CS1083 Assignment #9 - Winter 2021

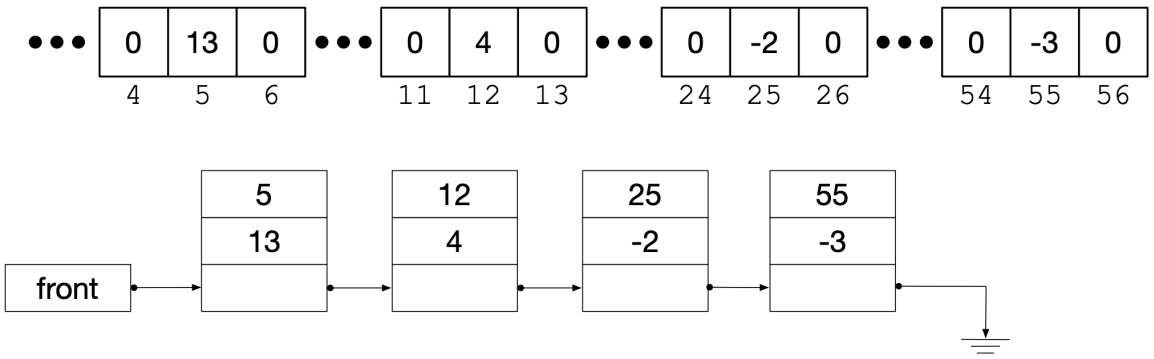
Due: Friday, March 26th before 4:00pm (Atlantic), submitted in the Assignment 9 dropbox in Desire2Learn. (Read the submission instructions at the end of this document carefully).

The purpose of this assignment is to introduce you to working with linked lists, including using recursion

This assignment is to be done individually. If you have questions, direct them to a tutor/assistant during a help session in the "Faculty of Computer Science Student Success Centre" team or to your course instructor.

Sparse Arrays

A sparse array is an array in which most of the elements are zero. To avoid the inefficiency of storing mostly zeros, the array can be represented as an ordered linked list of the nonzero elements, where each node contains both the index and the value, as shown below. Note that the list is ordered according to the index of each node.



Write a class called **Sparse.java** that implements storage and operations on a sparse array of integers represented as a linked list. You must implement your own linked list and are not permitted to use the java.util.LinkedList class. Your implementation must include the following methods (in addition to the constructor):

readList(String filename): Reads a file containing a sparse array represented as index,value pairs and inserts these pairs into a linked list.

mergeLists(Sparse other): Merges an other list with this list by inserting the values from the other list into this list. Makes use of the **insertValue()** method.

insertValue(int ind, int val): Inserts index and value into the list in order of the index. Adds value to node if node with same index exists. Removes node if value is or becomes zero. For instance, in the list pictured above, inserting the element 5,2 would result in the first node's value becoming 15, and inserting the element 25,2 would result in the node with index 25 being deleted.

printRecR(Node head): recursively prints the nodes in order, from first node to last. This is a private method that should be called by a public method **printRec()**

printRecBackwardsR(Node head): recursively prints the nodes in reverse order, from last node to first. This is a private method that should be called by a public method **printRecBackwards()**

Testing

Write a driver called **TestSparse.java** that reads two files containing sparse arrays, merges them, and prints the result forwards and backwards. Two files are shown below, which you can use to test your implementation. Include one other pair of files, so that your testing covers all cases, including inserting at the beginning and end of the list, and adding to an existing node where the result is either zero or nonzero.

The provided test case includes the files **test1a.in**:

```
0 6
12 36
25 2
316 18
```

and **test1b.in**:

```
5 13
```

```
12 4
25 -2
55 -3
```

Running the test driver with these files gives the following output:

```
$ java TestSparse test1a.in test1b.in
0: 6
5: 13
12: 40
55: -3
316: 18
now backwards:
316: 18
55: -3
12: 40
5: 13
0: 6
```

Your electronic submission (submitted via Desire2Learn) will consist of <u>two files</u>:

i. a written report. This should begin with a title page that includes: the course (CS 1083), your section (FR01B, FR02B, FR03B), the assignment number, your full name, and your UNB student number. The report must include:

   A single pdf file containing a listing of the source code for the **Sparse** and **TestSparse** classes, and output of the test driver for the test case above (test1a.in and test1b.in) and one other, captured from the terminal window.

   This written report should be prepared using a word processor; we recommend using Microsoft Word (i.e. create a .docx file for your report). Copy & paste your java source code & required output into the report document. Add appropriate headings for each part. Fix up the formatting where necessary, adjusting line breaks & page breaks to ensure that your document is easy to read. Use a monospaced font for your code to maintain proper indentation.) Once the report is complete and you've checked it all over, save the .docx file for your own records, and then **save a second copy in pdf format for submission**. (Note: Be sure to open that file in a pdf viewer to verify that the pdf was generated correctly.)

The SINGLE pdf file containing your report will be submitted to the appropriate assignment drop box on Desire2Learn. (It is important that you submit a pdf file and NOT the original Word document. This pdf will allow the marker to write comments directly on your work to give you better feedback.)

Note: name this report as follows: **YourName_CS1083_As9_Report.pdf**

ii. an archive file (**.zip**) that contains the **Sparse** and **TestSparse** classes and the two files of the second test case. You should not include the report document or the .class files in your archive. This archive should be submitted as a **single file** to the appropriate drop box on Desire2Learn.

Note: name this archive file as follows: **YourName_CS1083_As9_Archive.zip**