Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Computación
Estructuras de Datos y Algoritmos I

Trabajo Práctico Final

1. Motivación del problema

El Trabajo va a estar orientado al manejo de la informaicón de una *Agenda de Contactos*. Se puede diseñar cualquier nueva estructura de datos que considere apropiada, o se puede simplemente usar alguna de las desarrolladas en clase. Una *Agenda de Contactos* contiene datos que necesita como nombre, apellido, edad, género, número de teléfono, correo electrónico, fecha de cumpleaños, etc. Se puede suponer que el número de teléfono es único para todos los contactos.

2. Objetivos

El programa que se tendrá que realizar, será una *Agenda de Contactos* con la funcionalidad adecuada para manipularlos agregando una serie de funciones que requeriran de una estructura adecuada (o varias) para el manejo de la información.

3. Detalles de la implementación

Separemos las funciones en 3 categorías:

a) Básicas

- Interfaz de usuario: Un menú con opciones que permitan acceder a la funcionalidad del programa.
- Buscar: Dado un atributo de un contacto, encontrar otro (o todos) los atributos de ese contacto.
- Inserción: Insertar un nuevo contacto en nuestra Agenda.
- Eliminación: Eliminar un contacto de nuestra Agenda usando algún atributo específico que permita identificarlo de manera unívoca.
- Edición: Actualización de un atributo de un contacto dando un atributo específico que permita identificarlo de manera unívoca.
- Carga: Lectura de contactos desde un archivo externo con un formato que debe ser especificado en la documentación. El volumen de datos a cargar puede ser variable.
- Grabación: Escritura de la Agenda de Contactos en un archivo externo (debe contener el mismo formato que la Carga).

b) Intermedias

- Max (Min): Encuentra el máximo (mínimo) de un atributo específico
- Average: Calcula el promedio de un atributo (numérico).
- Deshacer: Se puede deshacer la última acción de inserción o eliminación realizada.
- Rehacer: Rehace la última acción deshecha.
- AND: Dado el valor de 2 o más atributos encuentra los Contactos que coincidan con todos los valores dados.

- OR: Dado el valor de 2 o más atributos encuentra los Contactos que coincidan con, al menos, un valor.

c) Avanzadas

- Busqueda por patrones: Dado un valor para un atributo de la forma X*, encuentra todos los Contactos cuyo atributo comience con X, siendo X un string. Por ejemplo, si buscamos los teléfonos de la forma: 34157* deberíamos obtener todos los contactos cuyos teléfonos comiencen con el patrón dado.
- : Ordenar: la lista de Contactos debería poder ordenarse por cualquier atributo. Esto debería funcionar para todos los atributos, mostrándolos ordenados por el atributo pedido.
- Conexión: Supongamos que las relaciones de todos los Contactos se almacenan en el libro de contactos, y todos los contactos de tus Contactos también están en la*Agenda de Contactos*. Dado dos de tus contactos: A y B, la función indica si A puede obtener los datos de B sin ti. Por ejemplo, supongamos que tenemos a cuatro contactos, representados como A, B, C, D y, nosotros estamos representados como Y y, las siguientes relaciones de amistad están almacenadas en la *Agenda de Contactos*

$$\begin{array}{c} \\ Y \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{ccccc} Y & A & B & C & D \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

donde 1 indica que el contacto de la fila y columna son contactos directamente y 0 en caso contrario. En este caso, se puede ver que D puede contactar a B sin Y ya que D puede contactar a A y, a través de él, llegar a B. Observe que las relaciones de contacto debe ser almacenadas.

Se pide que implemente todas las funciones básicas, 2 intermedias y 1 avanzada.

4. Evaluación

Para la evaluación del Trabajo Práctico se tomarán en cuenta los siguientes elementos:

a) Estructura de datos usada

b) Algoritmo propuesto

c) Eficiencia

d) Tiempo (Complejidad)

e) Calidad del código entregado