CSI 2110 Tutorial (Section A)

Yiheng Zhao
yzhao137@uottawa.ca
Office Hour: Fri 13:00-14:00
Place: STE 5000G

2019-09-27

Previous Exercise:

4.45. An array A contains n-1 unique integers in the range [0, n-1], that is, there is one number from this range that is not in A. Design an O(n) – time algorithm for finding that number. You are only allowed to use O(1) additional space besides the array A itself.

Hint: because the elements in A are unique, compute the sum of all elements with total available element in the range.

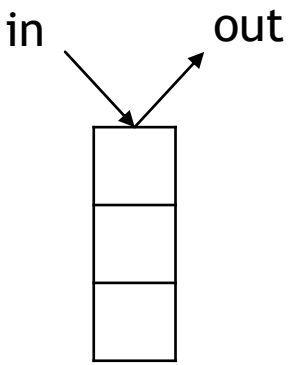
Example: assume n = 4 and A = {0, 3, 1}, range = [0, 3]
the sum of A is 0+3+1 = 4; the sum of total available element is 0+1+2+3 = 6. The missing element of array A is 6–4 = 2

Solution:

```
int sol(int[] A, int n){
    int tot = n*(n-1)/2, sum = 0;
    for(int i=0; i<A.length; i++)
        sum += A[i];
    return tot – sum;
}
```

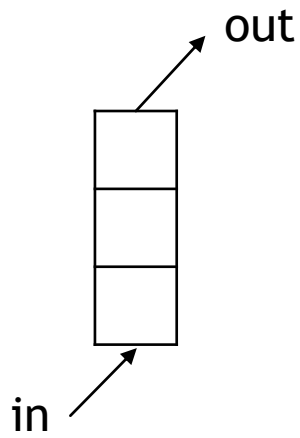
Review Stack, Queues, Deques

Stack



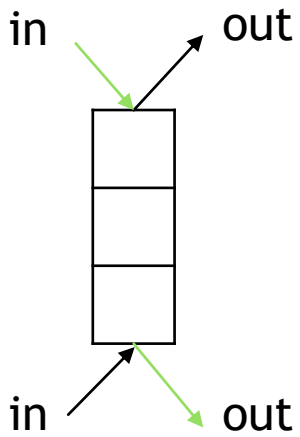
first in last out

Queue



first in first out

Deque



Main Functions:

push(ele)	enqueue(ele)	insertFirst(ele)
pop()	dequeue()	insertLast(ele)
top()	front()	removeFirst()
		removeLast()
		first()
		last()

Review List

Array List:
(Continuous Address)

get(index)
set(index, val)
add(index, val)
remove(index)

Linked List:
Positional List
(Discrete Address)

header

trailer

first()
last()
before(node)
after(node)

addFirst(val)
addLast(val)
addBefore(node, val)
addAfter(node, val)

set(node, val)
remove(node)

Exercise 6.1

Suppose an initially empty stack S has performed a total of 25 push operations, 12 top operations, and 10 pop operations, 3 of which returned Null to indicate an empty stack. What is the current size of S?

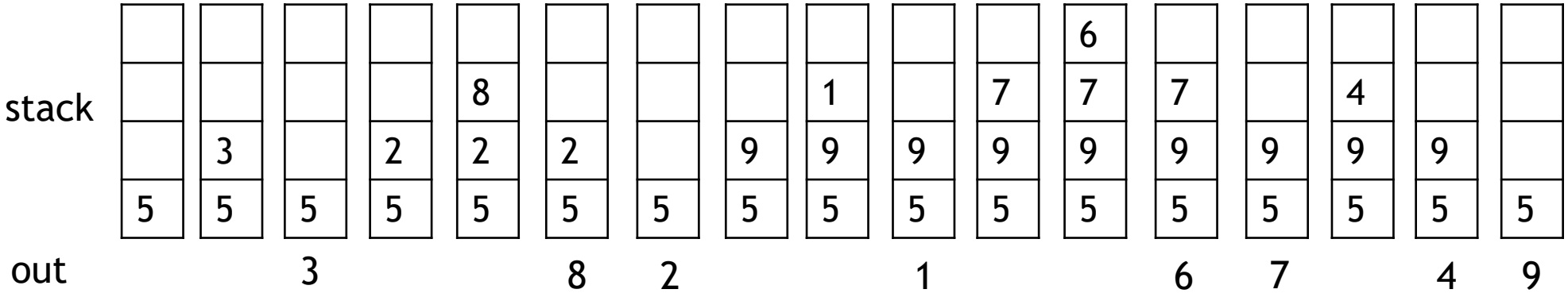
Hint: only consider how many of elements are added and how many are actually removed.

Solution:

size = 25 – (10 - 3) = 18

Exercise 6.3

What values are returned during the following series of stack operations, if executed upon an initially empty stack? push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop()?



Exercise 6.13

Suppose you have a deque D containing the numbers (1,2,3,4,5,6,7,8), in this order. Suppose further that you have an initially empty queue Q. Give a code Fragment that uses only D and Q (and no other variables) and results in D storing the elements in the order (1,2,3,5,4,6,7,8).

One Solution:

1) Put 1, 2, 3 from D into D (from the last position):

2) Put 4, 5 from D into Q

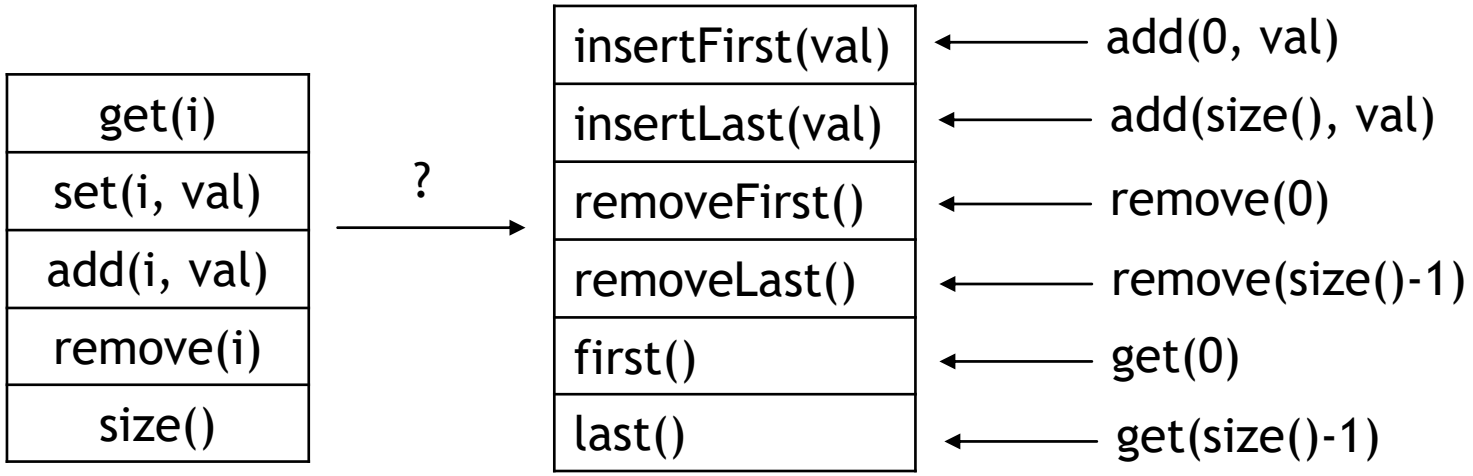
3) Put 4 from Q into D (from the first pos) and insert 5 from Q in D (from the last pos)

4) Insert 4, 6, 7, 8 from D into D (from the last position)

Exercise 7.3

Give an implementation of the deque ADT using an array list for storage

Hint:



Exercise 7.11

Describe an implementation of the positional list methods addLast and addBefore realized by using only methods in the set {isEmpty, first, last, before, after, addAfter, addFirst}

Solution:

```
addLast(val):
    if(isEmpty())
        addFirst(val)
    else
        addAfter(last(), val)
```

```
addBefore(node, val):
    if(node == first())
        addFirst(val)
    else
        addAfter(before(node), val)
```

Exercise 7.12

Suppose we want to extend the PositionalList abstract data type with a method, indexOf(node), that returns the current index of the node. Show how to implement this method using only other methods of the PositionalList interface

Hint: you can use functions as below:

first()	addFirst(val)	set(node, val)
last()	addLast(val)	remove(node)
before(node)	addBefore(node, val)	
after(node)	addAfter(node, val)	

One Solution:

```
indexOf(node):
    int num = 0;
    pointer = first()
    while(pointer != node){
        num ++;
        pointer = after(pointer);
    }
    return num;
```

Exercise R-6.14

Repeat exercise 6.13 using the deque D and an initially empty stack S.