

AQA Computer Science A-Level
4.3.2 Tree-traversal
Concise Notes



Specification:

4.3.2.1 Simple tree-traversal algorithms

Be able to trace the tree-traversal algorithms:

- pre-order
- post-order
- in-order.

Be able to describe uses of tree-traversal algorithms. Pre-Order: copying a tree. In-Order: binary search tree, outputting the contents of a binary search tree in ascending order. Post-Order: Infix to RPN (Reverse Polish Notation) conversions, producing a postfix expression from an expression tree, emptying a tree.



Tree Traversal

- Tree-traversal is the **process of visiting/updating/outputting each node** in a **tree**.
- It is an **algorithm**.
- **Unique** to trees.
- Must start at the **root**.
- Works around the tree **anticlockwise**.
- Form of **depth-first traversal**.
- Three types are **prefix**, **infix** and **postfix**.

Prefix Traversal

- Used for **all trees**.
- Mark the **left** of each node.
- Used for **copying a tree**.

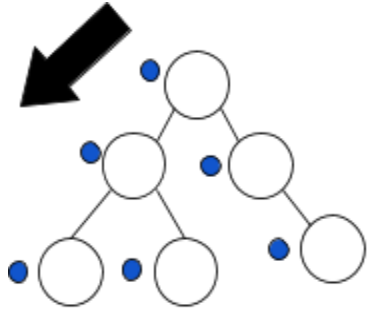
Infix Traversal

- Only well defined for **binary trees**.
- Mark the **bottom** of each node.
- Can be used on **binary search trees** to **output the contents in ascending order**.

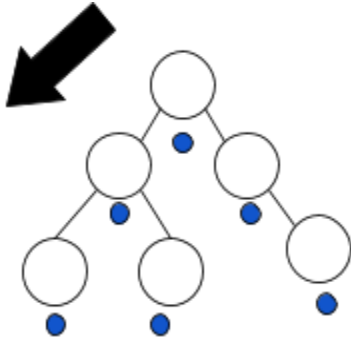
Postfix Traversal

- Can be applied to **any tree**.
- Mark the **right** of each node
- Used for **emptying a tree**.
- Used for **infix to RPN (postfix) conversions**.
- Can be used to create an **expression** from an **expression tree**

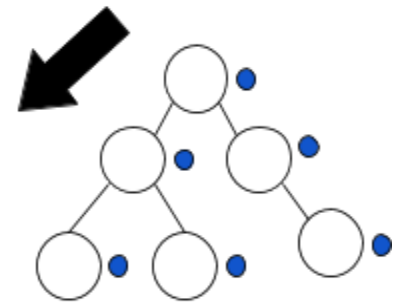




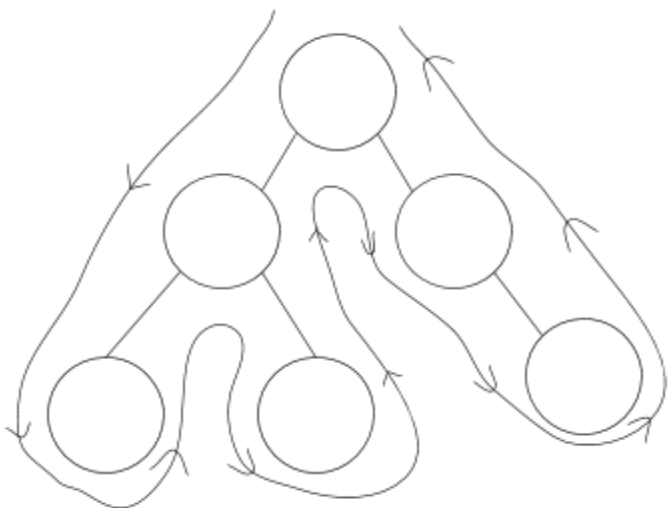
Pre-Order



In-Order



Pre-Order



The journey around a tree always occurs like this.

