

25 Lecture - CS301

Important Subjective

- 1. What is an expression tree?**
An expression tree is a binary tree representation of expressions, where the leaves are operands, and internal nodes are operators.
- 2. How can we evaluate an expression tree?**
We can evaluate an expression tree recursively by evaluating the left and right subtrees and then applying the operator in the root.
- 3. What is a prefix expression?**
A prefix expression is an expression where the operator comes before the operands, for example, + 2 3.
- 4. How do we convert a prefix expression to an expression tree?**
We start from the leftmost operand and create a new node for each operator encountered, with the left child being the next operand and the right child being the next operator or operand.
- 5. What is a postfix expression?**
A postfix expression is an expression where the operator comes after the operands, for example, 2 3 +.
- 6. How do we convert a postfix expression to an expression tree?**
We start from the leftmost operand and create a new node for each operator encountered, with the right child being the next operand and the left child being the next operator or operand.
- 7. What is an infix expression?**
An infix expression is an expression where the operator comes between the operands, for example, 2 + 3.
- 8. How do we convert an infix expression to an expression tree?**
We use the shunting-yard algorithm to convert an infix expression to postfix notation and then create an expression tree from the postfix expression.
- 9. What is the height of an expression tree?**
The height of an expression tree is the maximum depth of any leaf node in the tree.
- 10. What is the time complexity of evaluating an expression tree?**
The time complexity of evaluating an expression tree is $O(n)$, where n is the number of nodes in the tree.