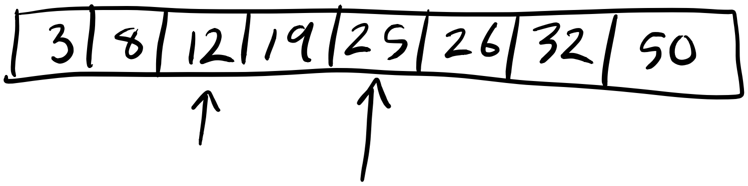
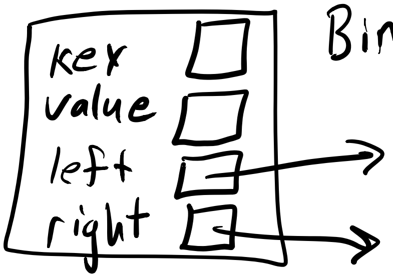
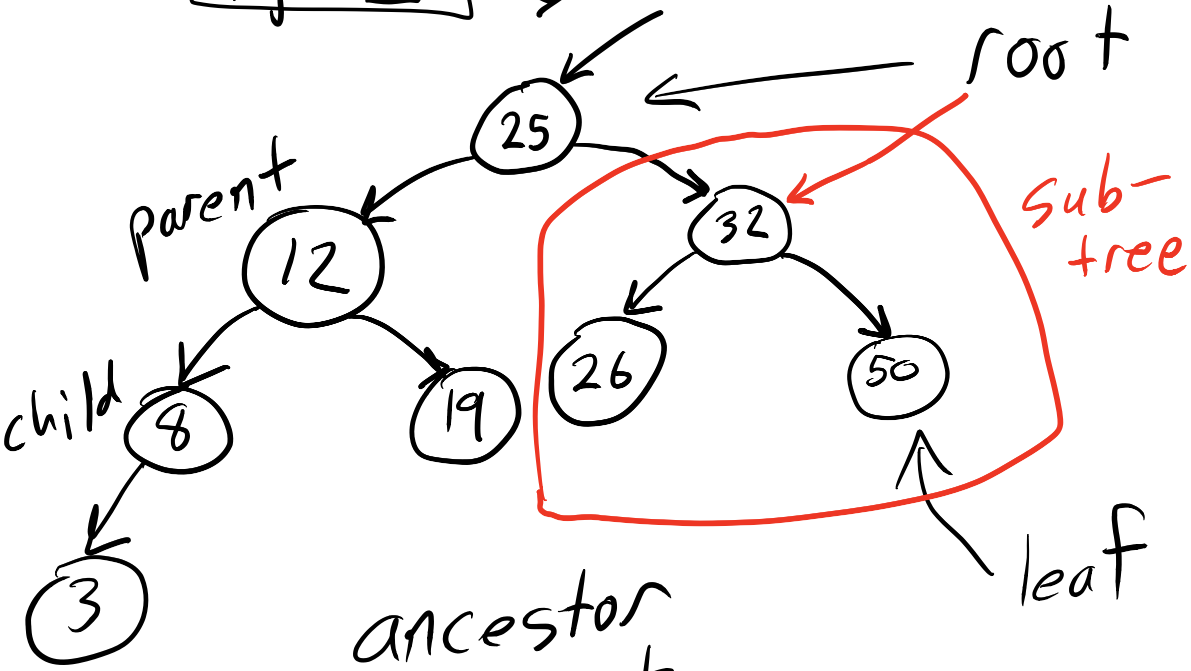


20

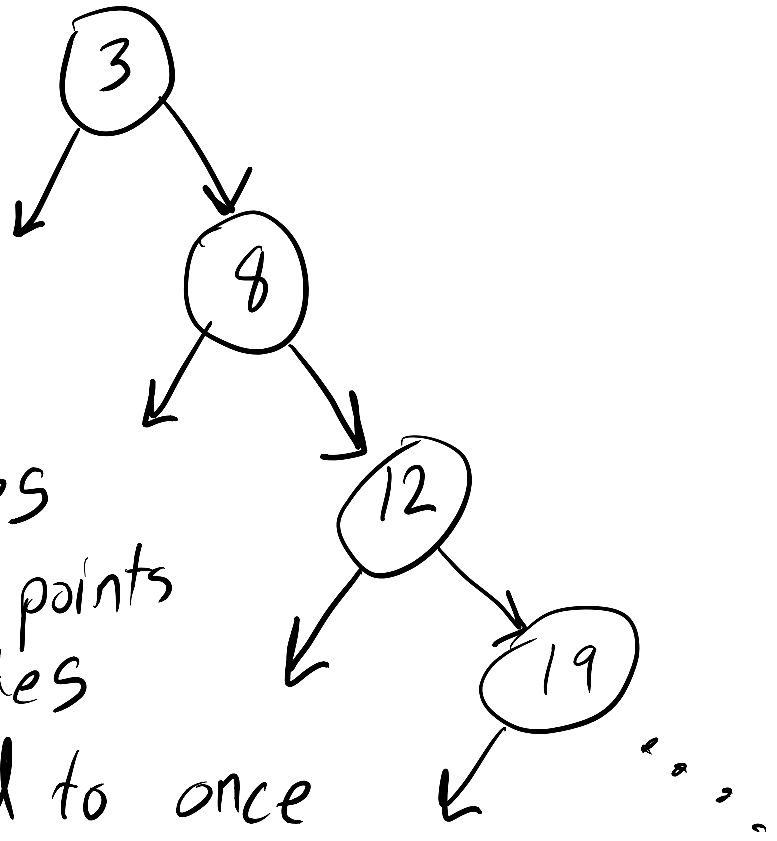
$\log_2(1,000,000,000) \approx 30$



Binary Search Tree Node



ancestor
descendant



Tree

made of nodes

- each node points to other nodes
- only pointed to once

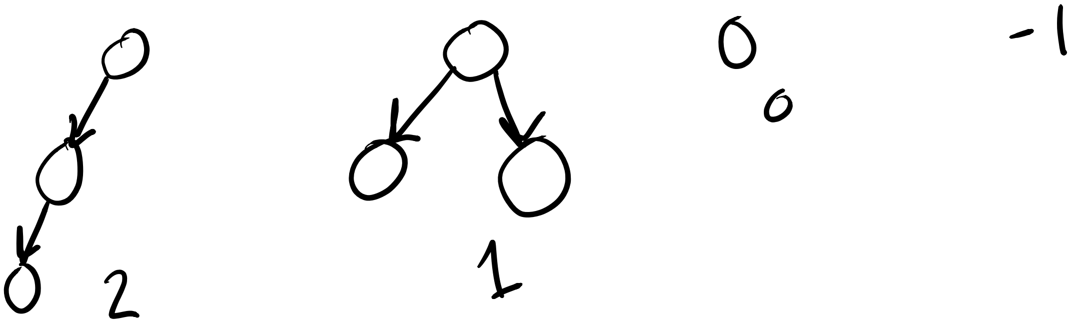
Binary Tree

- each node has at most two children
  - left child
  - right child

Binary Search Tree

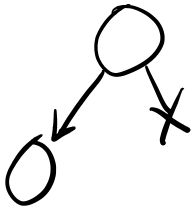
invariant: at every node
all left-descendants are smaller
all right-descendants are larger

height: max number of edges from the root to any leaf



```
value Method get (key)
    return find In Subtree(root, key)
end Method

Method find In Subtree(node, key)
    if node == null
        throw exception ☹
    if node.key == key
        return node.value
    if node.key < key
        return find In Subtree(node.right, key)
    if node.key > key
        return find In Subtree(node.left, key)
```



```
method insert (key, value)
    insertInSubtree(root, key, value)
```