

Laboratorio Nro. 5 Grafos

Sebastian Castaño Orozco
Universidad Eafit
Medellín, Colombia
scasta31@eafit.edu.co

Dennis Castrillón Sepúlveda
Universidad Eafit
Medellín, Colombia
dcastri9@eafit.edu.co

1) Simulacro de proyecto

1.1 Se entrega en la carpeta código el ejercicio propuesto. A continuación, se muestra el código.

2) Simulacro de maratón de programación

2.1 Se entrega el código de solución del problema de decidir si dado un grafo conexo arbitrario, ese grafo se puede colorear de 2 distintos colores. A continuación, el código.

```
"""
Estructura de datos y algoritmos 1
Laboratorio 5
Punto 2.1
Sebastian Castaño Orozco 201610054014
Dennis Castrillón Sepúlveda 201610035014
"""

import numpy as np

class Graph:
    def __init__(self, size):
        self.size = size
        aux = []
        matriz = []
        for i in range(self.size+1):
```

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

```

        aux = []
        for j in range(self.size+1):
            aux.append(None)
        matriz.append(aux)
    array = np.array(matriz)
    self.matriz = array

def addArc(self, vertex, edge, weight=1):
    vertex_exist = False
    edge_exist = False
    fila = None
    columna = None
    for i in range(1,len(self.matriz),1):
        if vertex == self.matriz[0][i]:
            vertex_exist = True
        if edge == self.matriz[0][i]:
            edge_exist = True
    if vertex_exist == False:
        for i in range(1,len(self.matriz),1):
            if self.matriz[0][i] == None:
                self.matriz[0][i] = vertex
                self.matriz[i][0] = vertex
                break
    if edge_exist == False:
        for i in range(1,len(self.matriz),1):
            if self.matriz[0][i] == None:
                self.matriz[0][i] = edge
                self.matriz[i][0] = edge
                break
    for i in range(len(self.matriz)):
        if self.matriz[i][0] == vertex:
            fila = i
        if self.matriz[0][i] == edge:
            columna = i
    self.matriz[fila][columna] = weight
    self.matriz[columna][fila] = weight
    return (self.matriz)

n = int(input("Ingrese número de nodos: "))
g = Graph(n)

```

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

```

arcos = int(input("Ingrese número de arcos: "))
for i in range(arcos):
    arcos_pares = str(input("Ingrese arco (Nodo inicial seguido por nodo final Ej:
23): "))
    vert1 = arcos_pares[0]
    vert2 = arcos_pares[1]
    b = g.addArc(vert1,vert2)
a = b[1:,1:]
colores = True
for i in range(len(a)):
    for j in range(len(a)-1):
        if a[i][j] == a[i][j+1]:
            colores = False

if colores == True:
    print("Los nodos del grafo:\n " + str(b) + "\n pueden ser coloreados con dos
colores")
else:
    print("Los nodos del grafo:\n " + str(b) + "\n NO pueden ser coloreados con dos
colores")

```

2.2 Se entrega el código de solución del problema Funny Game en la carpeta ejercicioEnLinea. Cabe resaltar que este código se extrae de internet con el fin de calcular mas adelante su complejidad y entender su funcionamiento.

2.3 Se entrega el código de solución del problema Claw Decomposition en la carpeta ejercicioEnLinea. Cabe resaltar que este código se extrae de internet con el fin de calcular mas adelante su complejidad y entender su funcionamiento.

2.4 Se entrega el código de solución del problema Lightning Away en la carpeta ejercicioEnLinea. Cabe resaltar que este código se extrae de internet con el fin de calcular más adelante su complejidad y entender su funcionamiento.

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



3) Simulacro de preguntas de sustentación de Proyectos

3.1 Escriban una explicación del numeral 1. Digan cómo funciona, cómo está representado el mapa de la ciudad, por ejemplo, utilizaron matrices, listas, tablas de hash, ¿por qué?

El numeral 1 consiste en calcular las posibles rutas que podrían tomar ciertas personas que van hacia un destino determinado, priorizando el uso de la menor cantidad de vehículos sin superar por un factor p el valor original de demora que toman esas personas para ir al destino en su carro de manera individual. Además de la condición de no superar el cupo máximo de 5 personas por vehículo.

El algoritmo se encarga entonces de leer el archivo txt en el que se encuentran las coordenadas x,y de cada ubicación del punto de origen de cada persona. Luego se encuentra el tiempo que se demora la persona en ir a cada uno de los destinos, el punto de llegada o destino común, y a su vez, el punto de origen de las demás personas que van hacia ese destino. Luego, se realizan las diferentes permutaciones entre las personas y se evalúa el tiempo total que tomaría cada persona realizando los distintos caminos y se compara este tiempo con el tiempo original que le toma a la persona para llegar al destino.

Luego, de comparar este tiempo para la persona origen, se deben comparar los demás tiempos de las personas que iría recogiendo en el camino, priorizando que ese tiempo no supere el tiempo original multiplicado por un factor p (1.2 o 1.7) definido por las personas.

Posteriormente, se escoge la mejor ruta, priorizando entonces que las personas que estén mas lejos del destino común sean las que recojan a las que están cerca de su camino o ruta habitual, sin superar el cupo máximo del carro y minimizando el tiempo total de recorrido para cada uno de los integrantes de la ruta.

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

El mapa de la ciudad está representando mediante coordenadas geodésicas. El algoritmo se implementó a través de listas de listas o matrices, debido a que de esta manera se puede almacenar la información según un orden preestablecido por nosotros mismos.

3.2 Si representamos el mapa de Medellín del numeral 1 con matrices de adyacencia, ¿Cuánta memoria consumiría? Tengan en cuenta que hay alrededor de 300,000 vértices

Si se representara el mapa de Medellín en una matriz de adyacencia con 300.000 vértices ocuparía un aproximado de 3,6 E¹¹ bytes, equivalentes a 360 Gigabytes

3.3 ¿Cómo solucionaron el problema de que los identificadores de los puntos del mapa no empiezan en cero?

Este problema se solucionó de manera anticipada ya que para la implementación se usó listas de listas y no grafos.

3.4 Expliquen la estructura de datos que utilizan para resolver el problema, y cómo funcionan los algoritmos realizados en el numeral 2.1 y los ejercicios opcionales que hayan hecho del punto 2. Esto en 3 a 6 líneas de texto.

3.5 Calculen la complejidad del ejercicio 2.1

La complejidad del algoritmo 2.1 es $O(n^2)$ ya que se recorre cada posición de la matriz generada (grafo) y en este caso, las matrices son cuadradas, recorriendo la cantidad de filas multiplicada por la cantidad de columnas, siendo las filas=columnas.

3.6 Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del numeral 3.5

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



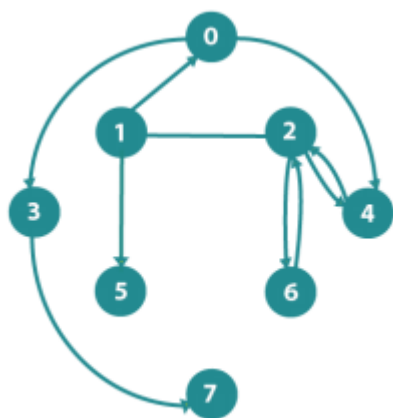
ESTRUCTURA DE DATOS 1

Código ST0245

Para este caso, n hace referencia a la cantidad de vértices. En este caso, el usuario define la cantidad de vértices cuando inicializa el grafo, internamente el programa hace la matriz del tamaño n , donde n es el numero de vértices. Por ende, la complejidad será recorrer esta matriz, vértice a vértice.

4) Simulacro de Parcial

4.1



	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								

4.2

0 > [3,4]

1 > [0,2,5]

2 > [1,4,6]

3 > [7]

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

4 > [2]

5 > []

6 > [2]

7 > []

4.3

b) $O(n^2)$

4.4

4.4.1 ii) 1, 4, 5, 0, 2, 3

4.4.2 i) 1, 4, 5, 0, 2, 3

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