

Estructura de Datos y Análisis de Algoritmos Manual de Usuario Laboratorio 2

Ariel Ignacio Tirado Maturana

Profesor:

Jacqueline Kohler Casasempere

Alejandro Cisterna Villalobos

Ayudante:

Gerardo Zuñiga Leal

Santiago - Chile
2-2016

TABLA DE CONTENIDOS

Tabla de Contenidos.....	1
Índice de Figuras	1
CAPÍTULO 1. INTRODUCCIÓN	3
CAPÍTULO 2. Como compilar y ejecutar	4
2.1 Linux (Ubuntu).....	4
2.1.1 Compilar.....	4
2.1.2 Ejecutar.....	4
2.2 Windows	5
2.2.1 Compilar.....	5
2.2.2 Ejecutar.....	5
2.3 Entradas y salidas del programa	6
CAPÍTULO 3. Posibles errores	7

ÍNDICE DE FIGURAS

<i>Imagen 2-1: Como compilar en Linux.</i>	4
<i>Imagen 2-2: Como ejecutar en Linux.....</i>	4
<i>Imagen 2-3: Como compilar en Windows.</i>	5
<i>Imagen 2-4: Como ejecutar en Windows.</i>	5
<i>Imagen 2-5: Archivo de entrada.</i>	6
<i>Imagen 2-6: Resultado del algoritmo.....</i>	6

CAPÍTULO 1. INTRODUCCIÓN

En el presente manual de usuario, se le indicará a la persona que utilice el programa adjunto como ejecutarlo, que cosas debe tener en consideración a la hora de su ejecución y se le indicarán los posibles errores a la hora de ejecutarlo.

El programa solicitado a los alumnos del curso de Análisis de Algoritmos y Estructuras de Datos para su tercera entrega de laboratorio consiste en la creación de un algoritmo, escrito en el lenguaje de programación C, el cual tiene como objetivo verificar si un grafo ingresado es conexo o no, ordenar sus vértices según su centralidad de grado y finalmente, ordenarlos según su centralidad betweenness, mostrando estos resultados por pantalla.

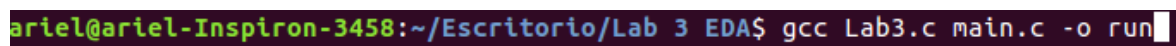
CAPÍTULO 2. COMO COMPILAR Y EJECUTAR

2.1 LINUX (UBUNTU)

2.1.1 Compilar

Para compilar en alguna distribución de Linux, más específicamente Ubuntu, debe acceder a la terminal del Sistema operativo y acceder desde ahí a la carpeta donde están ubicados los archivos *Lab3.c*, *main.c* y *Lab3.h*. Para esto debe ingresar el comando *cd* seguido de la dirección de la carpeta almacenadora (Cuando se inicia la terminal, por defecto se está en la raíz del Sistema Operativo).

Luego de realizar este procedimiento, se debe proceder a compilar los archivos. Para esto debe ingresar el comando indicado en la siguiente imagen:



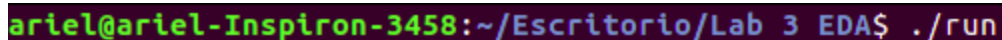
```
ariel@ariel-Inspiron-3458:~/Escritorio/Lab 3 EDA$ gcc Lab3.c main.c -o run
```

Imagen 2-1: Como compilar en Linux.

Al presionar el botón *Enter* se compilará el programa y se generará un archivo ejecutable llamado *run*.

2.1.2 Ejecutar

Luego de haber compilado el programa, se debe proceder a ejecutarlo. Para esto se debe escribir en la línea de comandos el nombre que se le dio al ejecutable a la hora de compilar, como se muestra en la imagen:



```
ariel@ariel-Inspiron-3458:~/Escritorio/Lab 3 EDA$ ./run
```

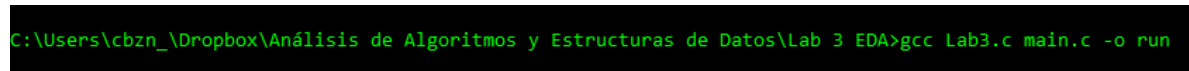
Imagen 2-2: Como ejecutar en Linux.

Luego de presionar el botón *Enter*, el programa se ejecutará.

2.2 WINDOWS

2.2.1 Compilar

Para compilar en Windows, debe acceder al Símbolo del Sistema, ingresando la línea *cmd* en el buscador del Sistema Operativo. Luego de haber ingresado al Símbolo del Sistema, debe buscar la carpeta donde están ubicados los archivos *main.c*, *Lab3.c* y *Lab3.h*, esto se logra usando el comando *cd* seguido de la dirección de la carpeta almacenadora. Una vez que se encuentre en la carpeta, se debe compilar el archivo usando el comando indicado en la siguiente imagen:



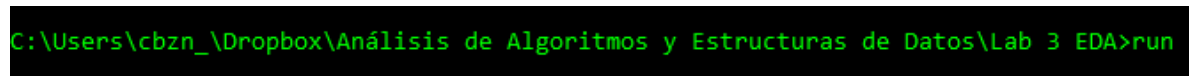
```
C:\Users\cbzn_\Dropbox\Análisis de Algoritmos y Estructuras de Datos\Lab 3 EDA>gcc Lab3.c main.c -o run
```

Imagen 2-3: Como compilar en Windows.

Al presionar el botón *Enter* los archivos se compilarán y se creará un archivo ejecutable llamado *run*.

2.2.2 Ejecutar

Luego de haber compilado los archivos, el programa estará listo para ser ejecutado. Para realizar este paso, el usuario debe ejecutar el archivo ejecutable creado anteriormente, como se muestra en la siguiente imagen:



```
C:\Users\cbzn_\Dropbox\Análisis de Algoritmos y Estructuras de Datos\Lab 3 EDA>run
```

Imagen 2-4: Como ejecutar en Windows.

2.3 ENTRADAS Y SALIDAS DEL PROGRAMA

Se considera el siguiente archivo de texto como ejemplo para mostrar los resultados del programa.

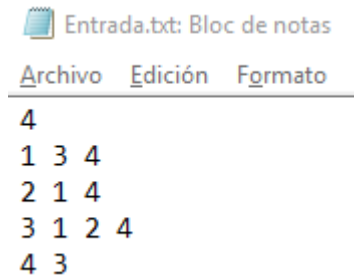
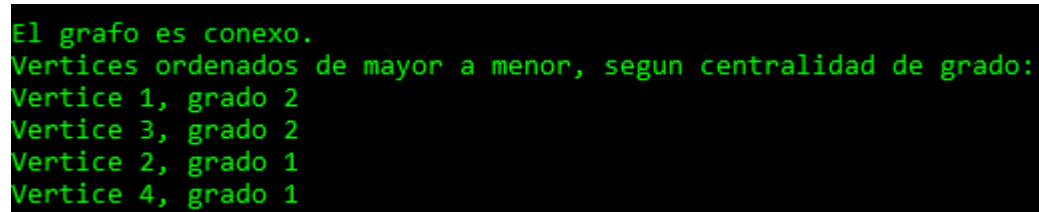


Imagen 2-5: Archivo de entrada.

Generando la siguiente salida por pantalla:



```
El grafo es conexo.
Vertices ordenados de mayor a menor, segun centralidad de grado:
Vertice 1, grado 2
Vertice 3, grado 2
Vertice 2, grado 1
Vertice 4, grado 1
```

Imagen 2-6: Resultado del algoritmo.

CAPÍTULO 3. POSIBLES ERRORES

A continuación se indicará al usuario un listado de posibles errores que pueden ocurrir con el uso del programa:

- El archivo contiene un formato definido y el programa solo puede leerlo en ese formato. El formato considerado contempla en la primera línea la cantidad de vértices del grafo, seguido de un salto de línea. Luego viene una línea por cada vértice con todos sus vértices adyacentes, separados por un espacio, y con un salto de línea luego del último vértice adyacente. No cumplir con este formato para los archivos de entrada puede provocar una lectura errónea de datos, lo que puede desencadenar en bucles infinitos en la ejecución.