Data Structures

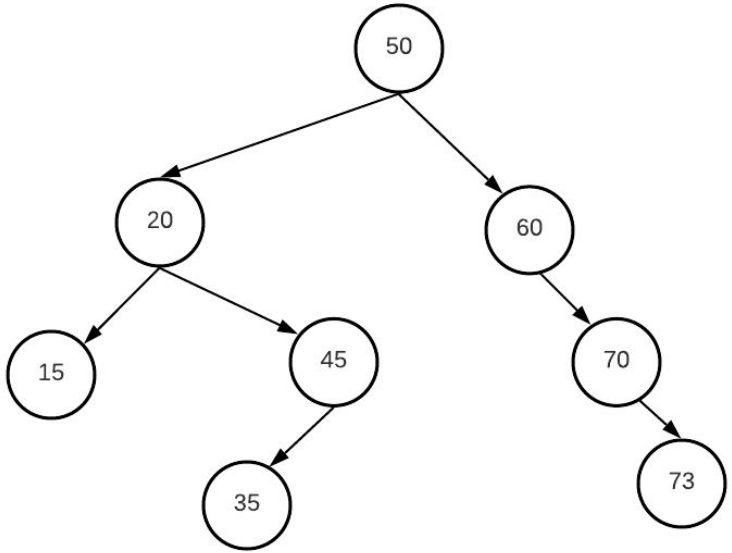
Minimum & Successor 1

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Find the minimum of BST

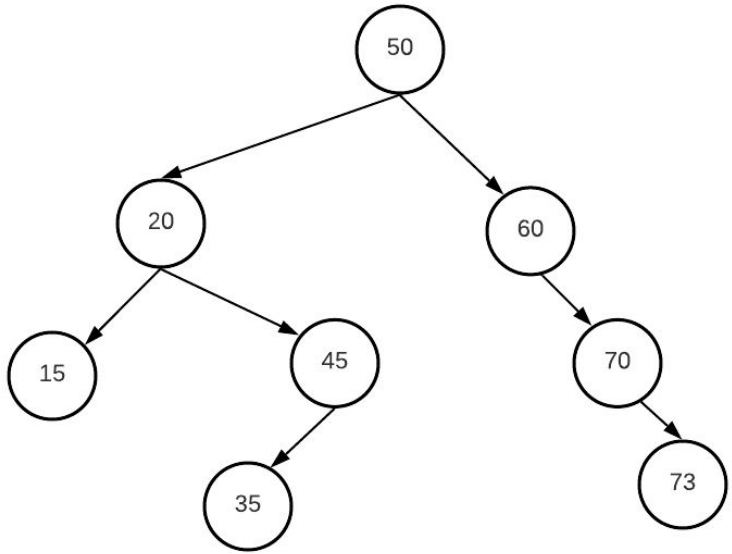
- In O(h), find the min of a tree
- Simply keep going to the left until there is no more left node!
- **Why?**
- For every node, the minimum of its 2 children is in the **left** subtree




```
def min(self):
    cur = self.root
    while cur and cur.left:
        cur = cur.left
    return cur.val
```

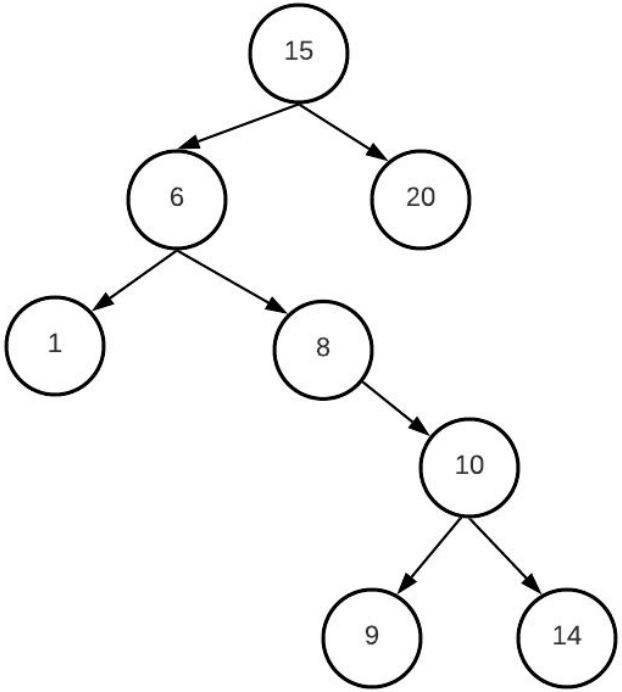
Observations

- The minimum node will **never** have a left child
  - Otherwise, it cannot be the minimum in a BST!
  - Similarly, max node don’t have a right child
- In a chain of left nodes, every node will be smaller than ALL previous values [decreasing]
  - Chain: [50 ⇒ 20 ⇒ 15]
  - Chain: [45 ⇒ 35]
- Similar logic applies in a chain of right nodes
  - Chain: [50 ⇒ 60 ⇒ 70 ⇒ 73]
  - E.g. 73 is greater than all previous values



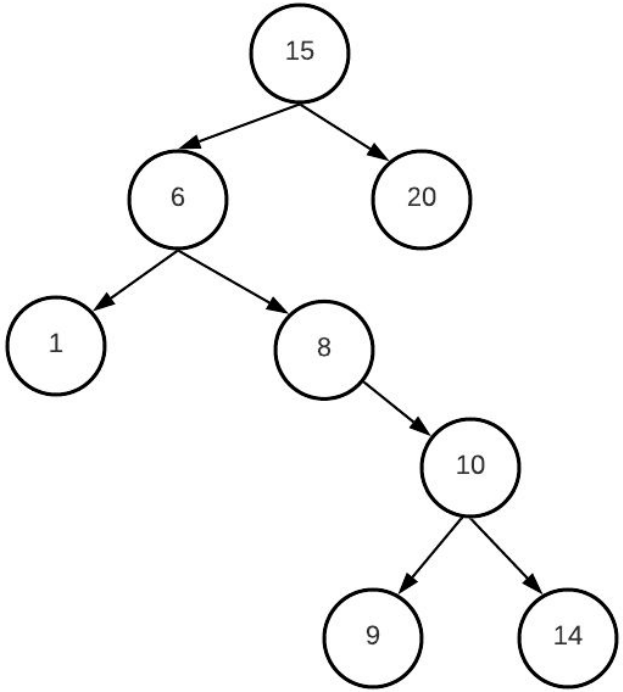
Inorder **Successor** in a Binary Search Tree

- Given node x, find node y that is the smallest y > x [in O(h)]
  - Inorder: 1 6 8 9 10 14 15 20, but O(n)
- Let's first find the node
- Where are greater values than me?
  - Any value in the right subtree will be greater, but is not necessarily the successor!
  - Or maybe somewhere between me and parent?!
  - Or maybe right of some parent?!
- But we only seek the smallest y > x?
  - There are only 2 cases then!
  - Think more about your right child!



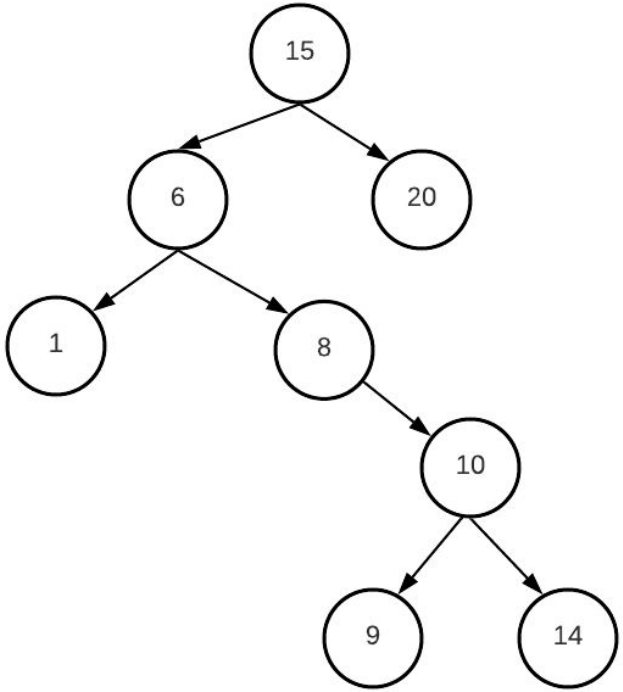
Inorder **Successor** in Binary Search Tree

- Consider x = Node(8)
- It has a right subtree
  - We have values: [9, 10, 14], all must be > 8
  - Answer is min(right) = 9



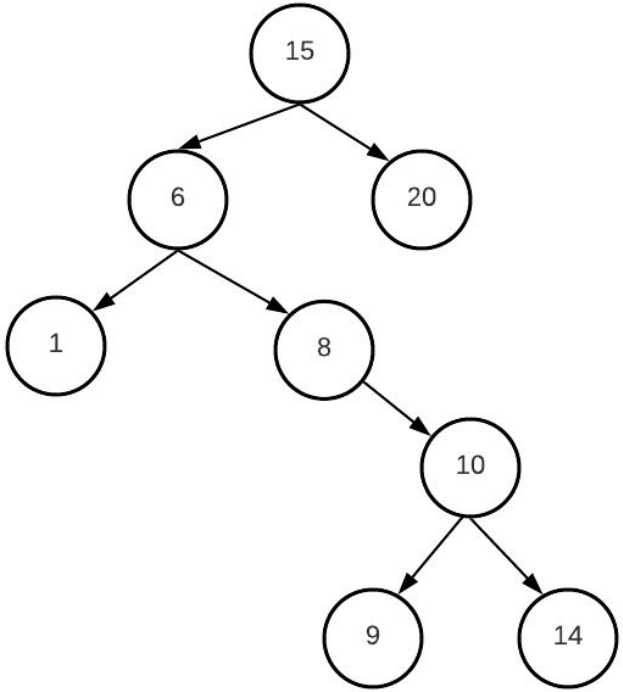
Inorder **Successor** in Binary Search Tree

- Consider x = Node(14)
- At 14: Is it a right child? Yes ⇒ go up
- At 10: Is it a right child? Yes ⇒ go up
- At 8: Is it a right child? Yes ⇒ go up
- At 6: Is it a right child? No
- What is the parent? 15 ⇒ Successor



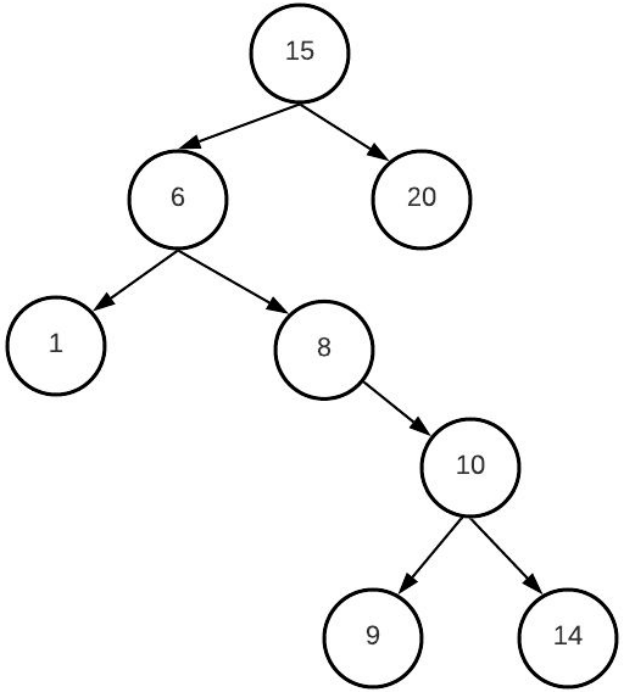
Inorder **Successor** in Binary Search Tree

- Consider x = Node(9) - No right Child
  - Work your way up
- At 9: Is it a right child? No
- What is parent? 10 ⇒ Successor



Inorder **Successor** in Binary Search Tree

- Consider x = Node(20) - No right Child
  - Work your way up
- At 20: Is it a right child? Yes ⇒ go up
- At 15: Root!
  - 20 is on its right, so 20 is bigger!
  - But 15 is the root; there are no further 'up' or 'parent' nodes
- No Successor for 20
  - Only the max value in the tree has NO successor



"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."