

Homework 2

Topic: Binary Search Trees and Tree Traversals

Q1: Build a Binary Search Tree

Insert the following values into an initially empty **Binary Search Tree (BST)**, then draw the final BST:

50, 30, 70, 20, 40, 60, 80

Q2: Build a Binary Search Tree

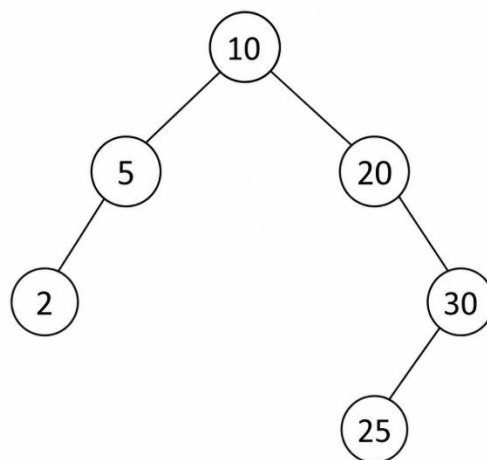
Insert the following values into an initially empty **Binary Search Tree (BST)**, then draw the final BST:

45, 25, 65, 15, 35, 55, 75, 10, 20, 30, 40

Q3: Tree Traversal

For the following binary tree, write the result of:

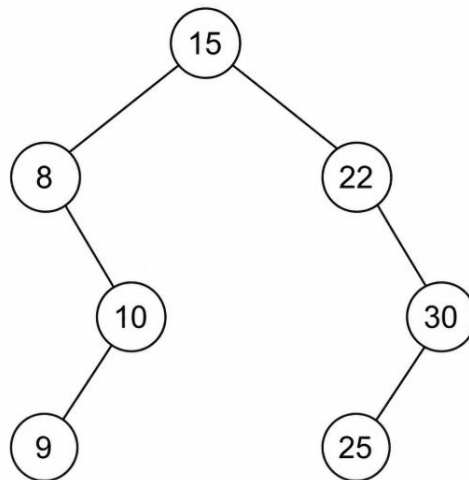
1. Preorder Traversal
2. Inorder Traversal
3. Postorder Traversal



Q4: Tree Traversal

For the following binary tree, write the result of:

1. Preorder Traversal
2. Inorder Traversal
3. Postorder Traversal



Q5: Search in a Binary Search Tree

Using the BST constructed in **Q1**, show the search path for each value:

1. 40
2. 60
3. 90

For each search, state whether the value is **found** or **not found**.

Q6: Delete from a Binary Search Tree

Using the BST constructed in **Q2**, delete the following values and draw the tree after each deletion:

1. Delete 10
2. Delete 25

Q7: Short Answer Questions

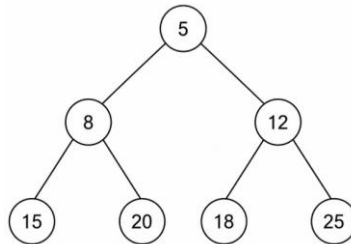
Answer briefly:

1. What is the difference between a Binary Tree and a Binary Search Tree?
2. Why is searching in a balanced BST faster than searching in a degenerate BST?
3. What is the time complexity of searching in a balanced BST?
4. Which traversal of a BST produces the values in sorted order?

Topic: Min-Heaps

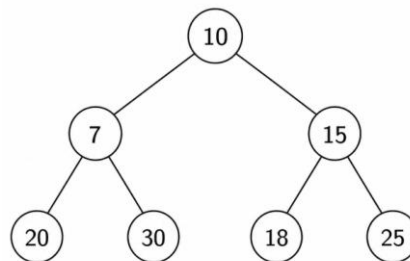
Q8: Identify a Min-Heap

Determine whether the following binary tree is a **Min-Heap** or **not**.
Justify your answer.



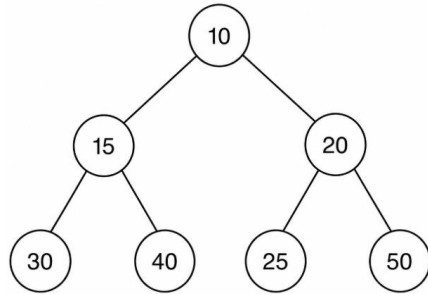
Q9: Identify a Min-Heap

Determine whether the following binary tree is a **Min-Heap** or **not**.
Justify your answer.



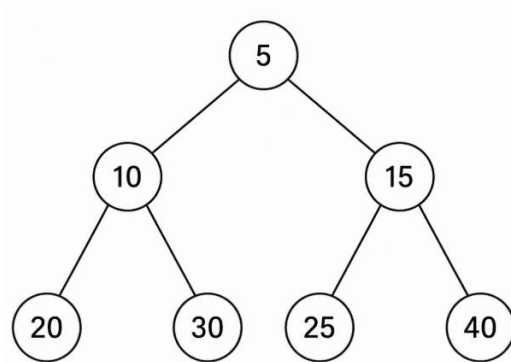
Q10: Insert into a Min-Heap

Given the following **Min-Heap**, insert the value **7**.
Draw the heap after insertion.



Q11: DeleteMin from a Min-Heap

Given the following **Min-Heap**, perform **deleteMin**.
Draw the heap after deleting the minimum value.



Q12: Build a Min-Heap

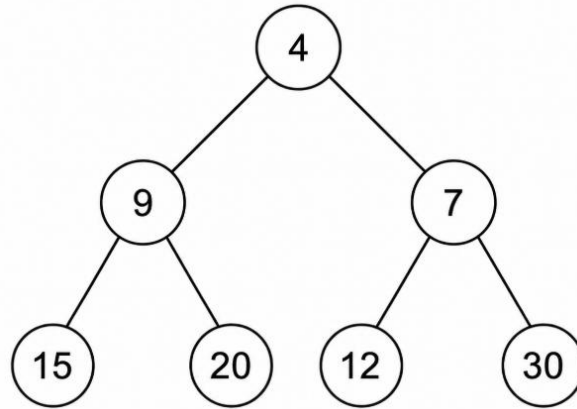
Build a **Min-Heap** from the following values using insertion one by one:

35, 20, 15, 10, 30, 25, 5

Draw the final heap.

Q13: Array Representation of a Min-Heap

Given the following **Min-Heap**:



Write its array representation.

Q14: Short Answer Questions

Answer briefly:

1. What is the main property of a **Min-Heap**?
2. Why is the minimum value always stored at the root of a Min-Heap?
3. Why is a heap useful for implementing a **Priority Queue**?
4. What is the time complexity of inserting an element into a Min-Heap?
5. What is the time complexity of **deleteMin** in a Min-Heap?
6. What is the time complexity of building a heap using **Floyd's Method**?