Laboratorio Nro. 1 Implementación de Grafos

Santiago Soto Marulanda
Universidad Eafit
Medellín, Colombia
ssotom@eafit.edu.co

Kevyn Santiago Gomez Patiño
Universidad Eafit
Medellín, Colombia
ksgomezp@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

1. Tenemos la clase DigraphAM y DigraphAl las cuales son para matrices de adyacencia y listas de adyacencia respectivamente, las cuales heredan de una clase abstracta Digraph que contiene su método constructor, en el caso de DigraphAm este crea una matriz de tamaño (size x size) la cual es la que utilizaremos para representar el grafo siendo size el número de nodos del grafo y para DigraphAL esta inicializa una lista de arreglos de una lista de arreglos de parejas (donde la construcción de la clase pareja está construida por dos elementos siendo el primero el nodo hacia donde va la relación y el segundo el peso del arco) con size numero de elementos dentro de la lista de arreglos. Estas clases tienen los métodos de addArc donde se le pasa por parametro dos vértices que están conectados y el peso de la conexión; el método getSuccesors el cual nos permite saber hacia qué nodos salen relaciones desde el nodo en que nos fijamos y por último el método getWeight que pasando por parámetro dos nodos nos permite saber cual es el peso de la relación entre estos dos.
2. Es mejor utilizar matrices de adyacencia para grafos que tienen pocos vértices ya que se puede acceder a cada vértice en O(1), pero cuando son grafos de mayor tamaño es mas optimo usar listas de adyacencia ya que las matrices ocupan mucha memoria.
3. Para este ejercicio es mejor utilizar listas de adyacencia ya que el grafo contiene 300000 nodos y cada nodo está conectado con pocos nodos, entonces si utilizaramos matrices de adyacencia para representar este grafo estaríamos desperdiciando una cantidad increíble de memoria..
4. Es mejor utilizar listas de adyacencia ya que los grafos no siempre están conectados todos sus vértices entre sí entonces si utilizaramos matrices de adyacencia para representar grafos que no tienen tanta relación entre sus nodos estaríamos desperdiciando una cantidad increíble de memoria.
5. Es mejor utilizar matrices de adyacencia ya que es más fácil obtener su posición.

6. Se está representando el grafo en forma de lista de adyacencia, dicha lista esta hecha con un arreglo de listas enlazadas, cada posición del arreglo representa una vertice y las listas enlazadas que están en cada posición representan las aristas, para resolver el problemas, primero se leen los datos, luego se crea el grafo, después de esto se recorre cada vértice, y con ayuda de un arreglo se lleva el control de los colores de cada vértice, y con una serie de comparación se determina si el grafo es bicolor o no.
7. $T(n) = 2(n*m) + C$
   $O(n) = 2(n*m) + C$ //Aplicación notación O
   $O(n) = 2(n*m)$ //Regla de la suma

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

O(n) = n*m //Regla del producto

8. **n** es el número de vértices del grafo
   m es el número de aristas del grafo

4) Simulacro de Parcial

1.

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

2. *0-> [3,4]*
   1-> [0,2,5]
   2-> [1,4,6]
   3-> [7]
   4-> [2]
   5-> []
   6-> [2]
   7-> []

3. *b) O(n2)*

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

6) Trabajo en Equipo y Progreso Gradual

a) Actas de reunión

Integrante	Fecha	Hecho	Haciendo	Por Hacer
Santiago Soto	10/02/18		Ejercicio 2.1	Ejercicios 3.6
Santiago Soto	10/02/18	Ejercicio 2.1	Ejercicios 3.6	Ejercicios 3.7
Santiago Soto	17/02/18	Ejercicios 3.6	Ejercicios 3.7	Simulacro Parcial
Kevyn Gomez	10/02/18		Ejercicios 1	Ejercicios 3
Kevyn Gomez	18/02/18	Ejercicios 3	Simulacro Parcial	Puntos opcionales

b) El reporte de cambios en el código

```
commit 24bc2b60d08fbd4f21346757c5b3e7dbf25ab7d9 (HEAD -> master, origin/master, origin/HEAD)
Author: ksgomezp <ksgomezp@eafit.edu.co>
Date:   Sun Feb 18 22:07:36 2018 -0500

    Subiendo codigo  ejercicios 1.*

commit 98dcd8ac52970ae649bba7cd5f106cfc21b664f4
Author: ksgomezp <ksgomezp@eafit.edu.co>
Date:   Sun Feb 18 20:39:32 2018 -0500

    Subiendo Informe

commit 0c9ff66ca50e7eb9f028f2183090dc4f9e2e442b
Author: Santiago Soto <ssotom@eafit.edu.co>
Date:   Sun Feb 18 19:45:11 2018 -0500

    Ejercicio en Linea

commit 48a6ae161c799d0add7dffaabd4f80ec5e45af50
Author: Santiago Soto <ssotom@eafit.edu.co>
Date:   Sun Feb 18 19:35:29 2018 -0500

    Ejerciio en linea
```

c) El reporte de cambios del informe de laboratorio