Laboratory practice No. 3: LinkedList and Dynamic Arrays

Samuel David Ben Jacob Arango Henao
Universidad Eafit
Medellín, Colombia
Sdarangoh@eafit.edu.co

3) Practice for final project defense presentation

3.1.

Exercise	ArrayList	LinkedList
1.1	O(n)	O(n)

3.2. The implementation of the exercise 2.1 has the next functioning, first, the program asks you for the input, which I going to be the wrote text with the broken keyboard, then, the program will use the method *reader* for put each character in one node of a linked list, after that, the method *change* in the class switch will create an empty String: *text,* and will search when appears "*[" (start)* or "*]" (end)* in the linked list using *temp.next* for analyze the content in the node. In the beginning if there is not a *start* or *end* character, the method will add each letter to the empty String, creating the new text until it finds one of the special characters. When the method finds one of them whatever it is, the process is going to be almost the same, the program will change the value of the node, erasing the special character and adding each following letter until appears a new special character, when that happens, the method will add to our String *text* the new value of the node in the start or in the end, depending on the type of character, finally when the LinkedList arrives to the end, the program will show the final text, without special characters and following the *start* and *end* orders.

3.3. The complexity on the 2.1 exercise is O(N)

3.4. Our variable is N, which is the length of the input.

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) Practice for midterms

4.1.
i) b
ii) b O(N+M)

4.2. *c O(N)*

4.3. falta???

4.4. in

4.5. a {7,8,3,1,2,9}

4.6. a $O(N^3)$

4.7. falta???

4.8. c O(n) y O(1)

4.9. ->

4.9.1. a O(k)

4.9.2. b 9

4.9.3. b O(1)

4.10. ->

4.10.1. d O(n)

4.10.2. a 6

4.10.3. d $O(n^2)$

4.11. ->

4.11.1. b O(max(list)*n)

4.11.2. b O(n)

4.12. ->

4.12.1. !s1.isEmpty()

4.12.2. s1.pop()

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

4.12.3. s2.pop()

4.13. ->

4.13.1. IV) 0, 2, 4, 6, 8, 10

4.13.2. I) O(1)

4.14. ->

4.14.1. III) $O(n^2)$

4.14.2. IV) $O(n^3)$

4.15. IV) 5, 4, 3, 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