ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 3: Linked list and dynamic vectors.

Maria Isabel Arango Palacio
Universidad Eafit
Medellín, Colombia
miarangop@eafit.edu.co

Isabella Montoya Henao
Universidad Eafit
Medellín, Colombia
imontoyah@eafit.edu.co

3) Practice for final project defense presentation

3.1 Complexity of the algorithms

1. Medellin map

First of all, in this exercise, we had to clean up the database because the data format was latin-1 and it was presenting many problems when we were reading the data, so we actualice the data format to UTF-8 (a csv file) which helps to read the data easily because it is the codification format that actually Python3 works with.

Worst Case:

$T(n, m) = C + C_8 * n + C_{10} * m$
$T(n, m) = C_8 * n + C_{10} * m \rightarrow Sum\ law\ and\ common\ factor$
$T(n, m) = n + m \rightarrow Product\ law$
$O(n, m)$ *where* n *is the number of vertices and* m *is the number of arcos.*

2. Student notes

$C_1 = 1$ $C_7 = 1$
$C_2 = 1$ $C_8 = 1$
$C_3 = 1$ $C_9 = 1$
$C_4 = 1$ $C_{10} = 1$
$C_5 = 1$ $C_{11} = 1$
$C_6 = 1$ $C_{12} = 1$

Worst Case:

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

$T(n) = C_1 + C_2 + C_3 + C_{3.0} * n + C_4 + C_5 + C_6 + C_{6.0} * n + C_7 + C_8 + C_9 + C_{10} + C_{11} + C_{12}$
$T(n) = (C_{3.0} + C_{6.0}) * n \rightarrow Sum\ law\ and\ common\ factor$
$T(n) = n \rightarrow Product\ law$
$O(n)\ where\ n\ is\ the\ csv\ length$

3. Pivote

Worst Case:

$T(n) = C + C_5 * n + C_7 * (n - 1)$

$T(n) = C_5 * n + C_7 * (n - 1) \rightarrow Sum\ law$
$T(n) = n + (n - 1) \rightarrow Product\ law$
$T(n) = n \rightarrow Sum\ law$
$O(n)\ where\ n\ is\ the\ array's\ length$

4. Store of fridges

Worst Case:

$T(n, m) = C + C_{7.1} * n + C_{8.1} * m + C_{9.1} * m + C_{9.3} * m * n + C_{10.3} * m + C_{10.6} * m * n$
$T(n) = C_{7.1} * n + (C_{8.1} + C_{9.1} + C_{10.3}) * m + (C_{9.3} + C_{10.6}) * m * n \rightarrow Sum\ law\ and\ common\ f$
$T(n) = n + m + n * m \rightarrow Product\ law$
$T(n) = n * m \rightarrow Sum\ law$
$O(n * m)\ where\ m\ is\ the\ queue`s\ length\ with\ the\ number\ of\ requests\ and\ n\ is\ the\ stack's\ length\ with\ the\ number\ of\ refrigerators.$

5. DoublyLinkedList

Complexity of get an element:

$T(n) = C + C_{0.4} * n + C_{0.5} * n$
$T(n) = (C_{0.4} + C_{0.5}) * n \rightarrow Sum\ law\ and\ common\ factor$
$T(n) = n \rightarrow Product\ law$
$O(n)\ where\ n\ is\ the\ linked\ list\ length$

Complexity of add data in a index:

$T(n) = C + C_{2.5} * n + C_{2.12} * n$

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

$T(n) = (C_{2.5} + C_{2.12}) * n \quad \rightarrow Sum\ law\ and\ common\ factor$
$T(n) = n \quad \rightarrow Product\ law$
$O(n)\ where\ n\ is\ the\ linked\ list\ length$

Complexity of search an element:

$T(n) = C + C_6 * n$
$T(n) = C_6 * n \quad \rightarrow Sum\ law$
$T(n) = n \quad \rightarrow Product\ law$
$O(n)\ where\ n\ is\ the\ linked\ list\ length$

Complexity of delete an element:

$T(n) = C + C_{3.13} * n + C_{3.20} * n + C_{3.26} * n$
$T(n) = (C_{3.13} + C_{3.20} + C_{3.26}) * n \quad \rightarrow Sum\ law\ and\ common\ factor$
$T(n) = n \quad \rightarrow Product\ law$
$O(n)\ where\ n\ is\ the\ linked\ list\ length$

6. Attention in a cashier

Worst Case:

$T(n, m) = C + n + m$
$T(n, m) = n + m \quad \rightarrow Sum\ law$
$O(p)\ where\ p\ is\ n + m\ ,\ m\ is\ the\ queue's\ length\ and\ n\ is\ the\ most\ large\ file`s\ length$

3.2 Broken keyboard

To develop this exercise we implement a deque to store the characters of the string that we will return at the end. To store them we go through the string that is given to us and evaluate some cases. In the first place we evaluate if the character in a certain position is not the strat key or end key and we add it to an auxiliary string. In the second place, we evaluate if the character is the start key and then if the acc variable (which indicates to us if we should add at the start or the end of the deque) is 0 we append the auxiliary string at the end, if not at the start and change it to 1. In the third place, we evaluate if the character is the end key and then if the acc variable is 1 we insert the auxiliary string at the start of the deque, if not we insert it at the end and change acc to be 0. In the end, we fill a string with the characters in the deque and we return it as they ask in the exercise.

3.3 1. Broken keyboard

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

Worst Case:

$T(n) = C + C_{23} * m + C_{3.0} * n$
$T(n) = C_{23} * m + C_{3.0} * n \rightarrow Sum\ law,\ where\ n = m$
$\rightarrow n = m\ because\ the\ string's\ length\ will\ be\ as\ long\ as\ the\ LinkedList's\ length$

$T(n) = (C_{23} + C_{3.0}) * n \rightarrow Common\ factor\ law$
$T(n) = n \rightarrow Product\ law$
$O(n)\ where\ n\ is\ the\ string's\ length\ and\ m\ is\ the\ List's\ length$

4) Practice for midterms

4.1 Line 4 : res += int(vector[i])*(2**(len(vector) -1) -i)
Complexity: O(n)
4.2 C
4.3 4.3.1. IV
4.3.2 i
4.4 Line 21: output.append(token)
Complexity in the worst case for pop() : O(1)
4.5 *A*
4.6 *A*
4.7 *III*
4.8 *D*
4.9 4.9.1 *A* 4.9.2 *C* 4.9.3 *C*
4.10 4.10.1 *D* *4.10.2* *A* *4.10.3* *B*
4.11 4.11.1 *C* 4.11.2 *B*
4.12 4.12.1 *while(!(s1.isEmpty()));* 4.12.2 *s2.push(s1.pop());* 4.12.3 return s2. pop
4.13 4.13.1 iii, O(n^2) 4.13.2 iii, O(n^2)

6) Teamwork and gradual progress (optional)

6.1 Meeting minutes

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

ESTRUCTURA DE DATOS 1
Código ST0245

Isabella Montoya Henao	Entrante	1 h 2 min	viernes 3:12 p.m.
Isabella Montoya Henao	Saliente		viernes 3:11 p.m.
Isabella Montoya Henao	Saliente	2 s	viernes 3:10 p.m.
Isabella Montoya Henao	Entrante	1 h 25 min	jueves 2:07 p.m.
Isabella Montoya Henao	Entrante	23 min 43 s	lunes 8:03 p.m.
Isabella Montoya Henao	Entrante	1 h 6 min	lunes 6:33 p.m.
Isabella Montoya Henao	Entrante	22 min 5 s	lunes 10:07 a.m.
Isabella Montoya Henao	Entrante	9 min 49 s	lunes 9:52 a.m.
Isabella Montoya Henao	Saliente	1 h 23 min	19/03 9:52 a.m.
Isabella Montoya Henao	Llamada perdida		19/03 9:50 a.m.
Isabella Montoya Henao	Saliente	32 min 51 s	18/03 7:31 p.m.
Isabella Montoya Henao	Entrante		18/03 7:30 p.m.

6.2 History of changes of the code

Commits on Mar 28, 2021

Add files via upload
imontoyah committed 2 hours ago — Verified — 70dcad5

Commits on Mar 27, 2021

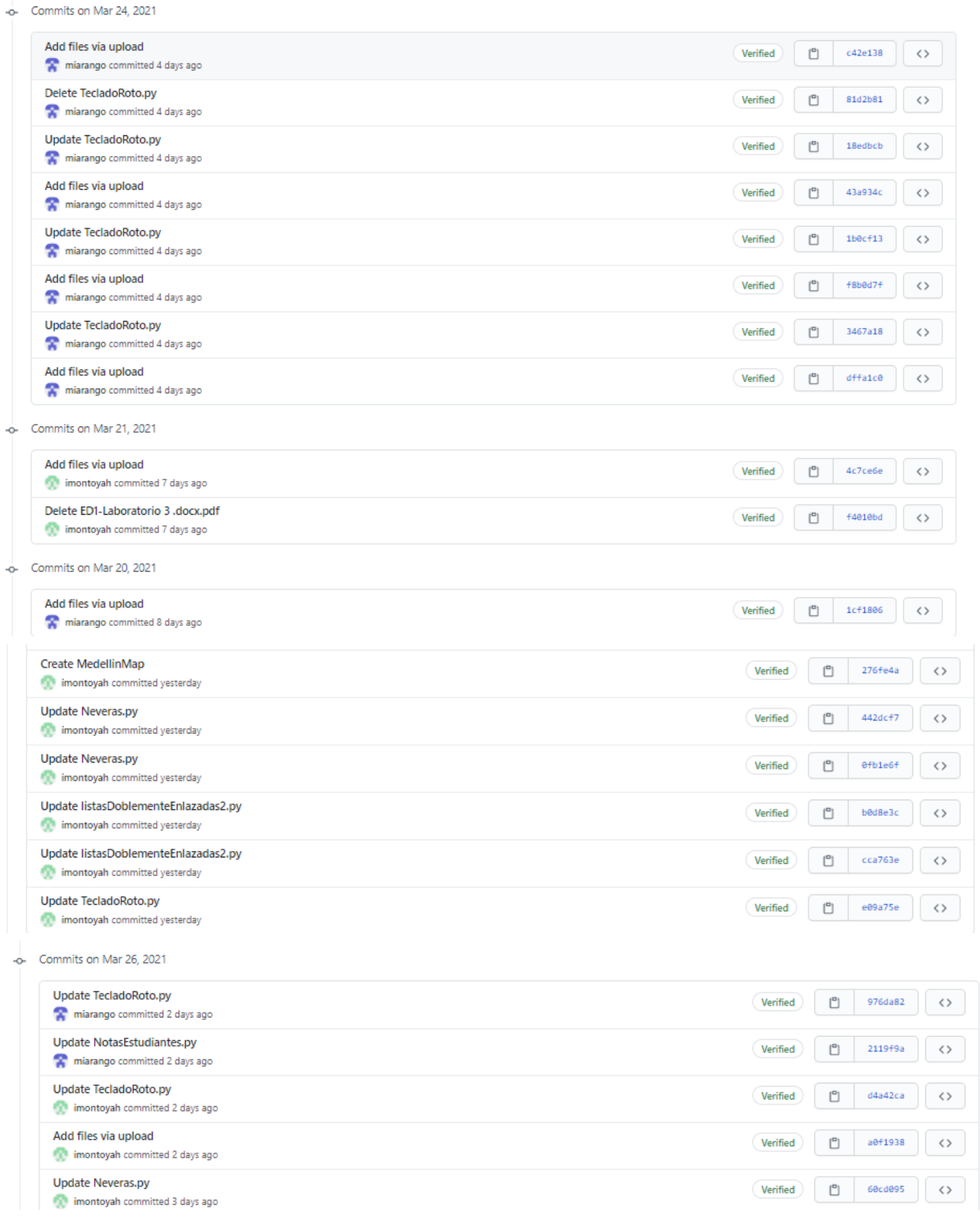
Update mapaMedellin.py
imontoyah committed yesterday — Verified — 416d14f

Update mapaMedellin.py
imontoyah committed yesterday — Verified — 8b76c4b

Delete hola
imontoyah committed yesterday — Verified — 56ba0bf

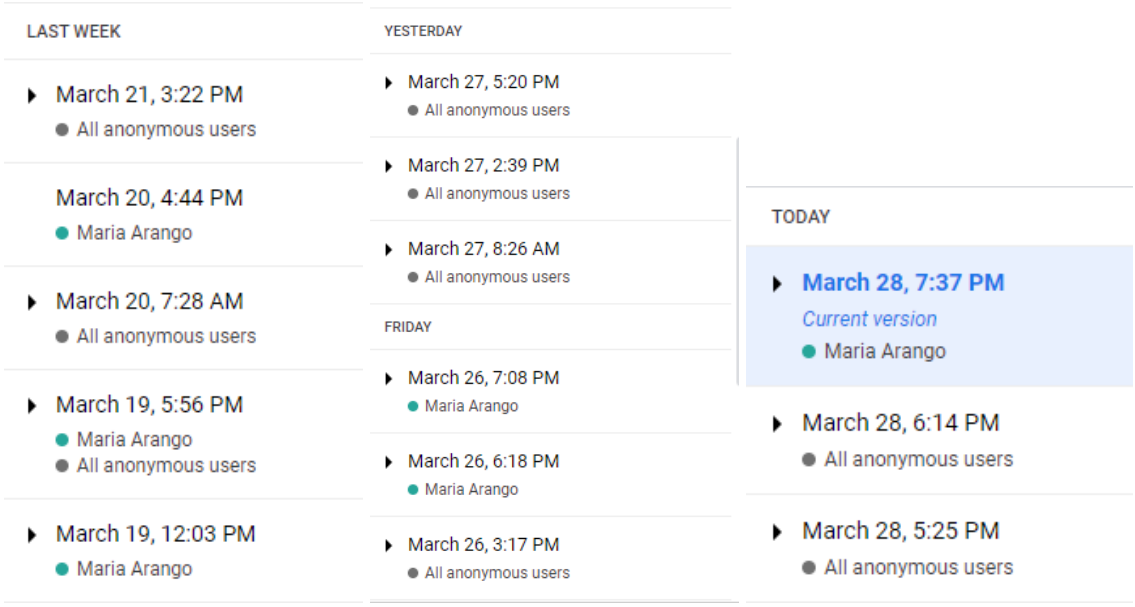
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 1
Código ST0245



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

6.3 History of changes of the report



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