

## Lab N. 5: Graph Implementation

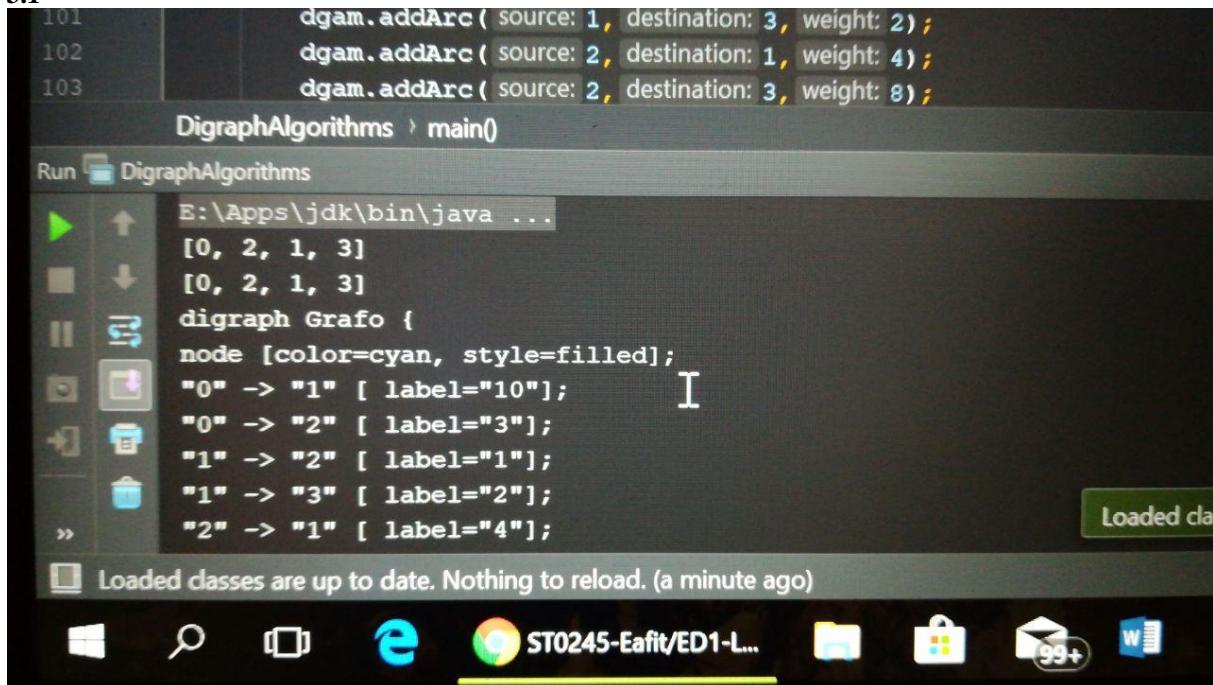
**Mateo Sánchez Toro**  
Universidad Eafit  
Medellín, Colombia  
msanchezt@eafit.edu.co

**Juan Pablo Giraldo Restrepo**  
Universidad Eafit  
Medellín, Colombia  
jpgiraldor@eafit.edu.co

**Juan Felipe Londoño Gaviria**  
Universidad Eafit  
Medellín, Colombia  
jflondonog@eafit.edu.co

### 3) Project questions drill

#### 3.1



```
101      dgam.addArc( source: 1, destination: 3, weight: 2 );
102      dgam.addArc( source: 2, destination: 1, weight: 4 );
103      dgam.addArc( source: 2, destination: 3, weight: 8 );

DigraphAlgorithms > main()

Run DigraphAlgorithms
E:\Apps\jdk\bin\java ...
[0, 2, 1, 3]
[0, 2, 1, 3]
digraph Grafo {
node [color=cyan, style=filled];
"0" -> "1" [ label="10"];
"0" -> "2" [ label="3"];
"1" -> "2" [ label="1"];
"1" -> "3" [ label="2"];
"2" -> "1" [ label="4"];
}

Loaded classes are up to date. Nothing to reload. (a minute ago)
```

#### 3.2

Basically, what the exercise does is that when done with matrixes, first the matrix is created with a size [size][size] where size is the number of values in the graph. The method `getWeight` is implemented whom returns the weight of each connection. After, `addArc`, adds the arcs to the graph. Last `getSuccesors` returns the successors of each and every one of the nodes. Now going to the graphs with lists, they are a bit similar. First an `ArrayList` of pairs is created and basically the same are implemented.

**DOCENTE MAURICIO TORO BERMÚDEZ**  
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627  
Correo: mtorobe@eafit.edu.co

### 3.3

Both are convenient because they worked as directed and non-directed. There are certain explicit situations in which one takes the decision regarding what's asked, but both are convenient for all situations.

### 3.4

Adjacency matrixes because it would show all the possible routes.

### 3.5

Adjacency matrixes because it would ease the way to view the average of friendships that a user has on Facebook.

### 3.6

Adjacency lists, because it would only show the shortest route.

### 3.7

$O(n+m)$

### 3.8

N is the number of nodes and m is the number of connections.

#### 4) Test drill

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

0->[3,4]

1->[0,2,5]

2->[1,4,6]

3->[7]

4->[2]

5->[]

6->[2]

7->[]


3) b

#### 5) Team Work

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

	<p style="text-align: center;"><b>UNIVERSIDAD EAFIT</b>  <b>ESCUELA DE INGENIERÍA</b>  <b>DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS</b></p>	<p><b>Código: ST245</b></p>
		<p><b>Estructura de Datos 1</b></p>

Mateo Sánchez Toro: made exercise 2 and helped in exercise 1

Juan Pablo Giraldo Restrepo: Made exercises 1 and 4 and also translated the template.

Juan Felipe Londoño Gaviria: made exercise 3, helped in exercise 1 and made the template in Spanish.

Communication was through WhatsApp and the lab was made by all the members of the group.

**DOCENTE MAURICIO TORO BERMÚDEZ**  
**Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627**  
**Correo: mtorobe@eafit.edu.co**