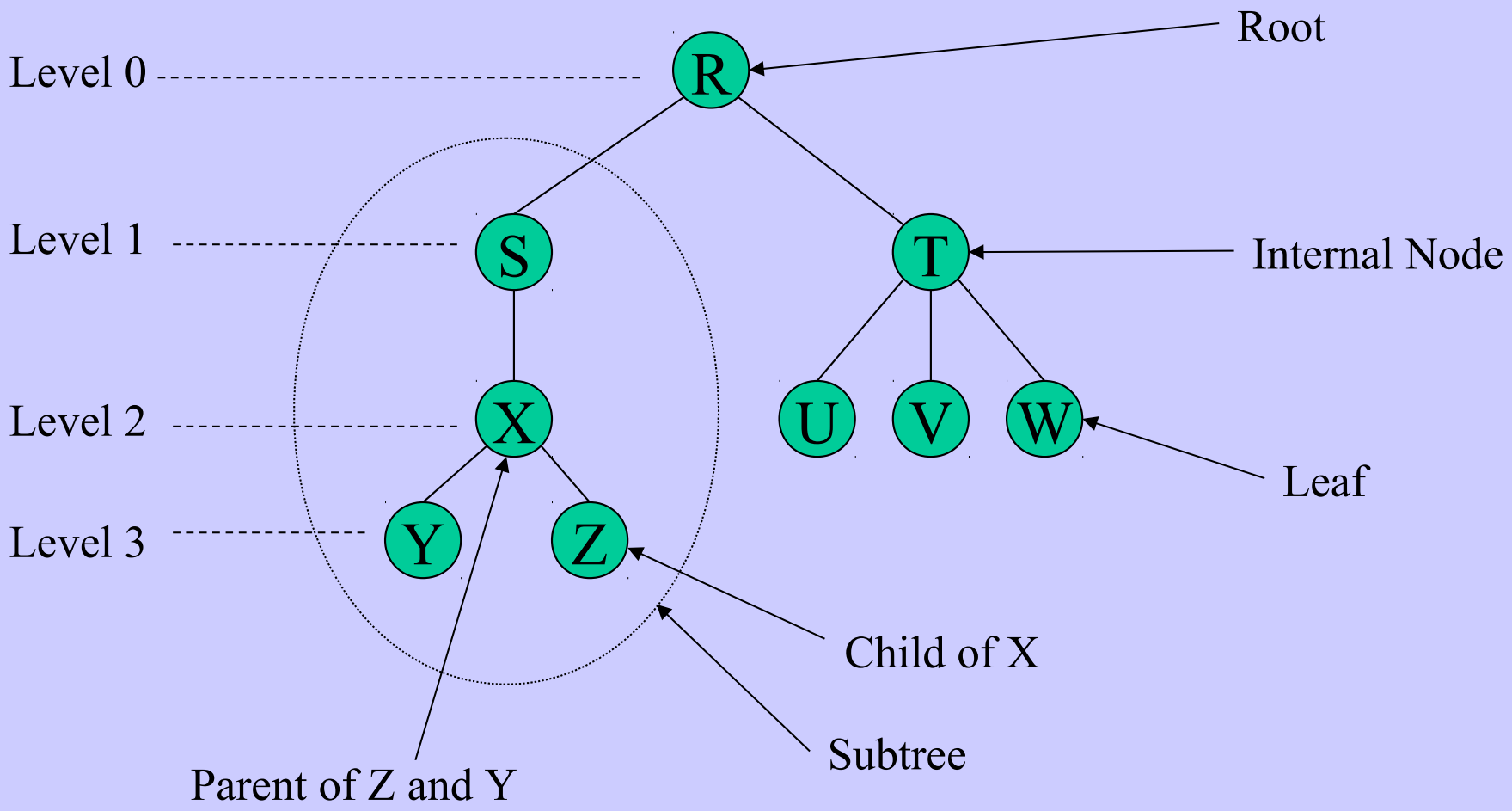
Self-Study Topic-2
(Tree Traversal: Preorder, postorder, inorder)

Tree Anatomy

The children of a node are, themselves, trees, called subtrees.

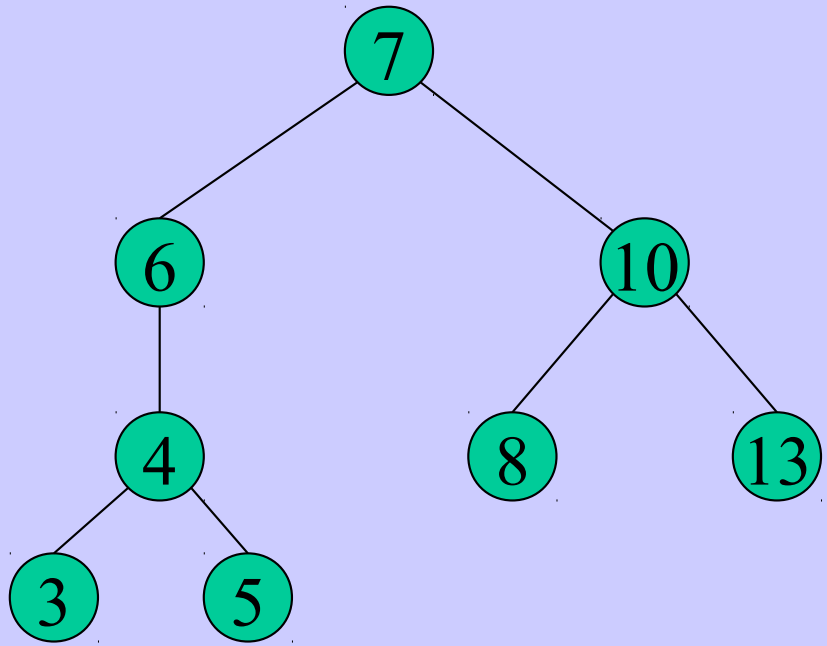


Tree Traversals

- One of the most common operations performed on trees, are a *tree traversals*
- A traversal starts at the root of the tree and visits every node in the tree exactly once
  - *visit means to process the data in the node*
- Traversals are either *depth-first* or *breadth-first*

Breadth First Traversals

- All the nodes in one level are visited
- Followed by the nodes at next level
- Beginning at the root
- For the sample tree
  - 7, 6, 10, 4, 8, 13, 3, 5

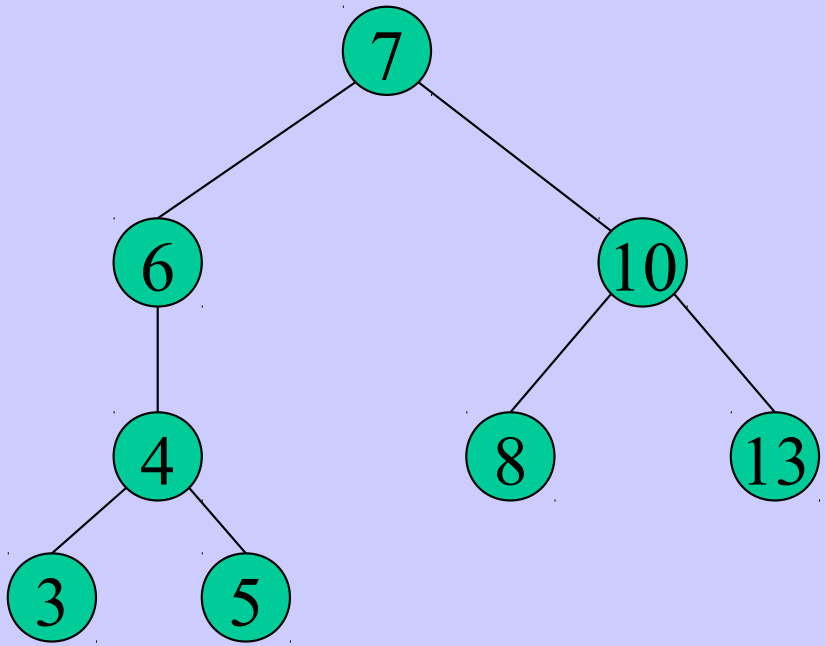


Depth-First Traversals

- There are 3 different depth-first traversals
  - pre-order traversal
  - in-order traversal
  - post-order traversal

Pre-order Traversal: VLR

- Visit the node
- Do a pre-order traversal of the left subtree
- Finish with a pre-order traversal of the right subtree
- For the sample tree
  - 7, 6, 4, 3, 5, 10, 8, 13



Preorder Traversal

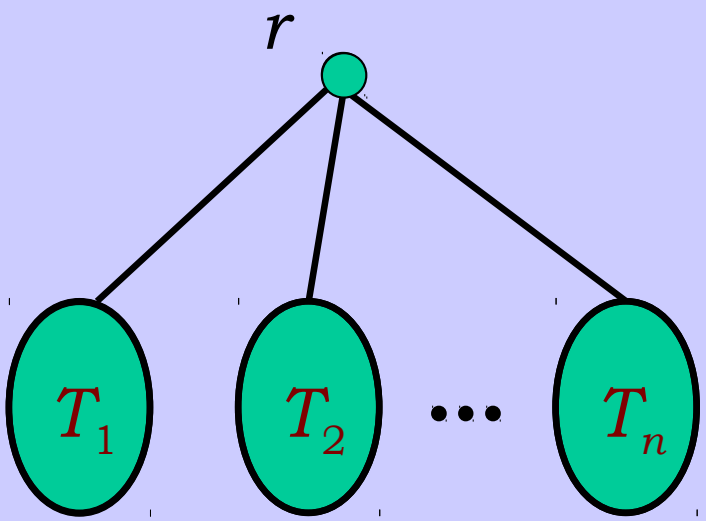
Step 1: Visit r

Step 2: Visit T_1 in preorder

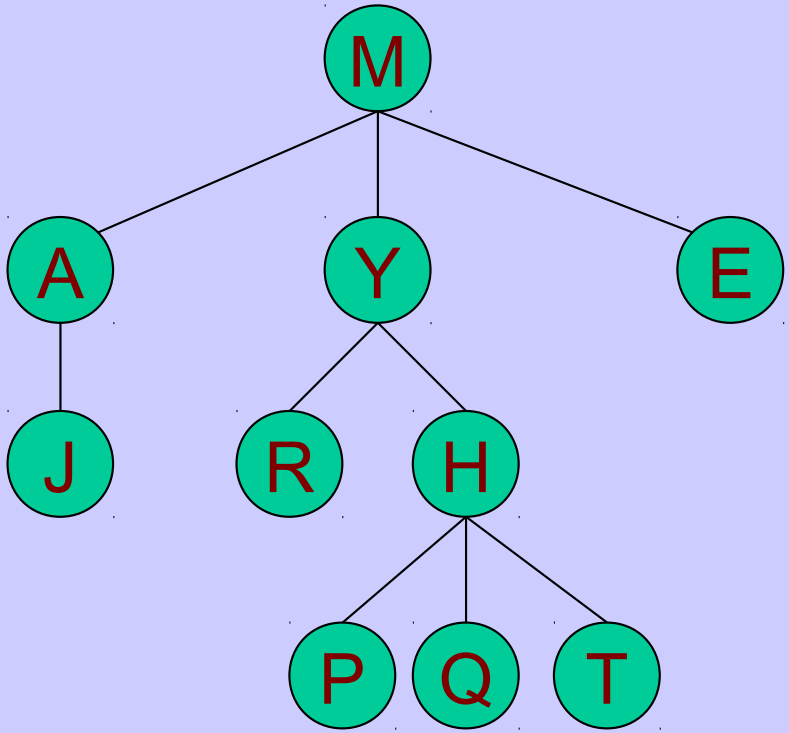
Step 3: Visit T_2 in preorder

⋮

Step n+1: Visit T_n in preorder

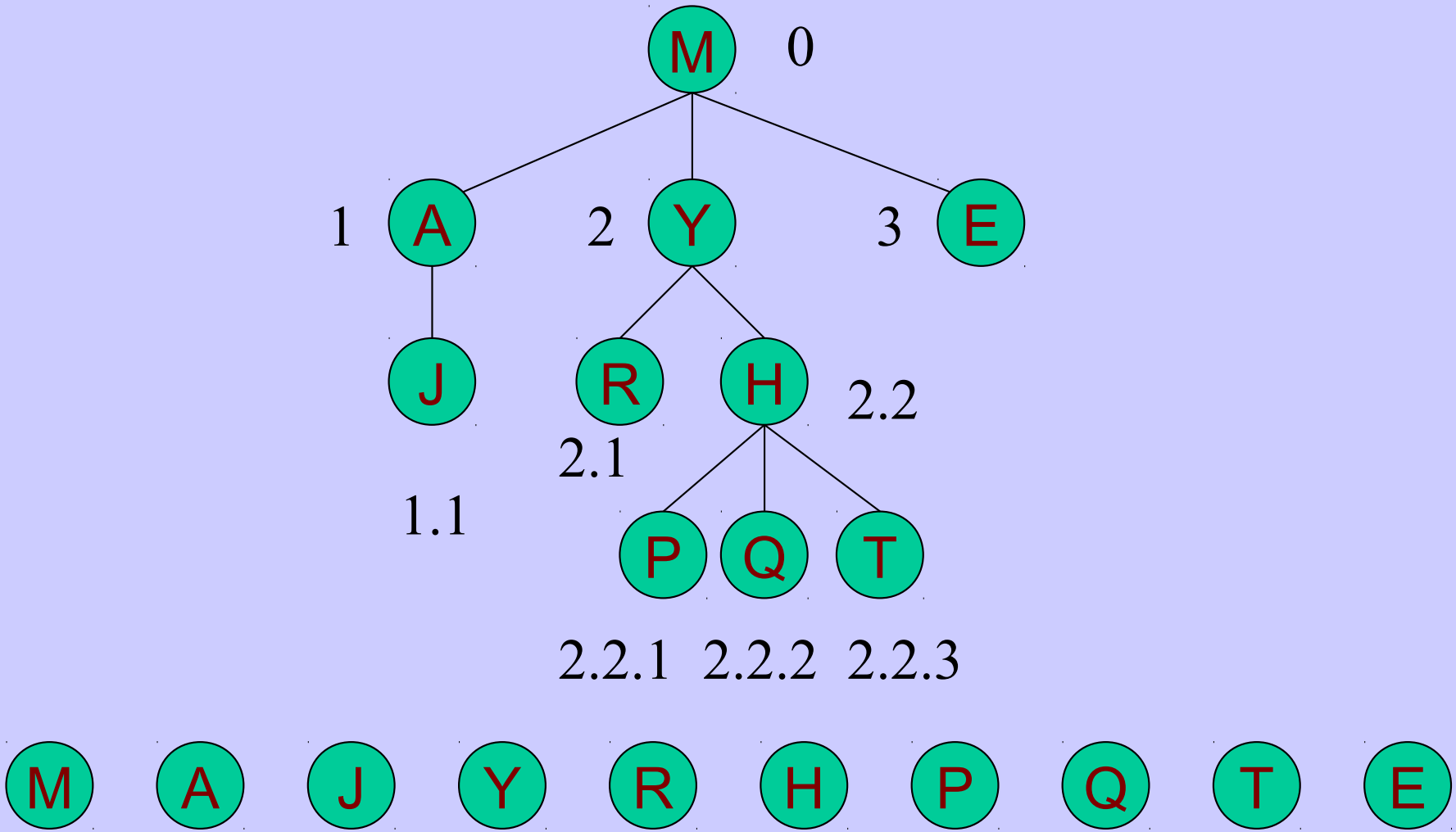


Example



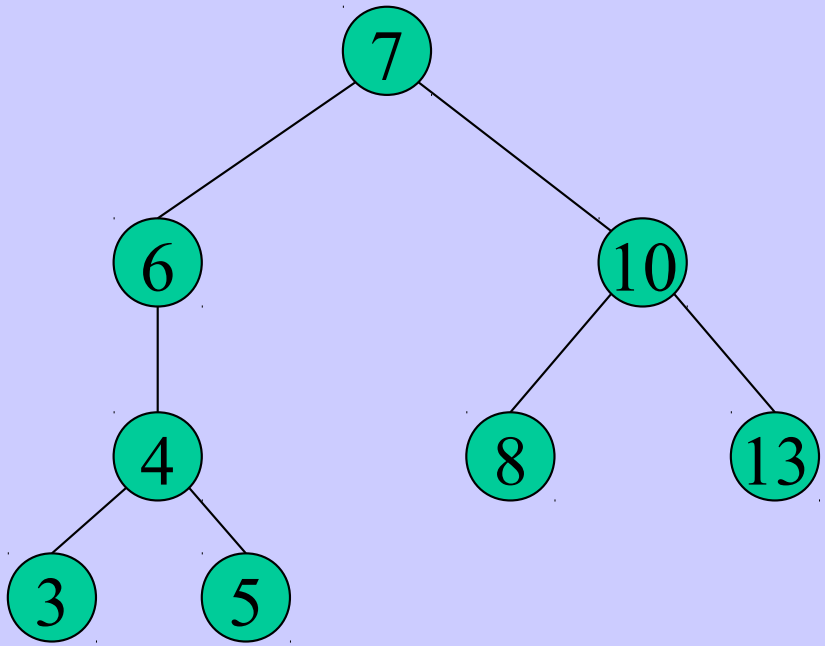
M A J Y R H P Q T E

Ordering of the preorder traversal is the same a the **Universal Address System** with lexicographic ordering.



In-order Traversal: LVR

- Do an in-order traversal of the left subtree
- Visit the node
- Finish with an in-order traversal of the right subtree
- For the sample tree
  - 3, 4, 5, 6, 7, 8, 10, 13



Inorder Traversal

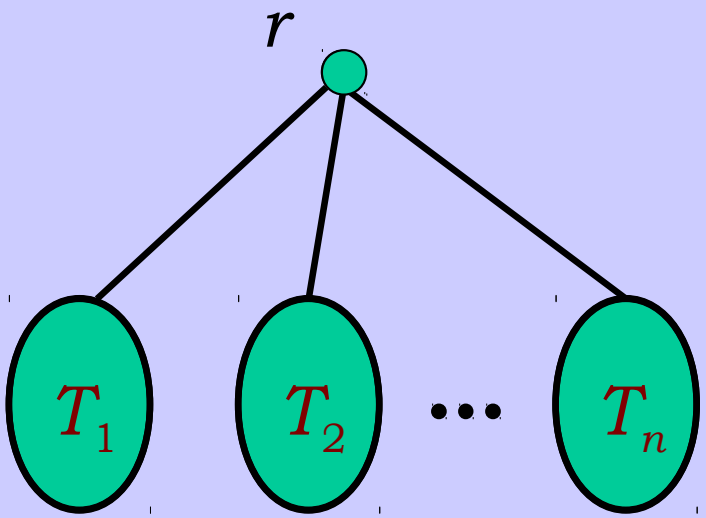
Step 1: Visit T_1 in inorder

Step 2: Visit r

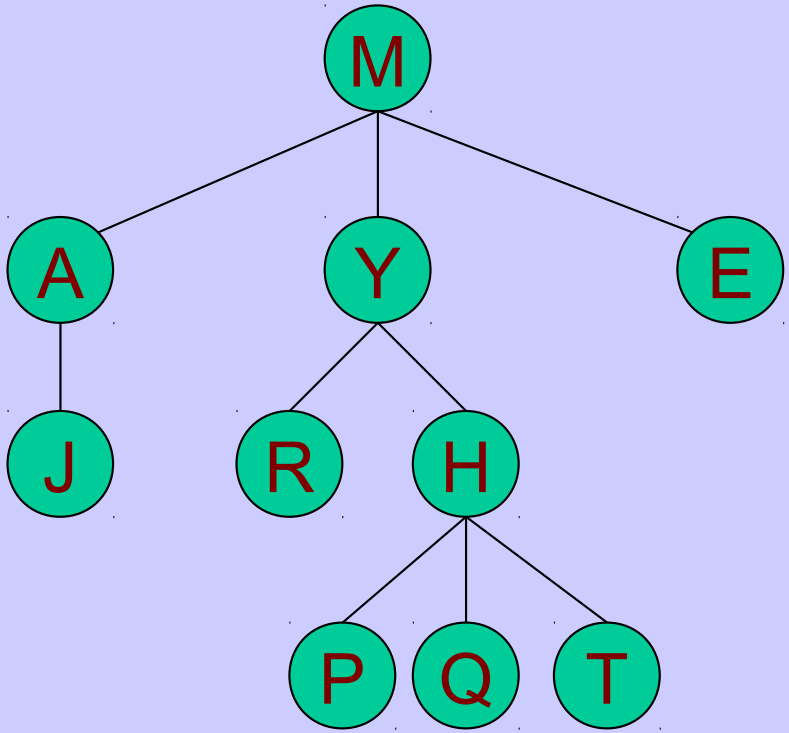
Step 3: Visit T_2 in inorder

⋮

Step n+1: Visit T_n in inorder



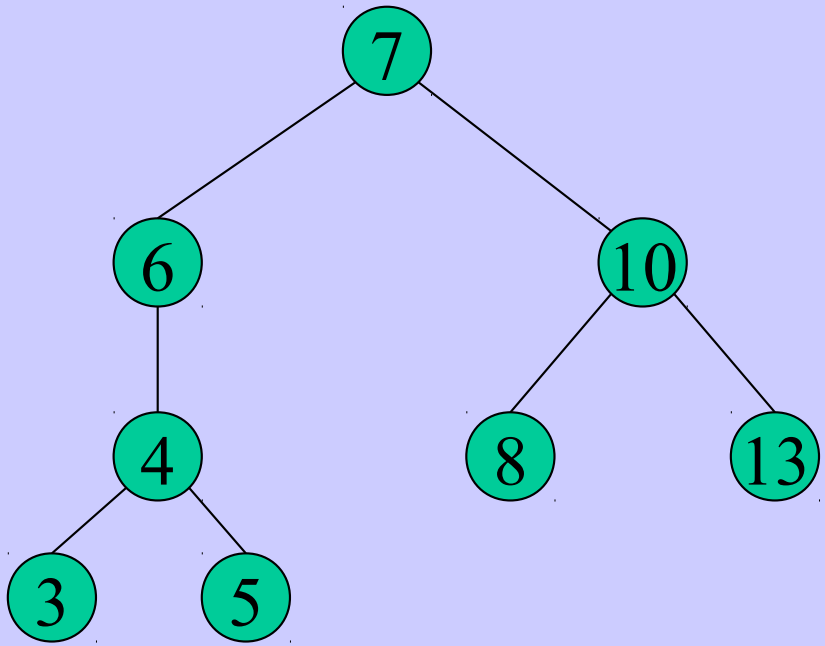
Example



J A M R Y P H Q T E

Post-order Traversal: LRV

- Do a post-order traversal of the left subtree
- Followed by a post-order traversal of the right subtree
- Visit the node
- For the sample tree
  - 3, 5, 4, 6, 8, 13, 10, 7



Postorder Traversal

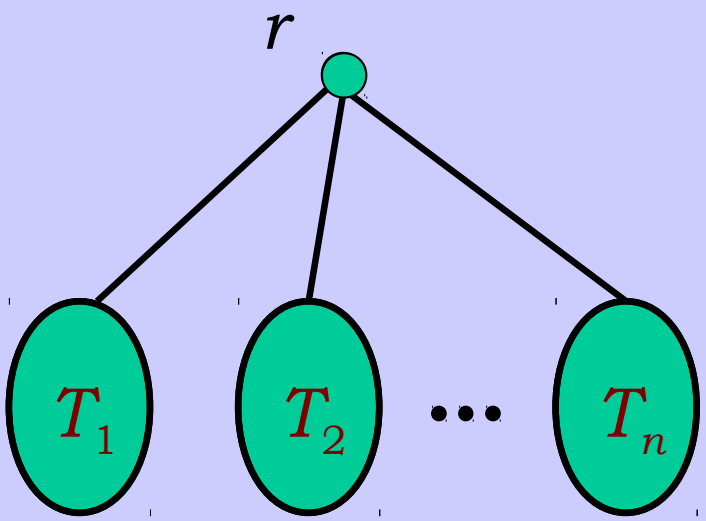
Step 1: Visit T_1 in postorder

Step 2: Visit T_2 in postorder

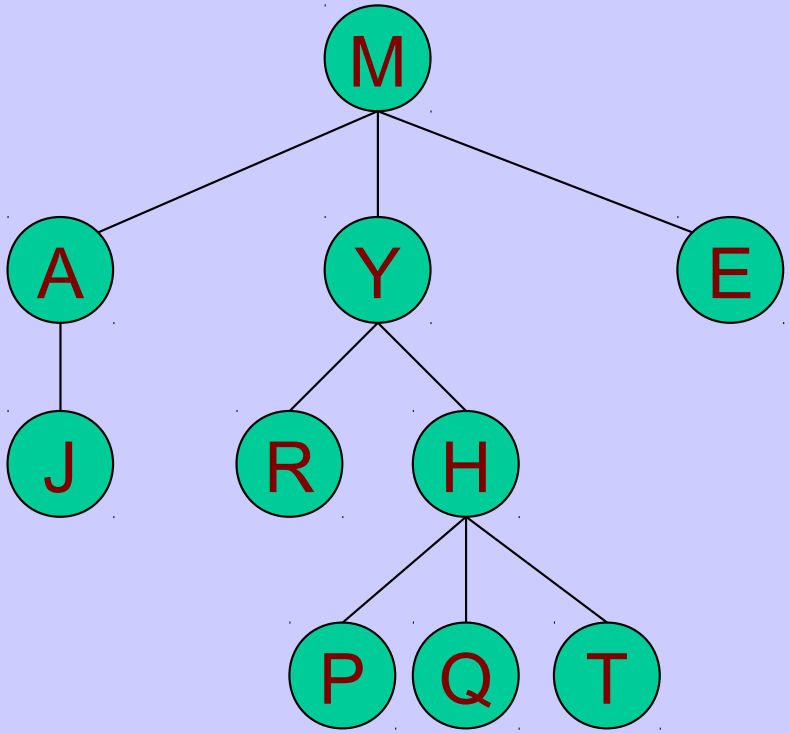
⋮

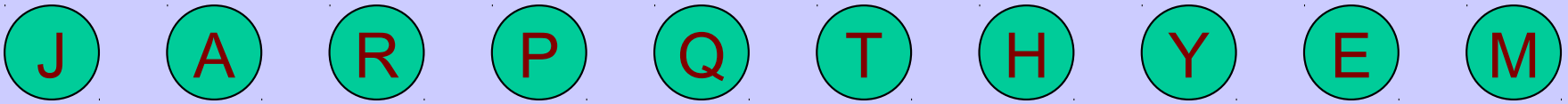
Step n: Visit T_n in postorder

Step n+1: Visit r



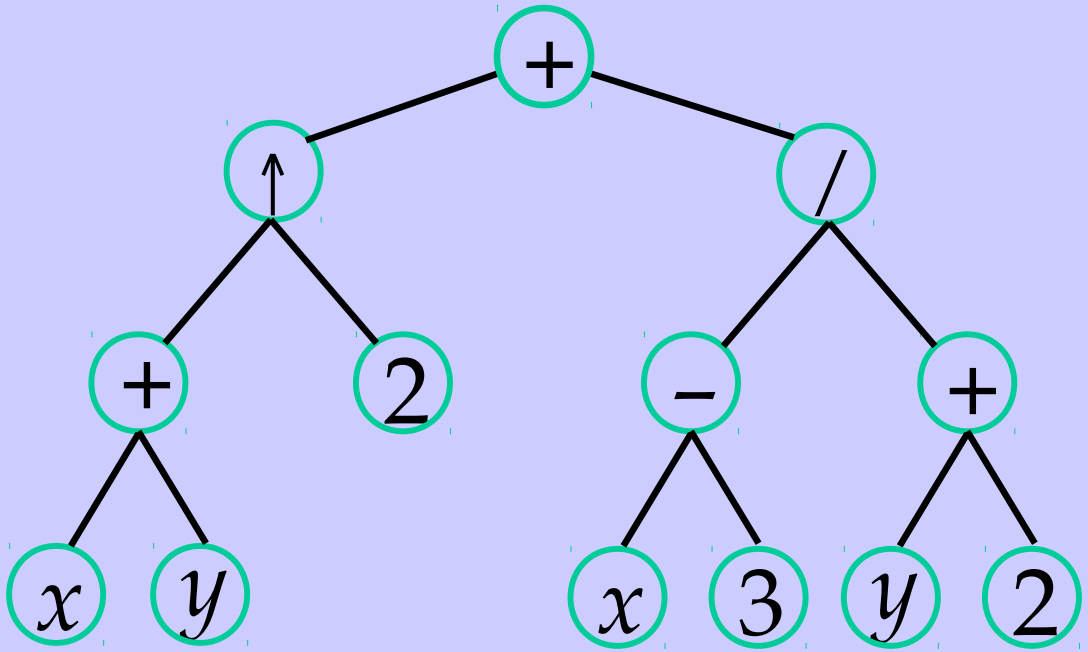
Example





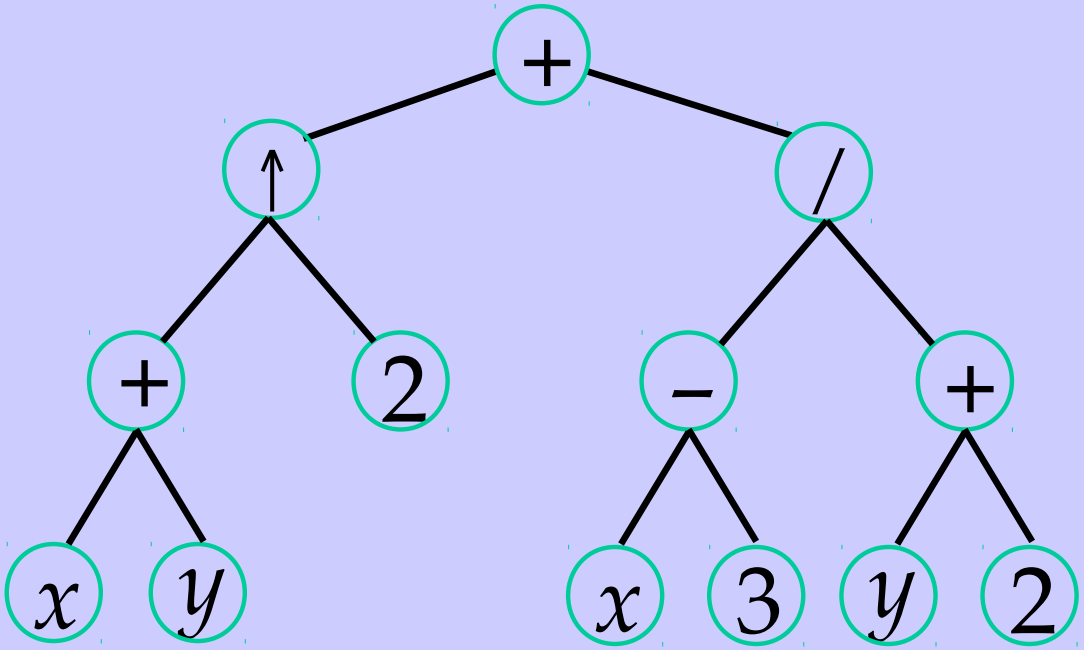
Example

$(x+y)^2 + (x-3)/(y+2)$



Infix Notation

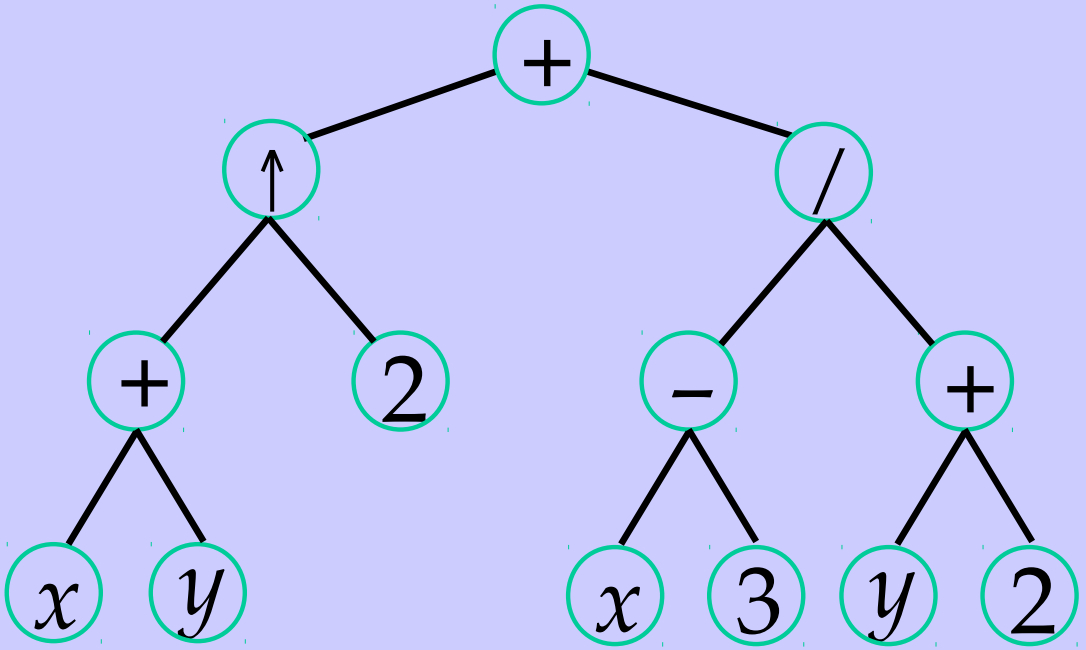
- Traverse in inorder (LVR) adding parentheses for each operation



$$(((x + y) \uparrow 2) + ((x - 3) / (y + 2)))$$

Prefix Notation (Polish Notation)

- Traverse in preorder (VLR)

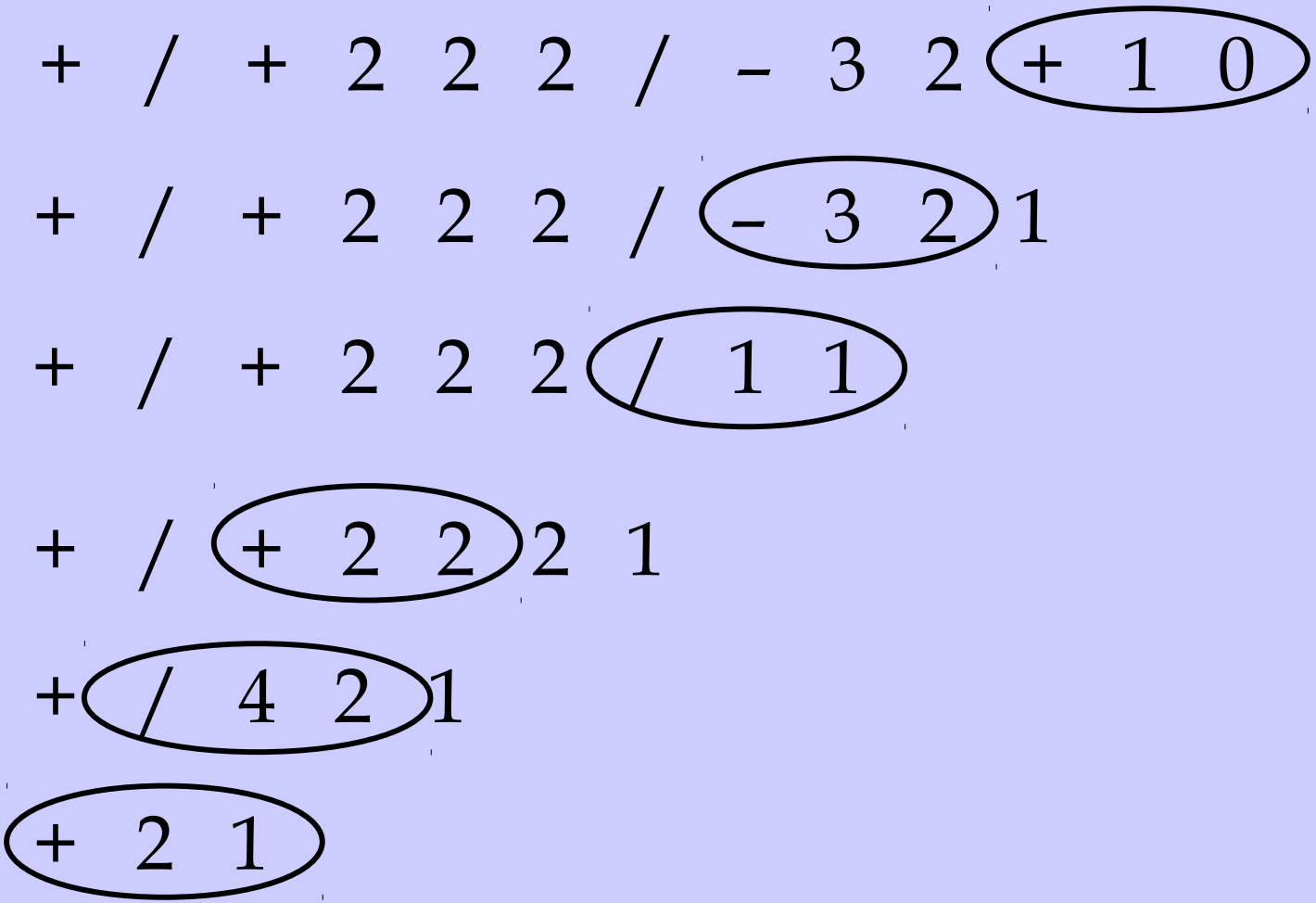


+ ↑ + x y 2 / – x 3 + y 2

Evaluating Prefix Notation

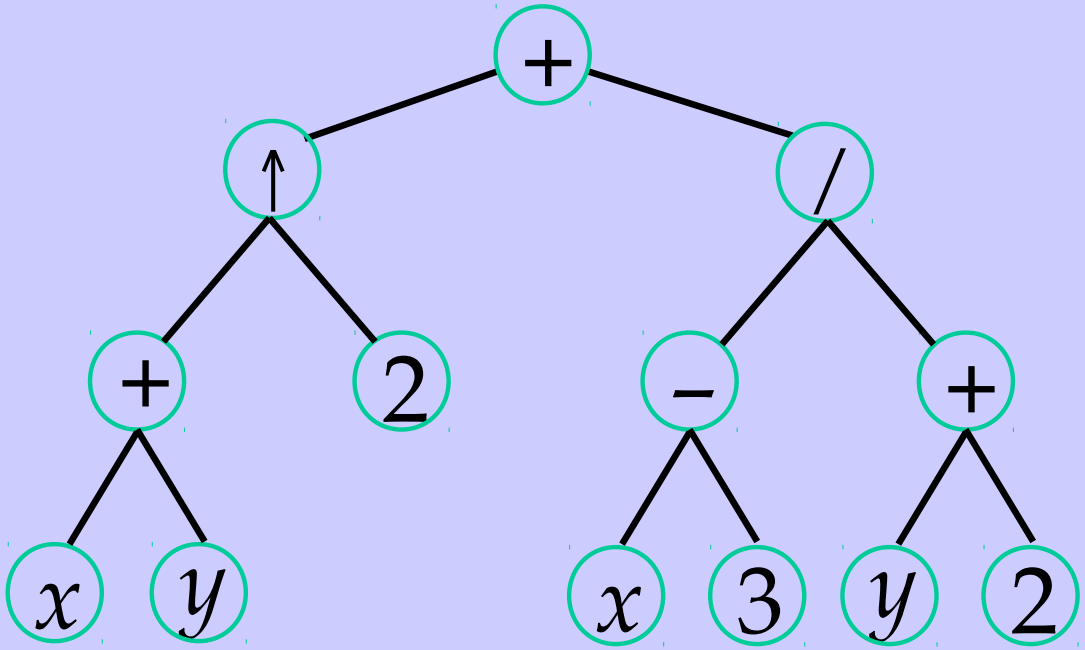
- In an prefix expression, a binary operator precedes its two operands
- The expression is evaluated right-left
- Look for the first operator from the right
- Evaluate the operator with the two operands immediately to its right

Example



Postfix Notation (Reverse Polish)

- Traverse in postorder (LRV)

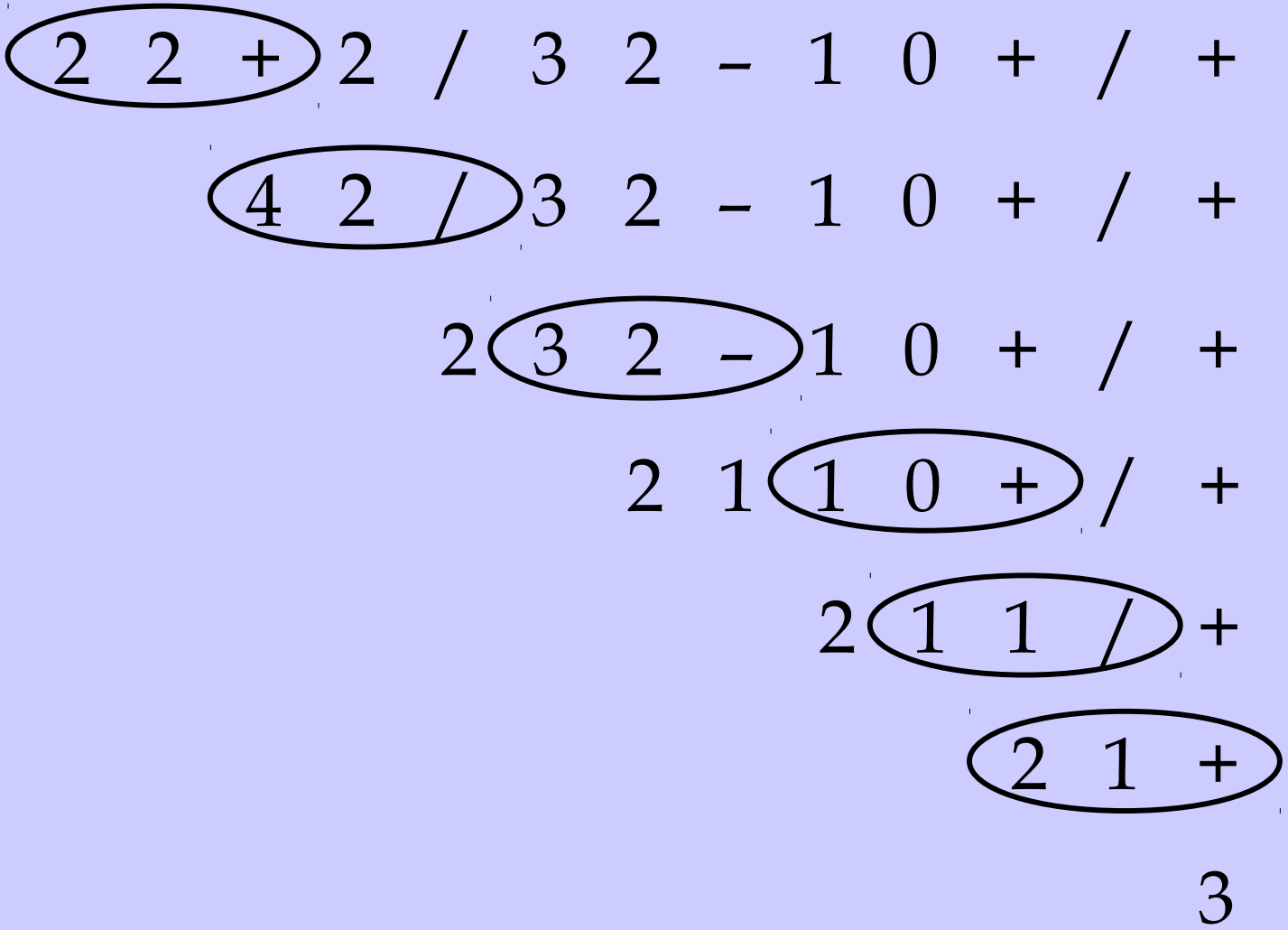


$$x\ y\ +\ 2\ \uparrow\ x\ 3\ -\ y\ 2\ +\ /\ +$$

Evaluating Postfix Notation

- In an postfix expression, a binary operator follows its two operands
- The expression is evaluated left-right
- Look for the first operator from the left
- Evaluate the operator with the two operands immediately to its left

Example



Thank You