PRACTICE PROBLEMS: LINEAR DATA STRUCTURES

A simple rule I follow: Honor the coder and their code. The constraints they endured are not ours to know. Make it better if you can.. —Woody Zulill

Course: CS 5002
Fall 2018
Due: No due date

OBJECTIVES

After you complete this assignment, you will be comfortable with:

- The notion of a data structures
- The need for different data structures
- Linked lists
- Doubly linked lists
- Stacks
- Queues

RELEVANT READING

- Introduction to Data Structures
- Basic Data Structures in Python
- Linked List
- Doubly Linked List
- Stack
- Queue

EXERCISES

Question 1

Write a function `count(int number)` that counts the number of times a given integer `number` occurs in a list.

(a) What is the run time of the given function, expressed as function as the size of the list, n.

The pseudocode for method `count(int number) is given below.`

```
    int count(int number) {
    Node currentNode = this;
    int count = 0;

    while (nodeNode != NULL) {
    if (current.data == number):
      count++;
    currentNode = currentNode.next;
  }
  return count;
}
```

Question 2

Write a function *getNth(int n)* that takes an integer index `n`, and returns the data value stored in the node at the `n`-th position. Your function should follow 0-based indexing, so for a list 42, 13, 666 `getNth()` with index 1 should return 13.

The pseudocode for method `getNth(int n) is given below.`

```
  int getNth(int index) {
  Node currentNode = this;
  int indexCounter = 0; // the index of the node we're currently looking at

  if(index > this.length)
    print("Error:index does not extist!")
    return NULL;

  while (currentNode != NULL) {
    if (count == index)
      return(currentNode.data);
    indexCounter++;
    currentNode = currentNode.next;
  }
}
```

Question 3

Write a function `insertNth(int n, Node node)` which inserts a new node at index `n` within a list. The caller can specify any index in the range [0..length], but should not be allow to specify an index outside of that range.

The pseudocode for method `insertNth(int n, Node n) is given below.`

```
void insertNth(int index, Node n) {
  Node currentNode = this;
  int i;

  // position 0 is a special case...
  if (index == 0){}
     pushNode(n);
  else {
    for (int = 0;  i < index-1; i++) {
      if(currentNode != NULL);
        currentNode = currentNode.next;
    }
    if(currentNode != NULL)
      currentNode.push(data);
```

Question 4

Write a function `sortList()` which, given a list, rearranges its nodes so that they are sorted in increasing order.

Solution omitted.

Question 5

Given two lists, write a function `mergeLists(list1, list2)` that merges their nodes to make one list, taking nodes alternately between two lists. So, given two lists with elements 1, 5, 10 and 2, 4, 6, functions `mergeLists(list1, list2)` should return 1, 2, 5, 4, 10 ,6. If one list is shorted than the other, once you run out of the elements in one list, take all of the remaining elements from the other list.

(a) What is the run time of the given function, expressed as function as the size of the lists, n_1 and n_2.

The pseudocode for method `mergeLists(list1, list2) is given below.`

```
  List SortedMerge(List list 1, List list2) {
  Node dummy;   // a dummy first node to hang the result on
  Node tail = dummy;      // Points to the last result node --
                    // so tail.next is the place to add
                    // new nodes to the result.
  dummy.next = NULL;

  while (1) {
```

```
  if (list1 == NULL) {  // if either list runs out, use the other list
   tail.next = list2;
   break;
  }
  else if (list2 == NULL) {
   tail.next = list1;
   break;
  }
  if (list1.data <= list2.data) {
   MoveNode(tail.next), list1);
  } else {
   MoveNode(tail.next), list2);
 }
 tail = tail.next;
}
return(dummy.next);
}
```

Question	Points	Score
1	10	
2	10	
3	15	
4	15	
5	20	
Total:	70	