COP 3502 9/11/2023

1. Recitation Reminders
   - Get quizzes back
   - Get into group
2. P1 due soon.
3. P2 T/W -> Queues (W, F)
4. STACKS
   - Abstract Data Type (ADT)
   - Operations it supports
   - 2 applications
   - 2 possible implementations

ADT - Don't know <u>how</u> data is stored, but the <u>behaviors</u> are described.

1) push object onto top
2) pop the top item off the stack
*3) size function
4) full?
5) empty?

ought to be O(1) time!

~~X~~
6
5

Stack
push(5)
push(6)
push(7)
x = pop()
x [7]

Linked List Implementation

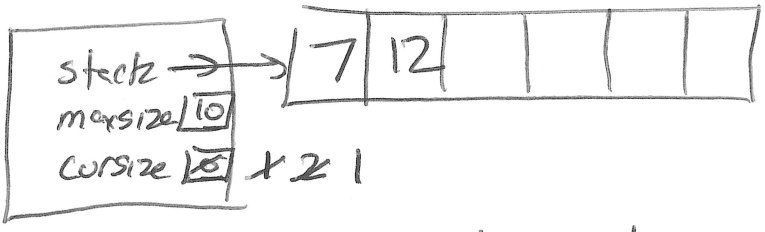
push = insert front
pop = delete front

push(7)
push(12)
x = pop()
x [12]
push(3)
push(8)

ptr → X
ptr → [7|X]
ptr → [12|→][7|X]
ptr → [7|X]
ptr → [3|→][7|X]
ptr → [8|→][3|→][7|X]

Stack: Last In, First Out (LIFO)

Array Implementation



X [12]
↓ in index

push → add item to index cursize.
(if array is full, realloc twice the size)

pop → return the item from index cursize-1.
~~them~~ before you return subtract 1 from cursize.

1st Application: Eval Postfix

7 6 + (operator is at the end)

Infix op1 op op2 ⟺ Postfix

7 + 6 ⟺ 7 6 +

Why postfix?

Infix ambiguous 3 + 7 * 5

→ order of operations is what removes ambiguity

Postfix 3 7 5 * + 3+(7*5)

3 7 + 5 * (3+7)*5

7 6 3 + * 2 3 - 1 + + 7 /

Algorithm

Read expr left to right

(1) if operand (number), push onto stack

(2) if operator (+, -, *, /), pop off top 2 items stack (op2, op1) then calculate op1 op op2, push result back top of the stack.

6+3 7+9

Stack	Stack	Stack	Stack
3		3	
6	9	2 (-1)	
7	7	63	63

1			
-1	0	7	
63	63	63	9
Stack	Stack	Stack	Stack
	-1+1		63/7

End als: should be 1 # left on stack ⇒ ANSWER

If ever try pop empty stack OR end w/more 1 item on stack, the expression is an invalid postfix expression

Sample Expression for program:

2 4 * 1 + 2 4 2 - + /

2 4 -3 1 -1 2 4 2 -2 -1 -4

4-2

		2			
		4	2		
4	1	2	2	4	
2	8	9	9	9	2 (int div)
Stack	Stack	Stack	Stack	Stack	Stack