

Programación 2

Práctico 1 - Estructuras Lineales en Memoria Dinámica

Objetivos

- Trabajar sobre estructuras lineales en memoria dinámica, accediendo directamente o usando operaciones.
- Poner el foco en la diferencia entre el pasaje de parámetros por copia y el pasaje de parámetros por referencia, teniendo en cuenta los efectos secundarios de cada uno.
- Trabajar con funciones y procedimientos tanto totales, como parciales (que contemplan precondiciones).

Ejercicio 1

Considere la representación para una Lista Encadenada de Enteros que se presenta:

```
struct nodo {
    int elem;
    nodo * sig ;
};
typedef nodo * lista ;
```

(a) Utilizando la representación de la Figura 1, implemente las siguientes funciones en forma iterativa y sin usar procedimientos auxiliares.

- last** : dada una lista no vacía, retorna su último elemento.
- average** : dada una lista no vacía, retorna el promedio de sus elementos.
- insert** : dados un entero x y una lista l ordenada: retorna el resultado de insertar a x en l ordenadamente.
- snoc** : que dados un entero x y una lista l, retorna el resultado de insertar a x al final de l.
- remove** : que dados un entero x y una lista l, retorna el resultado de eliminar a x de l.

(b) Las estructuras que devuelve su solución: ¿comparten memoria con los parámetros? En caso afirmativo: ¿qué problemas puede acarrear esto?

(c) Analice las operaciones anteriores indicando cuáles son totales y cuáles parciales. ¿Es posible en cada caso pensar en ambas variantes (total y parcial)?

Ejercicio 2

Escriba el pseudocódigo de funciones recursivas que obtengan:

- la cantidad de ocurrencias de un elemento x en una lista L.
- la suma de los elementos de una lista L.

Ejercicio 3

Escriba el pseudocódigo de una función recursiva que verifique si un elemento x se encuentra en una lista L.

Ejercicio 4

Utilizando la representación para Lista Encadenada de Enteros del ejercicio 1 implemente las siguientes funciones en forma iterativa. Las soluciones retornadas no deben compartir memoria con los parámetros.

- (a) **take** : que dado un natural i y una lista l , retorna la lista resultado de tomar los primeros i elementos de l .
- (b) **drop** : que dado un natural i y una lista l , retorna la lista resultado de eliminar los primeros i elementos de l .
- (c) **merge** : que dadas dos listas ordenadas l y p , genera una lista intercalando ordenadamente ambas listas.
- (d) **append** : que dadas dos listas l y p , retorna una lista que contiene a los elementos de l y luego a los elementos de p , en el mismo orden.

Ejercicio 5

(a) Utilizando la representación para Lista Encadenada de Enteros del ejercicio 1 implemente los siguientes procedimientos en forma iterativa y sin usar procedimientos auxiliares.

- i. **insert** : que dados un entero x y una lista l ordenada, inserta a x en l ordenadamente
 - ii. **snoc** : que dados un entero x y una lista l , inserta a x al final de l
 - iii. **remove** : que dados un entero x y una lista l , elimina a x de l
- (b) ¿Qué diferencias encuentra entre las soluciones de este ejercicio y las que realizó para el ejercicio 1?