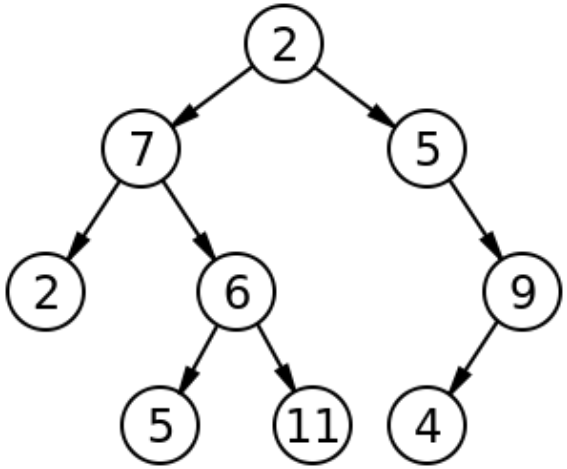
CSE 303X

ACCELERATED ADVANCED JAVA PROGRAMMING I/II/III

Binary Tree

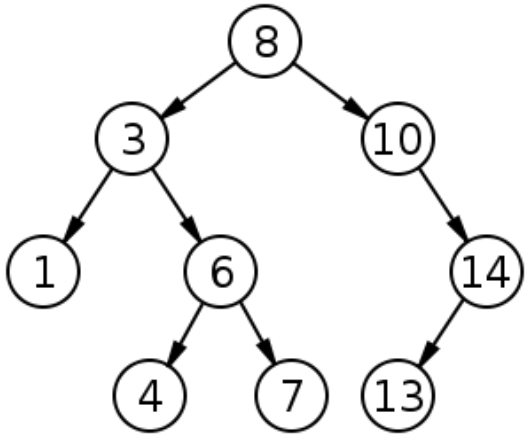
Binary Tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.



Binary Tree

Pre-order traversal

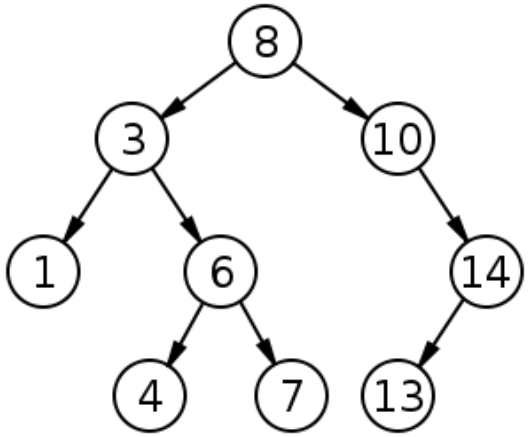
parent, left, right
8, 3, 1, 6, 4, 7, 10, 14, 13



Binary Tree

In-order traversal

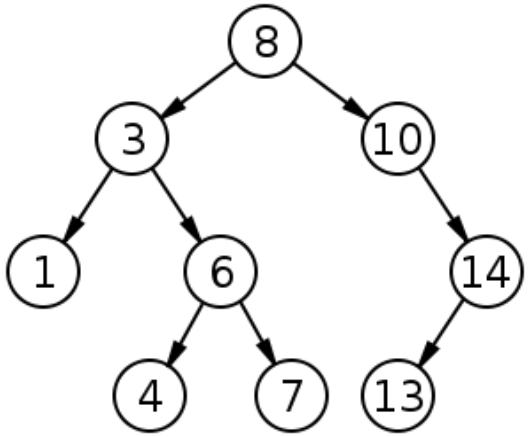
left, parent, right
1, 3, 4, 6, 7, 8, 10, 13, 14



Binary Tree

Post-order traversal

left, right, parent
1, 4, 7, 6, 3, 13, 14, 10, 8



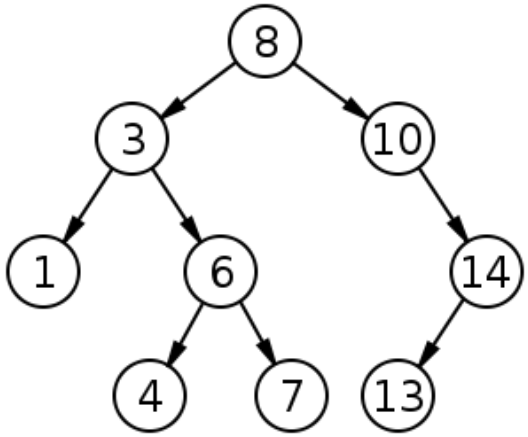
Binary Tree

Depth-first Search

Pre-order, in-order, post-order

Implemented via stack

8, 3, 1, 6, 4, 7, 10, 14, 13



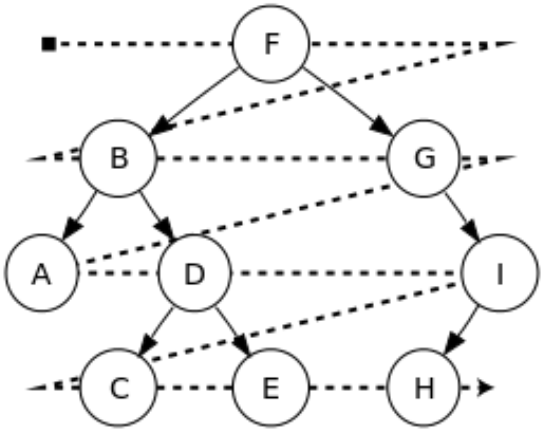
Binary Tree

Breadth-first Search

Implemented via queue

Search level by level, layer by layer

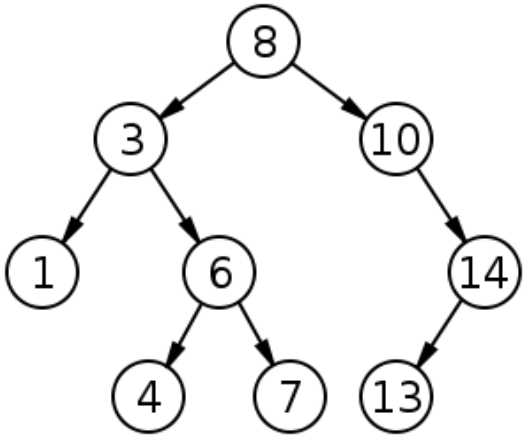
F, B, G, A, D, I, C, E, H



Binary Search Tree

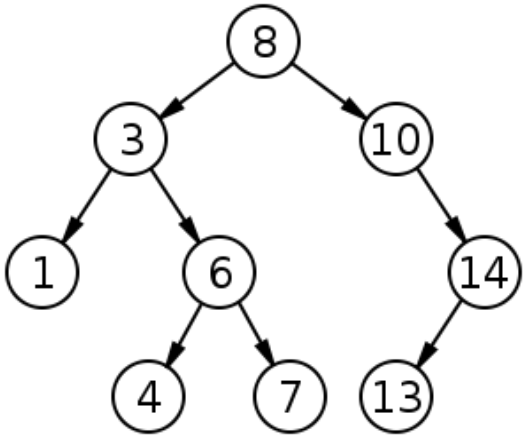
Binary Search Tree is a tree data structure that keeps their keys in sorted order, so that lookup and other operations can use the principle of binary search.

Left child node is always less than the parent node.
Right child node is always more than the parent node.



Binary Search Tree

Minimum key is at the very left (1) and Maximum key is at the very right (14).

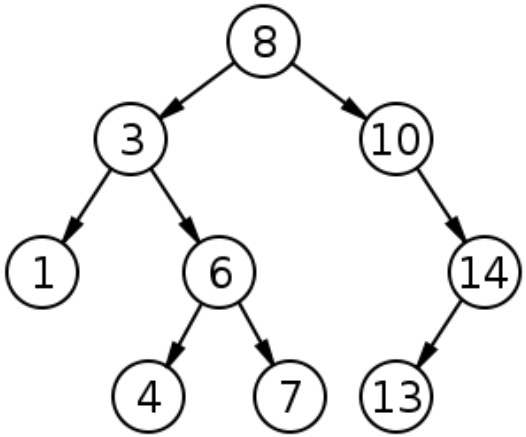


Binary Search Tree

The second min key is

at the very left of the min key's right subtree

If the right subtree of the min key does not exist, then the second min key is the parent of the min key

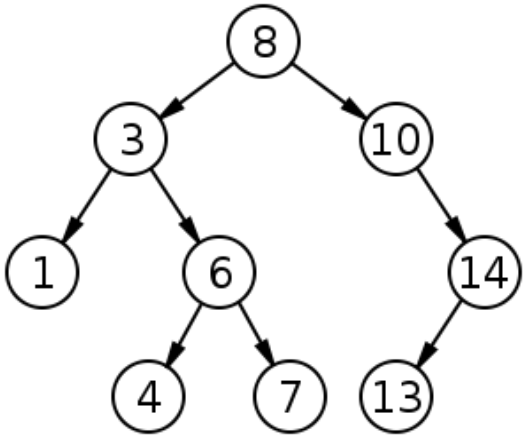


Binary Search Tree

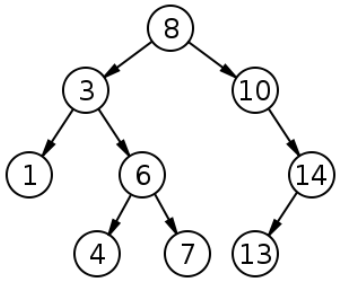
The second max key is

at the very right of the max key's left subtree

If the left subtree of the max key does not exist, then the second max key is the parent of the max key



Binary Search Tree



Find the Nth min key:

```
stk <- new stack
push all left nodes including root -> stk   (in this example,
                                              they are 8, 3, 1)

while (stk is not empty and n > 0):
        tmp <- stk.pop()
        n = n – 1
        if tmp's right child is not null:
                push all left nodes of tmp including tmp -> stk
return tmp's key
```