TIP102 | Intermediate Technical Interview Prep

Intermediate Technical Interview Prep Summer 2025 (@ Section 4b | Saturdays and Sundays 10AM - 12PM PDT)
Personal Member ID#: **115310**

Unit 9 Cheatsheet

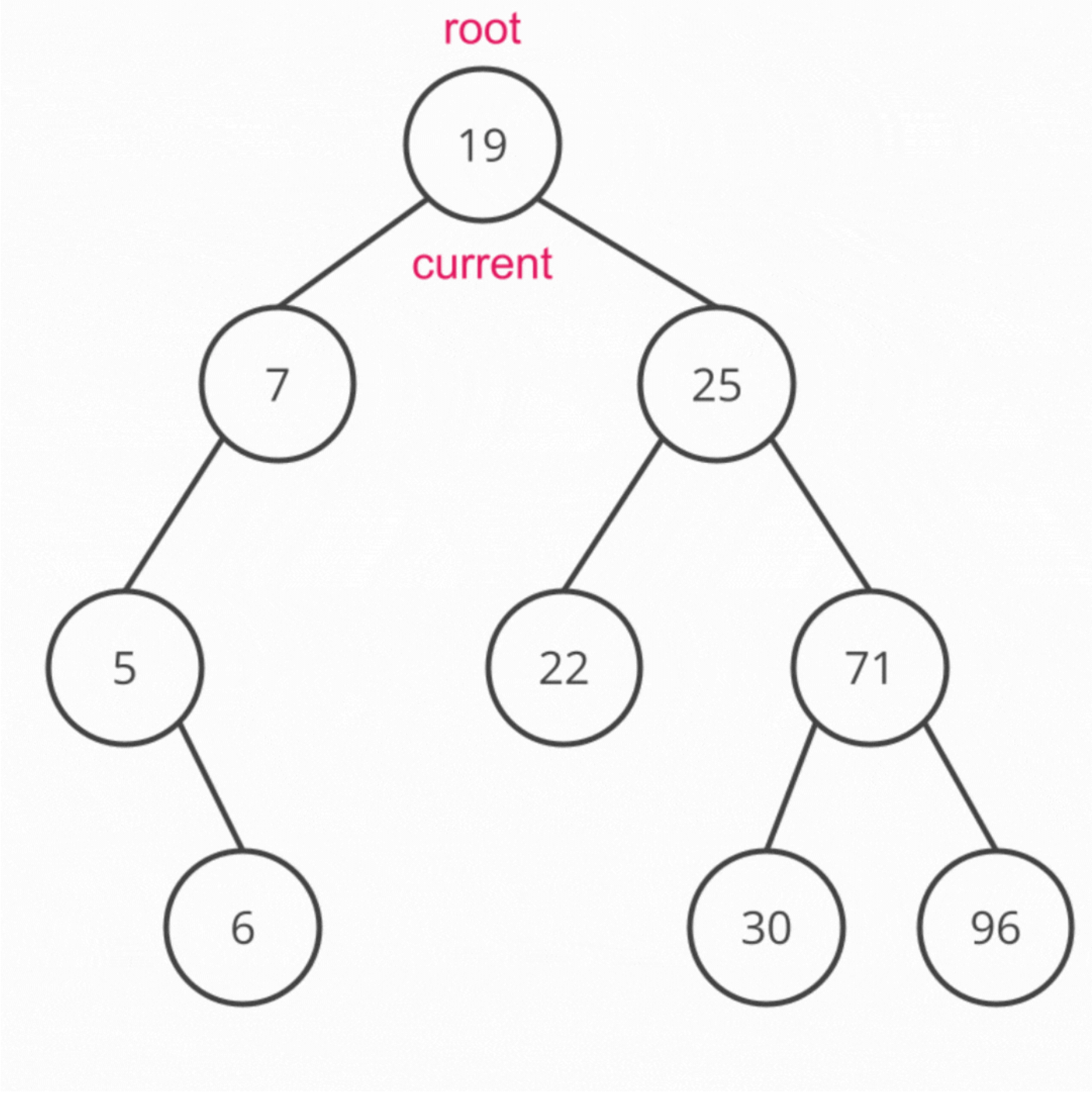
Overview

Here is a helpful cheatsheet outlining common syntax and concepts that will help you in your problem solving journey! Use this as reference as you solve the breakout problems for Unit 9. This is not an exhaustive list of all data structures, algorithmic techniques, and syntax you may encounter; it only covers the most critical concepts needed to ace Unit 9. In addition to the material below, you will expected to know any required concepts from previous units.

Standard Concepts

Breadth First Search

Breadth First Search (BFS), also known as Level Order Traversal, is a method for visiting all the nodes in a tree. In a breadth first search approach, we visit nodes level by level. We begin by traversing the tree's root node, then traversing the root's direct children from left to right, followed by the root's grandchildren, etc.



In the diagram above, nodes that are outlined in pink have been added to the queue. Nodes shaded in pink have been visited and removed from the queue. The root node, at level 1, is visited first. Then the root node's children at level 2, nodes 7 and 25, are visited. The pattern continues until the nodes have been explored in the following order: `[19, 7, 25, 5, 22, 71, 6, 30, 96]`.

BFS is typically implemented iteratively using a queue. The pseudocode for a Breadth First Search is as follows:

```
If the tree is empty:
    return an empty list

Create an empty queue
Create an empty list to store visited nodes

Add the root into the queue

While the queue is not empty:
    Pop the next node off the queue
    Add the popped node to the list of visited nodes

    Add the popped node's left child to the queue
    Add the popped node's right child to the queue
```

BFS can also be implemented recursively, but an iterative, queue based implementation is generally preferred because the order in which BFS visits nodes in a tree matches the FIFO insertion/removal order of a queue.

How to Pick a Traversal Method

There are four standard traversal algorithms for a binary tree. The first three - preorder, inorder, and postorder - are all depth first search traversals. The final algorithm is a breadth first search traversal.

For many problems, any traversal algorithm will lead to a solution. However, there are certain cases where a particular algorithm is preferred.

Depth First Search

In general, depth first search algorithms are preferred when the solution is expected to be deeper within the tree since the algorithm follows one branch as far as possible before backtracking and exploring other paths. In these scenarios, a breadth first approach may still find a solution, but more slowly since it traverses nodes closest to the root first.

Inorder traversals are commonly used for finding leaves, the height of the tree, or the diameter of the tree.

Inorder

Given a binary search tree, inorder will traverse the nodes in sorted ascending order.

Inorder traversals are commonly used for binary search tree tasks or converting a binary search tree to a sorted list.

Preorder

Given a binary tree, preorder will process the root of the tree before either subtree. It also processes nodes in the order they were inserted into the tree.

Preorder traversals are commonly used for tree copying, expression tree evaluation, and serializing a tree.

Postorder

Given a binary tree, postorder will process the subtrees before the root.

Postorder traversals are commonly used for deleting a tree and expression tree evaluation.

Breadth First Search

Given a binary tree, breadth first search traverses nodes higher up in the tree (closest to the root) first. It is preferred when you expect the solution to be closer to the root. It also explores nodes level by level, from left to right.

Breadth first search is commonly used for problems that require traversing by level.