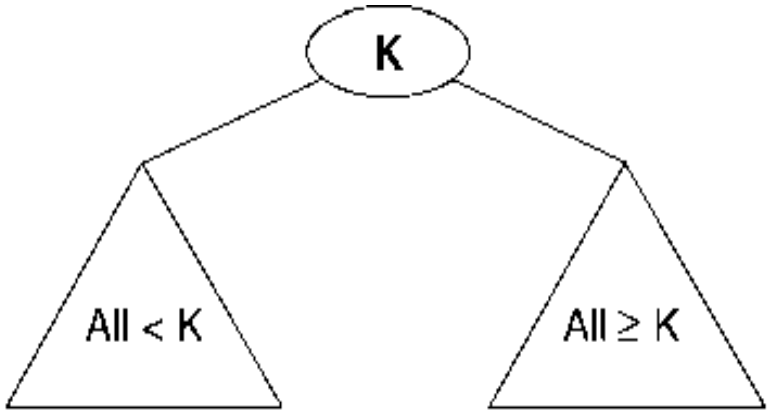
Introduction To Binary Search Tree

By Yash Gupta

BST Properties

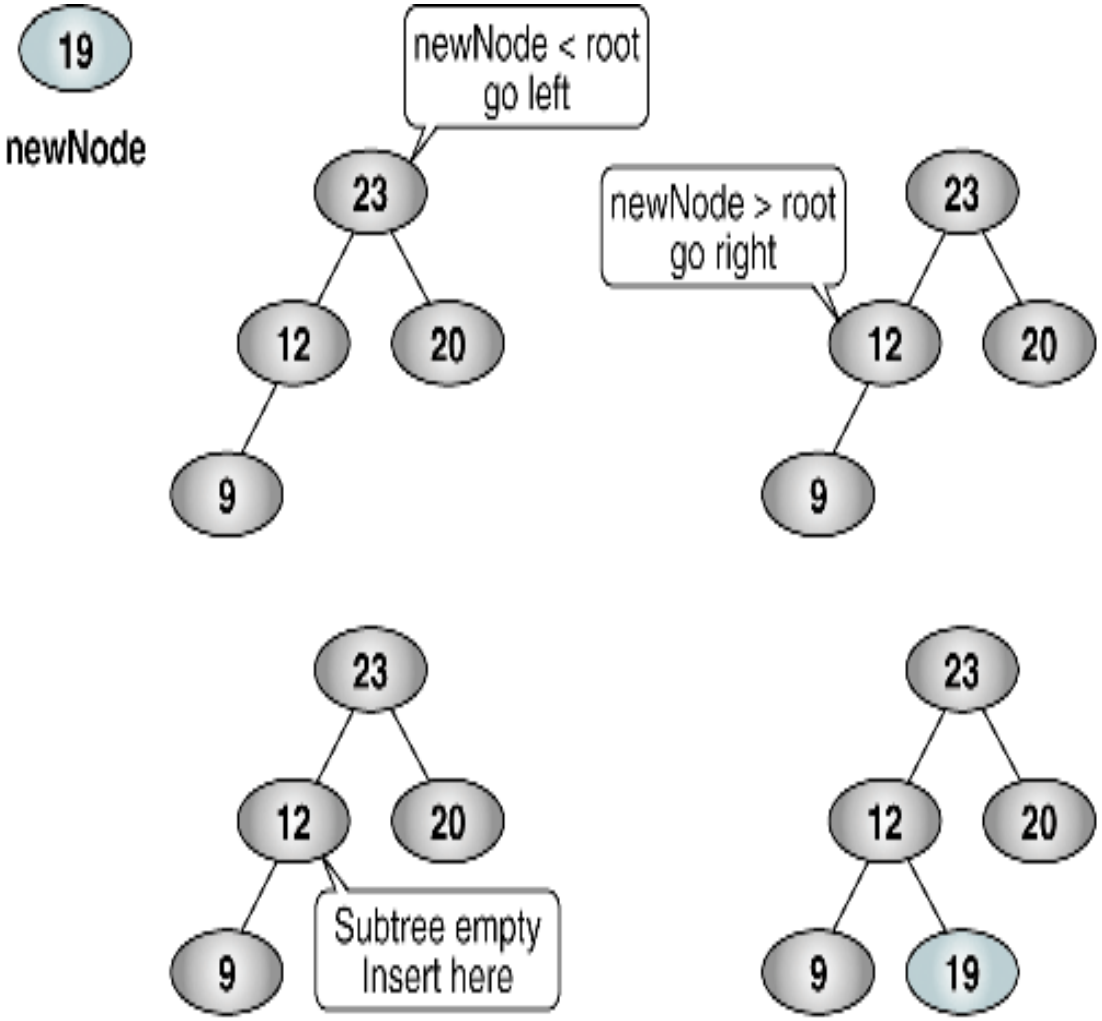
- All items in the left subtree are less than the root.
- All items in the right subtree are greater or equal to the root.
- Each subtree is itself a binary search tree.



BST Properties

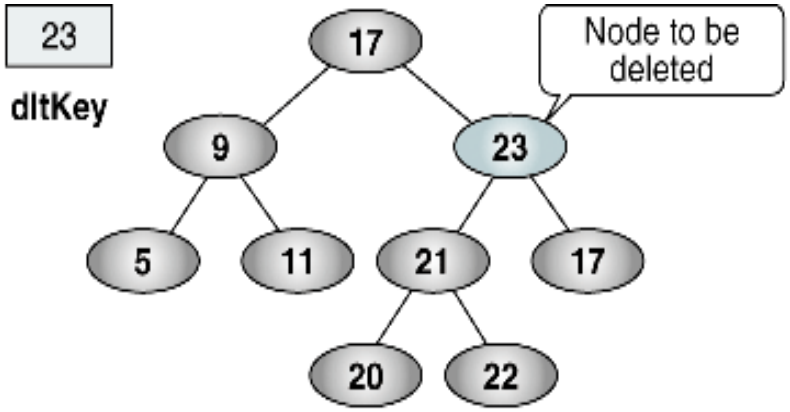
- Insertion
- Deletion
- Searching
- Traversal

BST Insertion

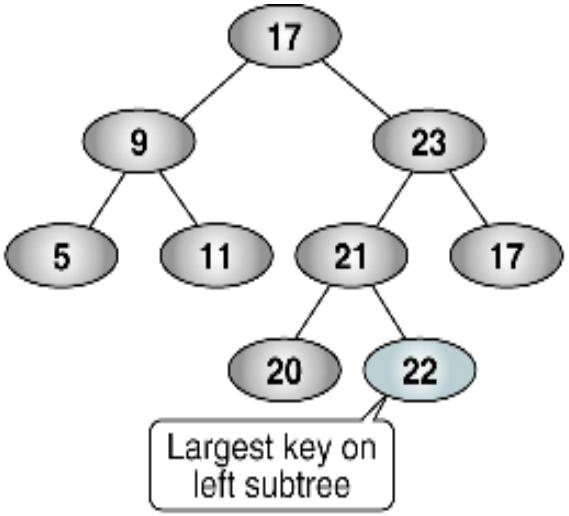


BST Deletion

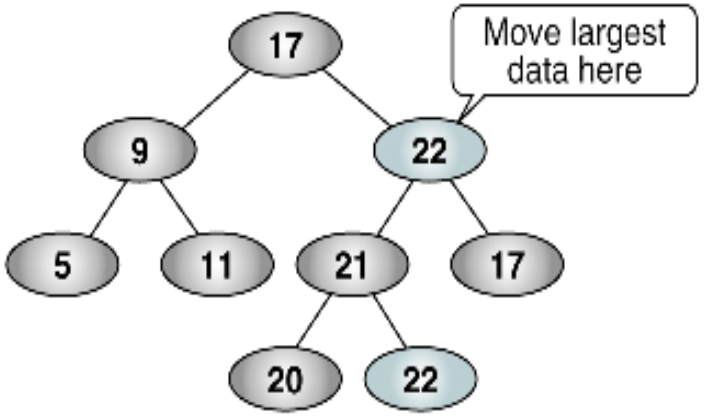
- Case 1 : Pure Leaf node case
  - In this case delete the node.
- Case 2 : Only one child
  - Replace the node with only child
- Case 3 : Has both children
  - Replace the node with its inorder successor or predecessor



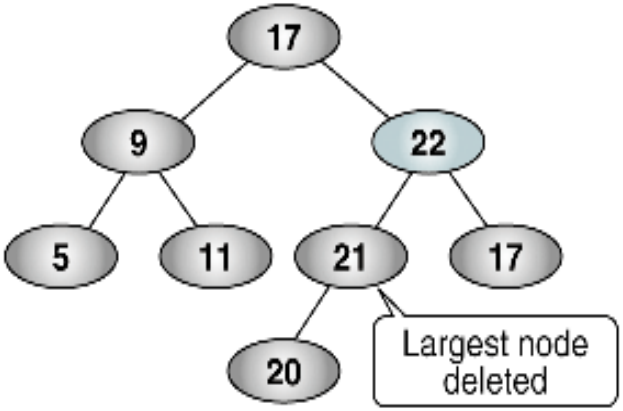
(a) Find dltKey



(b) Find largest



(c) Move largest data

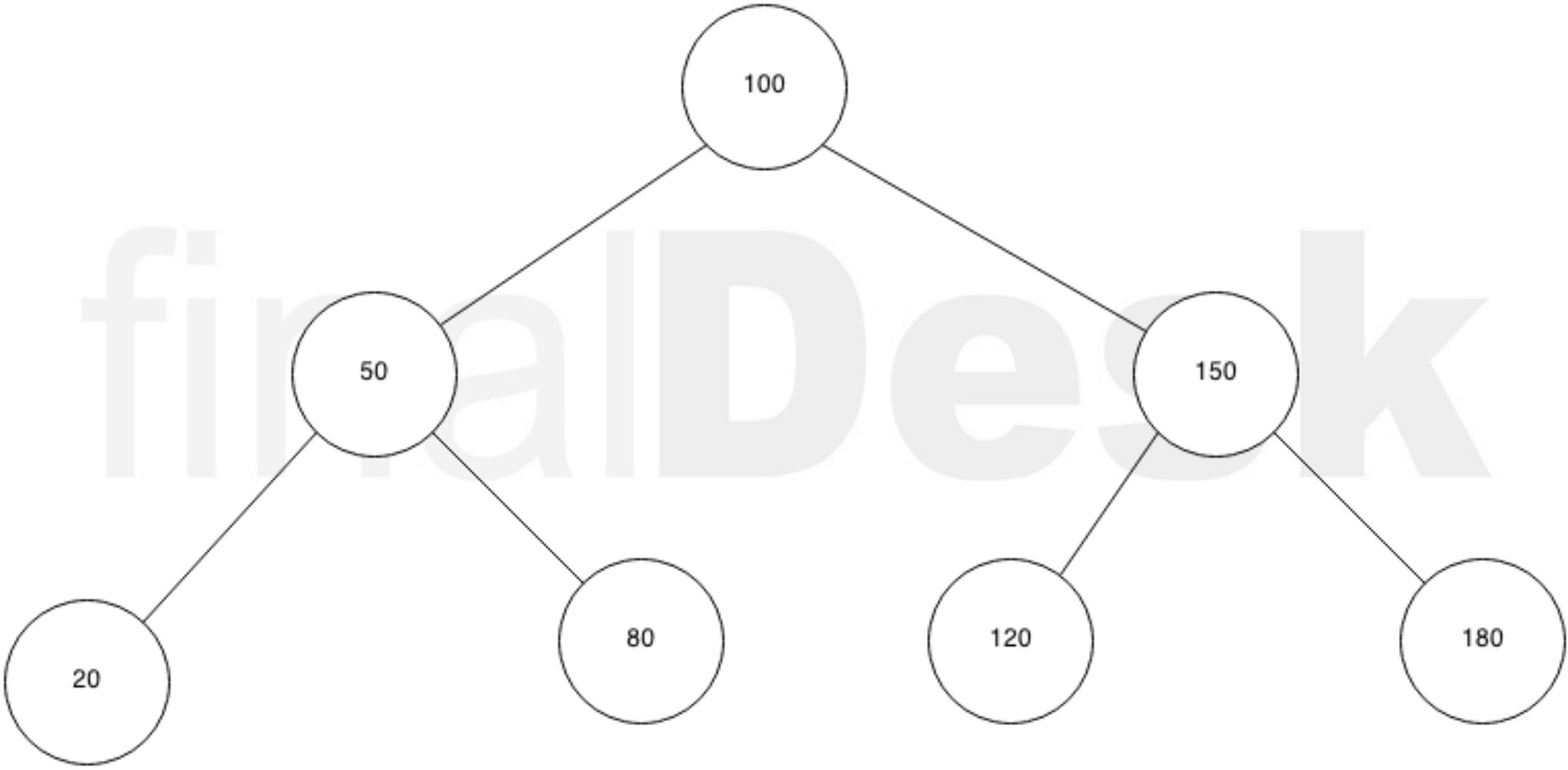


(d) Delete largest node

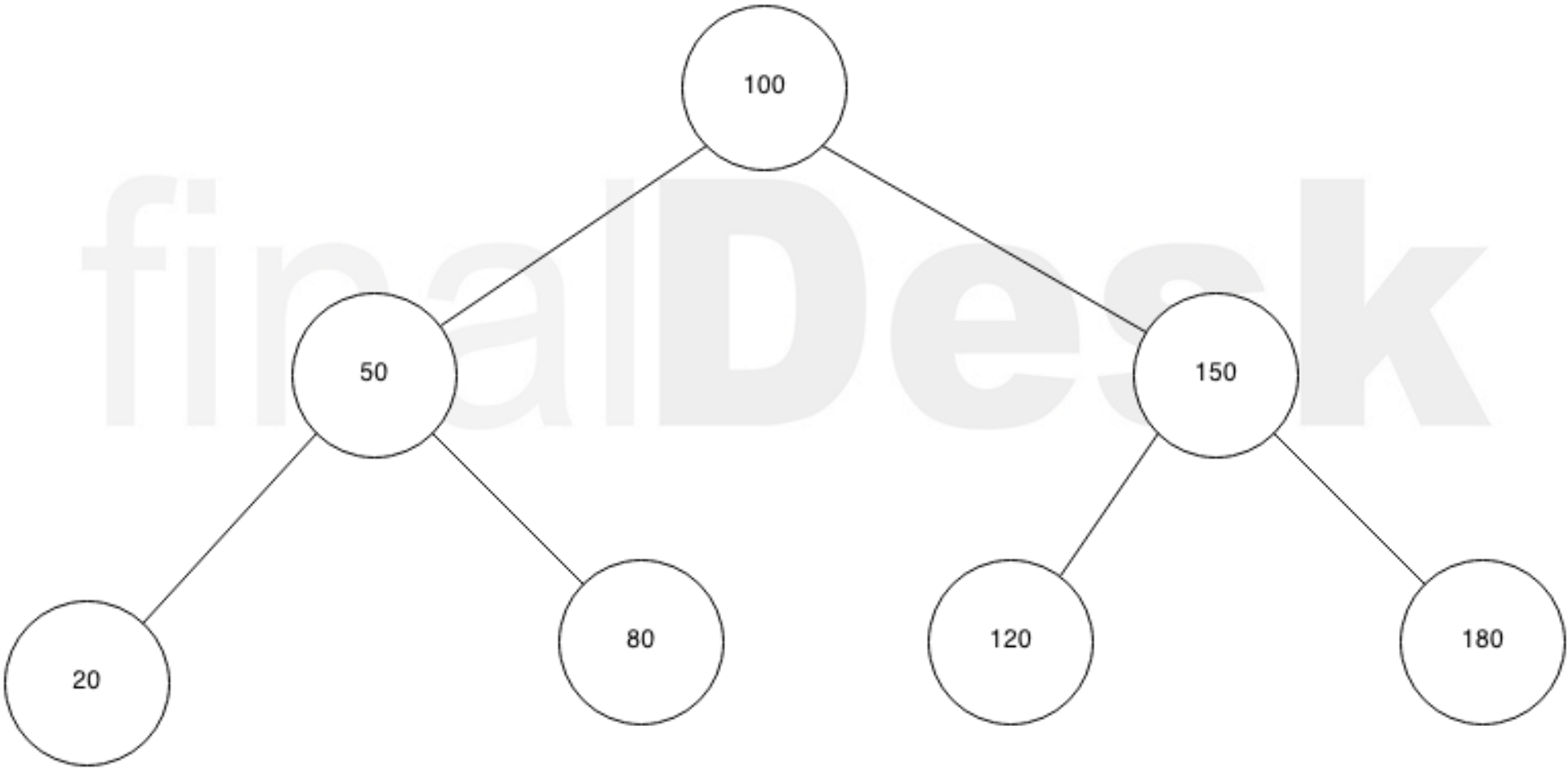
BST Tree Traversal

- Depth first Traversal
  - Pre-order Traversal
  - Post-order Traversal
  - Inorder Traversal
- Breadth-first Traversal

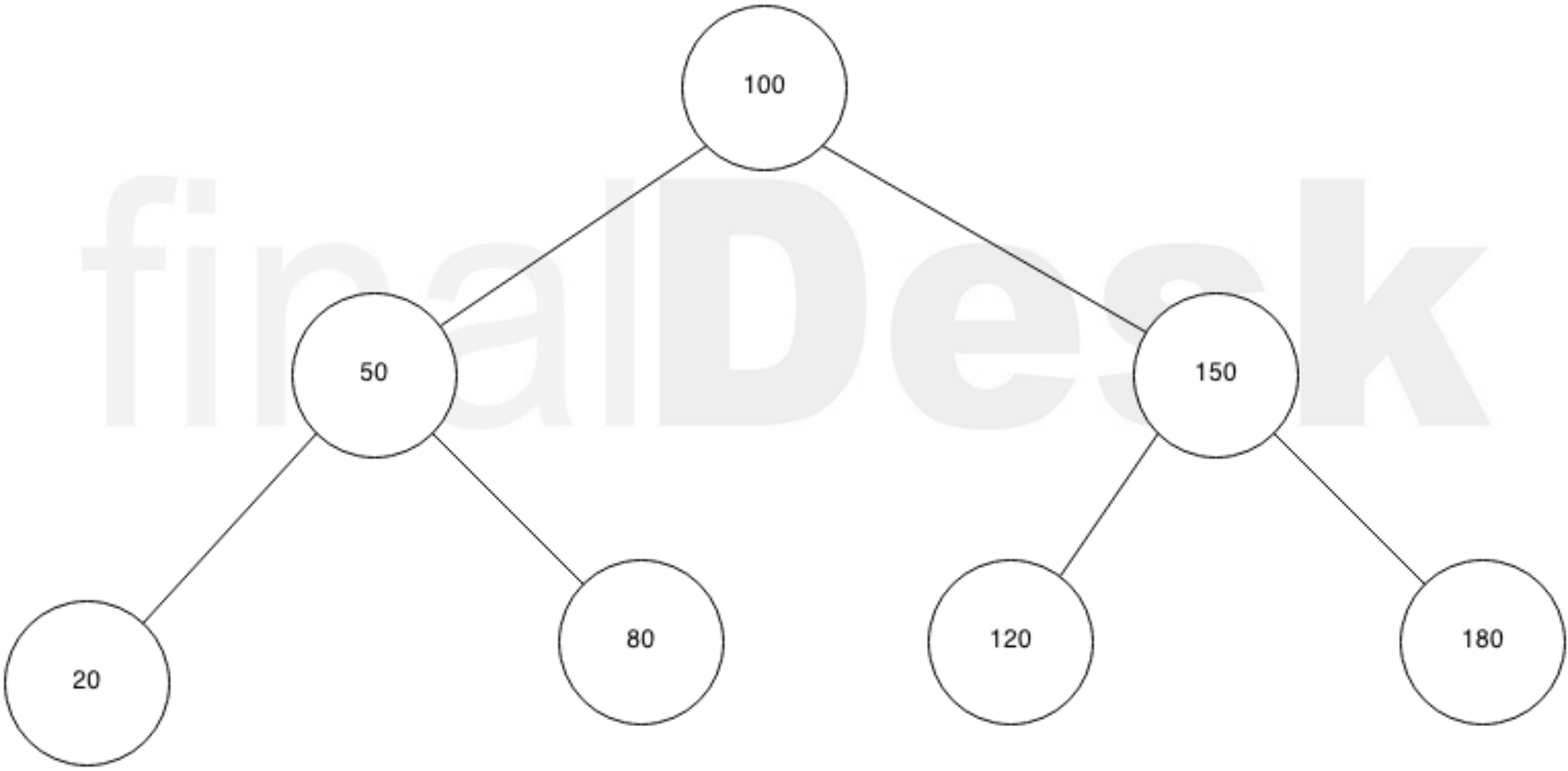
Pre-order (VLR)



Post-Order (LRV)



Inorder (LVR)



Analysis

Operation	Time Complexity		
	BEST	AVERGAE	WORST
INSERT	O(1)	-	O(n)
DELETE	O(1)	O(Log n)	O(n)
SEARCH	O(1)	O(Log n)	O(n)
TRAVERSAL	-	-	O(n)

Generate Tree

- Inorder – Preorder
- Inorder- Postorder
- Preorder - Postorder

Contact Info

- trainers@finaldesk.com
- rishabh@finaldesk.com
- nilesh@finaldesk.com
- jignesh@finaldesk.com
- yash@finaldesk.com
- anand@finaldesk.com