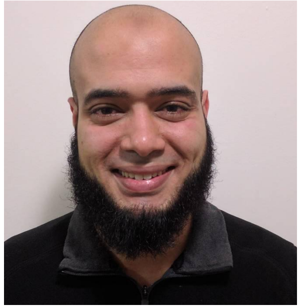
Data Structures

Minimum & Successor 2

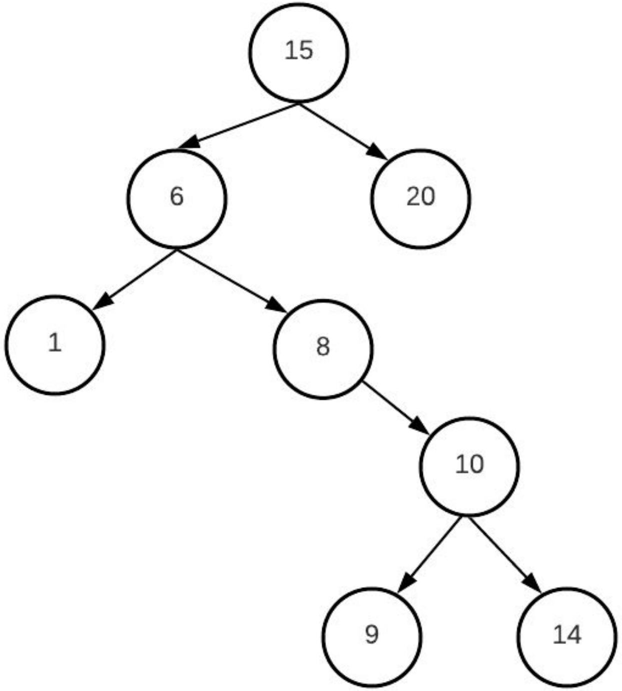
Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

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



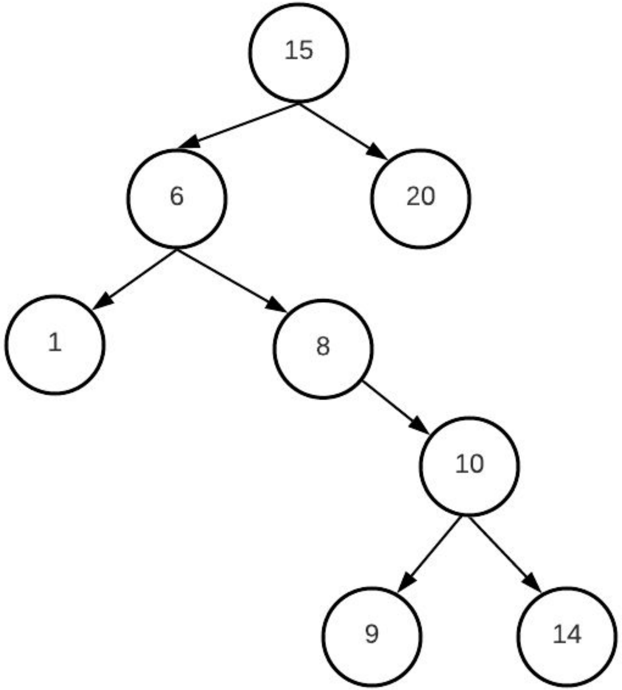
Why correct?

- We proved inorder traversal (LVR) is sorted!
- Given a value, its successor is one after it in traversal
- Actually the 2 cases are driven from inorder traversal behaviour :)
- Recall inorder traversal is
  - Process and print left
  - Print me
  - Process and print right
- Think about the correctness for 10 minutes



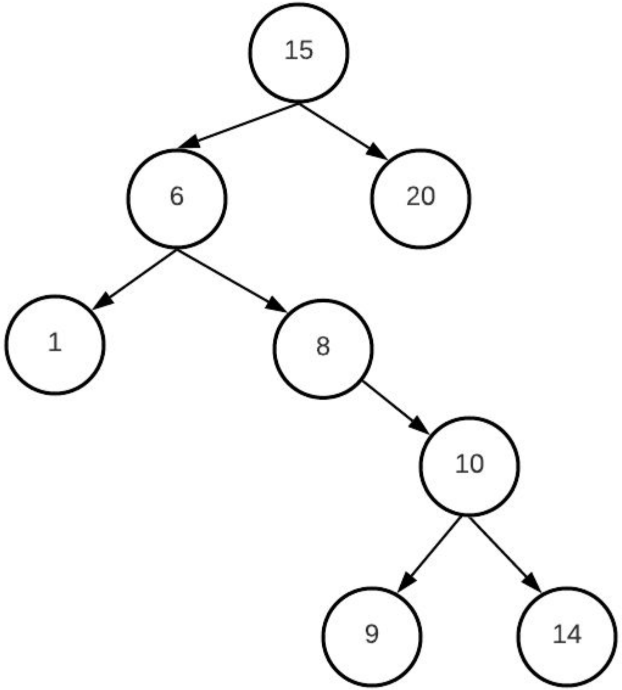
Why correct?

- Consider node (8)
- Case 1: we have a right subtree
- So far inorder printed: [1, 6, **8**]
    - Left till 8 is printed, then 8, then ?
- Now inorder will jump to my right to print it
    - Then my successor MUST be in my right subtree
- Which one? Logically, it's the minimum node in the right subtree
    - This logic also comes from our inorder traversal. Keep going left until there is no further left node - and then print it: it will be the min node!



Why correct?

- Consider node (14)
- Case 2: no right subtree
- So far inorder printed: [1, 6, 8, 9, 10, **14**]
  - We go left until 14 is printed. Then 14, then?
- Since we have no right subtree, the recursion ends, and we go back up the tree
- It's easier to think about this as a chain of right nodes: [6 ⇒ 8 ⇒ 10 ⇒ 14]
  - The recursion keeps going up the tree; ending recursive calls each time
- After 6 is done as well, go up again
- Left of 15 is done. Now print 15
  - So 15 first value printed after 14 ⇒ Successor



“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”