ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 3: LinkedList

David Calle Gonzales
Universidad Eafit
Medellín, Colombia
dcalleg@eafit.edu.co

Julian Ramirez Giraldo
Universidad Eafit
Medellín, Colombia
jramirezg@eafit.edu.co

3) Practice for final project defense presentation

EJERCICIO 1.1	ArrayList	LinkedList
SHOW NAME AND GRADE	O(n)	O(n)
GRADE, NAME AND COURSE	O(n)	O(n)

3.1
Both structures have a complexity O (n) that is quite good but in certain cases it is better to use ArrayList, for example, in the case of having to access LinkedList you need O (n) steps, however in ArrayList access needs Or (1), which decreases the complexity of the algorithm considerably.

3.2 The input is entered and the "sol" lists that will be the solution and "aux" that will serve as an auxiliary are created, in addition to the "st" condition that will be explained later and that starts with a False value.

then character by character will be analyzed to decide if it is a letter or one of the special symbols either start or end; in case of being a normal character, the ConditionedUnion function will be executed which, if there is something in the auxiliary list, will add this to the beginning of the solution; if the character read is "[", ConditionedUnion will be executed to prevent the case of 2 "[" in a row, and the auxiliary list will be cleared, also, the condition st, which indicates that the start of the line was entered, will become True .

Now, if the condition st is true, the following elements will be added to the auxiliary list until a "]" or end of line is reached, in which case the st will become false and the code will keep going.

At the end of the code a final ConditionedUnion will be executed to add what is missing in the auxiliary list.

3.3 in the worst case (there is a "[" at the beginning) T (n) = n * n + n
complexity is O(n^2) where n is the number of characters in the list.
3.4the first n is because of the for cycle, the second one is for adding each element to the aux list at the end of it, and the third one is for combining both aux and sol lists with ConditionedUnion, because the function has to go through every element in aux.

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

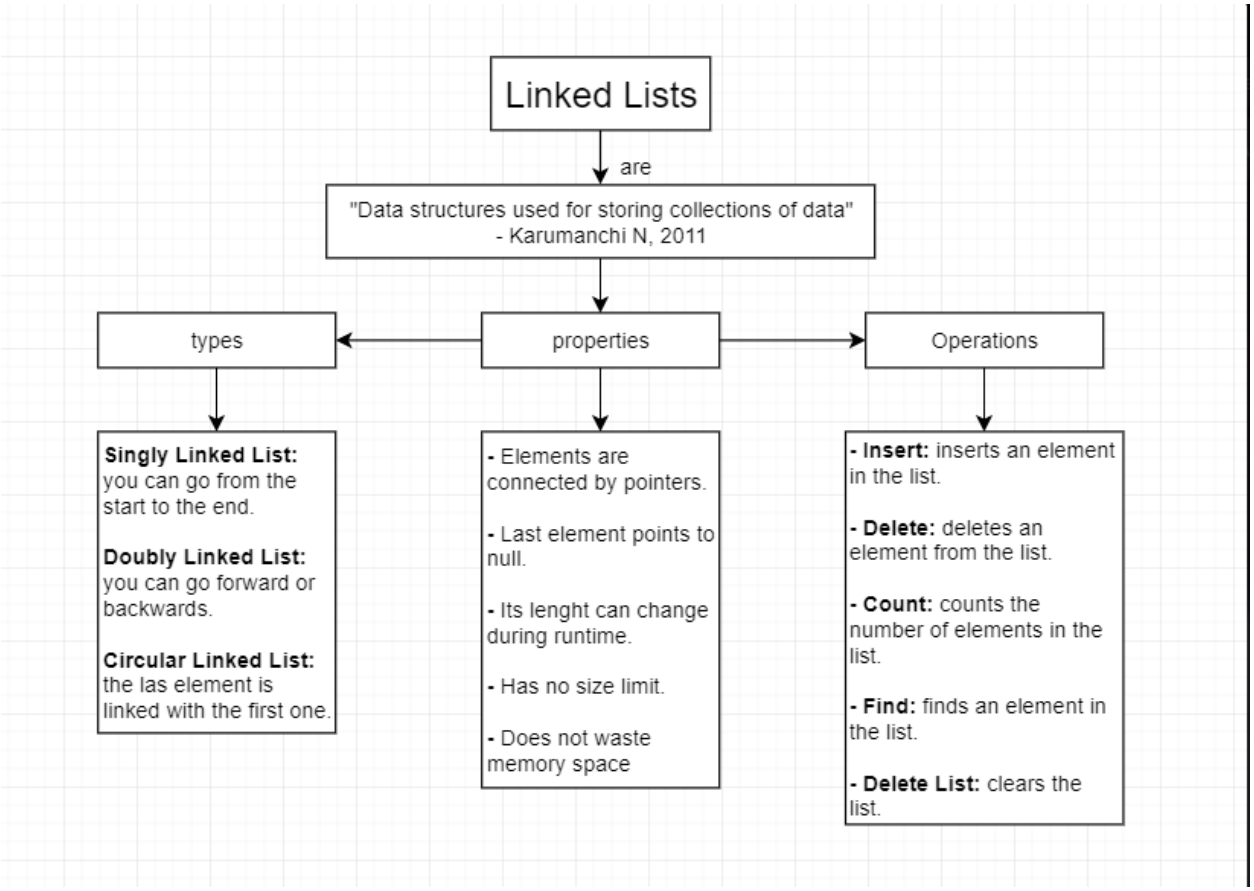
this could be more efficient using circly linked lists, this way the complexity could be O(n)

4) Practice for midterms

4.1) a
4.2) c
4.3 a) q.size() > 1
4.3 b) <=
4.3 c) q.remove()
4.3 d) q.remove()
4.4 a) lista.size()
4.4 b) lista.add(auxiliar.pop)
4.5.1) auxiliar1.zise() > 0 // auxiliar2.size() > 0
4.5.2) personas.offer(edad)
4.6) a
4.7) c
4.8.1) a
4.8.2) c
4.8.3) c
4.9.1) d
4.9.2) c
4.9.3) b
4.10.1) c
4.10.2) b
4.11.1) s1.size() > 0
4.11.2) s1.pop()
4.11.3) s2.pop()
4.12.1) iv
4.12.2) i
4.13.1) iii
4.13.2) iii
4.14) iii

5) Recommended reading (optional)

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



6) Team work and gradual progress (optional)

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

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