CS 146

Data Structures and Algorithms

Summer Semester 2015

Department of Computer Science
San José State University
Instructor: Ron Mak

Homework #2
Indexed List

Assigned:	Thursday, June 11
Due:	Monday, June 22 at 11:59 pm
	100 points max

The purpose of this assignment is to create a new **`List`** implementation that will combine the best performance features of **`ArrayList`** and **`LinkedList`**. Can we design a new **`List`** type that is fast for node access at an arbitrary position and for node insertions and deletions?

Some ideas for a new **`List`** implementation:

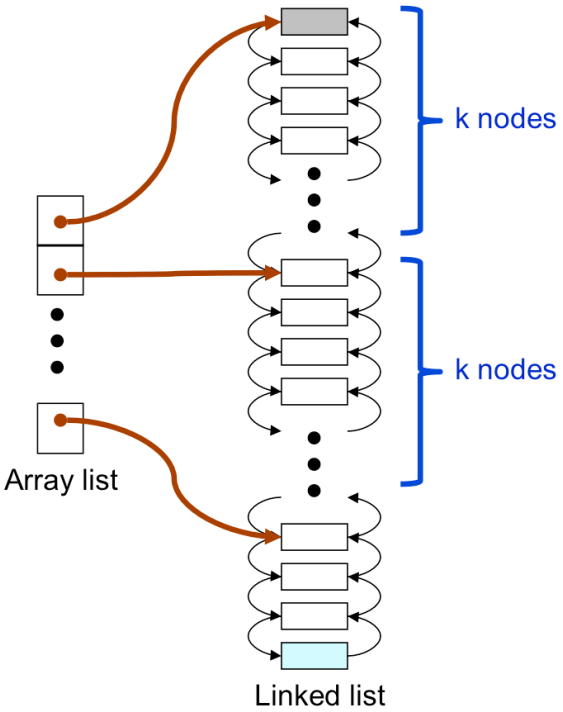
- Use an array list to speed up node access.
- Do insertions and deletions with a linked list.
- Take advantage of list iterators.

A new `IndexedList` type

Keep data in the linked list of N nodes. Create a separate array list whose elements are pointers to the linked list nodes. Have one pointer element for every k linked list nodes. What's an optimal value for k? What else should be in each array list element?

Suppose you need to access the i^{th} list node. Choose the array element that points to a linked list node that's closest to the i^{th} node. (How do you choose this element?) Follow the pointer to the node. Then use a list iterator to move forward or backwards to the i^{th} node.

How much does this improve node access time?



Design and implement this new **`IndexedList`** data type. It must implement the **`List`** interface. You can have the data nodes hold only **`Integer`** data – in other words, your array list and linked list implement **`List<Integer>`**.

The array list of pointers should be a hidden implementation artifact. There should be no access to the array list by programs that use your **`IndexedList`** data type.

Run tests to verify that

- You can **`get()`** and **`set()`** an arbitrary data node.
- You can **`add()`** and **`remove()`** an arbitrary data node.

Time how long these operations take with various sizes N of the linked list and different values of k for the array list. Is there an optimal value for k relative to N? What is the growth rate of $T(N)$ for node access and node deletion?

You only need to implement the **`List`** methods that you need for this assignment, such as **`get()`**, **`set()`**, **`add()`**, **`remove()`**, **`clear()`**, and **`size()`**. You can "dummy out" the remaining methods. Examples:

```
@Override public boolean addAll(Collection<? extends Integer> c) { return false; }
@Override public boolean contains(Object o) { return false; }
@Override public boolean containsAll(Collection<?> c) { return false; }
@Override public int indexOf(Object o) { return 0; }
@Override public boolean isEmpty() { return false; }
```

Teamwork

You may work individually as a team of one, or you can partner with another student as a team of two.

You can be on only one team at a time. If you partner with someone, both of you will receive the same score for this assignment. You'll be able to choose a different partner or work alone for subsequent assignments.

What to turn in

Create a zip file containing:

- Your Java source files.
- A sample output file. Use output redirection, or cut-and-paste into a text file. The output should show that you can **`get()`**, **`set()`**, **`add()`**, and **`remove()`** arbitrary nodes.
- A text file containing test output showing timings for data node access and insertions and deletions for various values of N and k. What is an optimal value for k?
- A short report (at most 3 pages) that describes your conclusions from doing this assignment.

Name the zip file after yourself or yourselves.
Examples: **`smith.zip`**, **`smith-jones.zip`**

Each team should email the zip file to ron.mak@sjsu.edu. Your subject line must be:

```
CS 146 Assignment #2 Your name(s)
```

Example:

```
CS 146 Assignment #2 Mary Smith & John Jones
```

If you work with a partner, you should email only one assignment between the two of you. Whoever emails the assignment should CC the partner so that when I send you your team score, I can just do a “Reply all”.