# Lab 1 Looking at Java’s ArrayList and LinkedList classes

# Week beginning 21/Sept/2020

# The following table is from Thinking In Java 4 by Bruce Eckel (http://sd.blackball.lv/library/Thinking\_in\_Java\_4th\_edition.pdf) in Choosing between Lists in Chapter 11.

# It gives the time in milliseconds for multiple runs of various methodsof ArrayList and LinkedList.

---------------------ArrayList ---------------------

size add get set iteradd insert remove

10 121 139 191 435 3952 446

100 72 141 191 247 3934 296

1000 98 141 194 839 2202 923

10000 122 144 190 6880 14042 7333

---------------------LinkedList ---------------------

size add get set iteradd insert remove

10 182 164 198 658 366 262

100 106 202 230 457 108 201

1000 133 1289 1353 430 136 239

10000 172 13648 13187 435 255 239

(Note: Example of application: using ArrayList/LinkedList to store a dictionary in spelling check)

In this lab we will write some code to compare ArrayList and LinkedList.

In your code, populate an ArrayList and a LinkedList with data from a large dataset.

There are datasets available to download from github (and elsewhere) e.g. <https://github.com/dwyl/english-words/blob/master/words_alpha.zip>

Download this file – Choose View Raw. Save the file to desktop and then extract it to the desktop. This is just a file with 370099 words that you could use as a dictionary.

Put the file in the root folder of your project in IntelliJ.

Sample code for reading from a file is given in filehandlingcode.txt.

So your main() method should be:

public static void main(String[] args) {

//declare an ArrayList instance

//declare a LinkedList instance

//populate both lists with contents of a data file

//generate a list of random numbers so that indices used for ArrayList and LinkedList get are the same (likewise for set)

// call testGetForArrayList, testGetForLinkedList, testSetForArrayList etc to test get, set, insert and remove for ArrayList and LinkedList.

}

To test get, use code like:

**public static void** testGetForList(List list, int []randomNos) {  
 **for**(**int** i = 0; i < randomNos.length; i++) {  
 list.get(randomNos[i])); *// list is your list  
 // (ArrayList or LinkedList)* }  
}

Write a method for each list type e.g. testGetForArrayList and testGetForLinkedList – each of these method calls testGetForList with the relevant list as argument.

To test set, use code like

**public static void** testSetForList(List list, int [] randomNos) {  
 **for**(**int** i = 0; i < randomNos.length; i++) {  
 list.set(randomNos[i], “test”); *// any String value will do*}

To test insert (add at index), use code like:

**public static void** testInsertForList(List list) {  
 **int** reps = //need to experiment with values here**for** (**int** i = 0; i < reps; i++)  
 list.add(5, “test”); *// use 5 to minimize random-access cost, 5 an arbitrary number*}

To test remove, use code like:

**public static void** testRemoveForList(List list) {  
 **while** (list.size() > 5)  
 list.remove(5); *// Use 5 to minimize random access cost // 5 an arbitrary number*}

Time the tests on the two types of list. Look at System class for a method that gives you can use to time method calls.

Fill in the following table (time in nanoseconds):

ArrayList

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| size | get | set | insert | remove |
| 10  100  1000  10000  100000 | 4500  4800  4700  6300  14600 | 3800  4200  4200  6100  11900 | 10300  10700  12900  45000  397500 | 10900  53600  455200  6223100  384026600 |

LinkedList

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| size | get | set | insert | remove |
| 10  100  1000  10000  100000 | 10700  11900  89200  792400  8866800 | 4400  7100  44400  366600  11487200 | 11100  12300  9800  11700  19200 | 8800  37500  320100  1851000  6994500 |

Check if your results are comparable to Eckels.