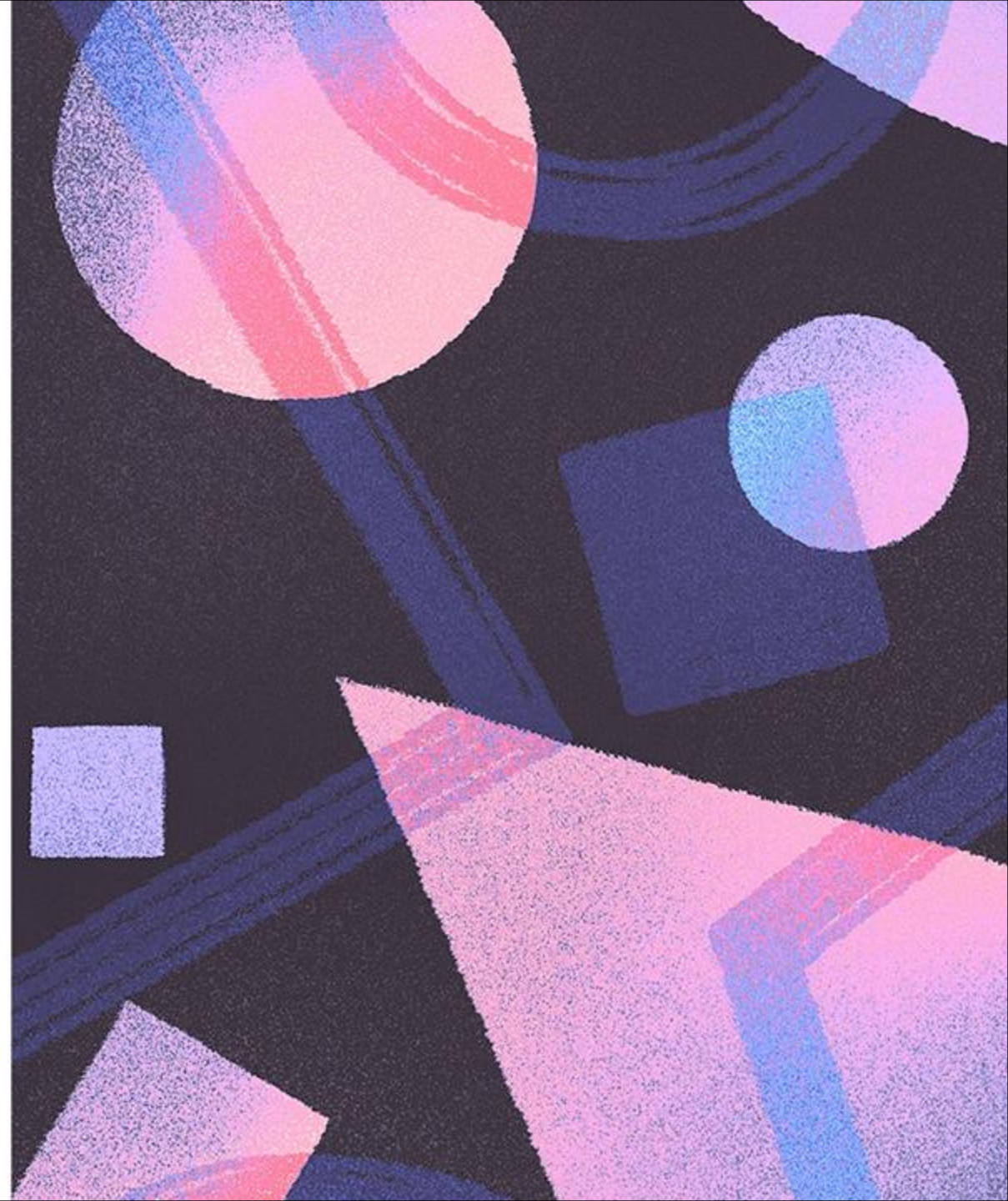


# Data Structures Lab

## Lecture 7

> Binary Search Tree (BST)



# Recall: Binary Tree

What is a Binary Tree?

A Binary Tree is a hierarchical data structure where:

- Each node has at most two children
- Children are called left and right
- No restriction on how values are arranged

# Motivation: Why Do We Need BST?

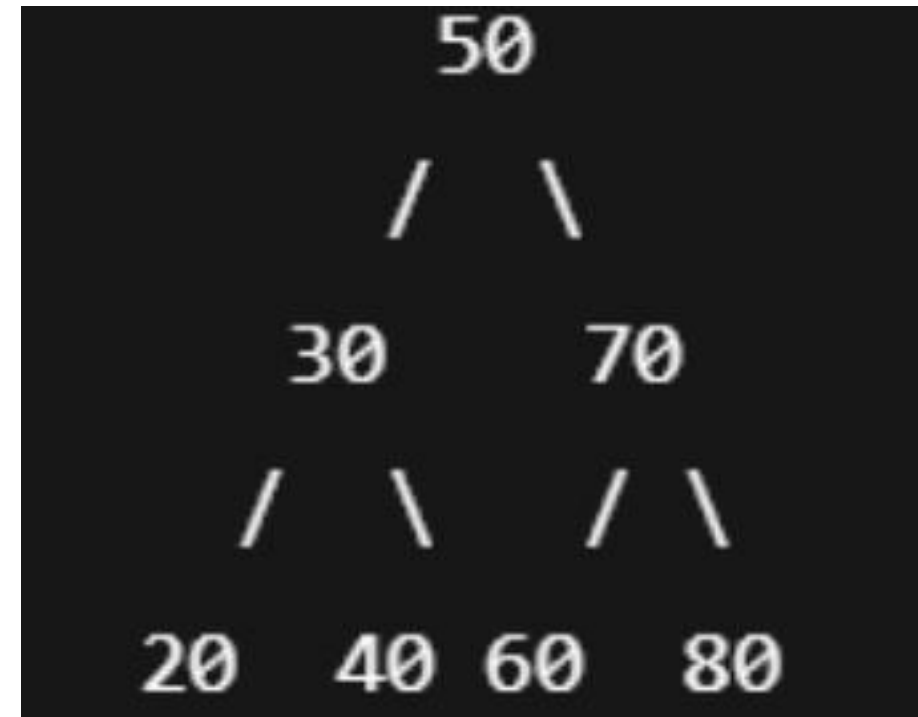
Problem with Binary Trees

- Searching for a value may require visiting all nodes
- Worst-case time:  $O(n)$

# Binary Search Tree

A Binary Search Tree (BST) is a binary tree where:

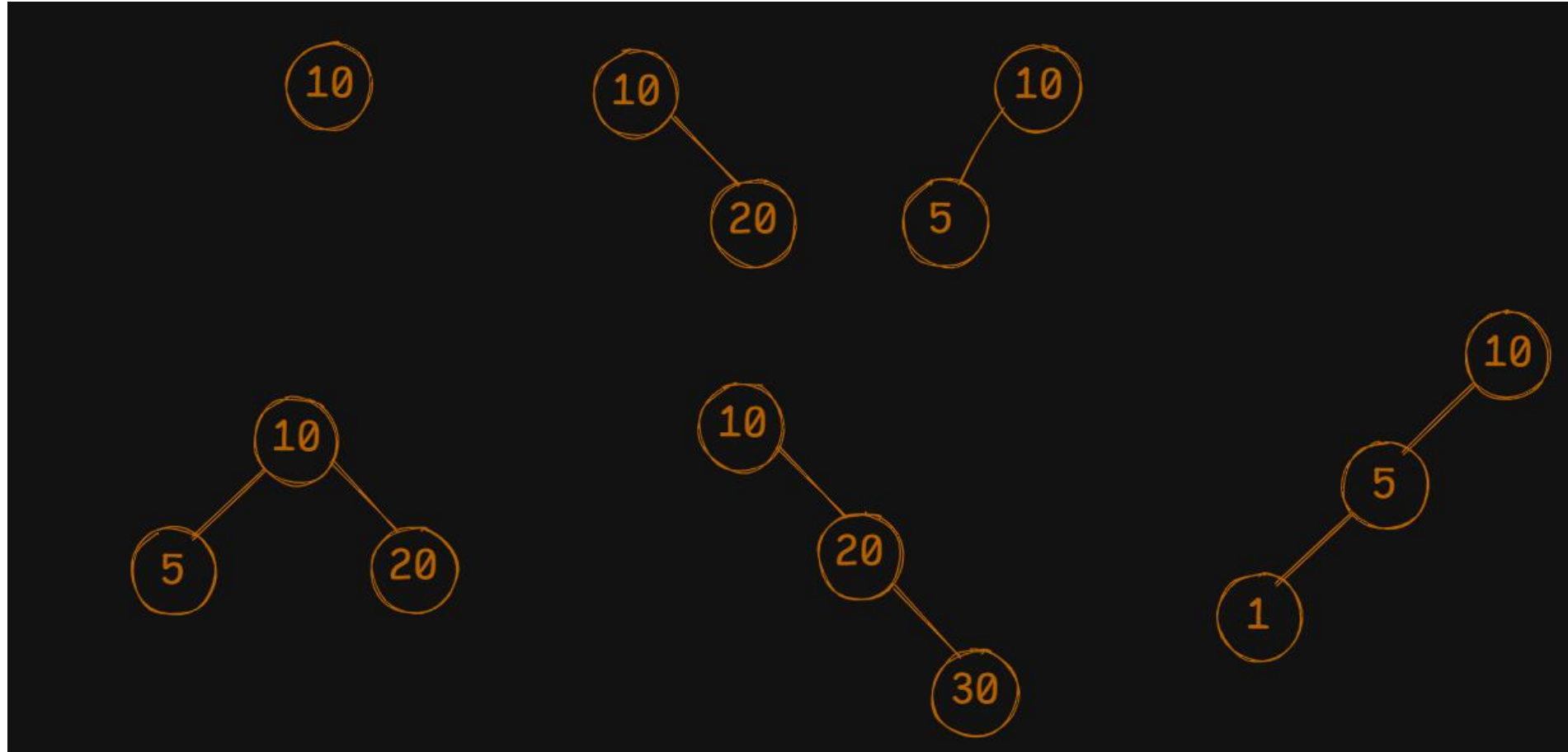
- All values in the left subtree are less than the node
- All values in the right subtree are greater than the node
- This rule applies recursively to all subtrees
- $\text{left} < \text{root} < \text{right}$



# Binary Tree vs BST

Feature	Binary Tree	Binary Search Tree
Children limit	Max 2	Max 2
Ordering	No	Yes
Searching	$O(n)$	$O(\log n)$ average
Inorder traversal	Random	Sorted
Usage	General structure	Fast search & sorting

# Types of BST (By Shape)



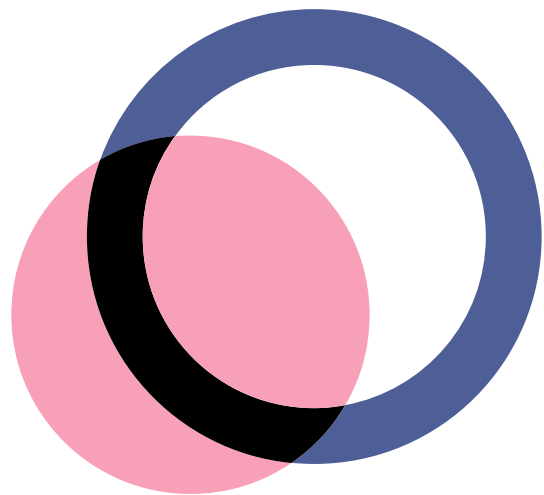
# BST Implementation

- Insert Operation
- Search Operation
- Delete Operation

# Summary

- ✓ BST = Binary Tree + Ordering
- ✓ Fast search, insert, delete (average)
- ✓ Inorder traversal = sorted data
- ✓ Used heavily in real systems
- ✓ Foundation for advanced trees





**THANK  
YOU**

