Introduction to Computer Science II
Lab 5
A Nice Linked List

1 The `NiceList` interface

We start with the `NiceList` interface[1], which defines a *linear, indexed container* that can store any collection of `Objects`. Like any interface, it doesn't define **how** those values are stored; instead, it defines how the container must behave by specifying the methods that it must implement:

- `public void insert (int index, Object value)`
  Insert a `value` at the given `index`.
- `public Object remove (int index)`
  Remove and return the value at the given `index`.
- `public void set (int index, Object value)`
  Replace the value at the given `index` with the new `value`.
- `public Object get (int index)`
  Return the value found at the given `index`.
- `public int length ()`
  Return the current length of the list.

Any class that implements these methods with this behavior is a `NiceList`.

2 `NiceArrayList` and `NiceLinkedList`

For this assignment, there are two classes that implement the `NiceList` interface: `NiceArrayList` and `NiceLinkedList`.

The `NiceArrayList` class uses basic arrays to store pointer to `Object` values. The array is often larger than needed, and the `_length` data member keeps track of how many values are actually stored at any given moment. When the array gets full and another value is inserted, the a new array, double the size of the old one, is created, and the pointers are copied from

[1] We use the silly prefix `Nice` because the interface `List` is already defined in Java. The `NiceList` interface is our simplified variant of the standard `List` interface.

the old to the new. This class is an example of one way to implement a `NiceList`. It is completely written, and you can look through it to see how it works and behaves.

The `NiceLinkedList` is only partially implemented. It uses a linked list (where the links are `NiceLink` objects) to store its pointers to `Object` values. The methods `set()` and `get()` are already implemented, as is the private helper method `walkTo()`. You can look at these to see how your code can move through the chained sequence of `NiceLink`s.

3 What you must do

Get the code: Use the following link to download a zip file with a bunch of source code:

`https://bit.ly/AMHCS-2020S-112-l5`

Complete the linked list implementation: You must complete write the `insert()` and `remove()` methods of the `NiceLinkedList` class.

A few critical things to notice about the linked list class that you must complete:

- There is a `_length` data member. Notice that some methods depend on this variable being up-to-date. The methods you are writing already contain code to update this data member; be mindful and when and how this value changes when you write and debug your code.
- The methods you must write must implement a specific behavior. Check the interface itself (in `NiceList.java`) and examine the comments to see what the behavior of each should be.
- There is no class with a `main()` method. Write one, and use it to test your own code.

4 How to submit your work

Submit your `NiceLinkedList.java` file via the CS submission system:

`https://www.cs.amherst.edu/submit`

This assignment is due on Sunday, Mar-29, 11:59 pm.