Data Structures

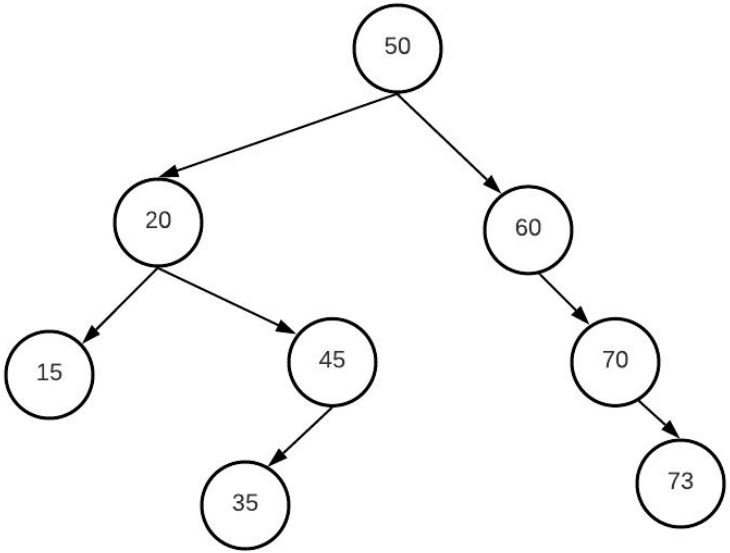
BST Homework 3

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 #1: BST from preorder v1

- BinarySearchTree(deque<int> &preorder)
- Implement a constructor that build a pst from its preorder
- You can find an easy O(n^2) solution with similar idea to our last section of binary tree from preorder+inorder
  - Hint: root split data to < and >
- Tree preorder: 50 20 15 45 35 60 70 73
- Don’t solve/listen to next problems before finishing this one



Problem #2: BST from preorder v2

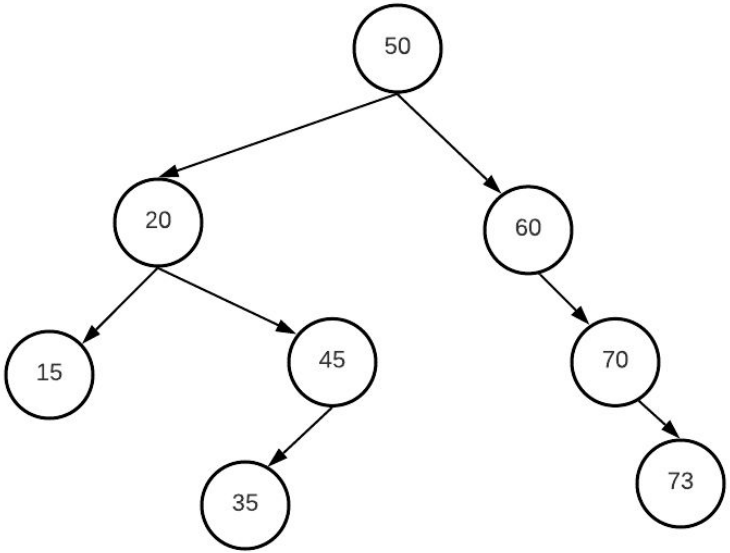
- In the previous code, you needed given a root, to iterate and find the first element greater than it within the node start/end indices
- But this made code O(n^2)
- To make it O(n), we need to find the index of the next greater element faster
- Recall, we implemented this in stack section
  - Implement vector<int> next_greater_idx(deque<int> v)
  - Return the next index of every element. Preprocessed once before building BST
- Change your code to be total O(n)

Problem #3: BST from preorder v3

- So far, we were trapped in our first idea and improving it
- There is a short clean recursive code that can build the task easier
- Recall is degenerate problem? We stressed each node can have [min, max]
- Assume all nodes values are inclusive [1, 1000]
- Implement: BinarySearchTree(deque<int> &preorder, int min = 0, int max = 1001)
- It recursively keep building/updating the min/max of each node

Problem #4: BST from level-order traversal

- BinarySearchTree(deque<int> level_order)
- Given level-order traversal, build the BST
- Tree level order: 50 20 60 15 45 70 35 73
- Tips:
  - Use node [min, max] to develop a simple iterative code to build the tree
  - Similar code to level-order traversal



"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."