# CSC 172– Data Structures and Algorithms

Lecture #11 Spring 2018

Please put away all electronic devices



# Agenda

• Shunting Yard Algorithm

How to Implement Stack and Queue
 – Cost of various operations

# **SHUNTING YARD ALGORITHM**

#### Infix vs Postfix Expressions

- Infix:  $5 + 4 \times 5/2 3$
- Postfix: 5 4 5 \* 2 / + 3 -

- Infix: (5+4)\*5/2 3
- Postfix: 5 4 + 5 \* 2 / 3 -

# **Postfix Expression Evaluation Algorithm**

- Initialize an empty stack
- While (there is still a token to read)
  - read the token t
  - if t is an operand, push it onto the stack
  - if t is an operator,
    - pop two operands from the stack, compute the result (using t) // if there is division by zero, scream
    - push the result back onto the stack // if there is less than two operands, scream
- In the end, if there is one number in the stack, output it.
  - // If there is more than one number in the stack, scream.

### Example We will see

• Infix:

$$1 - (2 + 3 + (4 - 5 \times 6) \times 7) + 8 \times 9$$

Postfix (Reverse-Polish Notation):
1 2 3 + 4 5 6 × - 7 × + - 8 9 × +

Evaluate the following reverse-Polish(postfix) expression using a stack:

 $1 \ 2 \ 3 \ + \ 4 \ 5 \ 6 \ \times \ - \ 7 \ \times \ + \ - \ 8 \ 9 \ \times \ +$ 



### Push 1 onto the stack 1 2 3 + 4 5 6 × $-7 \times + -89 \times +$



### Push 1 onto the stack 1 2 3 + 4 5 6 × $-7 \times + -89 \times +$



Push 3 onto the stack 1 2 3 + 4 5 6 × -7 × + -8 9 × +



Pop 3 and 2 and push 2 + 3 = 51 2 3 + 4 5 6 × - 7 × + - 8 9 × +



Push 4 onto the stack 1 2 3 + 4 5 6 × -7 × + -8 9 × +



Push 5 onto the stack 1 2 3 + 4 5 6 ×  $-7 \times + -89 \times +$ 



Push 6 onto the stack 1 2 3 + 4 5 6 × - 7 × + - 8 9 × +



# Pop 6 and 5 and push $5 \times 6 = 30$ 1 2 3 + 4 5 6 × - 7 × + - 8 9 × +



# Pop 30 and 4 and push 4 - 30 = -261 2 3 + 4 5 6 × - 7 × + - 8 9 × +



Push 7 onto the stack 1 2 3 + 4 5 6 × -7 × + - 8 9× +



# Pop 7 and -26 and push $-26 \times 7 = -182$ 1 2 3 + 4 5 6 × - 7 × + - 8 9 × +



### Pop -182 and 5 and push -182 + 5 = -1771 2 3 + 4 5 6 × $-7 \times + -89 \times +$



Pop -177 and 1 and push 1 - (-177) = 178 1 2 3 + 4 5 6 × - 7 × + - 8 9 × +



Push 8 onto the stack 1 2 3 + 4 5 6 ×  $-7 \times + -8 9 \times +$ 



### Push 1 onto the stack 1 2 3 + 4 5 6 × $-7 \times + -89 \times +$



# Pop 9 and 8 and push 8 × 9 = 72 1 2 3 + 4 5 6 × $-7 \times + -89 \times +$



# Pop 72 and 178 and push 178 + 72 = 2501 2 3 + 4 5 6 × - 7 × + - 8 9 × +



### How about Infix Expression?

• Shunting yard algorithm

• Convert infix to postfix

• Or, evaluate infix expressions directly

- Invented by Edsger Dijkstra
- Named the "shunting yard" algorithm because its operation resembles that of a railroad shunting yard.

 A Shunting yard is a railway yard found at some freight train stations, used to separate railway cars onto one of several tracks.

# Rough Idea (Shunting Yard Algorithm)

- Use 2 stacks: an operand stack, an operator stack
- If tok is an operand, push it on operand stack
- Else if tok is one of + \* /
  - while (precedence(tok) ≤ precedence(stack.peek())
    - stack.pop()
  - Push tok on top of operator stack
- Else if tok is one of ( [ {
  - Push tok on top of operator stack
- Else if tok is one of ) ] }
  - Evaluate operators on top until ( [ { seen, match up

### Example We will see

• Infix:

$$1 - (2 + 3 + (4 - 5 \times 6) \times 7) + 8 \times 9$$

Postfix (Reverse-Polish Notation):
1 2 3 + 4 5 6 × - 7 × + - 8 9 × +

### What would happen?



### What would happen?



### Quiz

- Next Token +
- Stack Contains \*(top), +

• What should you do?

Pop \*, + and then push +

 $1 - (2 + 3 + (4 - 5 \times 6) \times 7) + 8 \times 9$ 

Currently Handling	Operand List	Operator Stack
1	1	

 $1 - (2 + 3 + (4 - 5 \times 6) \times 7) + 8 \times 9$ 

Currently Handling	Operand List	Operator Stack
-	1	-

 $1 - (2 + 3 + (4 - 5 \times 6) \times 7) + 8 \times 9$ 

Currently Handling	Operand List	Operator Stack
(	1	- (

$$1 - (2 + 3 + (4 - 5 \times 6) \times 7) + 8 \times 9$$

Currently Handling	Operand List	Operator Stack
2	12	- (
Currently Handling	Operand List	Operator Stack
--------------------	--------------	----------------
+	12	- ( +

Currently Handling	Operand List	Operator Stack
3	123	- ( +

Currently Handling	Operand List	Operator Stack
+	123+	- ( +

Currently Handling	Operand List	Operator Stack
(	123+	- ( + (

Currently Handling	Operand List	Operator Stack
4	1 2 3 + 4	- ( + (

Currently Handling	Operand List	Operator Stack
-	123+4	- ( + ( -

Currently Handling	Operand List	Operator Stack
5	1 2 3 + 4 5	- ( + ( -

Currently Handling	Operand List	Operator Stack
*	1 2 3 + 4 5	- ( + ( - *

Currently Handling	Operand List	Operator Stack
6	1 2 3 + 4 5 6	- ( + ( - *

Currently Handling	Operand List	Operator Stack
)	123+456*-	- ( +

Currently Handling	Operand List	Operator Stack
*	1 2 3 + 4 5 6 * -	- ( + *

Currently Handling	Operand List	Operator Stack
7	1 2 3 + 4 5 6 * - 7	- ( + *

Currently Handling	Operand List	Operator Stack
)	1 2 3 + 4 5 6 * - 7 * +	-

Currently Handling	Operand List	Operator Stack
+	123+456*-7*+ -	+

Currently Handling	Operand List	Operator Stack
8	1 2 3 + 4 5 6 * - 7 * + - 8	+

Currently Handling	Operand List	Operator Stack
*	1 2 3 + 4 5 6 * - 7 * + - 8	+ *

Currently Handling	Operand List	Operator Stack
9	123+456*-7*+-89	+ *

Currently Handling	Operand List	Operator Stack
9	123+456*-7*+-89*+	

$$1 \ 2 \ 3 \ + \ 4 \ 5 \ 6 \ \times \ - \ 7 \ \times \ + \ - \ 8 \ 9 \ \times \ +$$

### **Practice Problem**

• 24 + 4-15 \* 8 / (2 + 5 \* 2 - 8) + 4

### 24 4 + 15 8 \* 2 5 2 \* + 8 - / - 4+

Currently Handling	Result Stack	Operator Stack
1	1	

Currently Handling	Result Stack	Operator Stack
-	1	-

Currently Handling	Result Stack	Operator Stack
(	1	- (

Currently Handling	Result Stack	Operator Stack
2	12	- (

Currently Handling	Result Stack	Operator Stack
+	12	- ( +

Currently Handling	Result Stack	Operator Stack
3	123	- ( +

Currently Handling	Result Stack	Operator Stack
+	15	- ( +

Currently Handling	Result Stack	Operator Stack
(	15	- ( + (

Currently Handling	Result Stack	Operator Stack
4	154	- ( + (

Currently Handling	Result Stack	Operator Stack
-	154	- ( + ( -

Currently Handling	Result Stack	Operator Stack
5	1545	- ( + ( -

Currently Handling	Result Stack	Operator Stack
*	1545	- ( + ( - *

Currently Handling	Result Stack	Operator Stack
6	15456	- ( + ( - *

Currently Handling	Result Stack	Operator Stack
)	15456	- ( + ( - *

Currently Handling	Result Stack	Operator Stack
)	1 5 4 30	- ( + ( -

Currently Handling	Result Stack	Operator Stack
)	15-26	- ( +

Currently Handling	Result Stack	Operator Stack
*	15-26	- ( + *
Currently Handling	Result Stack	Operator Stack
--------------------	--------------	----------------
7	1 5 -26 7	- ( + *

Currently Handling	Result Stack	Operator Stack
)	1 5 -26 7	- ( + *

Currently Handling	Result Stack	Operator Stack
)	1 5 -182	- ( +

Currently Handling	Result Stack	Operator Stack
)	1 -177	-

Currently Handling	Result Stack	Operator Stack
+	178	+

Currently Handling	Result Stack	Operator Stack
8	178 8	+

Currently Handling	Result Stack	Operator Stack
*	178 8	+ *

Currently Handling	Result Stack	Operator Stack
9	178 8 9	+ *

<b>Currently Handling</b>	Result Stack	Operator Stack
	178 72	+

Currently Handling	Result Stack	Operator Stack
	250	

# Project 2

- You can use either approach:
  - Convert the infix expression into Postfix and then evaluate the postfix expression

- Or

 Evaluate the infix expression using the Single pass method.

# Acknowledgement

- Douglas Wilhelm Harder.
  - Thanks for making an excellent set of slides for ECE
    250 Algorithms and Data Structures course
- Prof. Hung Q. Ngo:
  - Thanks for those beautiful slides created for CSC 250 (Data Structures) course at UB.
- Many of these slides are taken from these two sources.