Trees, Generics, and Collections

Searching a Sorted Array

- Linear Search
  - Iterate through array checking each element
  - O(N)
- Binary Search
  - Check the middle element
  - If it is the desired element, return with that index
  - If the element is bigger recurse on the right half
  - If the element is smaller recurse on the left half
  - O(log(n))

[2,3,5,7,19,32]

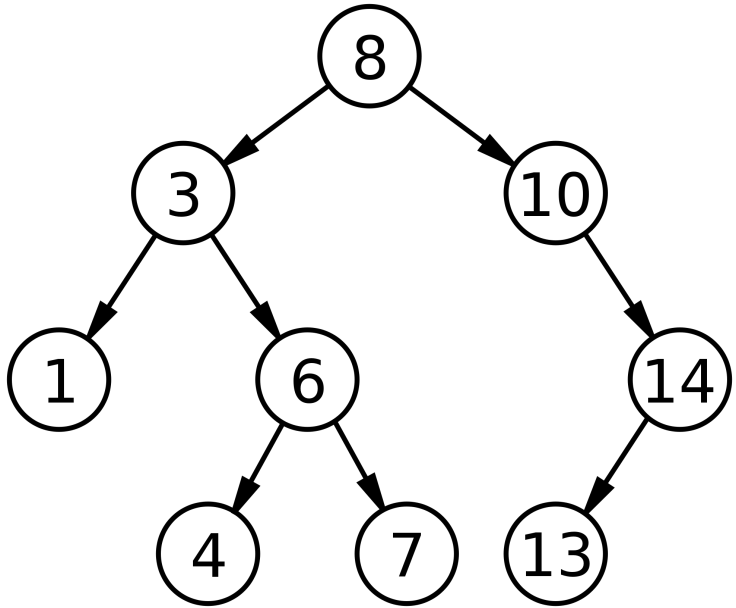
Find index of 3

Tree Terminology

- Parent and Child
  - If a node A contains a reference to another node B in the tree then A is B's parent and B is A's child
- Descendent
  - B is A's descendent if B is A's child of its child's child or its child's child's child and so on
- Root
  - The node at the top of the tree
  - Every node in a tree is either the root or a descendent of the root
- Leaf
  - Node with no children
- Height
  - Length of the longest path from root to leaf

Binary Search Tree

- A Node contains an integer value and two child nodes
- Given any node, every node in the right subtree is smaller and every node in left subtree is bigger
- Searching is done in O(h) where h is the max height of the tree



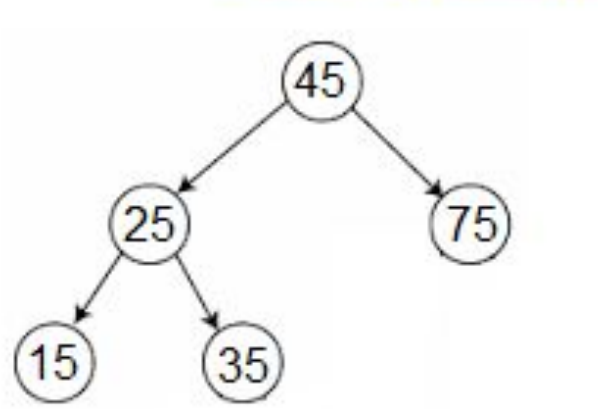
Traversals

- Process of visiting every node in the tree
- Three main types of traversals:
  - Pre-order: parent node, left node, right node
  - In-order: left node, parent node, right node
  - Post-order: left node, right node, parent node

Traversals

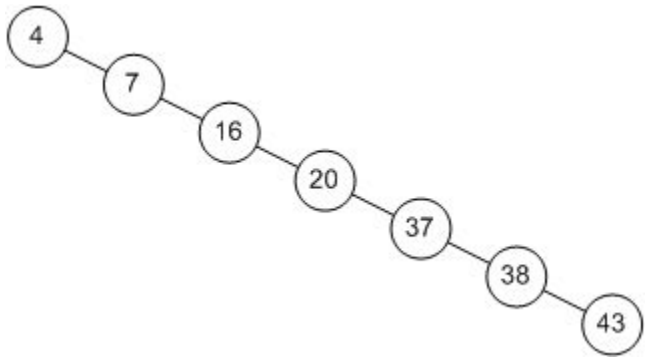
in-order traversal result:
15, 25, 35, 45, 75

Exercise: What do you get when you do a pre-order and post-order traversal?



Search Runtime Revisited

- What if we add the elements of a binary tree in order
- Can be as bad as a linked list



Balanced Binary Trees

- Maintain some invariant on the balancing of the tree and restructure the tree when it becomes too unbalanced
- Guaranteed O(log(n)) search
- AVL Tree
  - Height of subtrees varies by at most 1
- Red-Black Tree
  - Height of subtrees must be within a factor of 2 of each other
- 2-3 Tree
  - Allows for special 3 nodes with two values and three subtrees
  - All subtrees are of the exact same height

Uses For Balanced Binary Trees

- Sets
  - O(log(n)) adding and searching
  - Can easily iterate in order (with respect to the comparison)
- Maps
  - O(log(n)) adding and searching
  - Can easily iterate over keys in order (with respect to the comparison)

General Trees

- Not all trees are binary search trees!
- We can allow nodes to have arbitrarily many children
- Nodes do not need to follow the binary search
- In fact trees are great for representing programs (You will learn about Abstract Syntax Trees next week!!)

Generic Binary Trees

- We can abstract a Binary Search tree to work for any value type