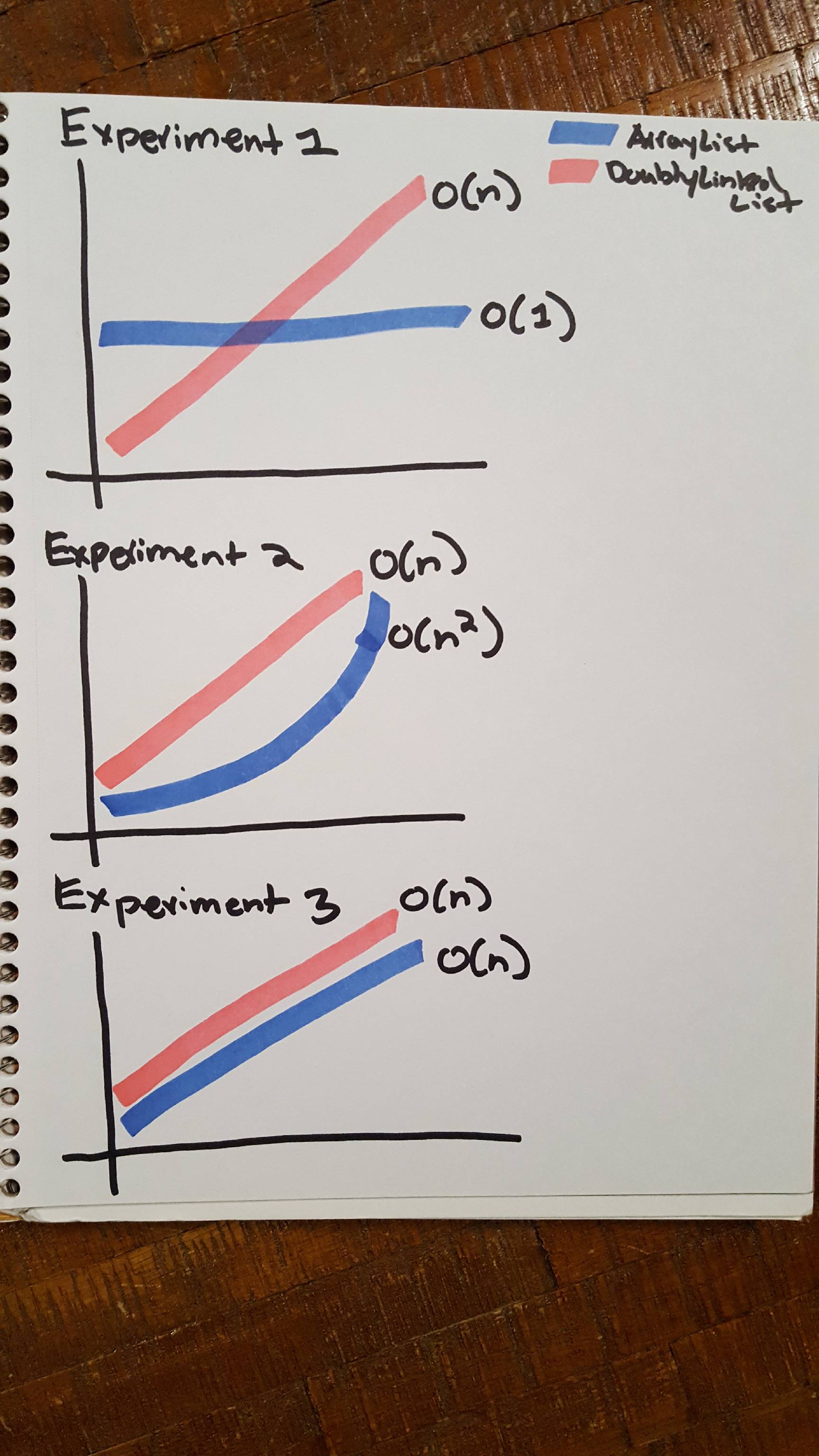
**Analysis**



**Empirical**

O(n)

O(n)

O(n)

O(nˆ2)

O(1)

O(n)

**Questions:**

1. My empirical test results gives a close accuracy to my analysis of the algorithms. Despite the random spikes, the graphs show a curve approximately what my analysis predicted.
2. Some of the random spikes surprised me and don’t make a lot of sense as to why they appear considering the numbers are averaged over 5000 trials.
3. The advantages of the arrayList is that the indexes are maintained so finding elements in the arrayList is very fast. However, the arrayList is not very space efficient and will always have memory tied up as excess. The doublyLinkedList is much harder to pull a random element out of the list because the only access to the element is from the forward and backward pointers. The linked list is much more space efficient however as the only extra space requirement is in the pointers between nodes.
4. The advantage of empirical testing is that there is solid evidence to point to and determine what type of curve the data is creating. The problem however is that random changes in the processor and memory can create random spikes that can make the data hard to interpret. The analytical approach is great for understanding exactly what is happening with the algorithm but potential hidden costs can be difficult to determine based on theory.

**Source Code:**

*Experiment 1*

package cs2321;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import net.datastructures.Position;

public class Experiment1 {

public static void main(String[] args) throws FileNotFoundException {

// TODO Auto-generated method stub

double start, stop;

ArrayList<Integer>[] array = (ArrayList<Integer>[]) new ArrayList[5000];

PrintWriter output = new PrintWriter("experiment1.txt");

for (int i = 0; i < 5000; i++) {

ArrayList<Integer> arr = new ArrayList<Integer>();

for (int j = 0; j < 100; j++) {

arr.addLast(j);

}

array[i] = arr;

}

for (int i = 0; i < 100; i++) {

start = System.nanoTime();

for (int j = 0; j < 5000; j++) {

array[j].get(i);

}

stop = System.nanoTime();

output.println((stop - start) / 5000 + "\t");

}

output.println("\t");

output.println("------------------------- \t");

output.println("\t");

DoublyLinkedList<Integer>[] array2 = (DoublyLinkedList<Integer>[]) new DoublyLinkedList[5000];

for (int i = 0; i < 5000; i++) {

DoublyLinkedList<Integer> arr = new DoublyLinkedList<Integer>();

for (int j = 0; j < 100; j++) {

arr.addLast(j);

}

array2[i] = arr;

}

Position<Integer> node;

for (int i = 0; i < 100; i++) {

start = System.nanoTime();

for (int j = 0; j < 5000; j++) {

node = array2[j].first();

for (int k = 0; k < i; k++) {

node = array2[j].after(node);

}

array2[j].after(node);

}

stop = System.nanoTime();

output.println((stop - start) / 5000 + "\t");

}

output.close();

}

}

*Experiment 2*

package cs2321;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

public class Experiment2 {

public static void main(String[] args) throws FileNotFoundException {

// TODO Auto-generated method stub

double start, stop;

ArrayList<Integer>[] array = (ArrayList<Integer>[]) new ArrayList[5000];

PrintWriter output = new PrintWriter("experiment2.txt");

for (int i = 0; i < 200; i++) {

for (int x = 0; x < 5000; x++) {

ArrayList<Integer> arr = new ArrayList<Integer>();

array[x] = arr;

}

start = System.nanoTime();

for (int j = 0; j < 5000; j++) {

for (int k = 0; k < i; k++) {

array[j].addFirst(k);

}

}

stop = System.nanoTime();

output.println((stop - start) / 5000 + "\t");

}

output.println("\t");

output.println("------------------------- \t");

output.println("\t");

DoublyLinkedList<Integer>[] list = (DoublyLinkedList<Integer>[]) new DoublyLinkedList[5000];

for (int i = 0; i < 200; i++) {

for (int x = 0; x < 5000; x++) {

DoublyLinkedList<Integer> arr = new DoublyLinkedList<Integer>();

list[x] = arr;

}

start = System.nanoTime();

for (int j = 0; j < 5000; j++) {

for (int k = 0; k < i; k++) {

list[j].addFirst(k);

}

}

stop = System.nanoTime();

output.println((stop - start) / 5000 + "\t");

}

output.close();

}

}

*Experiment 3*

package cs2321;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

public class Experiment3 {

public static void main(String[] args) throws FileNotFoundException {

// TODO Auto-generated method stub

double start, stop;

ArrayList<Integer>[] array = (ArrayList<Integer>[]) new ArrayList[5000];

PrintWriter output = new PrintWriter("experiment3.txt");

for (int i = 0; i < 200; i++) {

for (int x = 0; x < 5000; x++) {

ArrayList<Integer> arr = new ArrayList<Integer>();

array[x] = arr;

}

start = System.nanoTime();

for (int j = 0; j < 5000; j++) {

for (int k = 0; k < i; k++) {

array[j].addLast(k);

}

}

stop = System.nanoTime();

output.println((stop - start) / 5000 + "\t");

}

output.println("\t");

output.println("------------------------- \t");

output.println("\t");

DoublyLinkedList<Integer>[] list = (DoublyLinkedList<Integer>[]) new DoublyLinkedList[5000];

for (int i = 0; i < 200; i++) {

for (int x = 0; x < 5000; x++) {

DoublyLinkedList<Integer> arr = new DoublyLinkedList<Integer>();

list[x] = arr;

}

start = System.nanoTime();

for (int j = 0; j < 5000; j++) {

for (int k = 0; k < i; k++) {

list[j].addLast(k);

}

}

stop = System.nanoTime();

output.println((stop - start) / 5000 + "\t");

}

output.close();

}

}