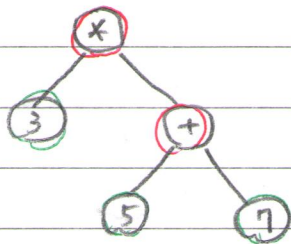


* expression tree revisited



$$3 * (5 + 7)$$

internal : operator

external : operands

Sub-tree : ()

print out infix notation

Infix Print (p) {

- if (is leaf (p)) print p → data; // operand
- else {

print "(";

- Infix Print (p → left);

- print p → data; // operator

- Infix Print (p → right);

print " ";

}

- : ^{inorder} traversal of the tree (print ⇒ visit)

visit sequence : 3, *, 5, +, 7

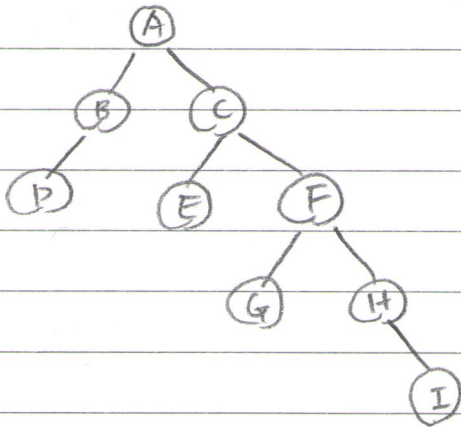
* postfix notation ⇒ postorder traversal

Postfix (p → left); Postfix (p → right); visit p → data;

prefix notation ⇒ prefix traversal

visit p → data; Prefix (p → left); Prefix (p → right);

*



in : DBA E C G F H I

post : D B E G I H F C A

pre : A B D C E F G H I

* why traversal : many bin. tree operations are similar to one of the traversal¹⁰

postorder on exp. tree \Rightarrow evaluation

preorder on two bin. trees \Rightarrow equality test

inorder on $\triangleleft_L < \text{root} < \triangleleft_R \Rightarrow$ ordered output