

Laboratorio Nro. 5 Grafos



Objetivos: 1. Comparar las ventajas y desventajas de implementaciones dinámicas y estáticas de estructuras de datos. 2. Escoger la estructura de datos apropiada para resolver un problema dado. 3. Resolver problemas fundamentales de grafos, incluyendo búsqueda DFS y BFS.



Consideraciones: Lean y verifiquen las consideraciones de entrega,



Leer la Guía



Trabajo en
Parejas



Si tienen reclamos,
regístrenlos en
<http://bit.ly/2g4TTKf>



Ver
calificaciones
en Eafit
Interactiva



En el GitHub
docente,
encontrarán la
traducción de
los Ejercicios
en Línea

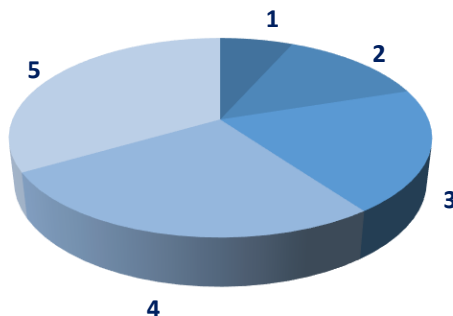


Hoy, plazo de
entrega



Subir el **informe pdf** en la carpeta **informe**, el **código del ejercicio 1** en la carpeta **codigo** y el **código del 2** en la carpeta **ejercicioEnLinea**

Porcentajes y criterios de evaluación



- 1. Simulacro sustentación proyecto
- 2. Análisis de complejidad
- 3. Código de laboratorio
- 4. Simulacro de parcial
- 5. Ejercicio con juez en línea

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

Resumen de ejercicios a resolver

1.1 Diseñar un algoritmo para disminuir el tráfico mediante la estrategia de vehículos compartidos. El algoritmo debe encontrar el mínimo número de vehículos particulares para que todos los dueños de vehículos particulares de una empresa puedan ir a trabajar recogiendo unos a otros respetando unas restricciones.

2.1 Decidir si dado un grafo conexo arbitrario se puede colorear con 2 colores.

2.2 [Ejercicio Opcional] <http://bit.ly/2gTLZ53>

2.3 [Ejercicio Opcional] <http://bit.ly/2hGqJPB>

2.4 [Ejercicio Opcional] <http://bit.ly/2hrrCfS>

2.5 [Ejercicio Opcional] <http://bit.ly/2k8CGSG>

3.1 [Ejercicio Opcional] Escriban una explicación **del numeral 1**. Digan cómo funciona, cómo diseñaron el algoritmo para lograr reducir el tráfico y respetar las restricciones.

3.2 Si representamos el mapa de Medellín del numeral 1 con matrices de adyacencia o con listas de adyacencia, ¿cuál sería la diferencia en el consumo de memoria?

3.3 [Ejercicio Opcional] 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.

3.4 Calculen la complejidad del ejercicio 2.1 y, si los hicieron, de los Ejercicios Opcionales

3.5 Expliquen con sus palabras las variables (*qué es 'n', qué es 'm', etc.*) del cálculo de complejidad del numeral 3.4

4. Simulacro de Parcial

5. [Ejercicio Opcional] Lectura recomendada

6. [Ejercicio Opcional] Trabajo en Equipo y Progreso Gradual

7. [Ejercicio Opcional] Utilicen la plantilla dispuesta en este idioma para el laboratorio.

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

1. Simulacro de Proyecto

Códigos para entregar en GitHub en la carpeta `codigo`

1 Simulacro de Proyecto: Códigos para entregar en GitHub en la carpeta código



Vean Guía
numeral 3.4



Código del ejercicio en línea
en **GitHub**. Vean Guía en
numeral 4.24



Documentación
opcional. Si lo
hacen, utilicen
Javadoc o
equivalente. No
suban el HTML a
GitHub.



No se reciben archivos
en **.RAR** ni en **.ZIP**



Utilicen Java, C++ o Python

El alto tráfico que vivimos en la ciudad de Medellín trae varios problemas como son: una disminución en nuestra calidad de vida, disminución en la calidad del aire, disminución de la productividad, entre otros. Una solución a este problema es que los dueños de vehículos particulares compartan sus carros para así no tener una persona por vehículo. En particular, esta solución es fácilmente aplicable a una empresa porque los trabajadores se conocen y llegan a la misma hora a trabajar. Una solución de este tipo también permitiría reducir los tiempos de espera en los parqueaderos de la empresa.



► **Diseñar un algoritmo** para disminuir el tráfico mediante la estrategia de vehículos compartidos. El algoritmo debe **encontrar el mínimo número de vehículos particulares** para que todos los dueños de vehículos particulares de una empresa puedan ir a trabajar recogiendo unos a otros. Respetas las restricciones a continuación.

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



Utilicen los conjuntos de datos que se encuentran en la carpeta *datasets*, en Github, para probar su algoritmo.



Para esto, suponemos que, actualmente, cada dueño de vehículo particular va solo a su lugar de trabajo y toma siempre la ruta más corta. También suponemos que cada dueño puede transportar máximo 4 pasajeros más en su vehículo. La restricción es que, para cada dueño de vehículo, el tiempo que se demora al trabajo, recogiendo a otros compañeros no aumente más de una proporción constante p .



Definición formal del problema:

Sea U las casas de los empleados dueños de un vehículo particular, d la ubicación de la empresa, y $V = U \cup \{d\}$ los vértices de un grafo completo $g = (V, E)$, cuyas aristas $E \subset V \times V \times \mathbb{N}$ tienen como peso la duración en minutos de la ruta más corta de un vértice $v_i \in V$ a un vértice $v_j \in V$. Encuentre \mathbb{P} , una partición del conjunto U , donde para cada $S_i \in \mathbb{P}$, $1 \leq |S_i| \leq 5$, y donde el número de particiones $|\mathbb{P}|$ sea el mínimo. Además, encuentre una permutación P_i , para cada $S_i \in \mathbb{P}$, que respete la siguiente restricción: Para cada elemento v_i de la permutación $P_i = [v_1, v_2 \dots v_n]$, de cada $S_i \in \mathbb{P}$, el nuevo tiempo para llegar a la empresa $t'_{i,d} = \sum_{j=i}^{n-1} t_{j,j+1} + t_{n,d}$ donde $(v_j, v_k, t_{j,k}) \in E$, cumpla con $t'_{i,d} \leq t_{i,d} \times p$ donde $(v_i, d, t_{i,d}) \in E$ y $p > 1$ es una proporción constante.

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

2. Simulacro de Maratón de Programación sin documentación HTML, en GitHub, en la carpeta `ejercicioEnLinea`

2 Simulacro de Maratón de Programación: sin documentación, en GitHub, dentro de ejercicioEnLinea



Vean Guía
numeral 3.3



No se requiere
documentación para
los ejercicios en línea



Utilicen Java,
C++ o Python



No se reciben archivos
en **.PDF** ni **.TXT**



No se reciben
archivos en **.RAR**
ni en **.ZIP**



Código del ejercicio en
línea en **GitHub**. Vean
Guía en numeral 4.24

2.1 Resuelvan el siguiente ejercicio

En 1976, el teorema de colorear un mapa con 4 colores fue probado con la ayuda de un computador. Este teorema muestra que un mapa puede ser coloreado solamente con 4 colores, de tal forma que no haya una región coloreada usando el mismo color que un vecino. Aquí hay un problema similar a ese problema, pero es mucho más simple.



Ustedes tienen que decidir si dado un grafo conexo arbitrario, ese grafo se puede colorear con 2 colores. Esto quiere decir, si uno puede asignar colores (de una paleta de 2 colores) a los nodos, de tal forma que no haya 2 nodos adyacentes del mismo color. Para simplificar el problema ustedes pueden asumir que:

- No hay un nodo que tenga un arco a sí mismo.
- El grafo es no dirigido, es decir que, si un nodo *a* está conectado a un nodo *b*, usted puede asumir que el nodo *b* también está conectado al nodo *a*.

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 DATOS1
Código ST0245

- c. El grafo será fuertemente conexo. Esto quiere decir, que hay al menos un camino de un nodo de grafo a cualquier otro nodo.



Entrada

La entrada consiste en varios casos de prueba. Cada caso de prueba comienza con una línea que tiene un número n ($1 < n < 200$) de nodos diferentes. La siguiente línea contiene el número de arcos.

Posteriormente, las siguientes líneas, cada una contiene 2 número que especifican que existe un arco entre dos nodos.

Un nodo en el grafo se representa con un número a ($0 < a < n$). Una entrada con $n = 0$ simboliza el fin de la entrada y no debe ser procesada.



Salida

Ustedes tienen que decidir si el grafo de entrada puede ser coloreado con dos colores o no, y deben imprimirlo como se muestra a continuación.



Ejemplos de las entradas

```
3
3
0 1
1 2
2 0
3
2
0 1
1 2
9
```

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 DATOS1
Código ST0245

8
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0



Ejemplos de la salida

NOT BICOLORABLE.
BICOLORABLE.
BICOLORABLE.



[Opc] Para los numerales 2.2 al 2.5, resuelvan los siguientes problemas

2.2

<http://bit.ly/2qTLZ53>

2.3

<http://bit.ly/2hGqJPB>

2.4

<http://bit.ly/2hrrCfS>

2.5

<http://bit.ly/2k8CGSG>

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 Proyecto en la carpeta informe

3 Simulacro de preguntas de sustentación de Proyecto: En la carpeta informe



Vean **Guía**
numeral 3.4



Exporten y entreguen
informe de laboratorio en
PDF, en español o Inglés




Si hacen el **informe en español**, usen la **plantilla en español**




No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**

Sobre el Simulacro de Proyectos

- 3.1**  **[Opc]** Escriban una explicación entre 3 y 6 líneas de texto **del código del numeral 1**. Digan cómo funciona, cómo hicieron para resolver el problema de reducir el tráfico y respetar las restricciones del problema. Pueden apoyarse realizando una imagen para facilitar su explicación.


- 3.2**  Si representamos el mapa de Medellín del numeral 1 con matrices de adyacencia o con listas de adyacencia, ¿cuál sería la diferencia en el consumo de memoria? ¿El grafo que trabajamos es un grafo completo o no?


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 DATOS1
Código ST0245

Sobre el simulacro de maratón de programación

3.3  **[Opc]** Expliquen, con sus propias palabras, 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 cada uno.

3.4  Calculen la complejidad del ejercicio 2.1 y, si los hicieron, de los Ejercicios Opcionales.

3.5  Expliquen con sus palabras las variables (*qué es 'n', qué es 'm', etc.*) del cálculo de complejidad del numeral 3.4. Ver ejemplo a continuación:



Ejemplo de esta respuesta:

“n es el número de elementos del arreglo”,

“V es el número de vértices del grafo”,

“n es el número de filas de la matriz y m el número de columnas”.

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

4. Simulacro de parcial en informe PDF

4 Simulacro de Parcial en el informe PDF

Resuelvan los ejercicios



Para este simulacro, agreguen **sus respuestas** en el informe PDF.



El día del Parcial no tendrán computador, JAVA o acceso a internet.



Si hacen el **informe en español**, usen la **plantilla en español**



Exportar y entregar informe de laboratorio en **PDF, en español o inglés**

Si hacen **el informe en inglés**, usen a **plantilla en inglés**



No apliquen **Normas Icontec** para esto

4.1 *Microsoft Game Studios* es, hoy en día, uno de las empresas que controlan la saga de videojuegos de *Age of Empires*. Una de la peculiaridades que tienen los videojuegos de esta saga es que los participantes se pueden ubicar en islas, distribuidas sobre el mar, que deben controlar para poder lograr la victoria. Dada una matriz booleana 2D, encuentra el número de islas. Un grupo de 1s conectados forma una isla. Por ejemplo, la siguiente matriz contiene 5 islas.

```
Entrada : mat[][] = {{1, 1, 0, 0, 0},
                    {0, 1, 0, 0, 1},
                    {1, 0, 0, 1, 1},
                    {0, 0, 0, 0, 0},
                    {1, 0, 1, 0, 1}}

Salida  : 5
```

El problema se puede resolver fácilmente aplicando *recorrido primero en profundidad* (en inglés, *DFS*) en cada punto. En cada llamado a DFS se visita una isla. Llamaremos

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

a DFS en el siguiente nodo no visitado. El número de llamados a DFS da el número de islas.

Si trabajas en Java, revisa este código:

```

1  class Islands {
2      static final int ROW = 5, COL = 5;
3
4      // Verificar si una celda (fila , columna)
5      // se le puede llamar un DFS
6      boolean isSafe(int M[][], int row, int col,
7                     boolean visited[][]) {
8          return (row >= 0) && (row < ROW) && (col >=
9                  0) && (col < COL) && (M[row][col] == 1 &&
10                 !visited[row][col]);
11     }
12
13     // Un DFS en una booleana matriz. Este
14     // considera como vecinos a los 8 adyacentes
15     void DFS(int M[][], int row, int col, boolean
16             visited[][]) {
17         int rowNbr[] = new int [] { -1, -1, -1, 0, 0,
18                                     1, 1, 1 };
19         int colNbr[] = new int [] { -1, 0, 1, -1, 1,
20                                     -1, 0, 1 };
21         visited[row][col] = true;
22         for (int k = 0; k < 8; ++k)
23             if (isSafe(M, row + rowNbr[k], col + colNbr
24                         [k], visited))
25                 .....
26     }
27
28     // Contar el numero de islas
29     int countIslands(int M[][]) {
30         boolean visited[][] = new boolean[ROW][COL];
31         int count = 0;
32         for (int i = 0; i < ROW; ++i)
33             for (int j = 0; j < COL; ++j)
34                 if (M[i][j] == 1 && !visited[i][j]) {
35                     DFS(M, i, j, visited);
36                     ++count;
37                 }
38         return count;
39     }
40 }

```

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

Si trabajas en Python, revisa este código:

```

1  class Graph:
2      def __init__(self, row, col, g):
3          self.ROW = row
4          self.COL = col
5          self.graph = g
6
7      # Verificar si una celda (fila, columna)
8      # puede ser incluida en el llamado al DFS
9      def isSafe(self, i, j, visited):
10         return (i >= 0 and i < self.ROW and
11                 j >= 0 and j < self.COL and
12                 not visited[i][j] and self.graph[i][j])
13
14     # Hacer DFS en una matriz 2D. Este
15     # considera como vecinos los 8 adyacentes
16     def DFS(self, i, j, visited):
17         rowNbr = [-1, -1, -1, 0, 0, 1, 1, 1];
18         colNbr = [-1, 0, 1, -1, 1, -1, 0, 1];
19         visited[i][j] = True
20         for k in range(8):
21             if self.isSafe(i + rowNbr[k], j + colNbr[k],
22                             visited):
23                 .....
24
25     # Retorna el
26     # numero de islas que hay en la matriz
27     def countIslands(self):
28         visited = [[False for j in range(self.COL)]
29                     for i in range(self.ROW)]
30         count = 0
31         for i in range(self.ROW):
32             for j in range(self.COL):
33                 if visited[i][j] == False and self.graph[i][j] == 1:
34                     self.DFS(i, j, visited)
35                     count += 1
36         return count

```

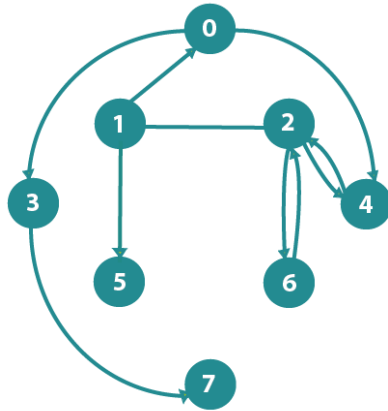
4.1.1 Completa la línea 18 en Java o la 22 en Python

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

- 4.2** Para el siguiente grafo, completen la representación de **listas de adyacencia**. Como el grafo no tiene pesos, sólo se colocan los sucesores en la lista. También completen la representación con **matrices de adyacencia**.

Consideren que, si no hay arco, por simplicidad, se deja el espacio en blanco.



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

0 ->[3,4]

1 ->

2 ->

3 ->

4 ->

5 ->

6 ->

7 ->

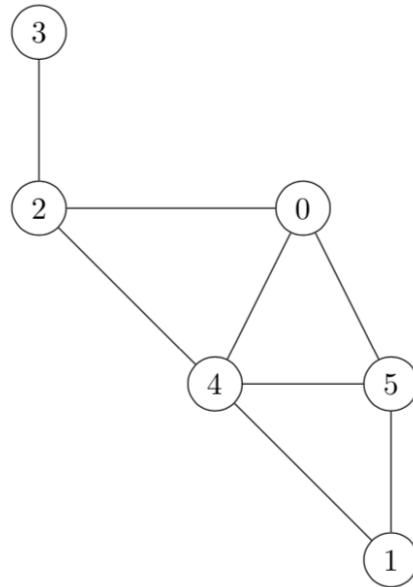
- 4.3** ¿Cuánta memoria (ojo, no tiempo sino memoria) ocupa una representación usando listas de adyacencia para un grafo dirigido con n vértices en el peor de los casos?

- a) $O(n)$
- b) $O(n^2)$
- c) $O(1)$
- d) $O(\log n)$
- e) $O(n \cdot \log n)$

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

4.4 Se tiene el siguiente grafo no dirigido:



4.4.1 ¿Cuál es un recorrido de *búsqueda primero en profundidad* del grafo anterior, si como nodo inicial se toma el nodo 1?

- i) 1, 5, 0, 3, 2, 4
- ii) 1, 4, 5, 0, 2, 3
- iii) 1, 4, 0, 3, 5, 2
- iv) 1, 5, 4, 0, 3, 2

4.4.2 ¿Cuál es un recorrido de *búsqueda primero en amplitud* del grafo anterior, si se toma como nodo inicial el nodo 1?

- i) 1, 4, 5, 0, 2, 3
- ii) 1, 5, 0, 2, 3, 4
- iii) 1, 4, 2, 0, 3, 5
- iv) 1, 3, 0, 4, 5, 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 DATOS1
Código ST0245

4.5 En la vida real, un grafo se utiliza para representar los mapas de *Google Maps*. Para cada una de las siguientes proposiciones, determine si son verdaderas o falsas.

4.5.1 Al representar un grafo completo con matrices de adyacencia y con listas de adyacencia, en ambos casos, el consumo de memoria (NO tiempo, sino memoria) es $O(n)$.

- (a) Verdadero
- (b) Falso

4.5.2 Cuando representamos un grafo con matrices de adyacencia, la complejidad asintótica, en el peor de los casos, de insertar un nuevo vértice es $O(n^2)$, donde n es el número de vértices.

- (a) Verdadero
- (b) Falso

Pista: Un grafo completo es aquel donde existe una arista entre cada par de vértices del grafo.

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



5. [Opcional] Lecturas Recomendadas

5 [Opc] Lecturas recomendadas



Vean *Guía* en **numeral 3.5 y 4.20**



Exportar y entregar informe de laboratorio en **PDF**, en **español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**



"El ejercicio de una profesión requiere la adquisición de competencias que se sustentan en procesos comunicativos. Así cuando se entrevista a un ingeniero recién egresado para un empleo, una buena parte de sus posibilidades radica en su capacidad de comunicación; pero se ha observado que esta es una de sus principales debilidades..." Tomado de <https://aprendeonlinea.udea.edu.co/revistas/index.php/ingeso/article/viewFile/13986/12398>



Lean a **"Robert Lafore, Data Structures and Algorithms in Java (2nd edition), Chapter 13: Graphs. 2002"** y sumen puntos adicionales, así:



Hagan un mapa conceptual con los principales elementos teóricos.

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

Otras sugerencias de lectura

Si desean otras lecturas, consideren las siguientes:



Thomas Cormen, Introduction to Algorithms (3th edition), Sections 23.2, 23.3 y 23.5. 2009, que pueden encontrar en biblioteca



“John Hopcroft et al., Estructuras de Datos y Algoritmos, Capítulo 7: Grafos no dirigidos. 1983” que pueden encontrar en biblioteca.

6. [Opcional] Trabajo en Equipo y Progreso Gradual

6 [Opc] Trabajo en Equipo y Progreso Gradual



Vean Guía en **numeral 3.5 y 4.20**



Exportar y entregar informe de laboratorio en **PDF**, en **español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**



El trabajo en equipo es imprescindible. "Algunos medios retratan la programación como un trabajo solitario, la realidad es que requiere de mucha comunicación y trabajo con otros. En la vida laboral serás parte de un equipo de trabajo y deberás comunicarte con otras personas". Tomado de <http://bit.ly/2qJKzJD>

- 6.1 ► Entreguen copia e integrantes que participaron
- 6.2 ► Entreguen el reporte de *git* con los cambios en el código y quién hizo cada cambio, con fecha, hora e integrantes que participaron
- 6.3 ► Entreguen el reporte de cambios del informe de laboratorio que se genera *Google docs* o herramientas similares

! **Nota:** Estas respuestas también deben incluirlas en el informe PDF

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

7. [Opcional] Laboratorio en Inglés con plantilla en Inglés

7 [Opc] Laboratorio en inglés



Vean *Guía* en **numeral 3.5 y 4.20**



Exportar y entregar informe de laboratorio en **PDF, en español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**



El inglés es un idioma importante en la Ingeniería de Sistemas porque la mayoría de los avances en tecnología se publican en este idioma y la traducción, usualmente se demora un tiempo.

Adicionalmente, dominar el inglés permite conseguir trabajos en el exterior que son muy bien remunerados. Tomado de goo.gl/4s3LmZ



Entreguen el código y el informe en inglés.

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

Ayudas para resolver los Ejercicios

Ayudas para el Ejercicio 1.....	<u>Pág. 26</u>
Ayudas para el Ejercicio 2.1.....	<u>Pág. 26</u>
Ayudas para el Ejercicio 2.2.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 2.3.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 2.4.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 3.4.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 3.5.....	<u>Pág. 28</u>
Ayudas para el Ejercicio 5.....	<u>Pág. 28</u>
Ayudas para el Ejercicio 6.1.....	<u>Pág. 28</u>
Ayudas para el Ejercicio 6.2.....	<u>Pág. 29</u>
Ayudas para el Ejercicio 6.3.....	<u>Pág. 29</u>



Ayudas para el Ejercicio 1



Pista 1: Vean en **numeral 4.13** “Cómo usar Scanner o BufferedReader”



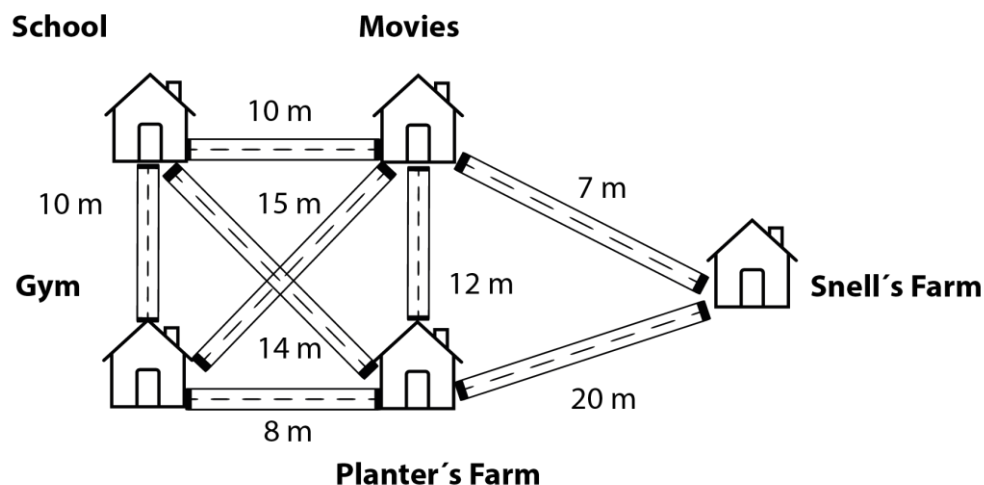
Pista 2: Es mejor usar *BufferedReader* porque es más rápido que *Scanner*. La idea es leer en una cadena de caracteres el contenido de cada línea y usando el método *split* de la clase *String* o usando *StringTokenizer*, dividir la cadena en partes cada que hay una coma (,).



Pista 3: El vértice con el identificador 1 es la empresa (d) y los identificadores del 2 al *n* son los dueños de los vehículos (U). Para cada vértice, se conoce la coordenada geodésica del lugar. Para cada arista del grafo, el peso representa el tiempo en minutos de la ruta más corta para ir de un vértice a otro.



Como un ejemplo, para el siguiente mapa, el archivo de entrada es el siguiente:



p 1.2

Vertices. Formato: ID, coordenada x, coordenada y, nombre

```

1 5.00000 2.00000 Snell
2 4.00000 1.00000 Movies
3 2.00000 5.00000 Planters
4 0.00000 2.00000 Gym
  
```

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 DATOS1
Código ST0245

5 2.00000 0.00000 School

Costo de Caminos Cortos. Formato: ID, ID, peso

1 2 7
1 3 19
1 4 27
1 5 17
2 1 7
2 3 12
2 4 15
2 5 10
3 1 19
3 2 12
3 4 8
3 5 14
4 1 27
4 2 15
4 3 8
4 5 10
5 1 17
5 2 10
5 3 14
5 4 10

Salida

La salida es una partición del conjunto de dueños de autos U . En cada línea se coloca una permutación de un subconjunto de la partición del conjunto U . Como un ejemplo, si una permutación es [4 3 2], quiere decir que el dueño 4 recoge al dueño 3, luego al dueño 2 y luego van a la empresa. Como otro ejemplo, si una permutación es [5] quiere decir que el dueño 5 va solo a la empresa.

Un ejemplo de salida válida para la entrada anterior, con $p = 1.2$, es:

4 3 2
5

Otro ejemplo de salida válida para la entrada anterior, con $p = 1.2$, es:

5 2
4 3

Si tuviéramos $p = 1.7$, para el ejemplo anterior, una posible respuesta es:

5 4 3 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



Ayudas para el Ejercicio 2.1



Pista 1: Usen un algoritmo para corroborar si es un grafo bipartito. Léase qué es un grafo bipartito en <http://bit.ly/2hGwAo2>



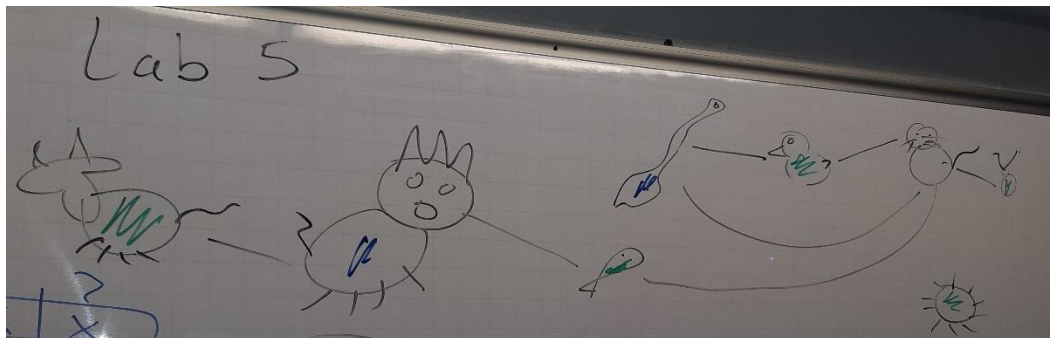
Pista 2: Si desean, pueden usar DFS o BFS para resolver este problema, pero existe otro tipo de algoritmos para resolverlo también



Pista 3: ¡Spoiler Alert! En este sitio web explican un algoritmo para verificar si un grafo es bipartito <http://bit.ly/2IOsQFZ>



Pista 3: La figura siguiente es un ejemplo de cómo colorear un grafo con dos colores. En dicho ejemplo, los animales de un zoológico son los vértices y los arcos representan la relación de depredación. Colorear el grafo con dos colores equivale a encontrar 2 jaulas tales que los animales de cada jaula no se coman entre sí.



Ayudas para el Ejercicio 2.2



Pista 1: Utilicen **Búsqueda en Profundidad** (Siglas en inglés DFS)



Pista 2: Vean Guía en **numeral 4.13** “Cómo usar Scanner o BufferedReader

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



Ayudas para el Ejercicio 2.3



Pista: Usen un algoritmo para corroborar si es un grafo bipartito. Léase qué es bipartito en <http://bit.ly/2hGwAo2>



Ayudas para el Ejercicio 2.4



Pista: Algoritmos para hallar componentes fuertemente conexos. Ordenamiento topológico. DFS. Léase en <http://bit.ly/2gTeJKh>



Ayudas para el Ejercicio 3.4



Pista: Vean **Guía en numeral 4.11** “Cómo escribir la complejidad de un ejercicio en línea”



Ayudas para el Ejercicio 3.5



Errores Comunes

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



Ayudas para el Ejercicio 5



Pista: Para que hagan el mapa conceptual se recomiendan herramientas como las que encuentran en <https://cacoo.com/> o <https://bit.ly/2QM9D3M>



Nota 1: Si desean otra lectura, consideren la siguiente: “*John Hopcroft et al., Estructuras de Datos y Algoritmos, Capítulo 6: Grafos dirigidos. Páginas 267 – 276. 1983*” que pueden encontrarla en biblioteca



Nota 2: Estas respuestas también deben incluirlas en el informe PDF



Ayudas para el Ejercicio 6.1



Pista 1: Vean **Guía en numeral 4.21** “Ejemplo de cómo hacer actas de trabajo en equipo usando Tablero Kanban”



Ayudas para el Ejercicio 6.2



Pista 1: Vean **Guía en numeral 4.23** “Cómo generar el historial de cambios en el código de un repositorio que está en svn”

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



Ayudas para el Ejercicio 6.3



Pista 1: Vean *Guía en numeral 4.22* “**Cómo ver el historial de revisión de un archivo en Google Docs**”

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

¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez

Teléfono: (+57) (4) 261 95 00 Ext. 9473

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agenden una cita dando clic en la pestaña

-*Semana*- de <http://bit.ly/2gzVg10>