Algorithm to construct an expression tree from a given postfix expression:

1. Create a Stack.

2. Read the given postfix one symbol at a time, and repeat Steps 3 to 4.

3. If the symbol is an operand, then

   - Create a single node expression tree with the operand.
   - Push the root pointer of the generated tree into the stack.

4. If the symbol is an operator, then

   - Pop out pointers to the two trees (T1(first popped out) and T2) from the stack.
   - Create a new tree with the operator as the root and whose left-child and right-child point to trees T2 and T1, respectively.
   - Push the root pointer of this new tree back into the stack.

5.The stack will contain a single tree; return it as the final expression tree.

Algorithm to compute value of an expression from the expression tree

```
Let ROOT be a pointer to the expression tree node.
1.  If ROOT is not NULL, then
        If symbol at ROOT is operand, then
            Return the operand.
2.  Else            // Symbol at ROOT is operator.
          Recursively compute the value of ROOT.left, operand1
          Recursively compute the value of ROOT.right, operand2
3.      Apply the operator at ROOT to the values stored in operand1 and operand2
4.      Return the result obtained in Step 3.
```