Laboratory practice No. 3: Linked List and Array List

Juan Sebastián Pérez Salazar
Universidad Eafit
Medellín, Colombia
jsperezs@eafit.edu.co

Yhoan Alejandro Guzmán García
Universidad Eafit
Medellín, Colombia
yaguzmang@eafit.edu.co

3) Practice for final project defense presentation

1.

Exercise	ArrayList complexity	LinkedList complexity
1.1	O(n)	O(n)
1.2	O(n)	O(n)
1.3	O(n*m)	O(n*m)
1.4	the point does not require to be implemented with ArrayList	O(n)

According to the table of the complexity of the methods, using the method implemented with ArrayList the effectiveness is the same as when we use the method implemented with LinkedList. However, there are some small features in time that allow us to decide which is preferable to use in comparison to the other, therefore, for exercise 1.1 it is better to use ArrayList because it is faster to access, for exercise 1.2 and 1.3 is better use LinkedList, because in these cases they are faster to eliminate or add data.

2. This method is based on a for, two booleans and a counter, the first thing that is done is to initialize the "counter" in 0, the boolean "begin" in "false" and the boolean "end" in "true". the cycle is executed and in this we ask in a conditional if the character that is being evaluated is "[" or "]". In case "[" the counter is defined in 0, "begin" in "true" and "end" in "false", in the opposite case, "begin" in false and "end" true. then two possible paths are executed, one when "begin" is true and the other when false. if "begin" is true, the character is added in the "cont" position and the counter increases, otherwise only the character is added at the end. The user has the possibility to call the “printText” method and print the resulting text.

3. **Code**

```
for(int i=0; i<s.length(); i++){// O(n)
        if(s.charAt(i)=='['){ //O(1)
           begin = true;
           end = false;
           cont=0;
           continue;
        }else if(s.charAt(i)==']'){ //O(1)
           begin = false;
           end=true;
```

```
            continue;
        }

        if(begin){ //O(1)
            chain.add(cont, s.charAt(i)); //O(m)
            cont++;
        }else chain.add(s.charAt(i)); //O(1)
    }
```

Complexity

The algorithm complexity is O(n*m)

4. In the case of the method used in number 2.1, the complexity is O(n * m), for this. "n" refers to what will be the length of the cycle which in this case will be the length of the string that has been entered, and "m" refers to what it will take in Linked List time to add an element to a position that is random. Since this algorithm does not work with iterators.
   Therefore, for each iteration of the cycle, an O (m) process will be performed, and the cycle total will be O(n), as the complexity of m is within the cycle, these multiply, and this generates a complexity of O(n * m).

4) Practice for midterms

1. C. they both have the same complexity
2. C. O(n)
3. Line 2: q.size()>0
   Line 3: <=
   Line 4: q.remove()
   Line 6: q.remove()
4. A) Line 3: lista.size();
   B) Line 7: lista.add(auxiliary.pop());
5. A) Line 12: auxiliar1.size()>0 Line 16: auxiliar2.size()>0
   B) Line 18: personas.offer(edad);
6. C. $O(n^2)$
7. C. $O(n^3)$
8. C. O(n)
9. 1) D. O(k)
   2) C. 12
   3) C. O(1)
10. 1) D. O(n)
    2) A. 6
    3) B. O(n)

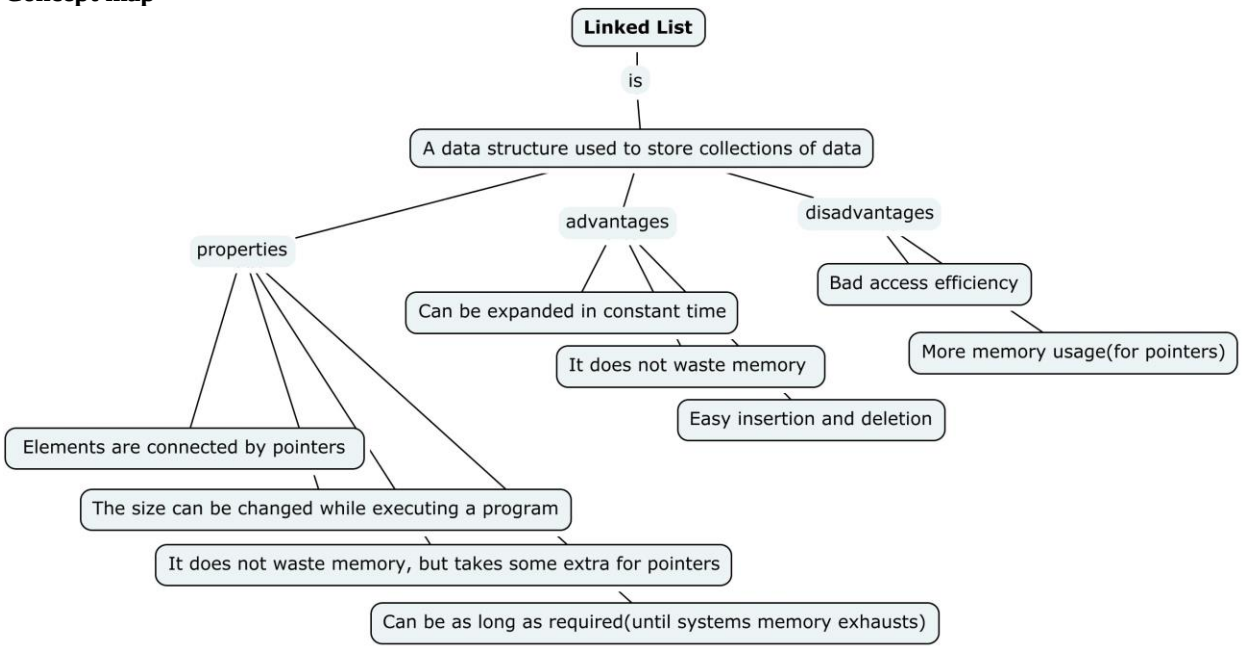
5) Recommended reading (optional)

a) Narasimha Karumanchi, Data Structures and Algorithms made easy in Java, (2nd edition), 2011. Chapter 3: Linked Lists

b) A linked list is a data structure used for storing collections of data. It is characterized by the way the elements are connected. Pointers connect its elements, which allows it to grow of shrink while

executing a program, and to be as big as required (until systems memory exhaust). The pointers also help to decrease the memory wastage, which is an advantage over regular array structures. Other advantages of the linked lists are that they can be expanded in constant time (which is not possible with arrays) and that insertion and deletion are easier and can be done in constant time. There are many structures for storing data, but the advantages that the linked list gives, makes it more interesting to use in some cases, for example, when insertion and deletion in constant time are priorities.

c) Concept map



6) Team work and gradual progress (optional)

a) Meeting work

Member	Date	Done	Doing	To do
Sebastián	30/09/2018	**I implemented arrayList with points 1.1 and 1.3**		Implement LinkedList with points 1.1 and 1.3
Sebastián	1/10/2018	**I implemented LinkedList with points 1.1 and 1.3**	Doing analysis of point 1.6	Test for points, 1.1, 1.3, 1.6
Sebastián	2/10/2018	**Test for points, 1.1, 1.3, 1.6**	Doing analysis of point 1.2	Make the laboratory report
Sebastián	3/10/2018	**Point 1.1, 1.2, 1.3, 1.6**	Laboratory report	Practice for midterms
Sebastián	4/10/2018	**Laboratory report and practice for mindterms**		Points 2.1 and 2.2

Sebastián	5/10/2018	**Points 2.1 and 2.2**		points 1.4, 1.5, 2.3
Yhoan	6/10/2018	**points 1.4, 1.5 and 2.3**		recommended reading
Yhoan	7/10/2018	**recommended reading**		upload the laboratory

b) History changes of code

History changes of code		
Version	**Code**	**Status**
1.0	1.1	
1.0	1.2	
2.0	1.2	
1.0	1.3	
1.0	1.4	
1.0	1.5	
1.0	1.6	
1.0	2.1	
1.0	2.2	
2.0	2.2	
1.0	2.3	

PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co