Lab Assignment 3

CS 301 – Data Structures

Implementation

You are given a file *linkedlist.java* (which you can download from canvas). The file contains a class `linkedlist` with the functions `insert`, `printList`, `deleteByKey`, `deleteAtPosition`, and `removeDuplicates`. Implement your solutions in the corresponding functions. **Do not make any changes outside of these functions (e. g. by adding helper functions); such changes will be undone.**

The main method already implemented at the end of the file *linkedlist.java* tests all of the functions you are asked to implement. Note that the purpose of the tests is for you to avoid major mistakes. **Passing all given tests *does not* imply that your algorithm is correct, especially that is has the expected runtime.**

Additional details

You will notice comments throughout the file that describe possible function implementations. These are meant to be helpful hints, but are not explicit instructions on how to implement a function. You are free to disregard and delete these comments, and use your own algorithm to implement functions. **You may *not*** change the implementation of the LinkedList data members or the Node class, however. Any function implementations you write must run in at most their upper bound of runtime, and be documented using JavaDoc comments, including a **description, @param, and @return**.

For the removeDuplicates function, you are given a sorted singly linked list (as chain of nodes), delete all duplicates such that each element appears only once. Return the head of the modified list. For example, when given the input

$$1 \to 1 \to 1 \to 2 \to 3 \to 3 \to 4 \quad \text{your function should return} \quad 1 \to 2 \to 3 \to 4.$$

Do not create new nodes and do not use additional data structures to store information. Your implementation should run in $\mathcal{O}(n)$ time. (As must all of your other functions, but this is the only function for which that should be a challenge)

Submission

For your submission, upload the file *linkedlist.java* with your implementation to canvas, as well as updating the comments at the top that contain your name, ID, etc.

This is an individual assignment. Therefore, a submission is required from each student.