

## AVL Tree – Theory-Based Interview Questions

### Basic Theory

1. What is an AVL tree?
2. How does an AVL tree differ from a regular Binary Search Tree (BST)?
3. Define **balance factor** in an AVL tree.
4. How is the balance factor calculated for a node?
5. What is the **maximum height** of an AVL tree with  $n$  nodes?
6. Why are rotations required in an AVL tree?
7. What are the types of rotations in AVL trees?
8. Explain **single rotations** with examples (LL and RR).
9. Explain **double rotations** with examples (LR and RL).
10. Can an AVL tree have duplicate values? If yes, how is it handled?

---

### Intermediate Theory

11. Compare **AVL tree vs normal BST** in terms of height and efficiency.
12. Compare **AVL tree vs Red-Black tree** – advantages and disadvantages.
13. What is the **time complexity** of search, insertion, and deletion in AVL trees?
14. How do you check if a given binary tree is an AVL tree?

15. Explain **how an AVL tree remains balanced after insertion**.
16. Explain **how an AVL tree remains balanced after deletion**.
17. What is the worst-case scenario for rotations in AVL tree insertions?
18. Explain **height calculation** in an AVL tree.
19. Why is it important to maintain a height-balanced tree in applications?
20. What happens to the balance factor after a rotation?

---

### Advanced Theory

21. Explain all cases of rotations after **deletion** in AVL tree.
22. Discuss **memory overhead** of AVL trees compared to BST.
23. How do AVL trees guarantee  **$O(\log n)$  height**?
24. Explain **real-world applications** where AVL trees are preferred.
25. How would you **convert a normal BST into an AVL tree**?
26. Explain the steps of **merging two AVL trees**.
27. Explain the steps of **splitting an AVL tree**.
28. How do rotations maintain **balance factor properties** across subtrees?
29. What are the limitations of AVL trees?
30. Explain **all advantages and disadvantages of AVL trees** in data structures.