

Implementation of data structures and algorithms
Fall 2019
Short Project 1: Linked lists
Thu, Aug 22, 2019

Version 1.0: Initial description.

Due: 11:59 PM Sun, Sep 1, 2019.

Submission procedure:

- * Create a folder whose name is your netid (NId).
- * Place all files you are submitting in that folder.
- * Use "package NId;" in all your java files.
- * There is no need to submit binary files created by your IDE (such as class files).
- * Include a text file named "readme.txt", that explains how to compile and run the code.
- * Zip the contents into a single zip or rar file.
- * If the zip file is bigger than 1 MB, you have included unnecessary files.
- * Delete them and create the zip file again.
- * Upload the zip or rar file on elearning.
- * Submission can be revised before the deadline.
- * The final submission before the deadline will be graded.
- * Only one member of each team needs to submit project.
- * Include the names of all team members in ALL files.

Team task:

1. Extend the SinglyLinkedList.java (in shared folder) to a doubly linked list (DoublyLinkedList class). Use class inheritance to avoid unnecessary duplication of code. Write a method dllIterator(), which extends the operations done by SLLIterator and adds methods hasPrev(), prev(), and add(x). The add method of the iterator should insert x before the element that will be returned by a call to next().

Additional Practice problems (no need to submit):

2. Add the methods addFirst(x), removeFirst(), remove(x) to SinglyLinkedList.java. addFirst(x) inserts a new item x to the beginning of the list. removeFirst() removes the first element of the list. Handle the edge case when the list has only one element. remove(x) deletes and returns the first occurrence of x from the list. If x is not found, it should throw a "No such element" exception. Write the functions to use as much of the existing code as possible.

3. Add indexing operations to SinglyLinkedList.java: get(index), set(index, x), add(index, x), remove(index). All operations throw "No such element" exception if the list is too short. get(index) returns the element at index (first element is at index 0). set(index, x) replaces the element at given index to be x. add(index, x) adds x as a new element at given index. remove(index) deletes and returns element at index.

4. Find errors in SinglyLinkedList.java and fix them.