

CS 231B Fall '24 Assignment 7: HashedList

In this assignment, we'll return to our old classic: Taking our latest structure and adding in the methods...

- `void addFirst(E item)`: makes item the first entry in the list
- `void addLast(E item)`: makes item the last entry in the list
- `E removeFirst()`: removes and returns the first entry in the list
- `E removeLast()`: removes and returns the final entry in the list

The structure that we've built in lecture and lab doesn't maintain any memory of the order in which things are inserted, so we'll have to add in that functionality. The trick is to build it into the *Nodes* that we maintain as the backbone of the structure. Specifically, we'll add two new fields to our Node class: a reference to the Node added before the current one and a reference to the one inserted after. Additionally, we'll maintain a reference in the HashedList class to the first and last Nodes to have been inserted. After building this, we could then use the HashedList to easily implement the Queue/Stack interfaces with (almost) constant runtime methods, while also getting an (almost) constant time `contains` and `remove` method! Very slick.

For the purposes of this assignment, you should ensure that your add methods only add *new* items, so your HashedList is still a *set*. If the item is already in the set, every add method does nothing.

To help get you started, [we've begun this HashedList class for you](#). You'll need to implement the methods with a `//TODO` comment in them (namely the four above, the `contains` method, and the `remove` method). **Make sure that you reset the number of buckets appropriately to keep the runtimes low and maintain reasonable space-efficiency!** (You will also need to give a `resize` method as part of this.)

Your submission this week should include:

1. the completed `.java` file.
2. A pdf containing your answer to the following questions: when did you choose to reset the number of buckets? How do your choices help ensure low runtimes and reasonable space efficiency?