EVALUACIÓN 2 - LABORATORIO
ESTRUCTURA DE DATOS
Docente Jazna Meza Hidalgo
Agosto 2022

APRENDIZAJE ESPERADO

Comprobar el correcto aprendizaje generado en clases de Estructuras de datos: manejo de grafos y tablas hash.

ESTRUCTURAS E IMPLEMENTACIÓN DISPONIBLES

Debe trabajar con la implementación de grafos y tablas hash entregadas en el laboratorio.

EJERCICIO 1 - GRAFOS - 4 PUNTOS

CONSIDERACIONES

1. Se tiene un laberinto formado por un conjunto de puertas y caminos o conexiones entre sus puertas.
2. El laberinto tiene UNA SOLA PUERTA DE ENTRADA y UNA SOLA PUERTA DE SALIDA.
3. El único movimiento válido es hacia adelante, es decir, no se permite retroceder.
4. Cada puerta está identificada por una letra. La entrada queda codificada como *e* y la salida como *s*. El resto de las puertas pueden ocupar para sus nombres el resto de las letras del abecedario.

Se dispone de 2 archivos de texto plano llamados: *puertas.txt* y *distancias.txt* que representan los nombres de las puertas y las distancias entre ellas de un laberinto. Ver la sección *FORMATO DE ARCHIVOS DE TEXTO* para más detalles de estos archivos.

FUNCIONES A IMPLEMENTAR

```
/* Obtiene el total de puertas usando la representación de la matriz de adyacencia */
int get_total_puertas(MATRIZ_ADYACENCIA, char);
/* Obtiene el total de puertas usando la representación de la lista de adyacencia */
int get_totalPuertasLA(LISTA_ADYACENCIA, char);
```

PRUEBA DE LA FUNCIONALIDAD

En la función main deberá implementar los siguientes requerimientos:

1. Cargar los datos desde los archivos de texto y crear 2 grafos: G_x representado usando **LISTA DE ADYACENCIA** y G_y usando **MATRIZ DE ADYACENCIA**. La figura 1 indica el contenido de los archivos de texto.
2. Mostrar la matriz de adyacencia
3. Mostrar la lista de adyacencia
4. Leer una letra que representa el nombre de una puerta dentro del laberinto. **Asuma que el nombre de la puerta EXISTE**.
5. Mostrar, usando G_x, la cantidad de puertas accesibles de formar directa desde la puerta indicada en el punto 4.
6. Mostrar, usando G_y, la cantidad de puertas accesibles de formar directa desde la puerta indicada en el punto 4.

FORMATO DE ARCHIVOS DE TEXTO

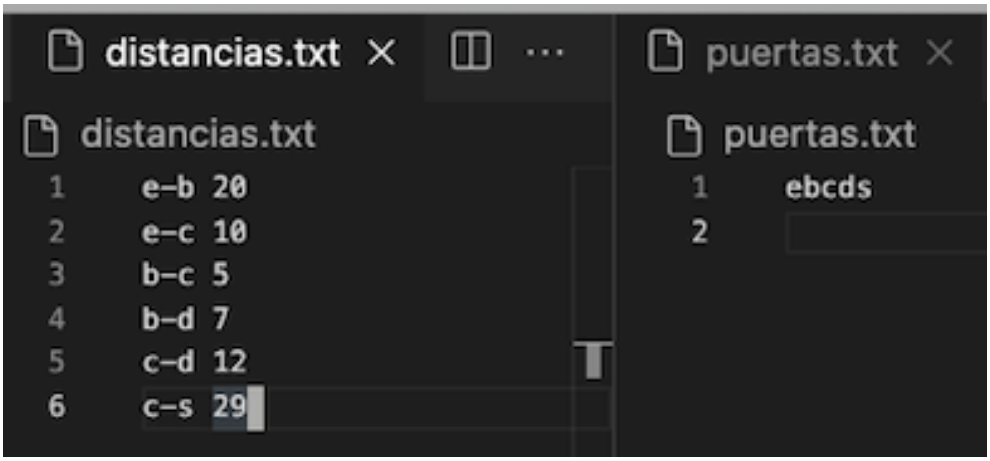

Figure 1: Formato de archivos de prueba

La figura 1 indica el formato y contenido de los archivos de texto.

- El archivo *puertas.txt* contiene una sola línea con una cadena de caracteres con el nombre de las puertas. El primer carácter es el nombre de la puerta de entrada y el último carácter es el nombre de la puerta de salida.
- El archivo *distancias.txt* contiene n líneas cada una con el formato de c_i-c_j *99* donde c_i es la puerta de inicio, c_j es la puerta de destino y 99 es un número que representa la distancia entre ambas puertas.

EJERCICIO 2 - HASHING - 2 PUNTOS

Considerando los puntos del plano cartesiiano almacenados en el archivo *puntos.txt* se pide:

1. Leer la capacidad que tendra la tabla *hash*
2. Definir una función *hash* para asignar claves a cada punto. La función debe ser única para estudiante, así es que se sugiere *creatividad* al momento de definirla.
3. Crear, usando los puntos del archivo, una tabla hash usando la función definida en el punto 2.
4. La tabla *hash* debe quedar guardada en el archivo *hash.txt*.
5. Leer desde teclado las coordenadas de un punto y buscar el punto dentro de la tabla *hash*, almacenada en memoria, indicando el éxito o fracaso de la búsqueda.

La figura 2 muestra un ejemplo de la tabla hash generada con la función hash $h(p, c) = MOD(p_x * 287 + p_y * 911, c)$ donde p es el punto y c es la capacidad de la tabla.

FORMATO DE ARCHIVO DE SALIDA

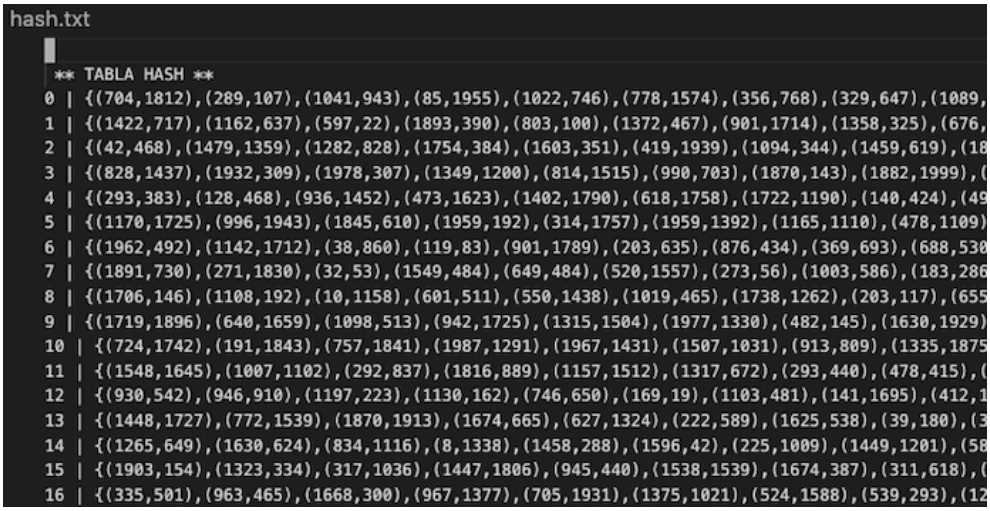

Figure 2: Formato de archivo GENERADO

La figura 2 contiene tantas líneas como la capacidad definida para la tabla *hash*. Cada línea está formada por la clave k_i y la lista de valores, en este caso puntos, cuya clave asignada fue k_i.

PLAZOS DE ENTREGA

En la plataforma MOODLE, **VIERNES 26 de agosto** hasta las 23:00. Se aceptan entregas posteriores de acuerdo a los siguientes descuentos:

FECHA DE ENTREGA	DESCUENTO A APLICAR
26 agosto a las 23:01 horas hasta el 26 agosto a las 23:59 horas	Descuento = 2 puntos
Después de las 23:59 horas del 26 agosto HASTA el 27 agosta a las 08:00 AM	Nota máxima a la que postula será un 3.0

CONDICIONES DE ENTREGA

1. El laboratorio es individual.
2. Entrega de código fuente con la debida documentación.
3. Las copias serán evaluadas con la nota mínima sin derecho a apelación.
4. Se debe programar en C.
5. Se debe respetar las estructuras y funciones entregadas.
6. Se debe respetar el tipo de estructura del parámetro de entrada de todas funciones, así como el tipo de datos de retorno. En caso de que no se respete ese requerimiento, la entrega **NO SE CORRIGE**.