

Tree

전위순회(pre-order)

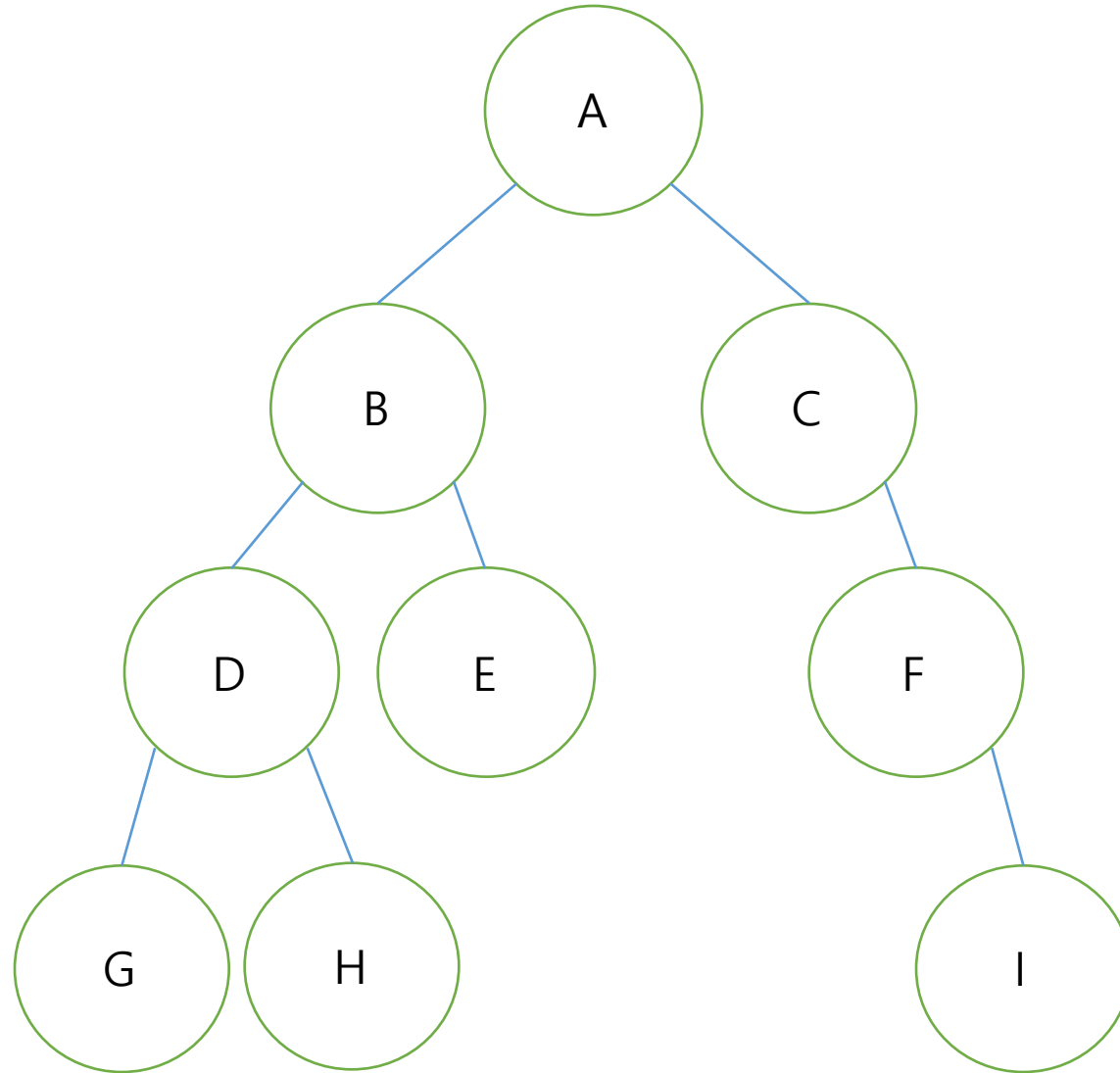
- Root를 방문한다
- 왼쪽 subtree를 방문한다
- 오른쪽 subtree를 방문한다

전위순회(pre-order)

Sudo code

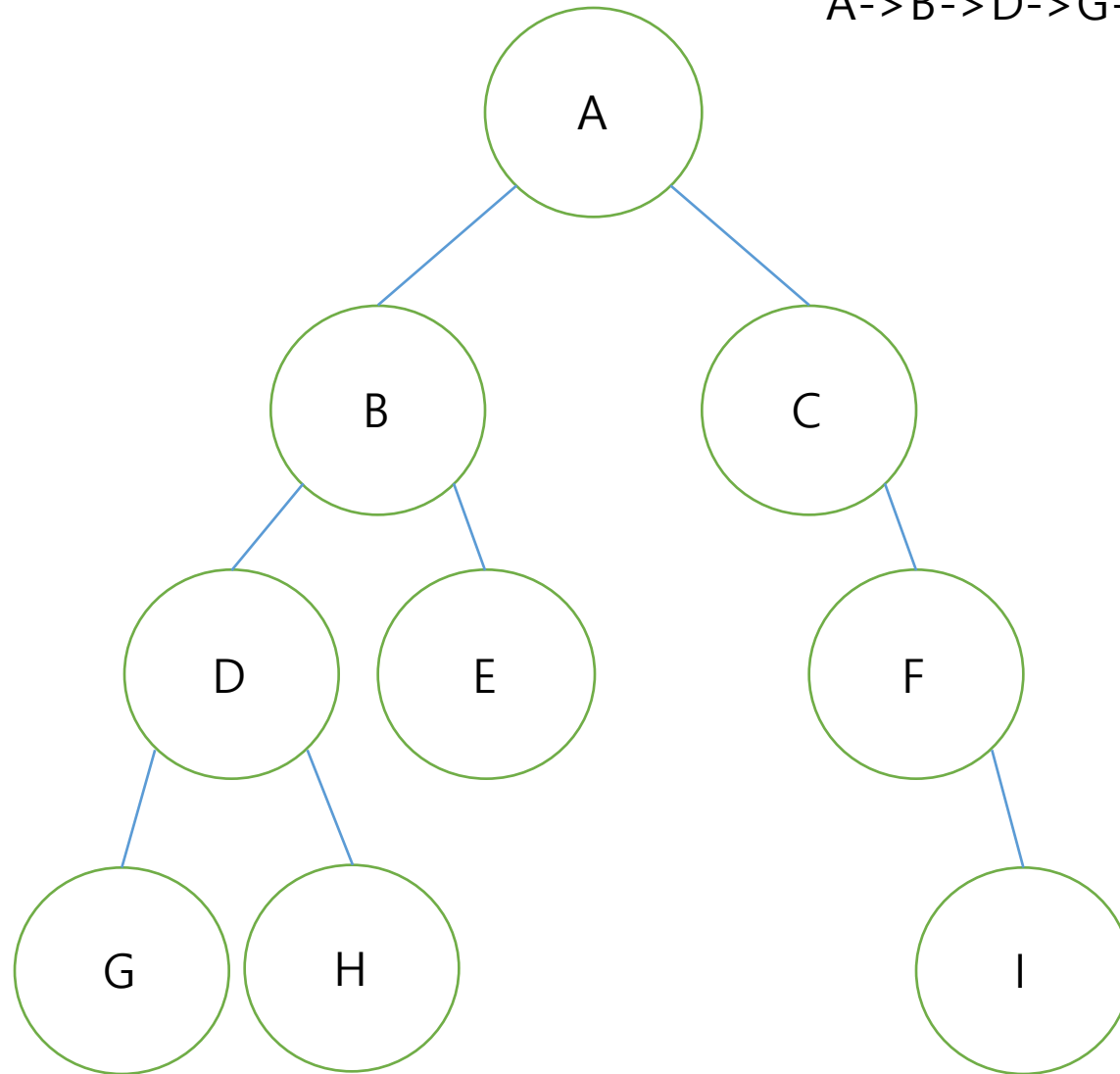
```
pre_order_traversal(current node){  
    if(current node is not leaf node){  
        visit(current node)  
        pre_order_traversal(left child node)  
        pre_order_traversal(right child node)  
    }  
}
```

전위순회(pre-order)



전위순회(pre-order)

A->B->D->G->H->E->C->F->I



중위순회(in-order)

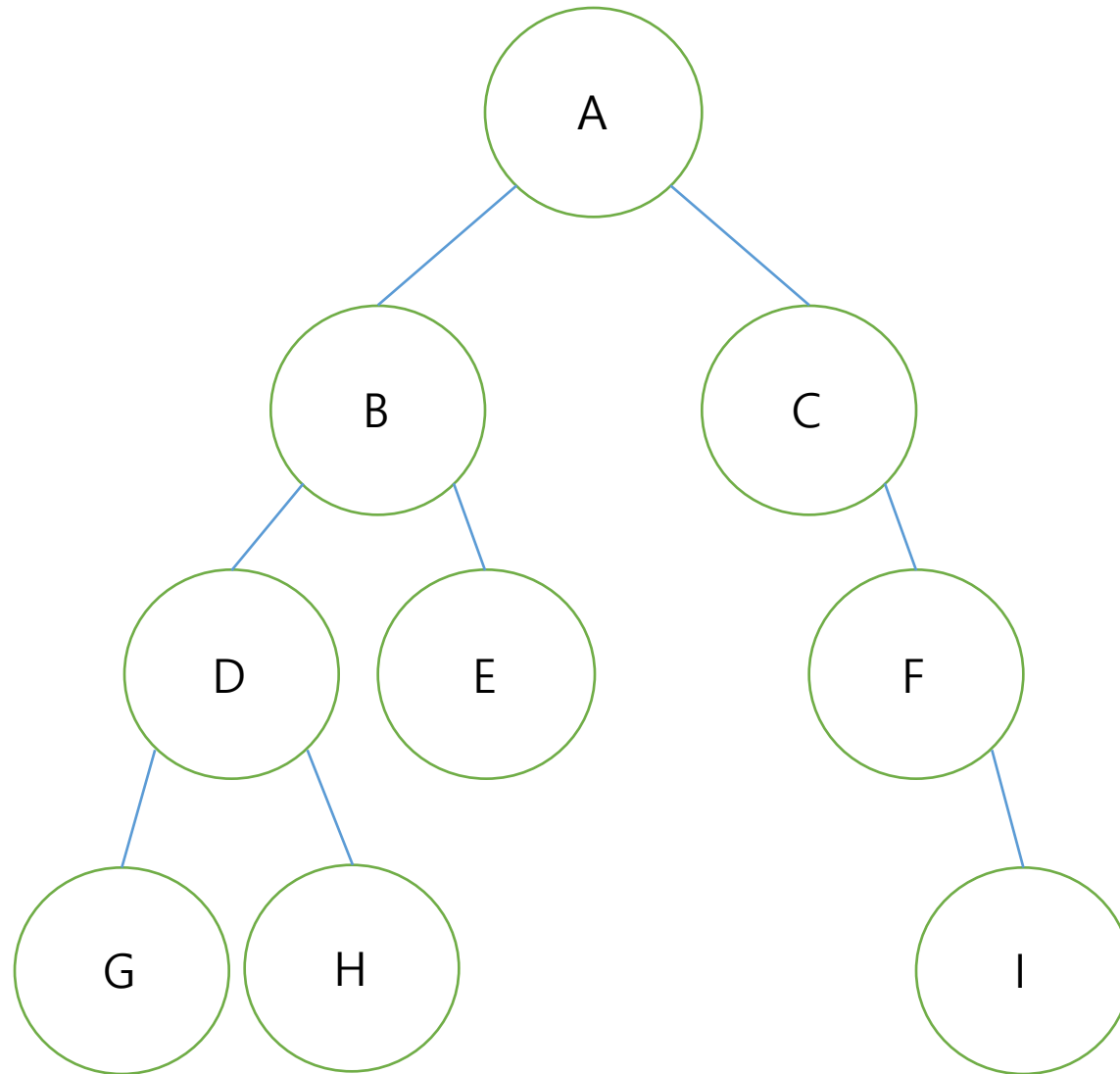
- 왼쪽 subtree를 방문한다
- Root를 방문한다
- 오른쪽 subtree를 방문한다

중위순회(in-order)

Sudo code

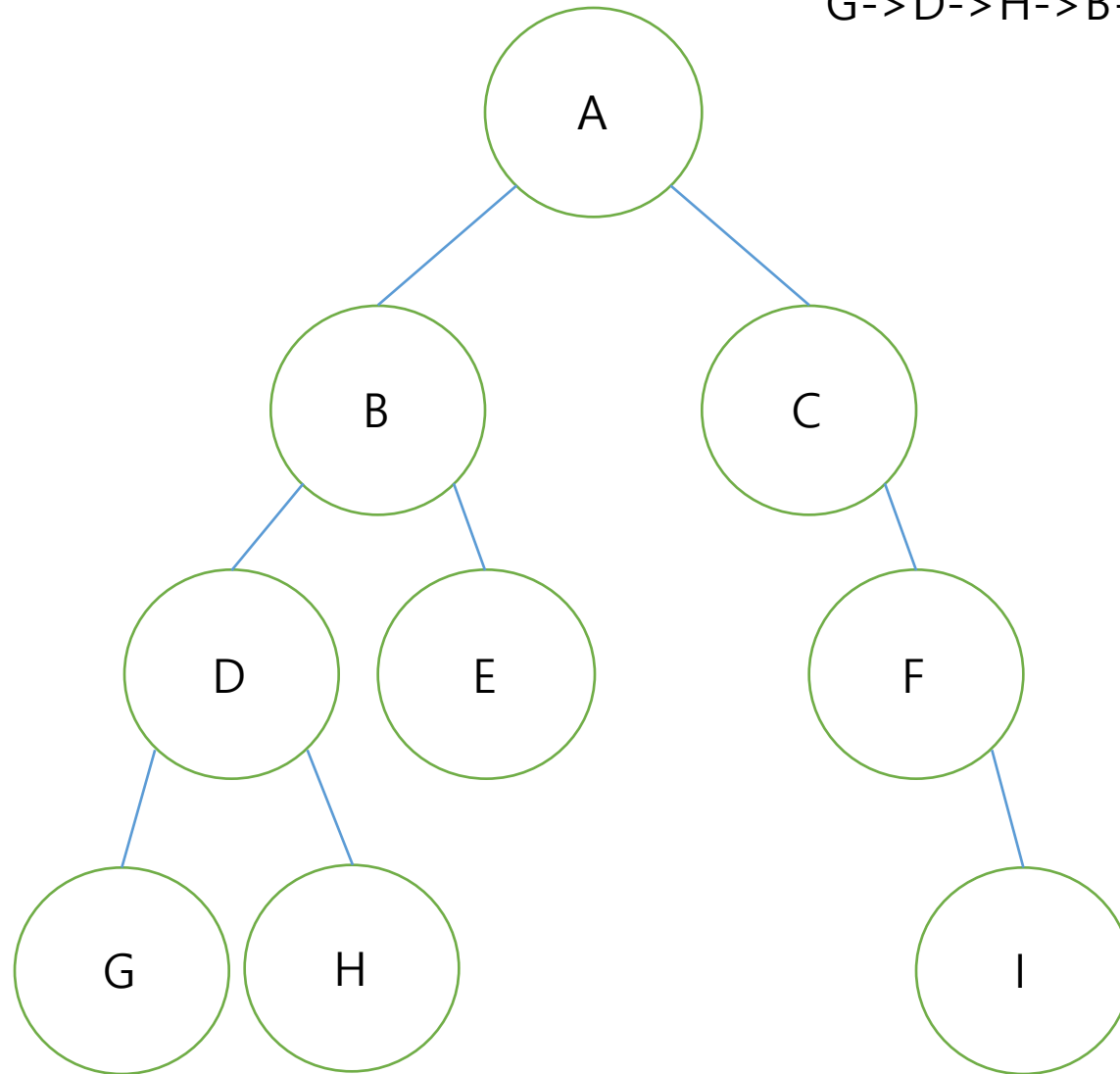
```
in_order_traversal(current node){  
    if(current node is not leaf node){  
        in_order_traversal(left child node)  
        visit(current node)  
        in_order_traversal(right child node)  
    }  
}
```

중위순회(in-order)



중위순회(in-order)

G->D->H->B->E->A->C->I->F



후위순회(post-order)

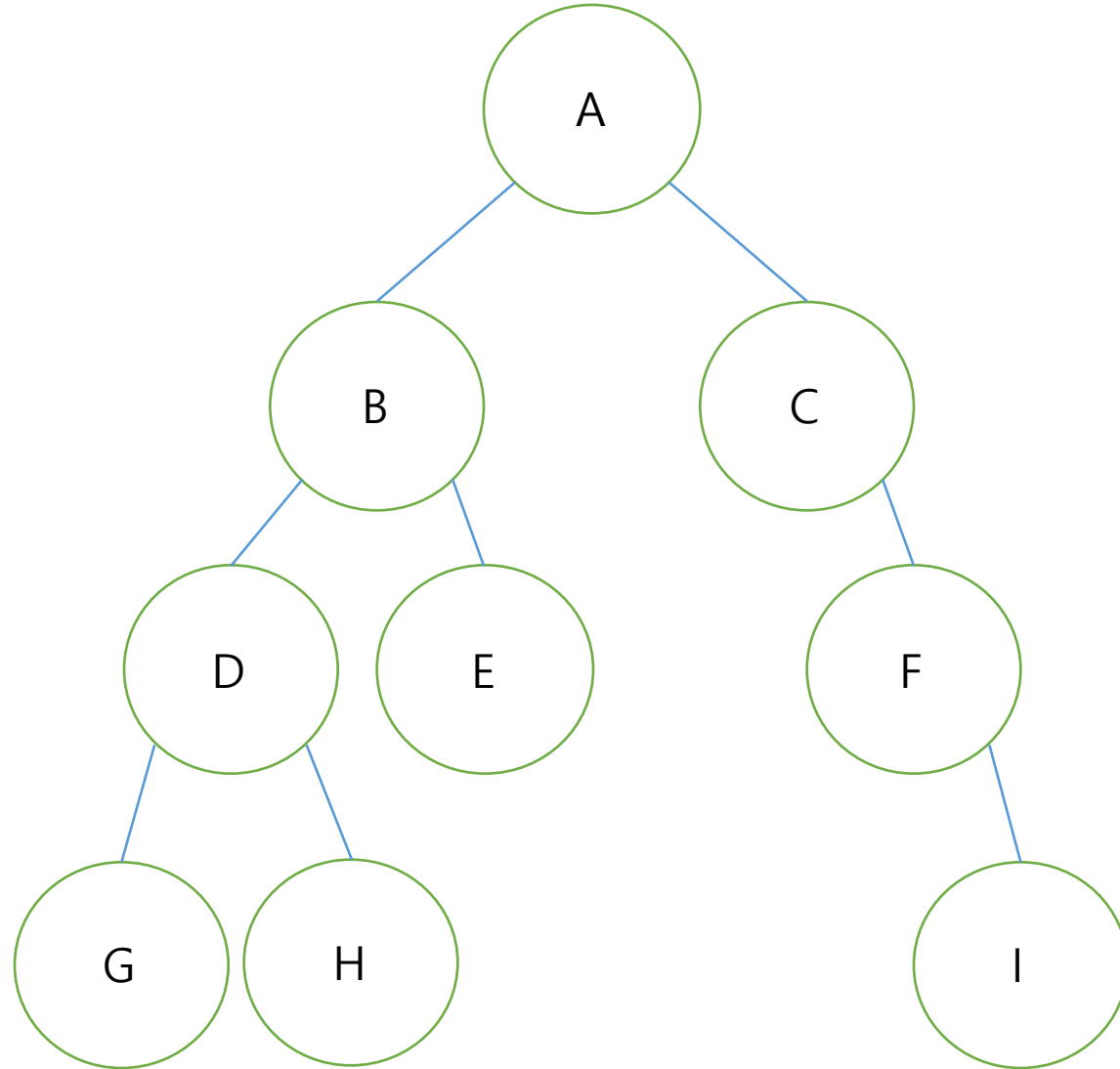
- 왼쪽 subtree를 방문한다
- 오른쪽 subtree를 방문한다
- Root를 방문한다

후위순회(post-order)

Sudo code

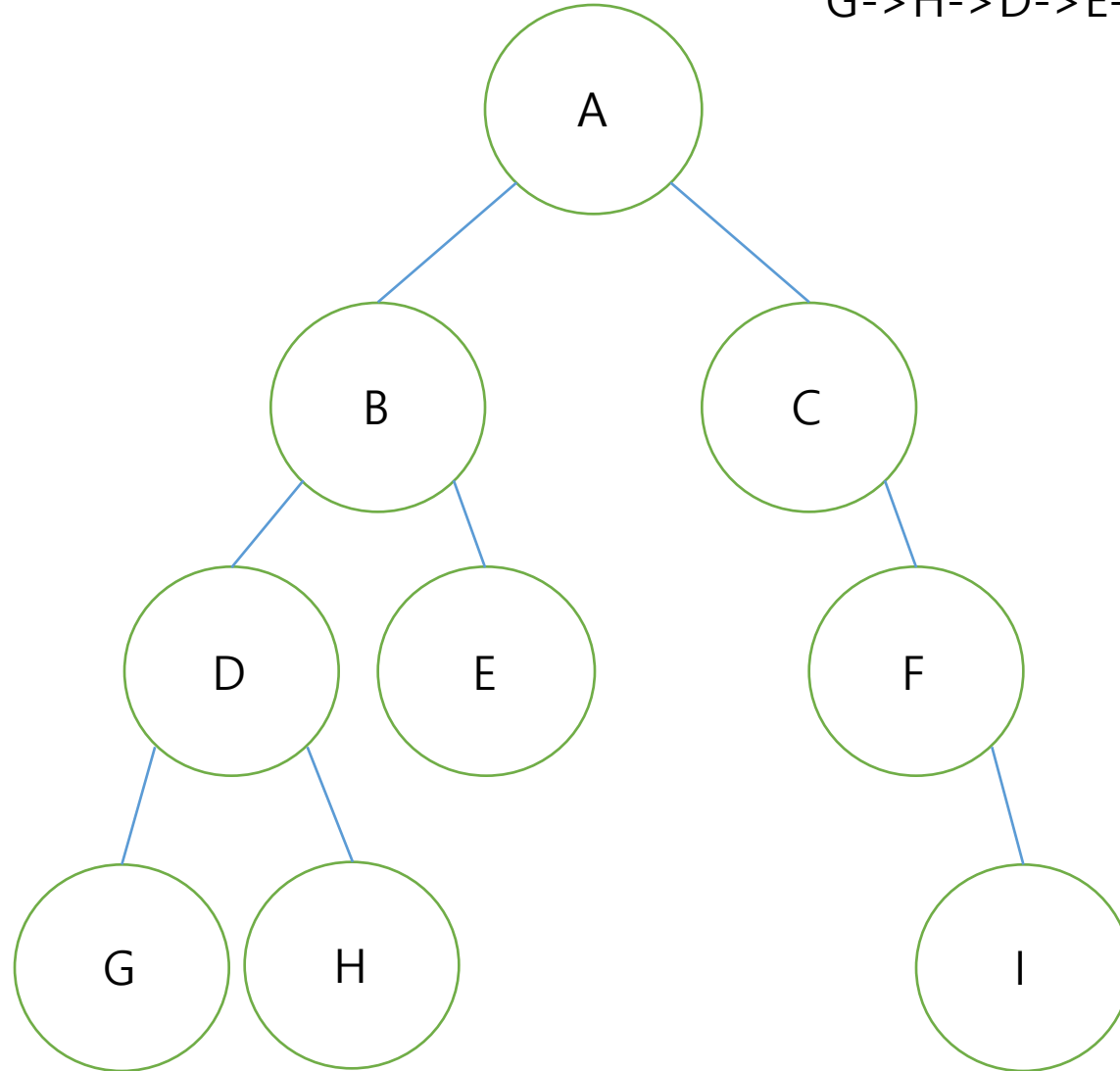
```
post_order_traversal(current node){  
    if(current node is not leaf node){  
        post_order_traversal(left child node)  
        visit(current node)  
        post_order_traversal(right child node)  
    }  
}
```

후위순회(post-order)



후위순회(post-order)

G->H->D->E->B->I->F->C->A

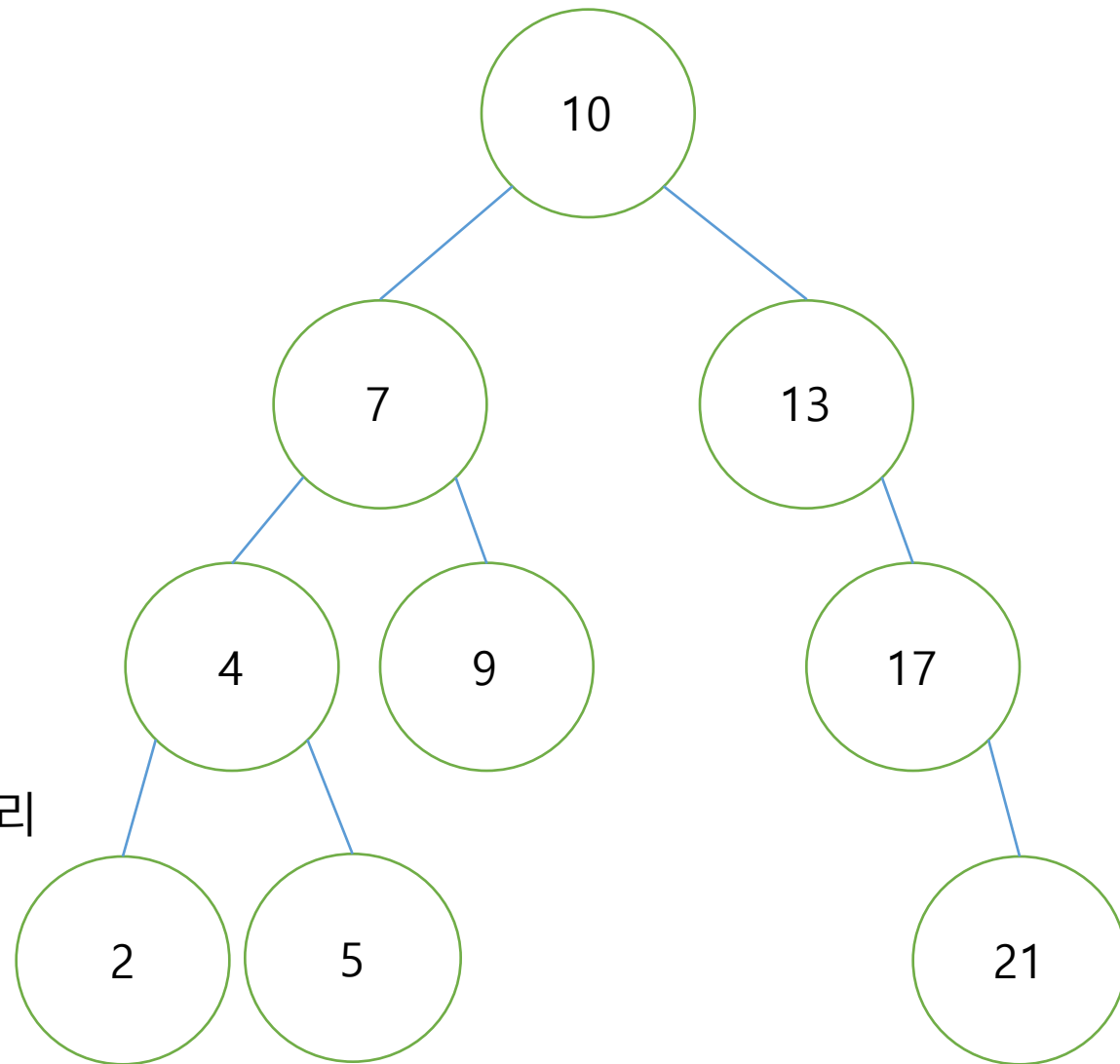


이진 트리(binary tree)

- 자식 노드가 2개 이하인 트리

이진 탐색 트리 (binary search tree)

- 이진 탐색 트리이며,
왼쪽 자식은 자신보다 작은값이
오른쪽 자식은 자신보다 큰 값이 저장되는 트리



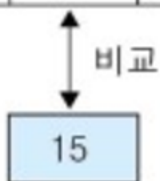
이진 탐색(binary search)

- 정렬된 배열에서 값을 탐색하는 방법

1	3	8	11	15	17	20
---	---	---	----	----	----	----

[그림 8-5] 이진 탐색을 위한 데이터 집합

1	3	8	11	15	17	20
---	---	---	----	----	----	----



1	3	8	11	15	17	20
---	---	---	----	----	----	----

탐색 영역



시간복잡도: $n\log(n)$