Binary Tree Traversal

CPE 103

Theresa Migler-VonDollen

Traversal

Definition

Tree traversal is a process in which each vertex in a tree is visited.

We will focus our attention on four types of traversal:

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

The first three traversals are considered *depth-first* traversals, while the last is considered a *breadth-first* traversal.

Pre-order Traversal

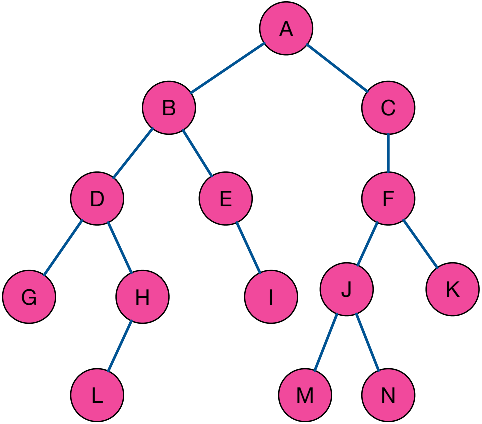
Pre-order(T, v)

Input: A tree, T, and a vertex v (initially the root).

Output: A list of the vertices in T (in pre-order).

```
if $v = null$ then
    Return
Visit $v$
Pre-order($T, v.left$)
Pre-order($T, v.right$)
```

Pre-order Traversal



Post-order Traversal

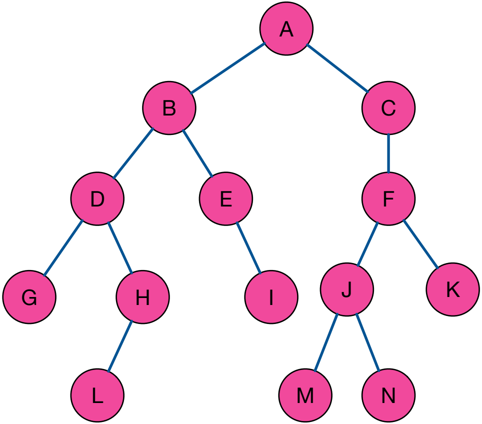
Post-order(T, v)

Input: A tree, T, and a vertex v (initially the root).

Output: A list of the vertices in T (in post-order).

```
if $v = null$ then
    Return
Post-order($T, v.left$)
Post-order($T, v.right$)
Visit $v$
```

Post-order Traversal



In-order Traversal

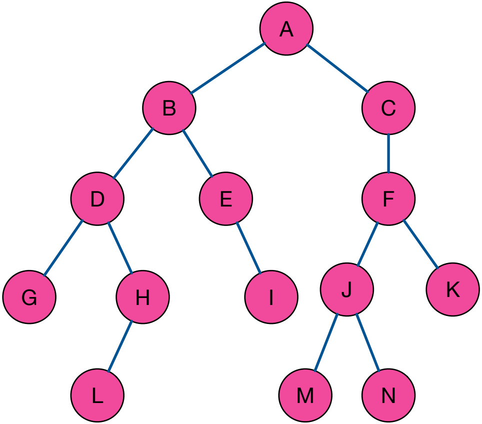
In-order(T, v)

Input: A tree, T, and a vertex v (initially the root).
Output: A list of the vertices in T (in in-order).

```
if $v = null$ then
    Return
In-order($T, v.left$)
Visit $v$
In-order($T, v.right$)
```

For a binary search tree, in-order traversal gives vertices in non-decreasing order.

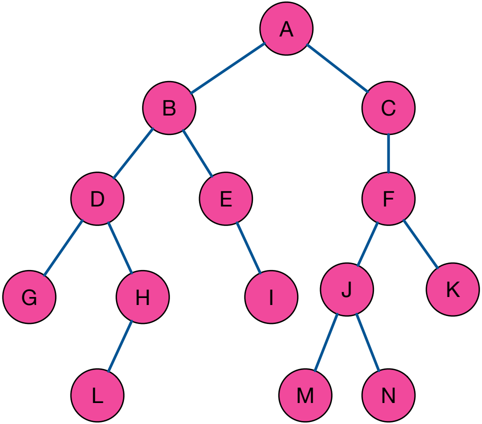
In-order Traversal



Level-order Traversal

```
Level-order($T, v$)
Input: A tree, $T$, and a vertex $v$ (initially the root).
Output: A list of the vertices in $T$ (in level-order).
Let $Q$ be an empty queue.
$Q.enqueue(v)$
while $Q \neq \emptyset$ do
    $x = Q.dequeue$
    Visit $x$
    if $x.left \neq null$ then
        $Q.enqueue(x.left)$
    if $x.right \neq null$ then
        $Q.exqueue(x.right)$
```

Level-order Traversal



Time complexity

What is the time complexity for the first three traversal methods?

- Each function is called once for each vertex.
- $O(n)$

What is the time complexity for level-order traversal?

- Each vertex is pushed onto the queue once and popped off the queue once.
- $O(n)$

Reconstruction

Suppose you had a graph with 14 vertices, $A, B, \ldots, N$. You wrote down the pre and post orderings for the vertices:
Preorder: $A, B, D, E, H, L, M, I, F, C, G, J, K, N$
Postorder: $D, L, M, H, I, E, F, B, J, N, K, G, C, A$
Then you lost your graph. Can you reconstruct the graph?