

COSC 201 – Lab #4

LinkedList vs ArrayList

Purpose: Practice using the Linked List class from the API

Tasks:

- 1.) Open Eclipse and start a new Java Project called LinkedList Project
- 2.) Add a new class called LLPractice.
- 3.) Your task: for this lab, you will be comparing the insertion and removal speeds of ArrayList and LinkedList.
 - a. First, create two lists: an ArrayList of Doubles called **testlist1** and a LinkedList of Doubles called **testlist2**. Note, if you aren't type specifying your lists correctly, there will be a deduction assessed.
 - b. Second, in main, randomly generate 1000000 doubles and insert them at random indices in **testlist1**. The easiest way to do this is to generate a random int from 0 to testlist.size()-1, inclusive, and use the add(index, element) method from LinkedList.
 - c. Third, remove 250000 of those elements, again picking random indices.
 - d. Fourth, do b. and c. with **testlist2**.
 - e. Add in timing code for both sets of code. To help you out, here's the timing code, with helpful comments on where to insert your code from b, c, and d.

```
public static void main(String[] args){  
  
    //declaration and instantiation of lists  
  
    long starttime = System.nanoTime();  
    //testlist 1 code  
    long totaltime = System.nanoTime() - starttime;  
    System.out.println("LinkedList insertion and removal  
time: " + totaltime + " ns");  
  
    starttime = System.nanoTime();  
    //testlist 2 code  
    totaltime = System.nanoTime() - starttime;  
    System.out.println("ArrayList insertion and removal  
time: " + totaltime + " ns");  
  
}
```

- 4.) In the comments of your submission (not in your code!) describe your observations in regards to the timing results that you see when you run your code.
- 5.) Turn in your properly commented code via Blackboard by 11:59 Tuesday.