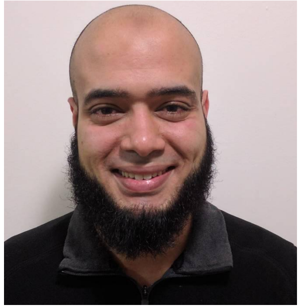
Data Structures
Solutions Drawings

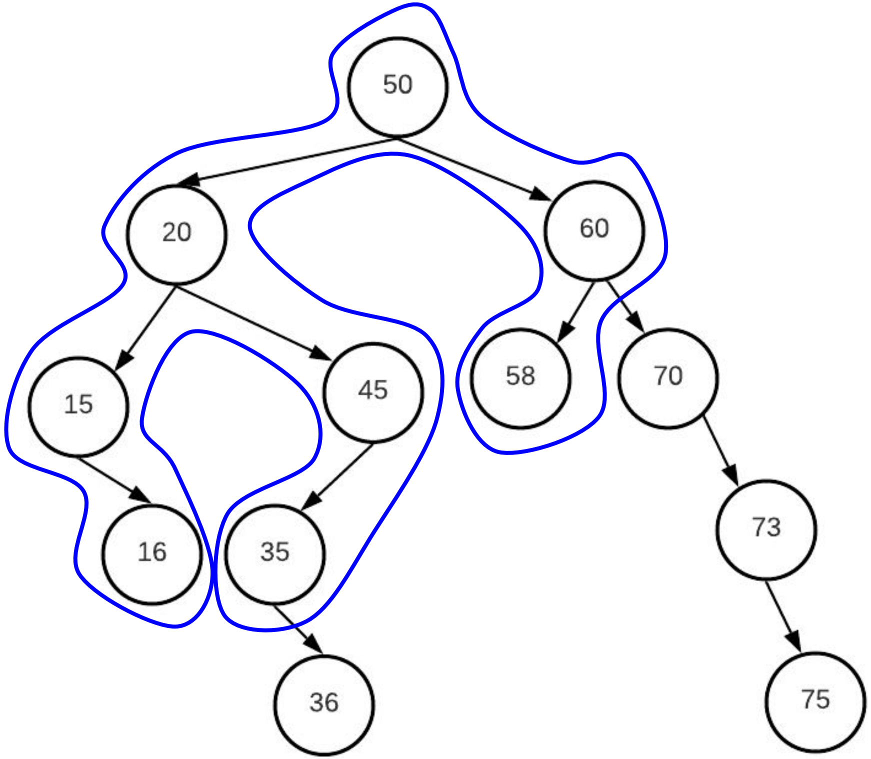
Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

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



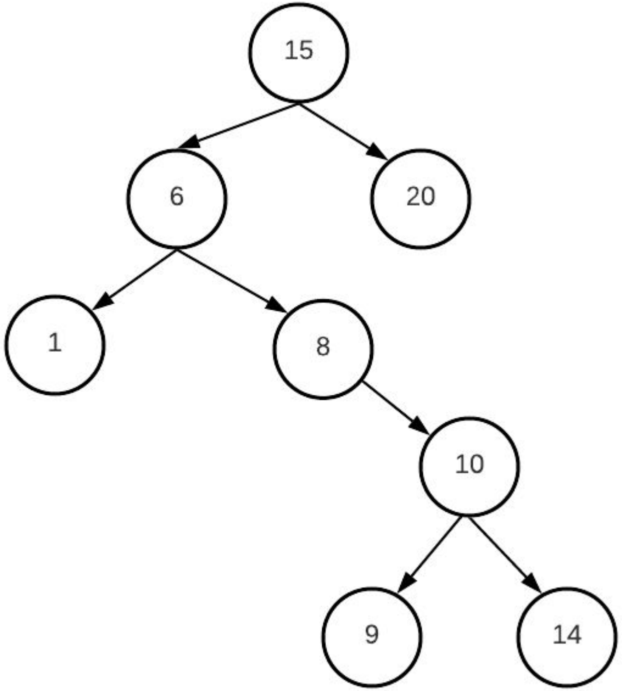
Problem #2: Queries of ancestors

- Develop a function that finds all of them such:
    - You don't do complete traversal. **Stop as early as possible** similar to lecture
        - {15, 20, 58}
    - Don't use parent pointer
    - Don't get chain of ancestors like lecture
- Tip:
    - We know inorder traversal already moves toward our successor. Why don't we just catch it once we find it
    - Code is a modified search function



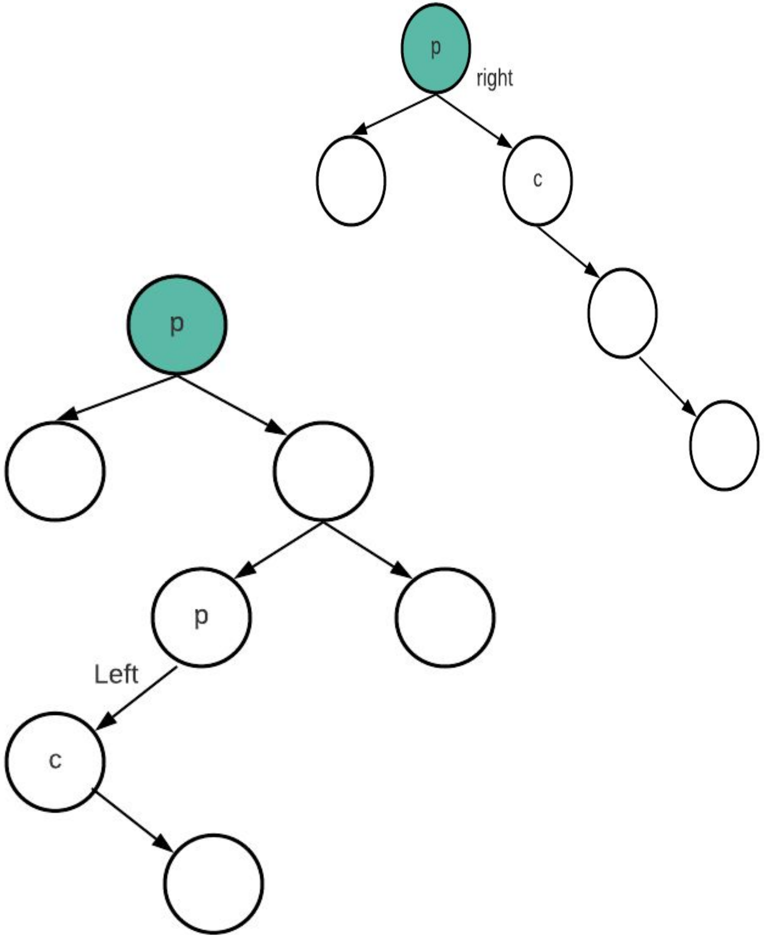
Problem #2: Queries of ancestors

- To find successor, we can modify the inorder traversal
- Keep adding the nodes you visit
- If at any node X, the last added node Y is the requested, then X is its successor
- The idea can be extended to list of queries but they must be sorted
- Don’t visit a branch that doesn’t have your next requested value!



Deletion

- First, remember that the successor node (c for child)
    - Can’t have left anymore
    - May have right child
- For the parent p, when we remove c, we want to connect its potential right
    - But connect to p->right or p->left
    - We need to find the conditions for that
    - Either c is directly the right of p
    - Or we actually kept going several left
    - Then is left child for p



“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”