# Lecture Notes, Tues May 23

## Linked List questions

1) Write a method to test if two linked lists are the same, i.e. they have the same values in the same order. In C++, you can call this method **operator==**, and test if lists **A** and **B** are the same by writing **A == B**.

2) Write these two related **List** methods

* A method that that searches for a given string in the list, and returns a pointer to the first node that contains it. If no node contains, **nullptr** is returned. Make this method **private**: we don’t want users of the **List** getting pointers to nodes (they could mess up the node!).
* A method that returns true if a given string s is in the list, and false otherwise.

3) Write a method called **push\_front\_new(s)** that adds **s** to the front of the list as long as **s** does *not* appear anywhere else in the list. If **s** is in the list it does nothing. **Hint**: This is easy If you have the methods from question 2.

4) Re-implement the size() method using **recursion** (and no loops). How does the recursive implementation compare to the non-recursive one?

5) Using **recursion** (and no loops), write a method that reverses the order of the strings on a list.