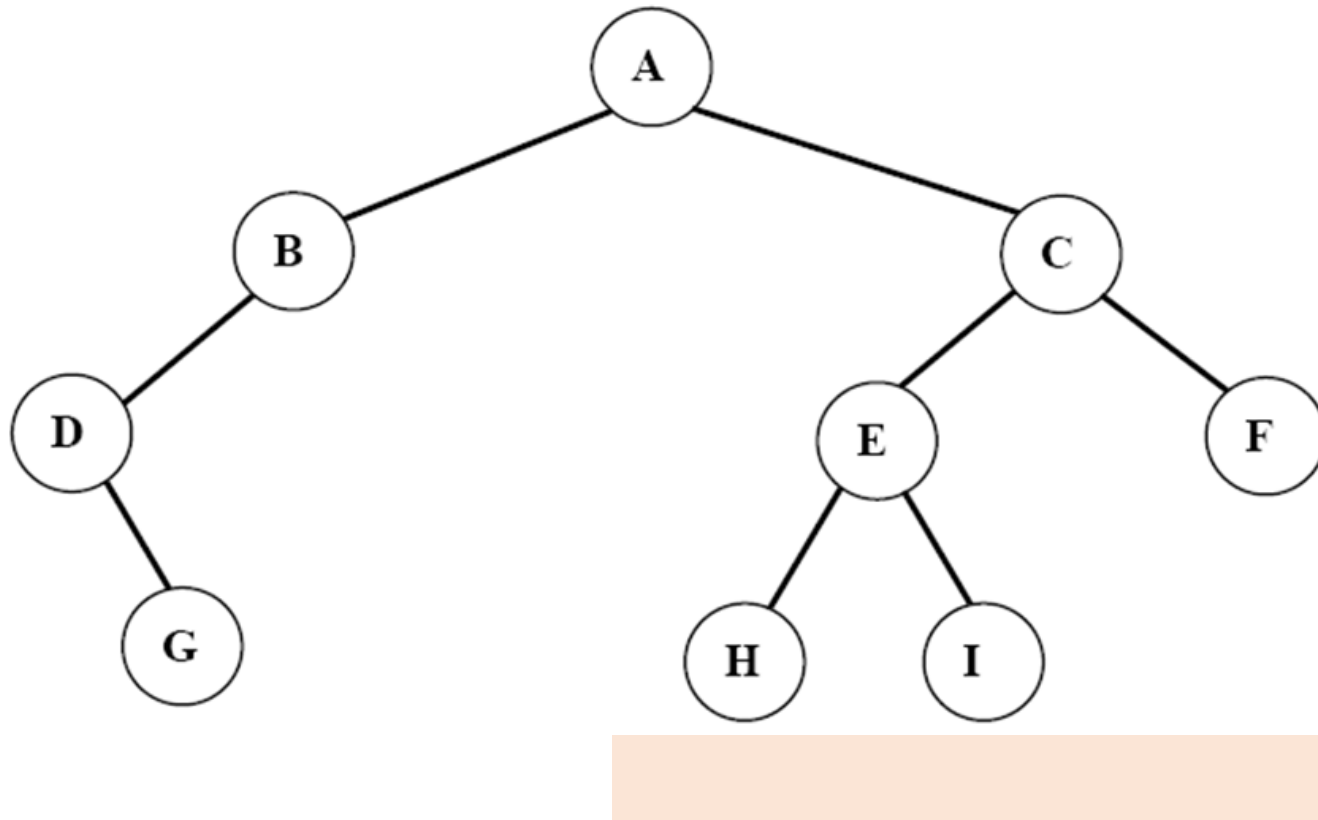
레벨 순회

■ 레벨 순회(level order)

- 각 노드를 레벨순으로 검사하는 순회 방법
- 지금까지의 순회 방법이 스택을 사용했던 것에 비해 레벨 순회는 큐를 사용하는 순회 방법



레벨 순회 예



레벨 순회 알고리즘

```
level_order(root)
```

```
    initialize queue;
```

```
    enqueue(queue, root);
```

```
    while is_empty(queue)≠TRUE do
```

```
        x ← dequeue(queue);
```

```
        print x→data;
```

```
        if(x → left != NULL) then
```

```
            enqueue(queue, x→ left);
```

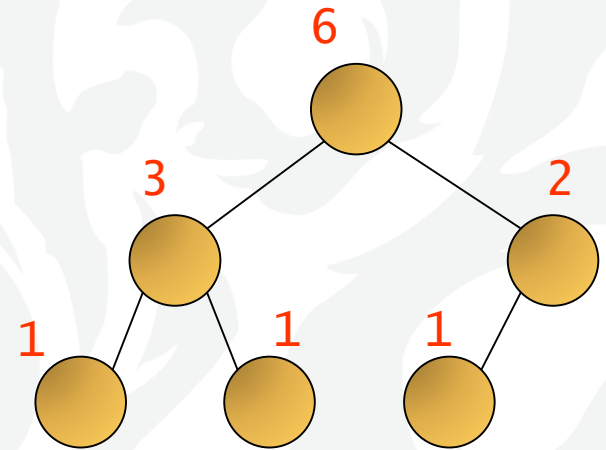
```
        if(x → right !=NULL) then
```

```
            enqueue(queue, x→ right);
```


이진트리연산: 노드개수

- 탐색 트리안의 노드의 개수를 계산
- 각각의 서브트리에 대하여 순환 호출한 다음, 반환되는 값에 1을 더하여 반환

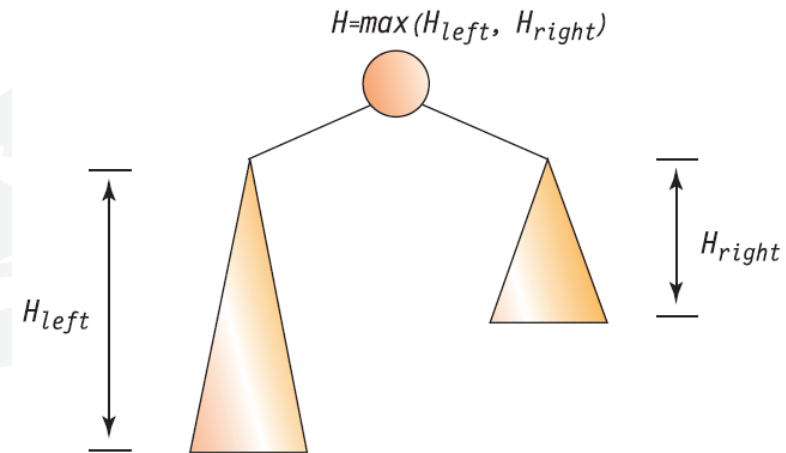
```
int get_node_count(TreeNode *node)
{
    int count=0;
    if( node != NULL )
        count = 1 + get_node_count(node->left)+
                get_node_count(node->right);
    return count;
}
```



이진트리연산: 높이

- 서브트리에 대하여 순환호출하고 서브 트리들의 반환값 중에서 최대 값을 구하여 반환

```
int get_height(TreeNode *node)
{
    int height=0;
    if( node != NULL )
        height = 1 + max(get_height(node->left),
                        get_height(node->right));
    return height;
}
```



Week 6: Tree 3

