

Práctica de laboratorio No. 1: Implementación de gráficos

Juan Felipe Ortiz Salgado
Medellín, Colombia
jfortizs@eafit.edu.co

Samuel Meneses Diaz
Medellin Colombia
smenesesd@eafit.edu.co

3) Práctica para la presentación final de la defensa del proyecto

3.1 La estructura de datos para representar la ciudad era priorizar el tiempo sobre la memoria porque facilita el acceso a los datos. Por este motivo, el algoritmo implementa una matriz de adyacencia. El tipo de datos que guardan la matriz es un tipo de datos abstracto, que recopila el peso y el nombre del arco.

3.2 La complejidad de la memoria de una matriz de adyacencia es $O(n^2)$. Para este caso específico, el gráfico tiene 300.000 vértices el consumo de memoria es de 300.000 x 2.

3.3. Se utilizó un operador condicional que evalúa si el nodo de recién leído era igual a 10.000, se cambia por un 0.

3.4. El algoritmo funciona de una manera sencilla. La idea principal es visitar cada nodo utilizando la primera búsqueda del método de recorridos de gráficos. De esta manera, for cada nodo se asignará un color y ellos visitan los nodos adyacentes del mismo de forma recursiva. Si un nodo tenía un color asignado anteriormente, el algoritmo lo comparará con el color del nodo anterior para determinar si tiene el mismo color. En el caso afirmativo, el método devolverá false.

3.5

```
-clase pública Algoritmo {
    Booleana estático público DFSColorFC(DigraphAM2 g)
        String[] Visitado á nueva cadena[g.size]; C_1
        Int origen = g.getFirst(); C_2
        return DFSColorFCAux(g, origen, visitados, "verde");
    }

    booleano estático privado DFSColorFCAux(DigraphAM2 g, int origen, String[] v, String color) á if(v[] ?
        null) // C_3
        for(Entero s : g.getSuccessors(origen)) // O(n)
            si(color.equals("verde")) // C_4
                v[origen] = "amarillo"; // C_5
            si(! DFSColorFCAux(g, s, v, "amarillo")) devuelve
                false; C_7
```

PhD. Mauricio Toro Bermúdez

Profesor | Escuela de Ingeniería | Correo electrónico de
informática y sistemas: mtorobe@eafit.edu.co | Oficina:
Edificio 19 – 627 Teléfono: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 2

Código ST0247

```

    }
    •ado C_8
      v[origen] = "verde"; // C_9

-if(! DFSColorFCAux(g, s, v, "verde")) devuelve false; // C_11
    }
  }
}
•elseá // C_12
  if(v[origen].equals(color)) // C_13
    devolver false; // C_14
  •elseá // C_15
    if(color.equals("verde")) // C_16
      devolver true; // C_17
    •elseá // C_18
      devolver true; // C_19
    }
  }
}
devolver true; // C_20
}
}

```

PhD. Mauricio Toro Bermúdez

Profesor | Escuela de Ingeniería | Correo electrónico de
informática y sistemas: mtorobe@eafit.edu.co | Oficina:
Edificio 19 – 627 Teléfono: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 2
Código ST0247

3.6 La variable (n) representa el entre los vértices que tienen el gráfico y puede representar el número de sucesores que tienen cada vértice.

4) Práctica para

exámenes Parciales 4.1

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

4.2

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

4.3

b) $O(n-2)$

PhD. Mauricio Toro Bermúdez

Profesor | Escuela de Ingeniería | Correo electrónico de
 informática y sistemas: mtorobe@eafit.edu.co | Oficina:
 Edificio 19 – 627 Teléfono: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 2
Código ST0247

PhD. Mauricio Toro Bermúdez

Profesor | Escuela de Ingeniería | Correo electrónico de
informática y sistemas: mtorobe@eafit.edu.co | Oficina:
Edificio 19 – 627 Teléfono: (+57) (4) 261 95 00 Ext. 9473

