고 급 문 제 해 결

<문제 9.13>

마커가 포함된 전위 순회를 사용해서 이진 트리 복원하기

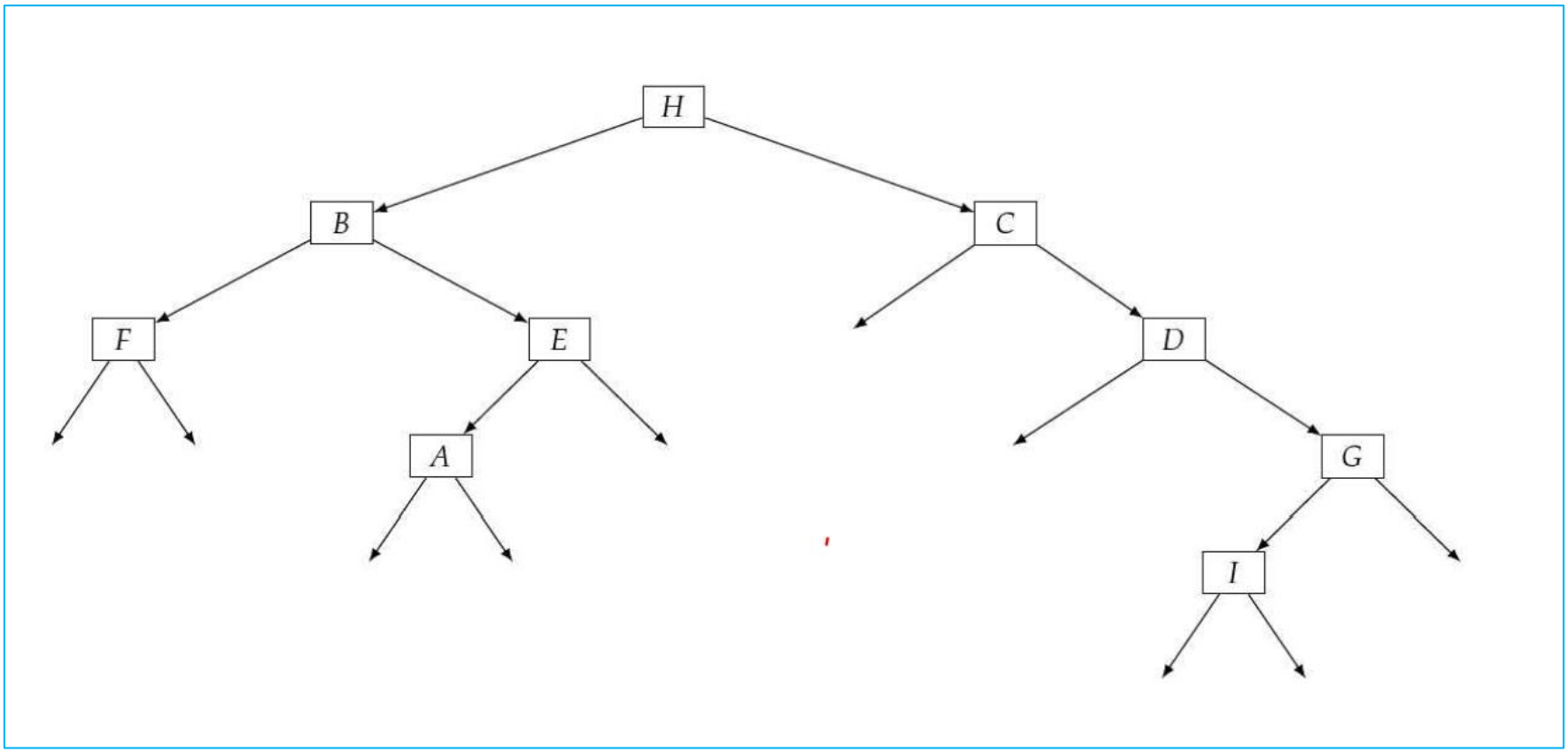
2018117543 한재성

Chapter9

Binary Tree

14.8 정렬된 배열에서 높이가 최소인 이진 탐색 트리 만들기

$\langle H, B, F, \texttt{null}, \texttt{null}, E, A, \texttt{null}, \texttt{null}, \texttt{null}, C, \texttt{null}, D, \texttt{null}, G, I, \texttt{null}, \texttt{null}, \texttt{null} \rangle$



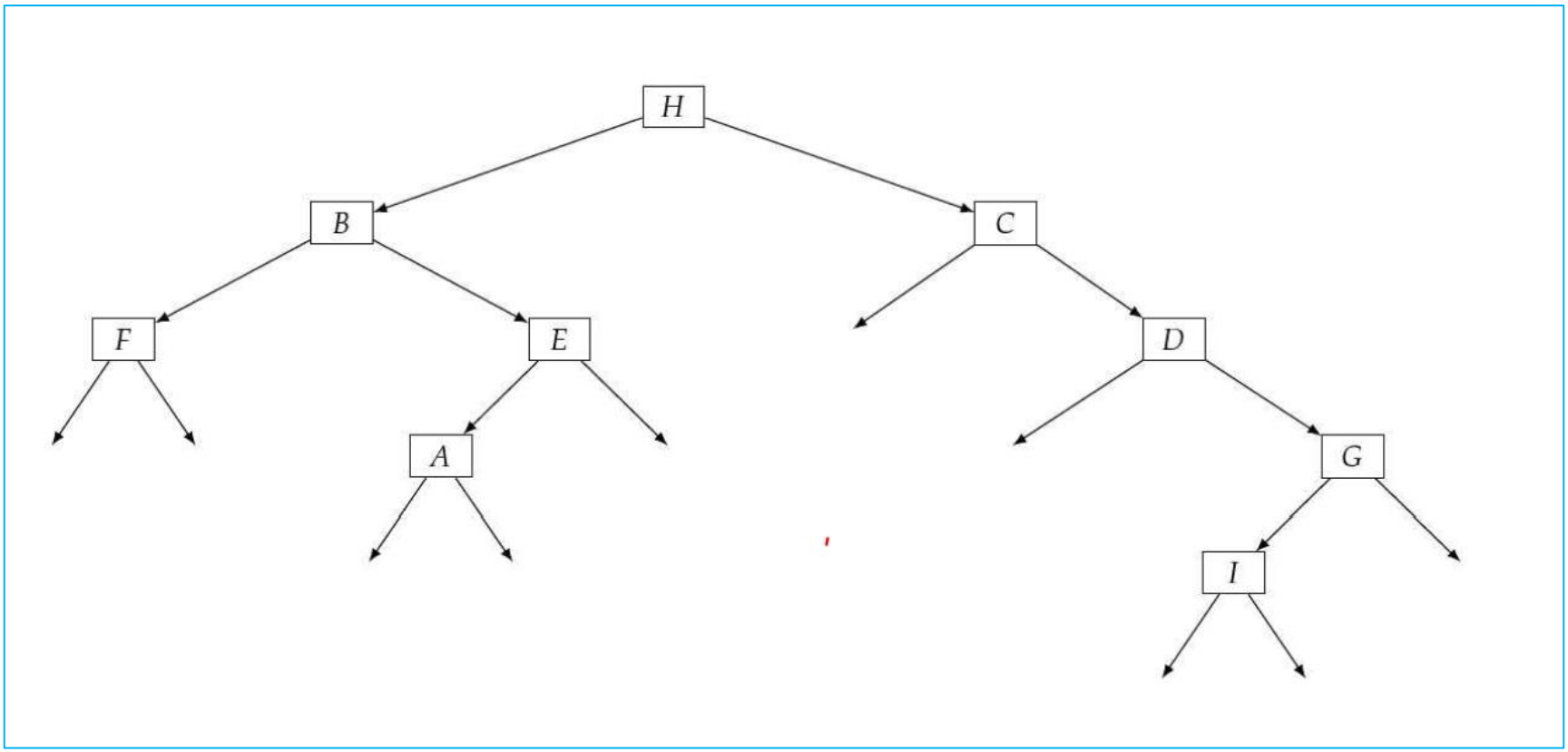
이진 트리의 성질

- Preorder 결과가 같더라도
  - ✓ 모양은 다를 수 있다.
- Unique한 모양을 알려면
  - ✓ 추가 정보가 필요
  - ✓ Preorder, Inorder
  - ✓ Postorder, Inorder
  - ✓ Marker(Null)

14.8 정렬된 배열에서 높이가 최소인 이진 탐색 트리 만들기 - idea

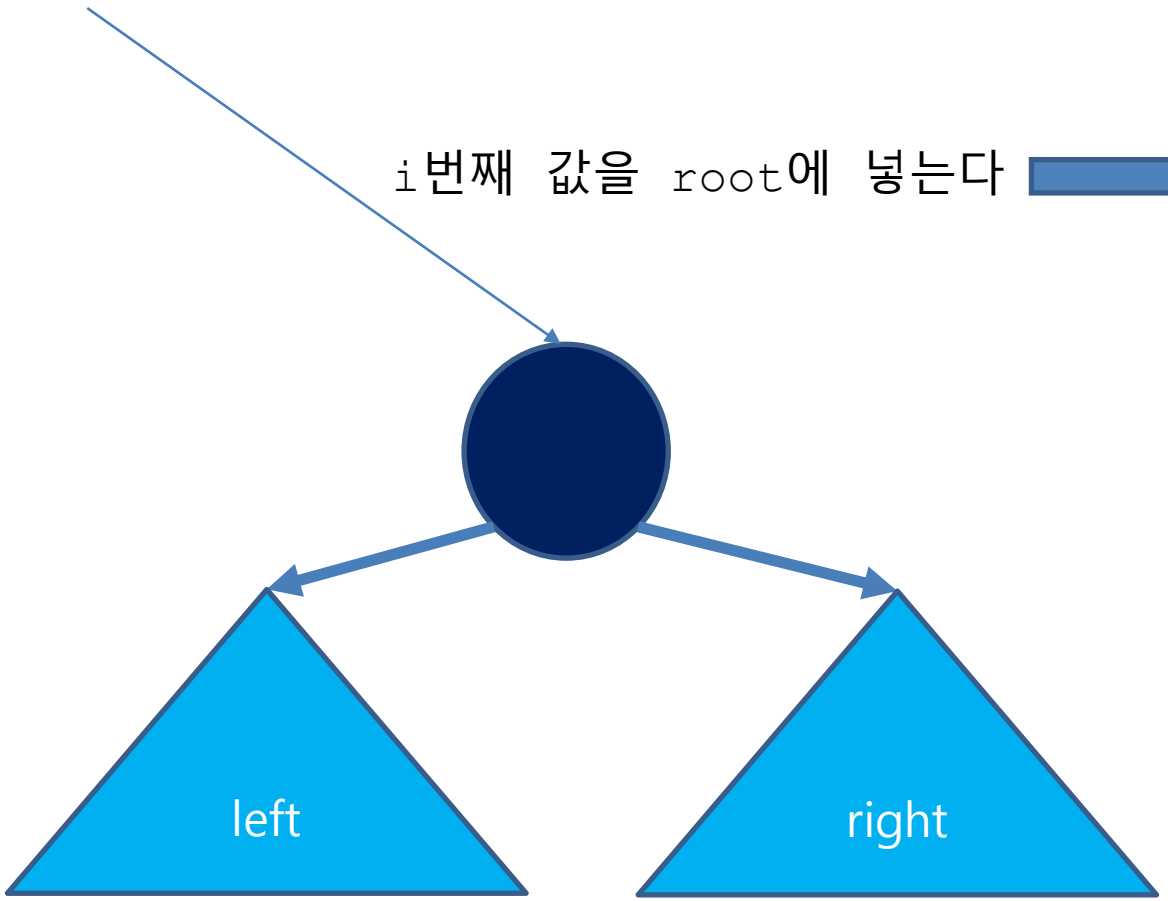
$\langle H, B, F, \texttt{null}, \texttt{null}, E, A, \texttt{null}, \texttt{null}, \texttt{null}, C, \texttt{null}, D, \texttt{null}, G, I, \texttt{null}, \texttt{null}, \texttt{null} \rangle$

Call DFS(Preorder traversal)



14.8 정렬된 배열에서 높이가 최소인 이진 탐색 트리 만들기

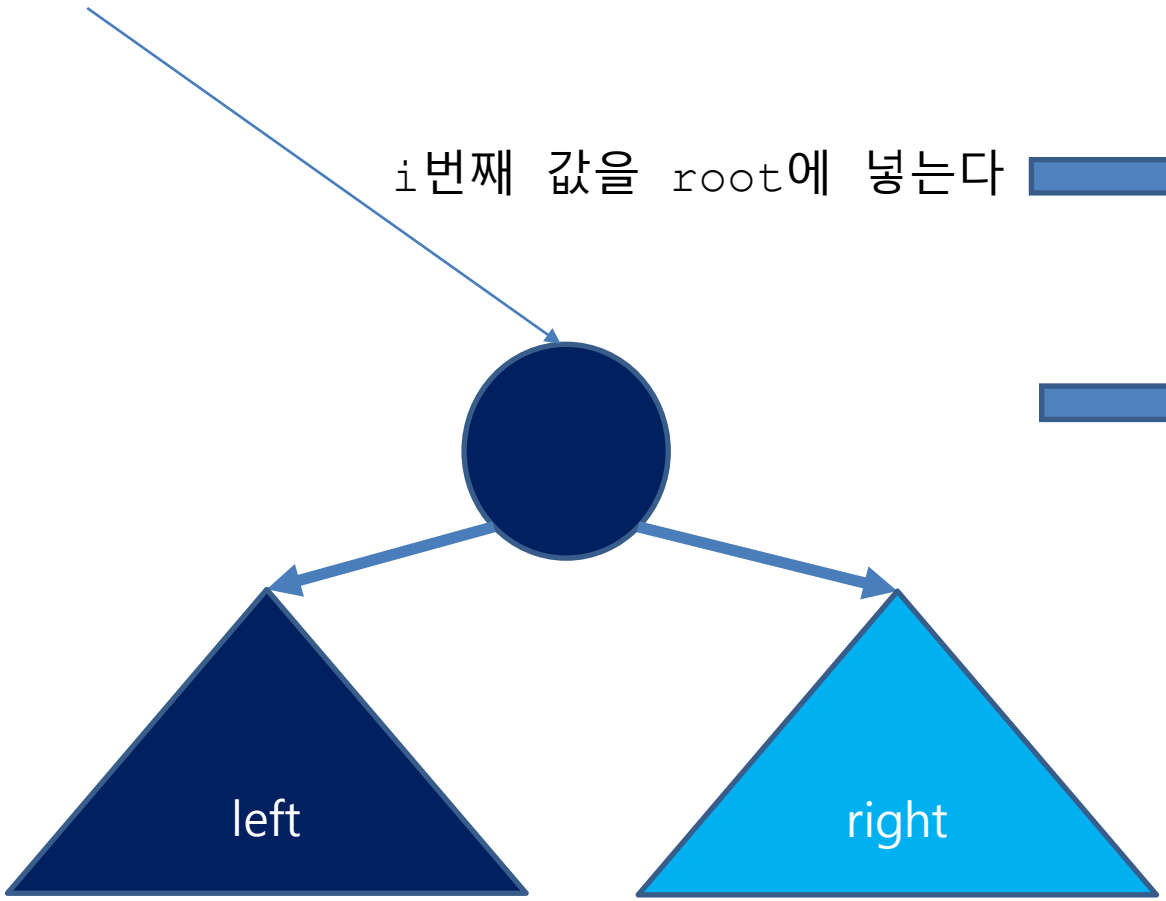
$\langle H, B, F, \texttt{null}, \texttt{null}, E, A, \texttt{null}, \texttt{null}, \texttt{null}, C, \texttt{null}, D, \texttt{null}, G, I, \texttt{null}, \texttt{null}, \texttt{null} \rangle$



Array의 i번째 값까지 TreeNode 만드는 데에 사용했다.

14.8 정렬된 배열에서 높이가 최소인 이진 탐색 트리 만들기

$\langle H, B, F, \text{null}, \text{null}, E, A, \text{null}, \text{null}, \text{null}, C, \text{null}, D, \text{null}, G, I, \text{null}, \text{null}, \text{null} \rangle$

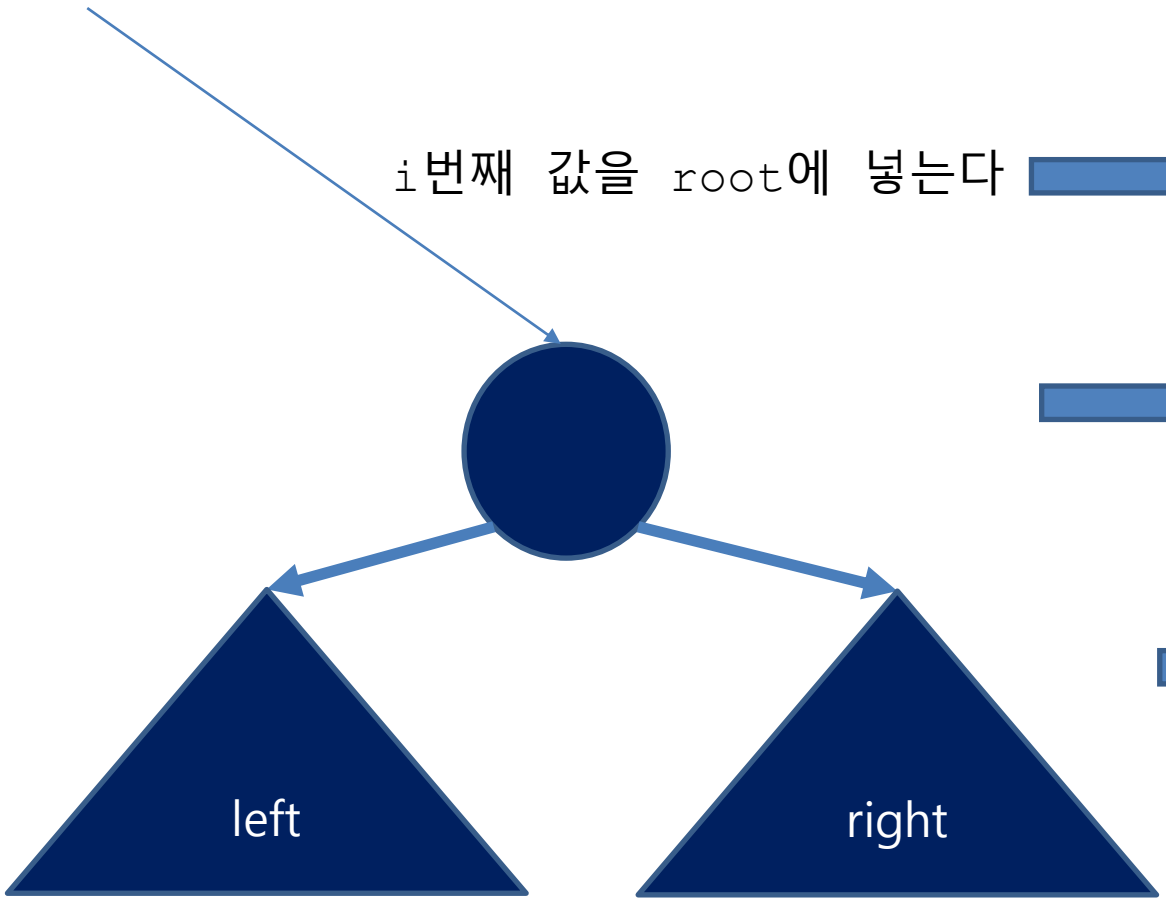


Array의 i번째 값까지 TreeNode 만드는 데에 사용했다.

Left tree를 모두 만들었을 때, l번째 index까지 활용했다.

14.8 정렬된 배열에서 높이가 최소인 이진 탐색 트리 만들기

$\langle H, B, F, \text{null}, \text{null}, E, A, \text{null}, \text{null}, \text{null}, C, \text{null}, D, \text{null}, G, I, \text{null}, \text{null}, \text{null} \rangle$



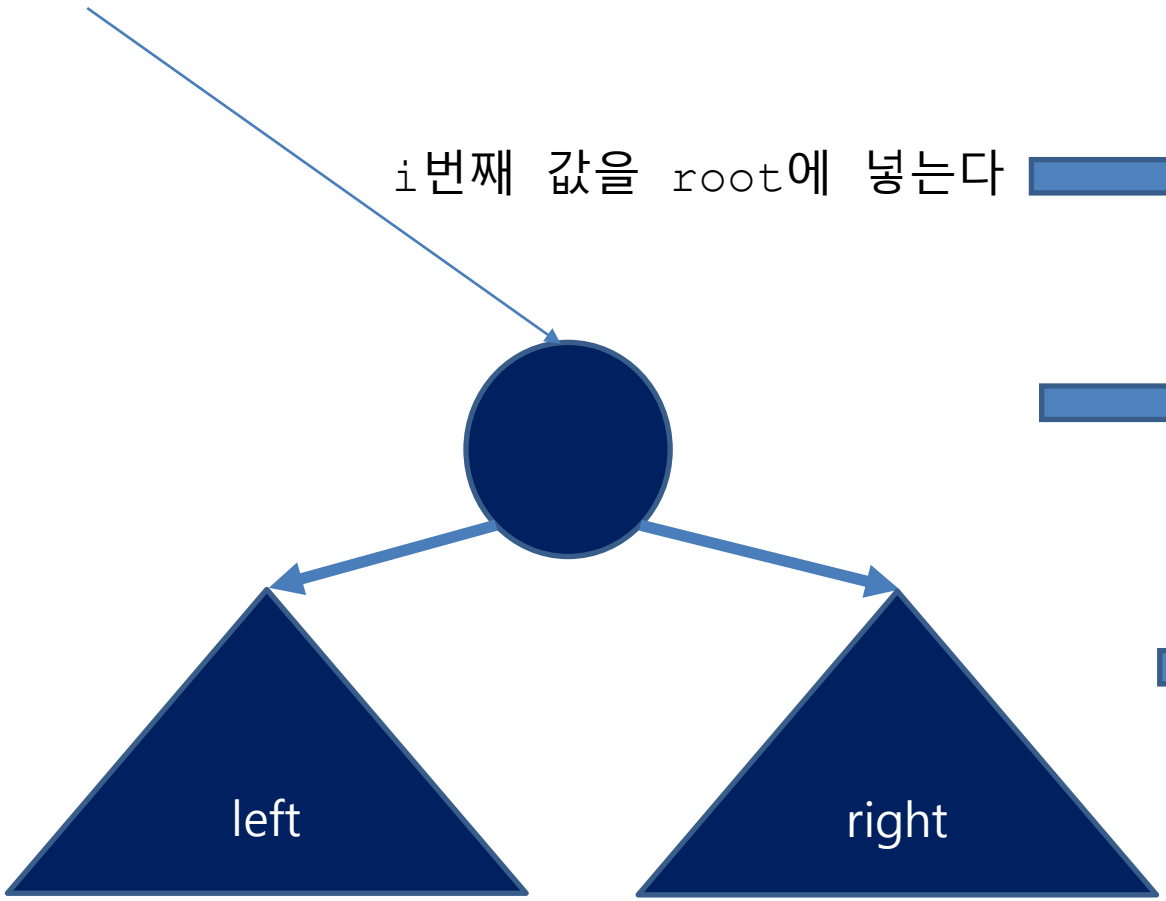
Array의 i번째 값까지 TreeNode 만드는 데에 사용했다.

Left tree를 모두 만들었을 때, l번째 index까지 활용했다.

Right tree를 모두 만들었을 때, r번째 index까지 활용했다.

14.8 정렬된 배열에서 높이가 최소인 이진 탐색 트리 만들기

$\langle H, B, F, \text{null}, \text{null}, E, A, \text{null}, \text{null}, \text{null}, C, \text{null}, D, \text{null}, G, I, \text{null}, \text{null}, \text{null} \rangle$



Array의 i번째 값까지 TreeNode 만드는 데에 사용했다.

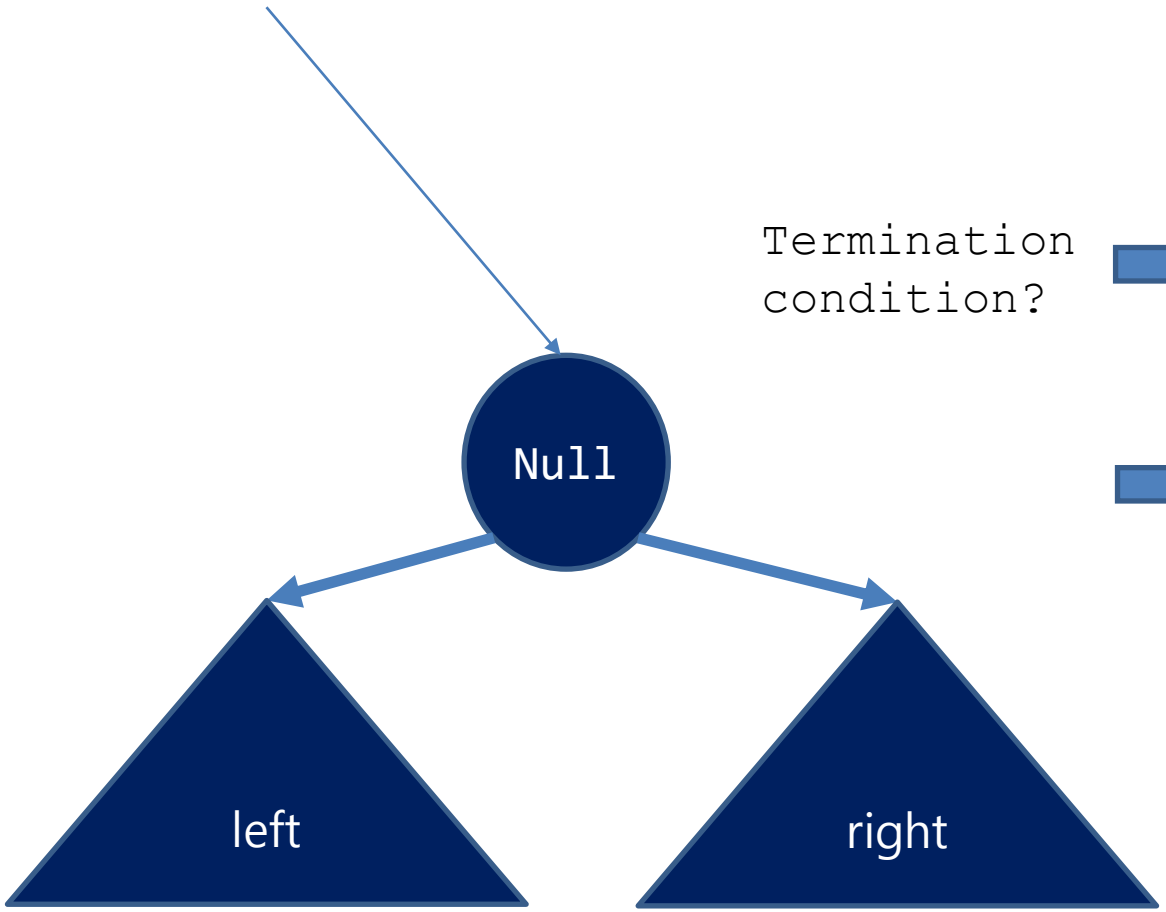
Left tree를 모두 만들었을 때, l번째 index까지 활용했다.

Right tree를 모두 만들었을 때, r번째 index까지 활용했다.

Return (root, r)
해당 트리는 r까지 활용해서 만들었다

14.8 정렬된 배열에서 높이가 최소인 이진 탐색 트리 만들기

$\langle H, B, F, \text{null}, \text{null}, E, A, \text{null}, \text{null}, \text{null}, C, \text{null}, D, \text{null}, G, I, \text{null}, \text{null}, \text{null} \rangle$



Termination condition?

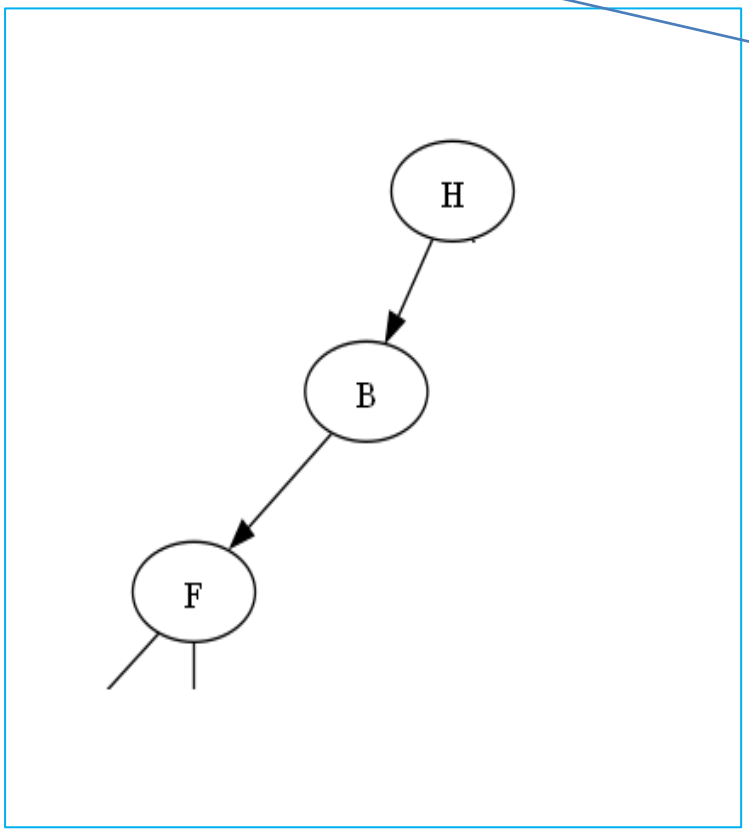
언제까지??
Null 까지.

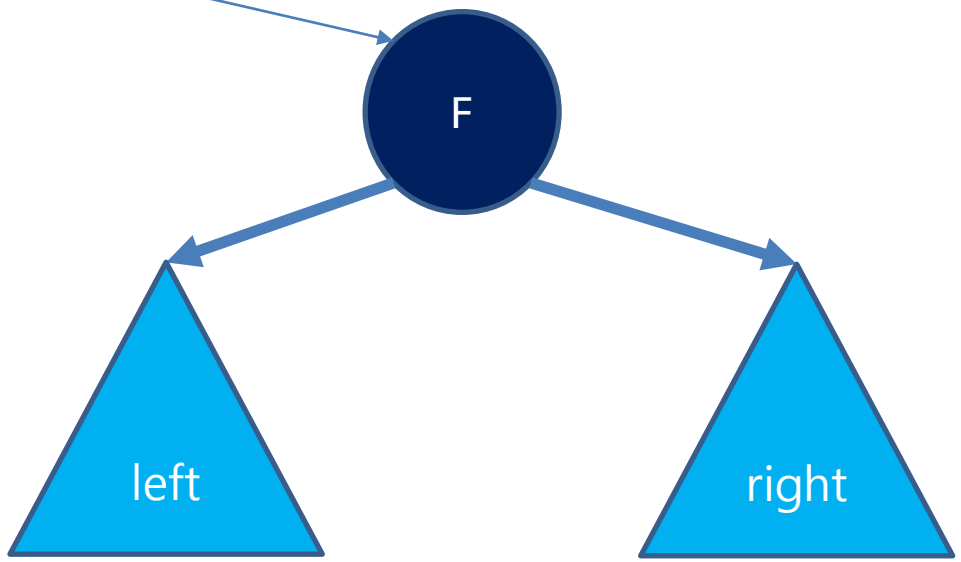
Return (Null, 3)

(3까지 이용해서 트리 만들었다.)

Example

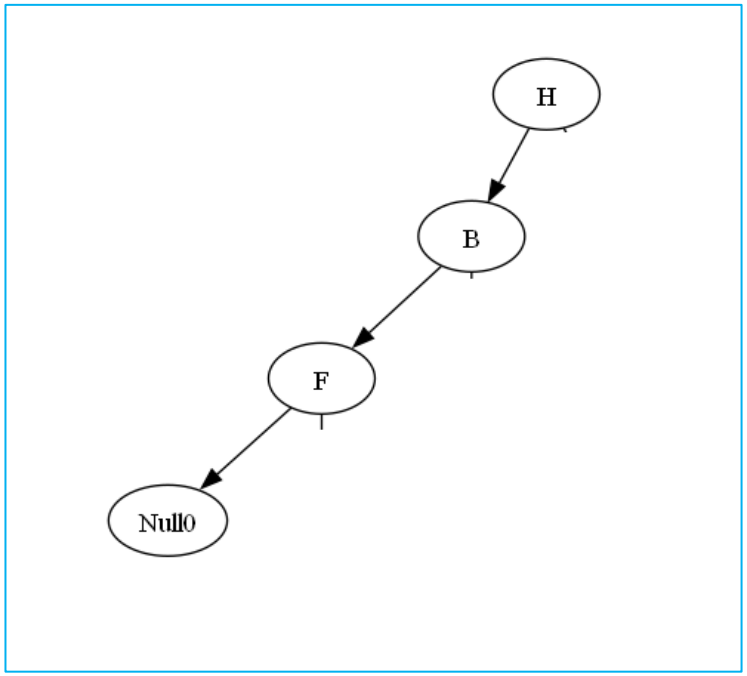
$\langle H, B, F, \texttt{null}, \texttt{null}, E, A, \texttt{null}, \texttt{null}, \texttt{null}, C, \texttt{null}, D, \texttt{null}, G, I, \texttt{null}, \texttt{null}, \texttt{null} \rangle$



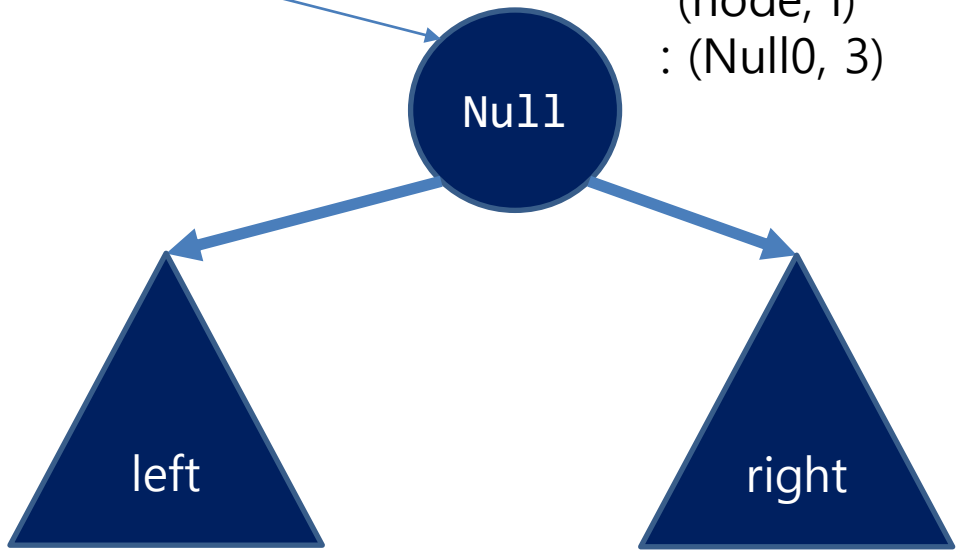


Example

$\langle H, B, F, \texttt{null}, \texttt{null}, E, A, \texttt{null}, \texttt{null}, \texttt{null}, C, \texttt{null}, D, \texttt{null}, G, I, \texttt{null}, \texttt{null}, \texttt{null} \rangle$

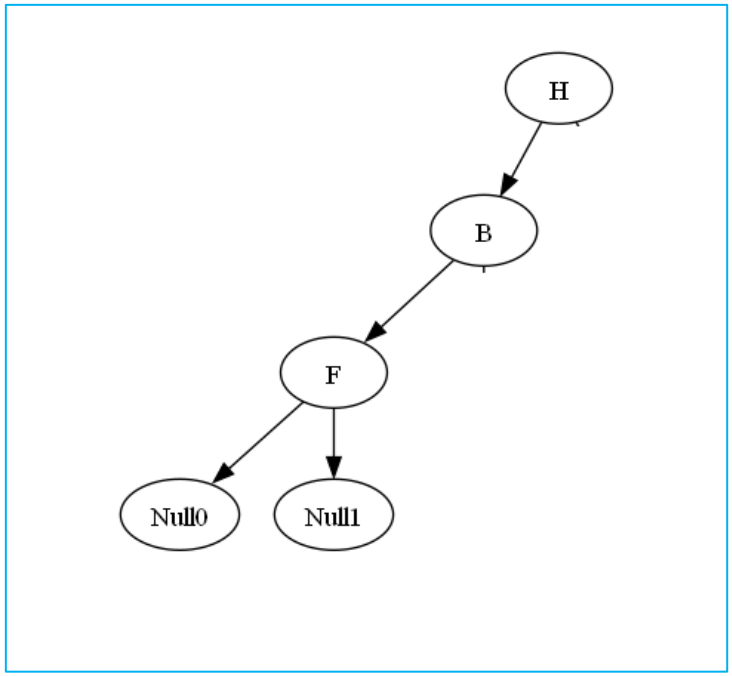


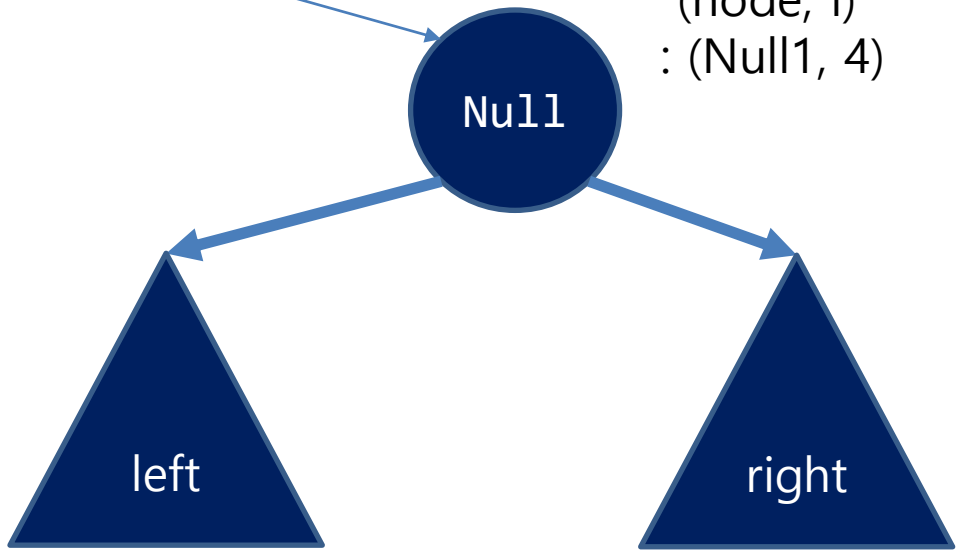
Return
(node, i)
: (Null0, 3)



Example

$\langle H, B, F, \texttt{null}, \texttt{null}, E, A, \texttt{null}, \texttt{null}, \texttt{null}, C, \texttt{null}, D, \texttt{null}, G, I, \texttt{null}, \texttt{null}, \texttt{null} \rangle$

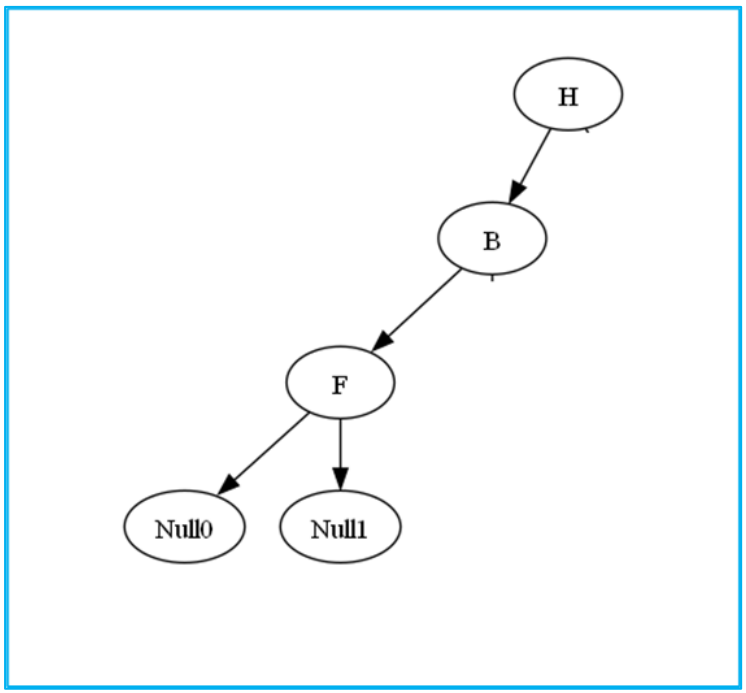


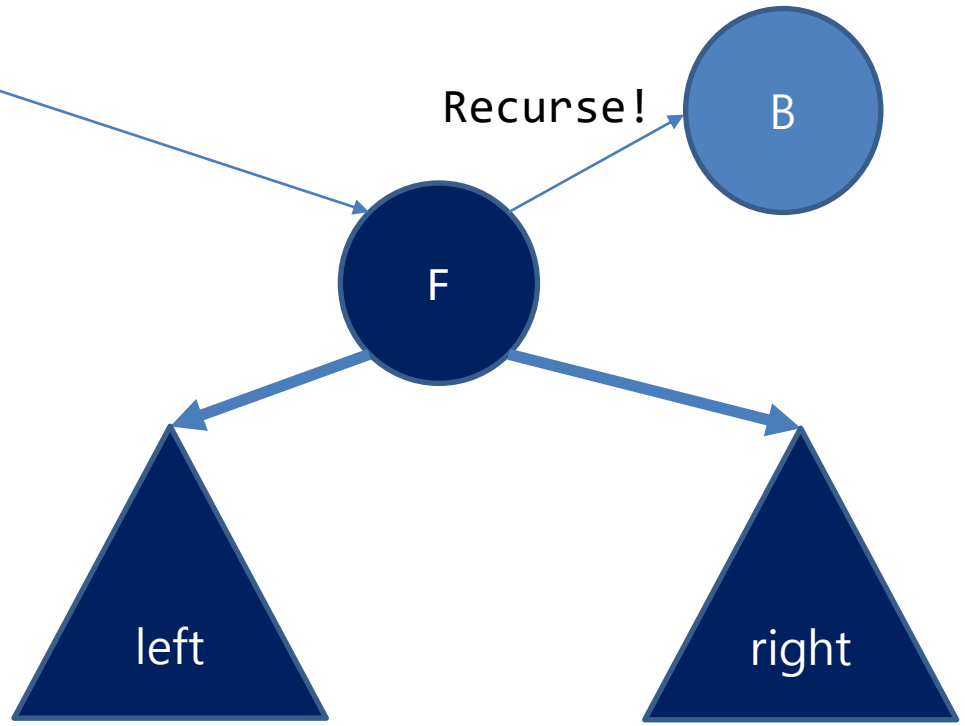


Return
(node, i)
: (Null1, 4)

Example

$\langle H, B, F, \text{null}, \text{null}, E, A, \text{null}, \text{null}, \text{null}, C, \text{null}, D, \text{null}, G, I, \text{null}, \text{null}, \text{null} \rangle$





Return
(F, 4)
해당 트리는 r(=4)까지 활용해서
트리를 만들었다.

code

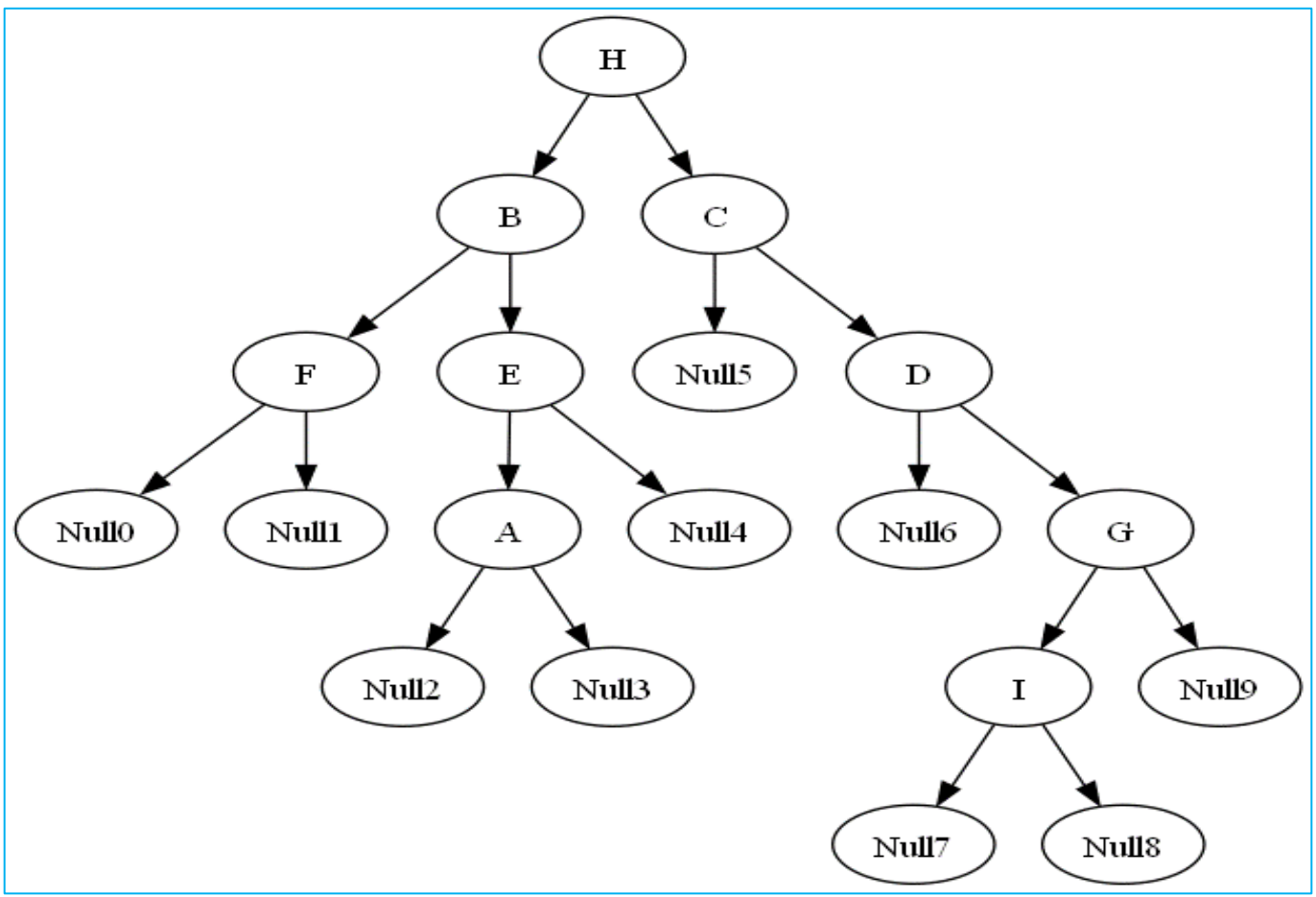
```
# Preorder : V-L-R
def dfs(i, arr):
    # 일단 i까지 잘 만들었다. (Root)
    root = TreeNode(arr[i])
    if root.key[:4] == 'Null':
        return root, i

    # left Tree -> l까지 이용해서 잘 만들었다.
    left_tree, l = dfs(i + 1, arr)
    # right Tree -> r까지 이용해서 잘 만들었다.
    right_tree, r = dfs(l + 1, arr)
    root.left = left_tree
    root.right = right_tree
    return root, r
```

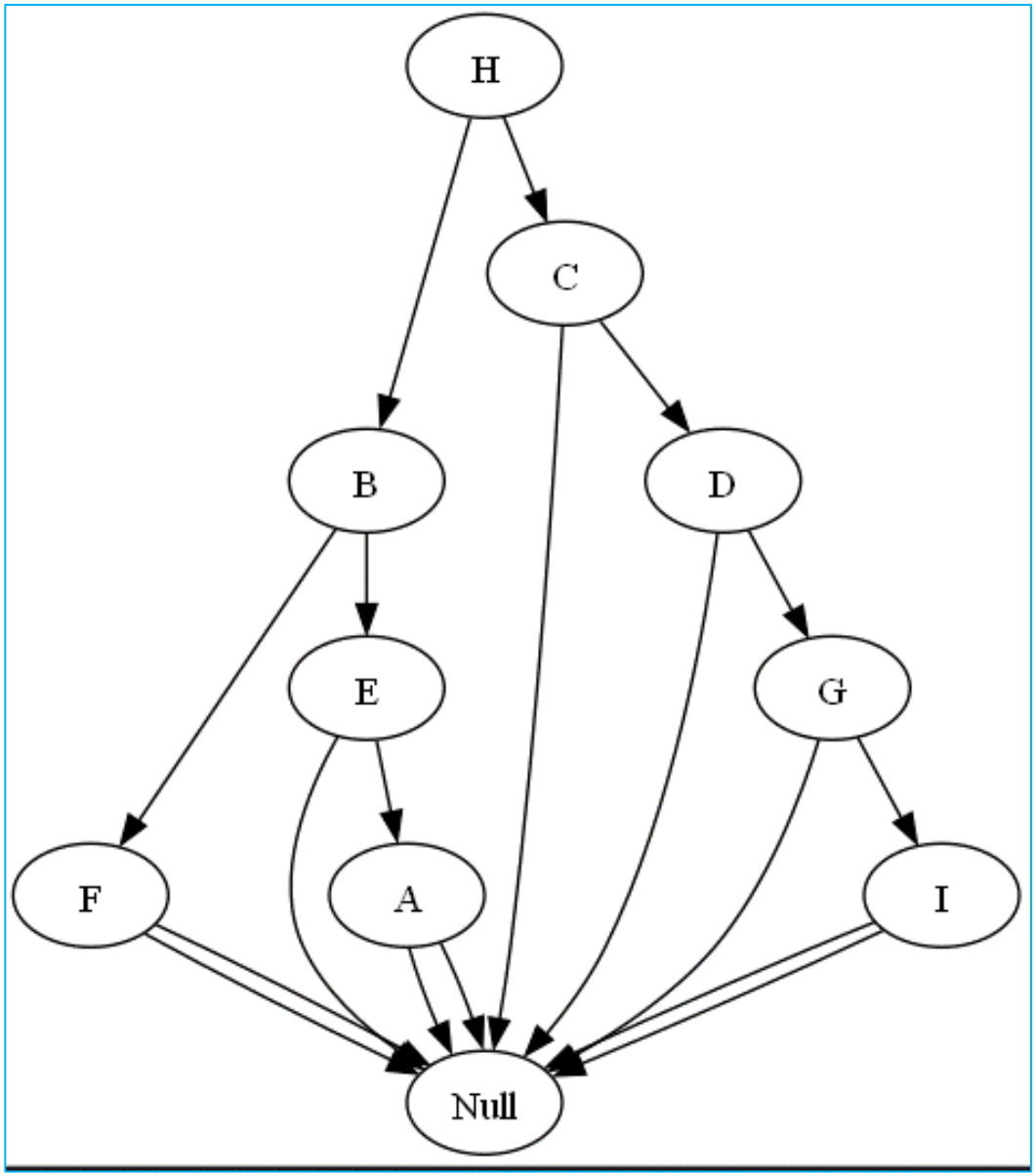
결 과

$$\langle H, B, F, \texttt{null}, \texttt{null}, E, A, \texttt{null}, \texttt{null}, \texttt{null}, C, \texttt{null}, D, \texttt{null}, G, I, \texttt{null}, \texttt{null}, \texttt{null} \rangle$$

Call DFS(Preorder traversal)



Null0, Null1, Null2 로 쓰지 않으면



Summary

- Preorder 결과가 같더라도
  - ✓ 모양은 다를 수 있다.
- Unique한 모양을 알려면
  - ✓ 추가 정보가 필요
  - ✓ Preorder, Inorder
  - ✓ Postorder, Inorder
  - ✓ Marker(Null)
    - Call DFS!(Preorder)

들어 주셔서 감사합니다