Laboratory practice No. 5: Graph Implementation

Juan Diego Gutiérrez Montoya
Universidad Eafit
Medellín, Colombia
jdgutierrm@eafit.edu.co

Juanita Vanegas Elorza
Universidad Eafit
Medellín, Colombia
jvanegase@eafit.edu.co

3) Practice for final project defense presentation

```
Ejersicio 1.3 =
[0, 2, 1, 3]
[0, 2, 1, 3]
```

1.

2. Exercise 1.1 is implemented for both linked list digraph and matrix digraph. LinkedList digraph works by instantiating an array of “Pareja” LinkedLists in DigraphAL’s constructor with a size of size parameter needed in the constructor. When adding an arc, a “Pareja” is added to the linked list in the arrays source position with the weight and destination. To implement the matrix digraph, a matrix is instantiated in the constructor with size length and size height. When adding an arc from source to destination, the position array[source][destination] is assigned the weight.

3. It is more convenient to use the matrix implementation on graphs with small size, because large graphs take a lot of space, when graphs are too large, the linked list implementation is more convenient.

4. It would be more convenient to use LinkedList to represent the city of Medellin because there are a lot of vertices and they connect to few vertices each.

5. It is better to use LinkedList because if the size of the graph is too big, it will take too much space.

6. It is better to use Matrices because there are few vertices.

7. Complexity

```
public static boolean ejercicio21aux(Digraph graph, int n, int[] color, boolean[] visited){
    if(!visited[n]){                                                          C1
        visited[n]=true;                                                      C2
        for(int i : graph.getSuccessors(n)){                                  C3*n
            if(color[i]==color[n]) return false;
    C4*n
            if(color[i]==0){                                                  C5*n
                color[i]=3-color[n];                                          C6*n
                return ejercicio21aux(graph,i,color,visited);
    (C7+T(m-1))*n
            }
        }
     }
     return true;                                                             C8
  }
```

Complexity = C1+C2+C3*n+ C4*n+ C5*n+ C6*n+(C7+T(m-1))*n
O(1+1+n+n+n+n+m*n))
O(m*n)
O(n*n)

8. Variables
n is the size of the graph
m is the non-visited vertices, since every vertices' visited, m = n;

4) Practice for midterms

1. A

	0	1	2	3	4	5	6	7
0				1	1			
1	1					1		
2					1		1	
3								1
4			1					
5								
6			1					
7								

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

2.

0	3,4
1	1,5
2	4,6
3	7
4	2
5	
6	2
7	

3. b

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