1.4 DATA TYPES & STRUCTURES · 1.4.2

Trees & traversals

BSTs, insertion, and the three traversals. Spec 1.4.2(b)(c).

01 Terminology

Tree	Hierarchy, **no cycles**.
Root	Single top node.
Leaf	Node with **no children**.
Binary tree	≤ 2 children per node.

02 Binary search tree

Rule	left subtree < node < right subtree.
Insert	From root: smaller → **left**, larger → **right**, until empty.
Search	Same compares; halves the tree each step (fast).
Balance	Sorted input → unbalanced chain → slow (like a list).

03 Traversals — when is the root visited?

Pre-order	**root** → left → right (root **first**).
In-order	left → **root** → right (root **middle**) → **sorted output**.
Post-order	left → right → **root** (root **last**).
DFS vs BFS	3 above are **depth-first** (stack/recursion); **breadth-first** = level by level (queue).

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on tree questions.

01

Traversal = **when the root is visited**: pre=first, in=middle, post=last.

02

In-order of a BST = sorted order. Learn this.

03

Insert: smaller **left**, larger **right**, compare from the root.

04

Sorted insertion → **unbalanced** tree → search as slow as a list.

05

DFS uses a **stack/recursion**; BFS uses a **queue**.

06

A tree has a **root + no cycles**; a graph need not.