



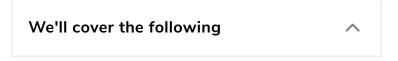






Solution Review: Evaluate Postfix Expression Using a Stack

This review provides a detailed analysis of a solution to the 'Evaluate Postfix Expression Using Stack' challenge.



- Solution: Numbers as Stack Elements
- Time Complexity

Solution: Numbers as Stack Elements

#

```
main.py
Stack.py
     from Stack import MyStack
  2
     def evaluate_post_fix(exp):
         stack = MyStack()
  5
         try:
  6
              for char in exp:
                  if char.isdigit():
                      # Push numbers in stack
  9
                      stack.push(char)
 10
                  else:
 11
                      # use top two numbers and evaluate
 12
                      right = stack.pop()
```

We check each **character** of the string from left to right. If we find a digit, it is pushed into the stack.

If we find an operand, we pop two elements from the stack (there have to be at least two present or else this postfix expression is invalid) and solve the expression. The resulting value is pushed back into the stack.

The process continues until we reach the end of the string.

Note: We have used a try except block. If the expression is correct, the result is returned and the try block is executed. Otherwise, it is caught in the except block.

Time Complexity#

Since we traverse the string of **n** characters once, the time complexity for this algorithm is O(n).

Interviewing soon? We've partnered with Hired so that companies apply to you instead of you applying to them. See how ①







← Back

Challenge 6: Evaluate Postfix Expressi...



Challenge 7: Next Greater Element Usi...



Completed



Report an Issue

