

Sorted Linked-Lists in JavaScript  
Cpt S 489

**(read all instructions carefully before writing any code; do all work individually!)**

The zip that contains these instructions also includes an HTML and JavaScript code file. The HTML page is set up to reference the JavaScript code file and use it to run tests. Do not change anything in this file. The JavaScript file is where you must write the code to implement a sorted, doubly-linked list.

This assignment is meant to assess your understanding of objects in JavaScript. Since the concept of classes in JavaScript may not have been covered fully in lecture by the time this project is assigned, the framework for the “SortedLL489” class is already in place for you. Do not delete any existing code in the .js file. Only add code.

Your task is to complete the “add” and “remove” function implementations. The “toString” function is already implemented for you and requires nodes in your linked-list to have the following members:

Node Object	
Member (case-sensitive!)	Purpose
value	Value stored in the node
next	Reference to next node in the linked-list (null for last node)
previous	Reference to previous node in the linked-list (null for first node)

Because you’re working with JavaScript, we don’t need to declare a node class. You must produce nodes as simple objects with the required members and insert them in the proper location into the linked-list. The linked-list is to be sorted in ascending order after each call to “add” or “remove”. Additional notes follow:

- Duplicate values ARE allowed in the list
- Both functions must run in worst case  $O(N)$  time and  $O(1)$  space

This project is worth 3 points. You need to have ALL (not “all but one or two”... ALL) basic test cases on the web page to be working to get at least 2 points. You get 1 point if at least 50% (but < 100%) of the basic test cases complete successfully. You get 0 points if at less than 50% of basic test cases pass/complete.

The basic test cases use numbers for node values. For the 3<sup>rd</sup> “challenge point”, you need to support linked-lists that have custom objects for the values in the nodes instead of just numerical values. ALL challenge cases must pass for the 3<sup>rd</sup> point to be awarded. Investigate the code and figure out how to make these cases work if you want the 3<sup>rd</sup> point!

Final note (and it’s important): The TA may use different and/or additional test cases than what’s included in the HTML page. Do additional tests to make sure your code works well.