MAKE
SCHOOL

TRAVERSING TREES

TRAVERSAL

Goal - visit each node once and only once

Three operations

- **visit** current node
- **traverse** to **left** node
- **traverse** to **right** node

TWO MAIN TYPES

Depth-first

Down first - visit child, then next descendent

Breadth-first

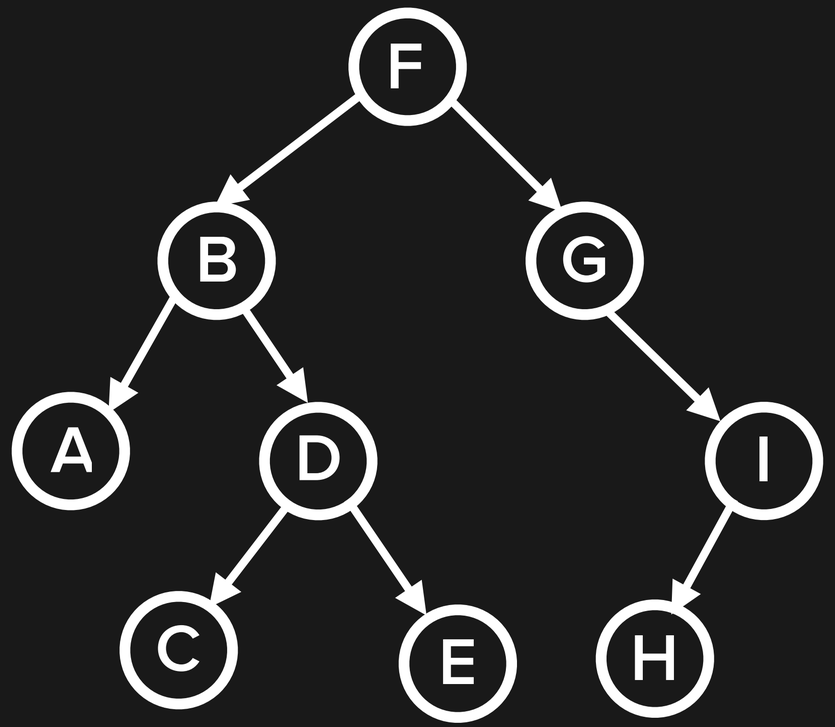
Across first - visit all siblings before going deeper

DEPTH-FIRST SEARCH

Always go left before going right

Three types of visitation

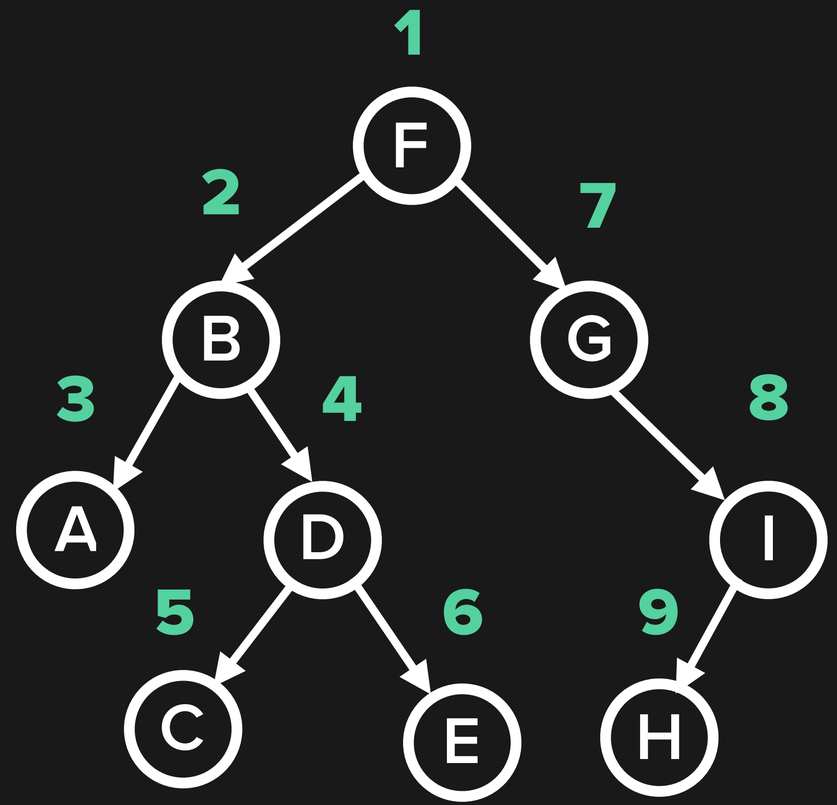
- Pre-order
- In-order
- Post-order



MAKE SCHOOL

PRE-ORDER DFS

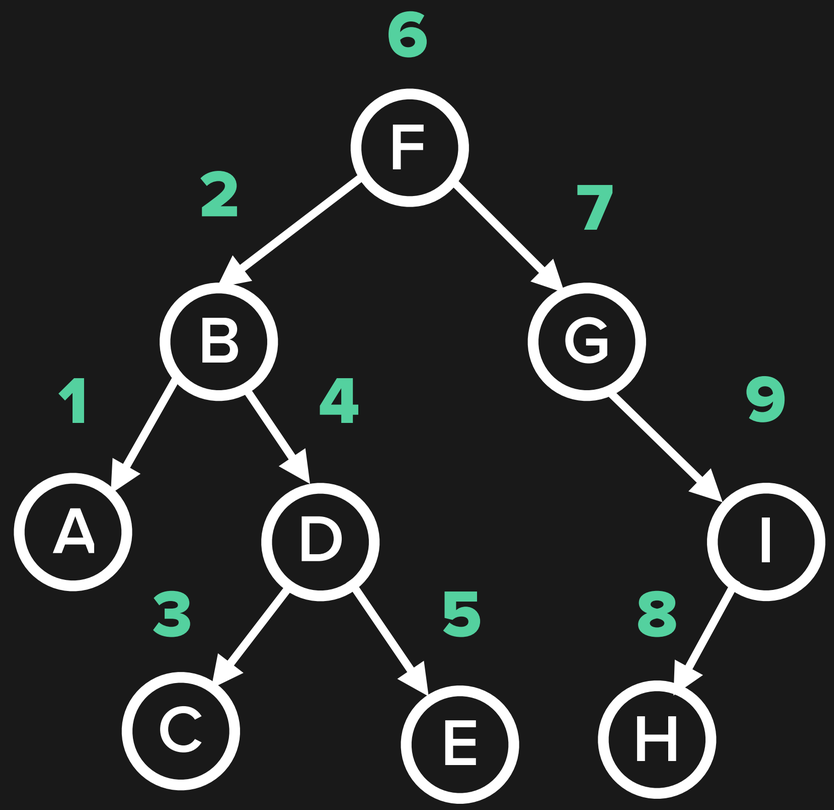
```
def pre_order_dfs(node):
    if node is not None:
        visit(node)
        pre_order_dfs(node.left)
        pre_order_dfs(node.right)
```



F B A D C E G I H

IN-ORDER DFS

```
def in_order_dfs(node):
    if node is not None:
        in_order_dfs(node.left)
        visit(node)
        in_order_dfs(node.right)
```

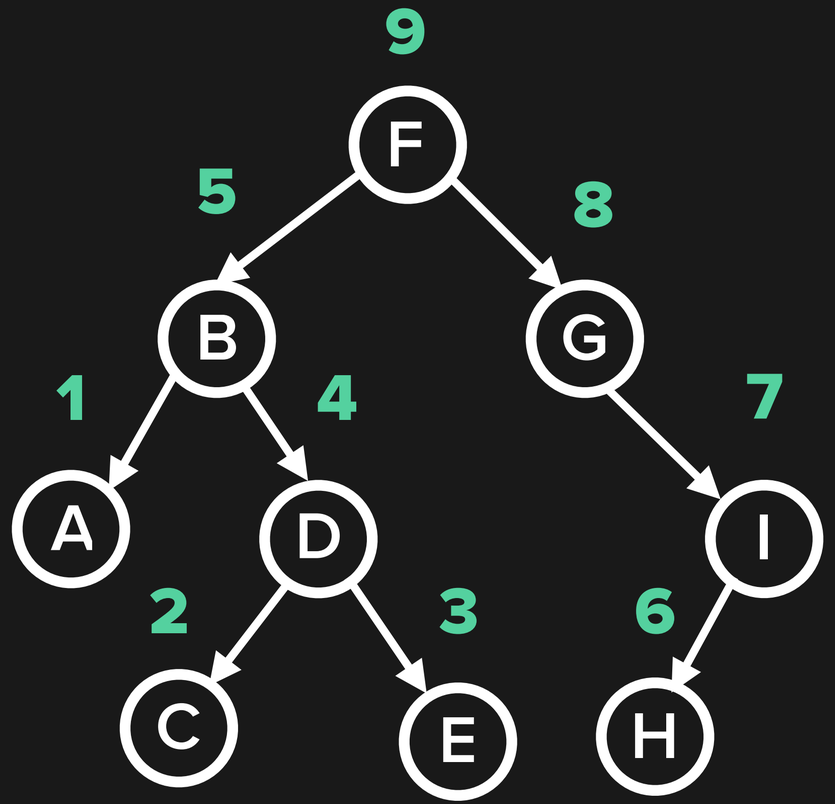


A B C D E F G H I

POST-ORDER DFS

```
def post_order_dfs(node):
    if node is not None:
        post_order_dfs(node.left)
        post_order_dfs(node.right)
        visit(node)
```

A C E D B H I G F

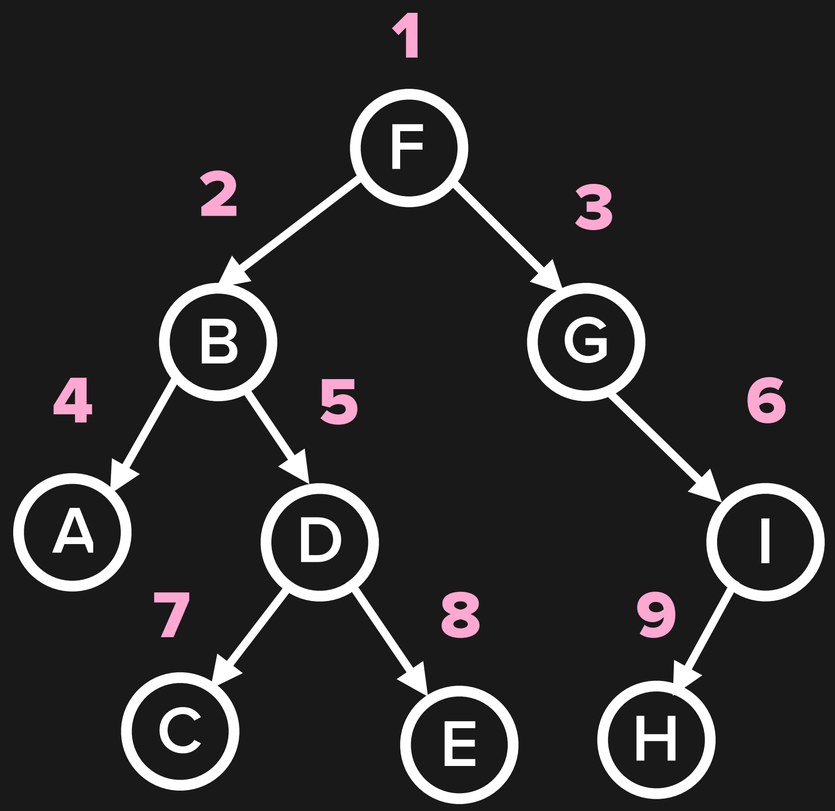


MAKE SCHOOL

BREADTH-FIRST SEARCH

```
from collections import deque

def bfs(root_node):
    queue = deque()
    queue.append(root_node)
    while len(queue) > 0:
        node = queue.popleft()
        visit(node)
        if node.left is not None:
            queue.append(node.left)
        if node.right is not None:
            queue.append(node.right)
```



F B G A D I C E H

MAKE SCHOOL

MAKE
SCHOOL