School of Computer Science

COMP20280: Data Structures and Algorithms I
Semester II

Assignment 1
On: 27/02/2020
Due: NA

The questions here should be tackled during the practical session. You can discuss the solutions with me or the TA's and there is no grade specifically associated with the work. Of course, you need to have a code repository of data structures at the end of the module which is assessed, and this is a good place to start building it.

Q1: _____ (**0points**)

Following the lecture notes, implement the full set of methods in the `SinglyLinkedList` class.

The `SinglyLinkedList` class should implement the `List` interface and the `Iterable` interface, and be a generic class. The `main` function should contain the following testing code:

```
public static void main(String[] args) {
        SinglyLinkedList<Integer> ll = new SinglyLinkedList<Integer>();
        //LinkedList<Integer> ll = new LinkedList<Integer>();

        ll.addFirst(0);
        ll.addFirst(1);
        ll.addFirst(3);
        ll.addFirst(4);
        ll.addFirst(5);
        ll.add(3, 2);
        System.out.println(ll);

        ll.addFirst(-100);
        ll.addLast(+100);
        System.out.println(ll);

        ll.removeFirst();
        ll.removeLast();
        System.out.println(ll);

        // Removes the item in the specified index
        ll.remove(2);
        System.out.println(ll);

        ll.removeFirst();
        System.out.println(ll);

        ll.removeLast();
        System.out.println(ll);

        ll.removeFirst();
        System.out.println(ll);

        ll.addFirst(9999);
        ll.addFirst(8888);
        ll.addFirst(7777);
```

```
        System.out.println(ll);
        System.out.println(ll.get(0));
        System.out.println(ll.get(1));
        System.out.println(ll.get(2));
        System.out.println(ll);
    }
```

Q2: _____ (**0points**)

Now the `SinglyLinkedList` is working, you should implement the `DoublyLinkedList` class. The `DoublyLinkedList` class should implement the `List` and `Iterable` interfaces, and the `main` should be the same as for the `SinglyLinkedList` (the results should also be the same). Implement the `DoublyLinkedList` class with `header` and `trailer` nodes.

Q3: _____ (**0points**)

Now implement the `CircularlyLinkedList` class. This class is very similar to the `SinglyLinkedList` class, with the addition of a `rotate` method. Use the `main` from the `SinglyLinkedList` class for testing (the results should also be the same).

Q4: _____ (**0points**)

What is the difference between a singly linked list and a circularly linked list?

Q5: _____ (**0points**)

In what situations would you prefer to use a linked list to an array?

Q6: _____ (**0points**)

Describe 2 possible use-cases for a circularly linked list (2-3 sentences for each).

Q7: _____ (**0points**)

Write a function which merges two sorted linked lists. For example, if the two lists are:

```
l1 = {2, 6, 20, 24};
l2 = {1, 3, 5, 8, 12, 19, 25};
```

The resulting list is:

```
result = l1.sortedMerge(l2);

// result = {1, 2, 3, 5, 6, 8, 12, 19, 20, 24, 25};
```