Estructuras de Datos y Algoritmos

Práctico de máquina 2 - Año 2023

Fecha de entrega: Jueves 4 de mayo de 2023 hasta las 8 hs.

El servicio mecánico de la concesionaria de vehículos *"El Bólido"* necesita un sistema capaz de manejar la información de los servicios realizados a los vehículos de sus clientes. La información que se mantiene sobre los mismos es: la patente del vehículo, que es único e identifica de forma unívoca toda la información asociada al mismo; marca y modelo del vehículo; año de fabricación; nombre del dueño; su número de teléfono; servicio mecánico realizado; el importe del servicio y la fecha en que se realizó. La empresa necesita poder realizar consultas sobre un vehículo dado, ingresar la información de nuevos vehículo y eliminar los datos de aquellos que se considere no mantener.

Suponga que se cuenta con las siguientes estructuras de datos para almacenar la información planteada:

a) Árbol Binario de Búsqueda (ABB).

b) Lista Invertida con Búsqueda Binaria por Bisección (LI).

c) Lista Secuencial Ordenada con Búsqueda Binaria por Trisección(LSOBT).

Para resolver el problema deberá escribir un programa en Lenguaje C en el que se deben implementar los operadores necesarios para el manejo de cada una de las estructuras propuestas.
La solución deberá presentar un menú principal que provea las siguientes opciones: **Administración de Estructuras** y **Comparación de Estructuras**.

Administración de Estructuras: debe permitir elegir la estructura sobre la cual se desea trabajar. Una vez seleccionada la estructura utilizada, se deberá presentar un menú que permita las siguientes operaciones: **ingreso de nuevos vehículo, eliminar vehículo existentes, determinar si a un vehículo se le ha realizado servicio mecánico**, **consultar la información completa asociada al vehículo** y la opción **Mostrar Estructura**. La opción **Mostrar Estructura** debe mostrar por pantalla el contenido de la estructura seleccionada listando la información completa de los elementos presentes en ella, en el orden en que están almacenados, para el árbol binario de búsqueda se debe implementar un barrido pre-orden mostrando para cada nodo ademas de los datos las patentes de sus hijos izquierdo y derecho.

Comparación de Estructuras: Para llevar a cabo la comparación de las estructuras, se utilizará una secuencia de operaciones que se detallan en un archivo de texto llamado *"Operaciones.txt"*, que será provisto por la cátedra. Esta secuencia de operaciones se deberá realizar sobre cada una de las estructuras, asegurando que las mismas **no contengan ningún dato inicialmente**.

Esta opción debe realizar y mostrar una comparación adecuada de lo que cuesta, en cada una de las estructuras, **realizar ingresos, eliminaciones y consultas de un vehículo dado**. En el análisis debe considerar el peor escenario y el comportamiento esperado en cada caso.
Para el cálculo de los costos de ingreso y eliminación: cada corrimiento de nupla tiene costo **1,5 (uno coma cinco)** y se considerará **1 (uno)** por cada modificación de punteros. En el caso de aplicar política de reemplazo en el árbol se deberá contar un corrimiento de nuplas por la copia de datos más la modificación de puntero.
Para las consultas el costo se determinará en celdas consultadas para todas las estructuras **1 punto por cada celda**.
Una vez finalizada esta operación **deben quedar** los datos resultantes de efectuar las operaciones del archivo para ser alcanzados desde la administración de cada una de las estructuras. Adicionalmente deberá realizar un análisis de los resultados obtenidos y sacar una conclusión de los mismos; dicha conclusión deberá quedar plasmada al principio de su programa principal (donde se encuentra el main) como comentario (**incluir resultados de la comparación**).

En el archivo *"Operaciones.txt"*, se describen las operaciones a realizar mediante un código de operación, seguido por los datos necesarios para la misma. Así, el código de ingreso de vehículo (1) estará seguido por la patente del vehículo a ingresar y todos sus datos (nupla completa), el código de eliminación (2) será seguido por la patente del vehículo a eliminar y todos sus datos (nupla completa), y el código de consulta (3) sólo seguido por la patente del vehículo a consultar.

El archivo de texto *"Operaciones.txt"* tiene el siguiente formato:

```
1                       /*Código primer operación (Alta)*/
AC123AA                 /*patente del vehículo*/
Chevrolet Onix          /*Marca y modelo del vehículo*/
2018                    /*Año fabricación del vehículo*/
Mariano Fazio Fernández /*nombre del dueño*/
261-4967890             /*teléfono del dueño*/
Cambio correa accesorios /*Servicio realizado al vehículo*/
5000.00                 /*importe del servicio*/
2019-02-13              /*fecha del servicio*/
3                       /*Código segunda operación (Consulta)*/
AC123LA                 /*patente del vehículo*/
                .
                .
                .
2                       /*Código i-esima operación (Baja)*/
AB213QA                 /*patente del vehículo*/
Ford Fiesta             /*Marca y modelo del vehículo*/
2017                    /*Año fabricación del vehículo*/
Jacobo Feldman          /*nombre del dueño*/
266-4967825             /*teléfono del dueño*/
Embrague                /*Servicio realizado al vehículo*/
42800.00                /*importe del servicio*/
2019-02-13              /*fecha del servicio*/
```

Consideraciones a tener en cuenta:

- Los grupos deben ser de 2 integrantes.
- Se esperan 250 vehículos.
- La confirmación del elemento en la rutina de baja en administración de estructuras debe realizarse por pantalla para todas las estructuras.
- No se utilizarán elementos ficticios de ningún tipo en ninguna de las estructuras
- Para **Búsqueda por Bisección** se utilizará la siguiente consigna: límite inferior inclusivo, límite superior exclusivo, segmento más grande a la derecha y testigo a izquierda.
- Para **Búsqueda por Trisección** se utilizará la siguiente consigna: límite inferior inclusivo, límite superior inclusivo, segmento más grande a la izquierda.
- La política de reemplazo en la baja del **ABB** será *menor de los mayores*.
- Sobre los datos:
  - La patente del vehículo es una secuencia de 7 caracteres.
  - La marca y modelo del vehículo puede contener un máximo de 60 caracteres.
  - El año de fabricación es un entero.

  - El nombre del dueño puede contener un máximo de 50 caracteres.
  - El número de teléfono puede contener un máximo de 15 caracteres.
  - El servicio mecánico realizado puede contener un máximo de 70 caracteres.
  - El importe del servicio es un real positivo.
  - La fecha en que se realizó es una secuencia de 10 caracteres (*AAAA-MM-DD*).
- El ingreso de datos **no debe ser sensible a mayúsculas y minúsculas**, esto significa que al buscar una patente de un vehículo deberá ser reconocido independientemente de cómo se ingresen las letras del mismo.
- El programa deberá desarrollarse en Lenguaje C, utilizando como herramienta para tal fin **Code::Blocks** (disponible en www.codeblocks.org).

Nota Importante: La entrega del práctico se realiza por medio de la página de la materia y se debe enviar el archivo fuente del programa. El nombre del archivo deberá estar conformado de la siguiente manera: ***P****nroP*-***Grupo****nroG* donde *nroP* es reemplazado por el número de práctico que se entrega y *nroG* por el número del grupo al que pertenece el programa. Por ejemplo, el nombre P1-Grupo22.c corresponde al práctico de máquina 1 enviado por el grupo 22. **Los programas cuyos nombres no respeten estas reglas de conformación no serán aceptados**.

Ejemplo de rutina para Lectura de Operaciones

El código que se presenta a continuación es una guía para programar una rutina que permita leer datos desde un archivo de texto. **Deberá adaptarlo a la situación planteada.**

```
int Lectura_operaciones()
{    .... //declaraciones
        FILE *fp;
        if (( fp = fopen ( ''Operaciones.txt'' , ''r'') )==NULL)
                return 0;
        else {
        while(!(feof(fp))){
            fscanf(fp, ''%d'', &codigoOperador);
            fscanf(fp, ''%c'', &barraene );
            fscanf(fp, ''%[^\n]'', &aux.CodArt );
            if(codigoOperador==1||codigoOperador==2){
                fscanf(fp,'' %[^\n]'',aux.TipoArt);
                fscanf(fp,'' %[^\n]'',aux.Marca);
                fscanf(fp,'' %[^\n]'',aux.Desc);
                fscanf(fp,''%f'',&aux.Valor);
                fscanf(fp,''%d'',&aux.Cant);
                //llamar al operador correspondiente (Alta o Baja)
                //en todas las estructuras
            }else if(codigoOperador==3){
                //llamar a Evocar en todas las estructuras
            }else{
            //error, código operación no reconocido
            }
            codigoOperador=0;
    }
                return 1;
        }
        fclose(fp);
}
```