

Laboratory practice No. 3: Linked Lists, Arraylist and Hash Tables

Gregorio Pérez Bernal
Universidad Eafit
Medellín, Colombia
Gperezb1@eafit.edu.co

Luisa Toro Villegas
Universidad Eafit
Medellín, Colombia
ltorov@eafit.edu.co

3) Practice for final project defense presentation

3.1

The exercise 1.1 was made using arrays, and it was found to have a complexity of $O(n)$, due to the fact that it is able to access the exact position where a student is, whereas implementing linked list was found to be $O(m*n)$ for it had to go through each node of attributes and each node per person.

	ArrayList	Linked List
1.1	$O(n)$	$O(m*n)$

Where n is the number of students on a class and m is the number of attributes per person

3.2 A linked list is created where each position is a character of the string imputed. The first loop of the method takes the string imputed and separates into characters and if the character is either a "[" or a "]" it assigns a boolean value for each bracket, and then at the end of the loop, if the Boolean is true, then the linked list will add the character at that position will be added based on a counter variable that has been implemented, and if the Boolean is false, the char will not has the index taken into account, that means, adding it at the end. Then, after the loop was done, a new linked list made up of characters will be created. Then another loop that will run through the length of the list until it is empty, will remove the first character by character and then, if the character that was removed is not a bracket, it will be added to the new list. Finally, another loop will travel through the list, printing it.

3.3 The complexity of exercise 2.1 is $O(n^2)$, where n refers to the length of the list. This occurs because it needs to go through the list $O(n)$ and then it needs to add an element in a particular part of the list $O(n)$.

3.4 n refers to the number of characters on the string

4) Practice for midterms

- 4.1 i B
ii B
- 4.2 C
- 4.3 i B
ii D
- 4.4 i token
ii C
- 4.5 A
- 4.6 A
- 4.7 NO EXISTE
- 4.8 C
- 4.9 i A
ii C
iii C
- 4.10 i D
ii A
iii B
- 4.11 i C
ii B
- 4.12 i s1.isEmpty()
ii s1.pop()
iii s2.pop()
- 4.13 i D
ii A
- 4.13 (again) i C
ii D
- 4.14 i C

5) Recommended reading (optional)

Conceptual map annexed on github.

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