Recitation 3: Stacks (Spring 2023 TA Version)

1. [10 minutes] Write the following push(), pop(), peek() methods to implement a stack using a doubly linked list (head is top of the stack)

public class intNode {

private int number; private intNode next; private intNode prev;

public intNode(int number) { this.number = number; next = null;

prev = null;

}

}

public class intStack {

private intNode top; public intStack() {

top = null;

}

public void push(int number) {

intNode add = new intNode(number);

if(head == null){

head , cursor, tail = add;

}else{

head.setPrev(add);

add.setNext(head);

head = add;

}

}

public int pop() {

intNode previousHead = head;

intnode newHead = prevHead.getNext();

newHead.setPrev(null);

head = newHead;

return prevHead.getData();

}

public int peek() {

return head.getData();

OR can do

data = pop();

push(data);

return(data);

}

}

1. [5 minutes] Write the order of complexity in Big-O for the following operations
   1. Searching for a value in a stack.

O(n)

* 1. Reversing an array using a stack.

O(2n) -> O(n)

* 1. Evaluating a postfix expression using a stack.

O(n)

* 1. Adding an element to a stack.

O(1)

* 1. Retrieving the bottom-most value in a stack.

O(n)

* 1. Removing a single element from a stack.

O(1)

1. [5 minutes] Evaluate the following postfix expression:

5 3 8 \* 9 15 \* 5 / + +

5+ (3\*8) + (9\*15/5) = 56

1. [5 minutes] Evaluate the following prefix expression:

+ - \* 8 3 / 6 3 4

( (8\*3) - (6/3) ) + 4 = 26

1. [5 minutes] Convert the following prefix expression to postfix:

/ \* A - B C \* / D E F

A\*(B-C) / ( (D/E)\*F ) INFIX

ABC-\* FDE / \* / POSTFIX

1. [5 minutes] Write a method to push a node to the bottom of a stack.

public void insertBottom(Node node, Stack s) {

Stack s2 = new Stack();

While(!isEmpty()){

S2.push(s2.pop);

}

s.push(node);

While(!isEmpty()){

S2.push(s2.pop);

}

**}**

1. [15 minutes] Write a method that evaluates a postfix expression using a stack and returns the result. Assume a valid postfix string is given and assume you have a stack with the following methods: .push(), .pop(), .peek():

Public static int evaluatePostfix (String str) {

**}**