

Laboratorio Nro. 5 Grafos

Miguel Ángel Sarmiento Aguiar
Universidad Eafit
Medellín, Colombia
msarmie4@eafit.edu.co

Marlon Pérez Ríos
Universidad Eafit
Medellín, Colombia
mperezr@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

- 3.1 Este algoritmo primeramente lee los dos archivos CSV que corresponden a los vértices y a los arcos. Después, corrige los nombres de cada una de las columnas. Posteriormente crea dos diccionarios (tablas de Hash): uno para los vértices, el cual asigna los nombres a cada uno de los ID, y se considera que las coordenadas no son información relevante, por lo que no se pone; y el segundo para los arcos, el cual asigna la distancia a cada una de las conexiones entre los vértices, considerando que el nombre no es relevante, por lo que no se pone tampoco.
- 3.2 Si se representa el mapa de Medellín con matrices de adyacencia se consumiría demasiada memoria, ya que al haber alrededor de 300 mil vértices, dicha matriz de adyacencia quedaría de 300 mil filas por 300 mil columnas, y los elementos dentro de ella serían en su gran mayoría ceros, debido a que cada lugar de referencia está conectado con unos cuantos lugares de referencia más por pocas calles. Por lo tanto, no es óptimo hacerlo de este modo.
- 3.3 El problema de que los identificadores de los puntos del mapa no empiecen en cero se resuelve al utilizar diccionarios, se asocian con el ID propio del vértice, y no con el número de fila al cual pertenecen.
- 3.4 Para resolver el problema se utiliza una lista madre que contiene en cada uno de sus ítems la matriz de adyacencia de cada uno de los grafos que se ingresan en la entrada. Este algoritmo primeramente coge la entrada, y según las reglas que se determinaron para la construcción de esta, crea una lista madre de matrices que ya se mencionó. Seguidamente, cada una de las matrices contenida en esta lista se ingresa al algoritmo responsable de determinar si el grafo asociado a dicha matriz es bipartito (dicho algoritmo fue copiado de internet y fue referenciado dentro de él mismo). Finalmente, se imprime el resultado de éste análisis.

3.5

```
class Graph():
    def __init__(self, V):
        self.V = V
        self.graph = [[0 for column in range(V)] \
```

```
# C1
# C2*n^2
```

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473



ESTRUCTURA DE DATOS 1

Código ST0245

```

for row in range(V)]

def isBipartite(self, src):
    colorArr = [-1] * self.V
    colorArr[src] = 1
    queue = []
    queue.append(src)
    while queue:
        u = queue.pop()
        if self.graph[u][u] == 1:
            return False
        for v in range(self.V):
            if self.graph[u][v] == 1 and colorArr[v] == -1:
                colorArr[v] = 1 - colorArr[u]
                queue.append(v)
            elif self.graph[u][v] == 1 and colorArr[v] == colorArr[u]:
                return False
    return True

```

C3*n
C4
C5
C6
C7*n
C8*n
C9*n
C10
C11*n^2
C12*n^2
C13*n^2
C14*n^2
C15*n^2
C16
C17

$$T(n) = C1 + C2*n^2 + C3*n + C4 + C5 + C6 + C7*n + C8*n + C9*n + C10 + C11*n^2 + C12*n^2 + C13*n^2 + C14*n^2 + C15*n^2 + C16 + C17$$

$$T(n) = O(C1 + C2*n^2 + C3*n + C4 + C5 + C6 + C7*n + C8*n + C9*n + C10 + C11*n^2 + C12*n^2 + C13*n^2 + C14*n^2 + C15*n^2 + C16 + C17)$$

$$T(n) = O(6*n^2 + 4*n)$$

$$T(n) = O(n^2)$$

3.6 n es el número de vértices del grafo

4) Simulacro de Parcial

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 -> [4, 6]

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 1

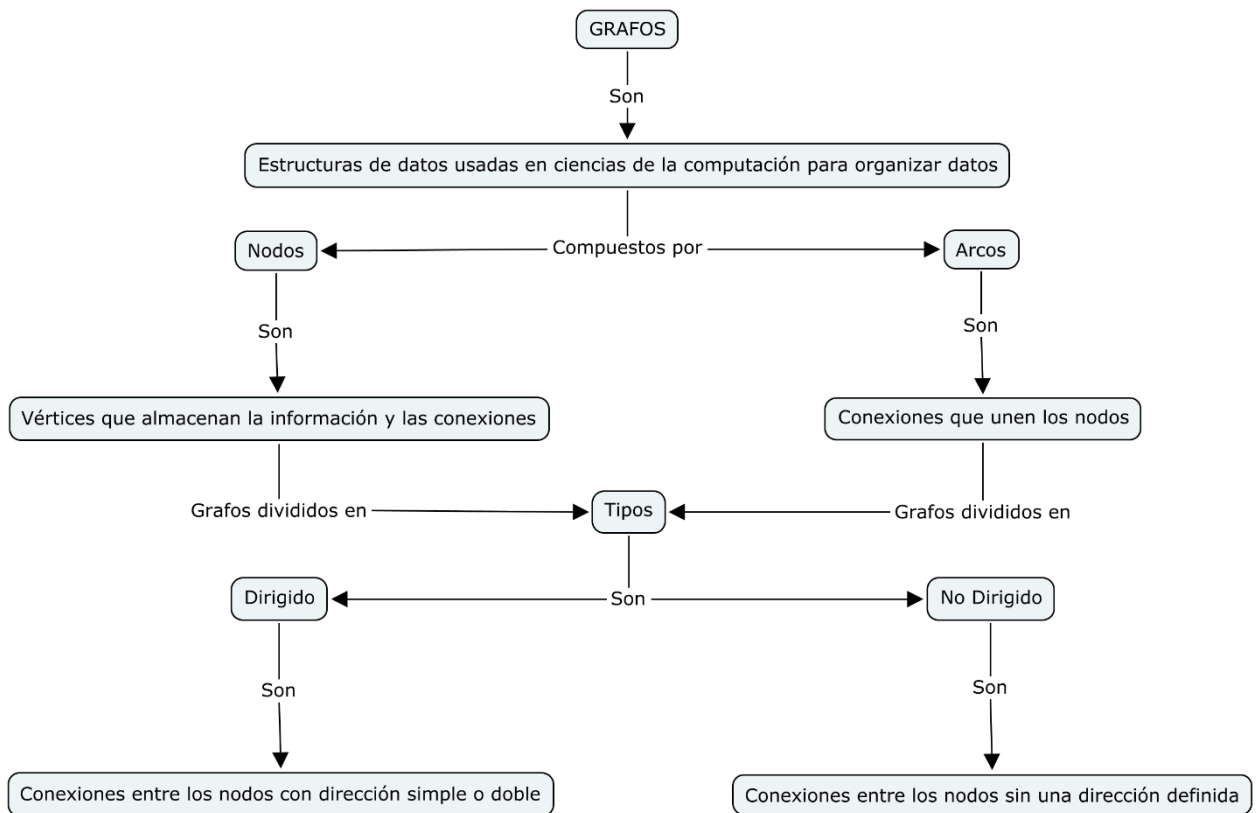
Código ST0245

3 -> [7]
 4 -> [2]
 5 -> []
 6 -> [2]
 7 -> []

4.3 b

4.4 4.4.1 ii
 4.4.2 i

5) Lecturas Recomendadas



PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
 Tel: (+57) (4) 261 95 00 Ext. 9473