

Facultad de Informática
Ingeniería en Informática de Gestión
Estructuras de Datos y de la Información
Grupos A y B, Marzo 2003

Ejercicios sobre listas y secuencias

Ejercicio 1 Especificar las siguientes operaciones sobre listas:

- dada una lista, calcular la lista formada por los n primeros elementos de dicha lista,
- dada una lista, calcular la lista obtenida quitando los n primeros elementos de dicha lista,
- calcular el elemento de la lista que aparece más veces repetido (en caso de haber más de uno, se devolverá el que primero apareció),

Ejercicio 2 Especificar una operación *ordenada* que compruebe si una lista está ordenada (suponiendo que la lista es ordenable) y operaciones que ordenen de menor a mayor los elementos de una lista según:

- el método de ordenación por inserción,
- el método de ordenación por mezclas,
- el método de ordenación rápida (quicksort)

Ejercicio 3 Un número natural enorme se puede representar como una lista circular con doble enlace implementada dinámicamente, de forma que cada nodo contenga cuatro dígitos del número excepto el que contiene los dígitos más a la izquierda, que podrá tener menos. Un número enorme vendrá entonces dado por un puntero al nodo que contiene los dígitos más a la derecha. Se pide:

- Implementar las listas circulares doblemente enlazadas.
- Diseñar un procedimiento que dados dos naturales enormes representados como se ha descrito anteriormente, calcule su suma, entendiendo que la representación de los argumentos no puede ser modificada.
- Utilizando el procedimiento anterior diseñar un procedimiento que dados dos naturales enormes calcule su producto sin modificar su representación.

Ejercicio 4 Considerar un tipo abstracto de datos para las matrices cuadradas de enteros dispersas (es decir, con la mayoría de elementos a 0) y con operaciones:

- construir una matriz nula,
- obtener el valor en una posición dada,
- modificar el valor de una posición dada,

- sumar y multiplicar dos matrices,
- hallar la transpuesta de una matriz.

Implementar el tipo usando *listas* de triplas (*fila, columna, valor*) donde solamente aparezcan las posiciones no nulas. Discutir la conveniencia de ordenar la lista siguiendo algún criterio.

Ejercicio 5 Especificar el tipo de las *listas generales* sobre un tipo parámetro dado, donde una lista general puede ser vacía, o si no lo es, está formada por elementos del tipo parámetro o bien por listas generales. Así las listas $[1, 2, 3]$, $[[1], [2], [3]]$, $[1, [2], 3]$ y $[[1, 2, 3], [[1], [2], [3]], [1, [2], 3]]$ son ejemplos de listas generales sobre naturales. Las operaciones de las listas generales son:

- crear la lista general vacía,
- añadir un elemento (del tipo parámetro) al principio de la lista,
- añadir una lista general al principio de la lista,
- operaciones para devolver y eliminar el primer componente de una lista general,
- operaciones para devolver el primero y último elementos del tipo parámetro de una lista general,
- operaciones de concatenación, longitud e inversa de una lista general,
- una operación que invierta una lista general, invirtiendo asimismo sus componentes,
- una operación que calcule el nivel de anidamiento de una lista general,
- una operación que calcule la longitud máxima de todas las listas que forman parte (tal vez anidadas) de una lista general,

Implementar este tipo de datos usando memoria dinámica. Para ello será necesario utilizar un tipo *union* (registro con variante) que permita distinguir un nodo conteniendo elementos del tipo parámetro de un nodo que a su vez apunta a una sublista general.

Ejercicio 6 Resuelve los siguientes problemas aplicando el esquema de recorrido de secuencias. En cada caso el contenido de la secuencia debe quedar inalterado.

1. Contar el número de apariciones de 'a' en una secuencia de caracteres dada.
2. Contar el número de apariciones de vocales en una secuencia de caracteres dada.
3. Dada una secuencia de enteros, contar cuántas posiciones hay en ella tales que el entero que aparece en esa posición es igual a la suma de todos los precedentes.

Ejercicio 7 Dado un número natural $n \geq 2$, se llaman *números afortunados* a los que resultan de ejecutar el siguiente proceso: se comienza generando una cola que contiene los números desde 1 hasta n , en este orden. Se elimina de la cola un número de cada dos (es decir, los números 1,3,5 etc.); de la nueva cola se elimina ahora un número de cada tres etc. El proceso termina cuando se va a eliminar un número de cada m y el tamaño de la cola es menor que m . Los números que queden en

la cola en ese momento son los afortunados. Diseña un procedimiento que reciba n como parámetro y produzca una secuencia formada por los números afortunados resultantes.

Indicación: para eliminar de una cola de n números un número de cada m , hay que reiterar n veces el siguiente proceso: extraer el primer número de la cola, y añadirlo al final de la misma salvo si le tocaba ser eliminado.