Laboratory practice No. 3: Linked lists and dynamic vectors

María Paulina Ocampo Duque
Universidad Eafit
Medellín, Colombia
mpocampod@eafit.edu.co

María José Gutiérrez Estrada
Universidad Eafit
Medellín, Colombia
mjgutierre@eafit.edu.co

3) Practice for final project defense presentation

3.1

COMPLEXITY

	ArrayList	LinkedList
1.1	O(n^2)	O(n)

3.2 Exercise 2.1

The algorithm will read the lines of the file and will be added to the linked list due to the condition. If the character found is a "[" or a "]",then the starting position will be move and written at the end or start, Start is considered when the opening bracket ([) appears and End is considered when the closing bracket (]) appears.
After all this, the resulting new string will appear

3.3 Complexity of Keyboard´s exercise

```
public static String brokenKb(String b) {
      LinkedList<String> list = new LinkedList<>();
      boolean Final = true;                          //c1
      int k = 0;                                     //c2
      for (int i = 0; i < b.length(); i++) {
         if (b.charAt(i) == '[') {                   //c3+n
            if (Final) {
               list.addLast(b.substring(k,i));       //c4+n
            } else {
               list.addFirst(b.substring(k,i));      //c5+n
  }

            k = i + 1;                               //n+1
            Final = false;
         }
         if (b.charAt(i) == ']') {                   //c7+n
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

```
            if (Final) {
                list.addLast(b.substring(k,i));    //c8+n
            } else {
                list.addFirst(b.substring(k,i));     //c9+n
            }

            k = i + 1;                                  //n+1
            Final = true;
        }
    }
    if (Final) {
        list.addLast(b.substring(k));   //C11+n
    } else {
        list.addFirst(b.substring(k));   //c12+n
    }
     String result= "";
    for (String a: list) {          //c13+n
        result+= a;
    }
      return result;
}
```

COMPLEXITY

- **O(n)**

The algorithm starts by going through a for and having a complexity of o(n) and then it prints the array and goes through another for.

3.4 The variable is n. The n represents the number of lines the keyboard reads(the size of the list) and the lenght of the string

4) Practice for midterms

4.1 Optional
4.2 Opt *c) O(n)*
4.3 **DOES NOT EXIST**
4.4 1. LINE 21 stack.pop();
2. *c) O(1)*
4.5 Optional
4.6 Opt *a) O(n^3)*
4.7 **DOES NOT EXIST**
4.8 c) O(n) y O(1)
4.9 **4.8.1** a) O(k)
4.8.2.c) 12
4.8.3 c) O(n)
4.10 **4.9.1** d) O(n^2)
4.9.2 a) 6
4.9.3 b) O(n)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4.11 **4.10.1** b) O(max(list)*n^2)
4.10.2 b) O(n)

4.12 **4.11.1 LINE 13** (s1!.isEmpty())
4.11.2 LINE 14 (s1.pop())
4.11.3 LINE 17 (s2.pop())

4.13 **4.12.1** iv) 0, 2, 4, 6, 8, 10
4.12.2 i) O(1)

4.13 **4.13.1** iii) O(n^2)
4.13.2 iii) O(n^2)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473