

Estructuras de Datos I

Ejercicios de listas



Universidad
Rey Juan Carlos

Bloque 2 – Estructuras de datos
lineales

Ejercicios

1. Escribir un procedimiento de búsqueda de un elemento en una lista no ordenada y en una ordenada, que devuelva un puntero a la posición que ocupa el elemento si está. Sino está devuelve NULL.
2. Sean X e Y dos cadenas de caracteres representadas mediante listas enlazadas. Escribir un procedimiento que encuentre el primer carácter de X que no se encuentra en la cadena Y. (Supóngase que cada nodo de las listas almacena un único carácter)

Ejercicio 3

(**Prueba escrita práctica, 1.5 punto**) Supóngase una lista estática que simula memoria dinámica capaz de almacenar hasta 10 elementos de tipo entero. Supóngase un estado inicial (diagrama superior), y el siguiente estado (diagrama inferior) al que se llega tras aplicar una operación sobre la lista.

ESTADO INICIAL

pos	1	2	3	4	5	6	7	8	9	10
Elemento→	2	3	4	5	6	7	8	9	10	11
Siguiente→	7	5	6	1	3	8	10	0	2	0

cabLibres: 9

cabOcupados: 4

ESTADO FINAL

pos	1	2	3	4	5	6	7	8	9	10
Elemento→	2	3	4	5	6	7	8	9	10	11
Siguiente→	7	5	6	9	3	8	10	0	2	0

cabLibres: 4

cabOcupados: 1

- a) (**0.5 puntos**) ¿Qué operación se ha aplicado para pasar de un estado al siguiente?
Justifique su respuesta **con precisión**.

- b) **(1 punto)** Codifique la operación que logra pasar de un estado a otro y que pueda aplicarse de forma general a cualquier estado de la lista estática que simula memoria dinámica.

Ejercicio 4

- Especifica algebraicamente la operación entremezclar, que dadas dos listas devuelve otra con los elementos de las dos de entrada intercalados.

Ejercicio 5

- Dadas las siguientes ecuaciones que forman parte de una especificación algebraica, se pide:

`Misterio(Construir(e, estructura), e1) = Construir(e1, estructura)`

- a) Explicar qué hace la operación `Misterio`.
- b) Implementar la operación `Misterio`. La complejidad de la implementación debe ser $O(1)$.

Ejercicio 6

Consideremos la operación Posfijo que tiene la declaración siguiente:

$$\text{Posfijo: TipoLista} \times \text{TipoElemento} \rightarrow \text{TipoLista}$$

que a partir de la lista nos devuelve la sublista que hay a partir del elemento. Si el elemento no existe, nos devolverá la lista vacía. Ejemplos:

$$\begin{aligned}\text{Posfijo}([2,4,5,6,8], 5) &= [6,8] \\ \text{Posfijo}([2,4,5,6,8], 2) &= [4,5,6,8] \\ \text{Posfijo}([2,4,5,6,8], 9) &= []\end{aligned}$$

Se pide:

- Especificación algebraica de dicha operación
- Implementación en Pascal de dicha operación a partir de las operaciones del TAD TipoLista visto en clase.

Ejercicio 7

Dada la siguiente estructura de lista estática simulando dinámica doblemente enlazada con cursor cabecera:

1		2		3		4		5		6	
Ana		Pedro		Juan		Eva		Luis		Carlos	
2	3	5	4	2	6	2	0	0	2	2	0

cabLista: 5
cabLibres: 1

Escribir en una línea los valores numéricos de los enlaces en los nodos y en la siguiente línea los valores de las variables *cabLista* y *cabLibres* tras eliminar el elemento de la cabecera de la lista (no implementar)

Ejercicio 8

Consideremos la operación Prefijo que tiene la declaración siguiente:

Prefijo: TipoLista x TipoElemento \rightarrow TipoLista

Esta operación dada una lista y un elemento, nos devuelve la sublista de elementos que hay desde el comienzo de la lista hasta el elemento dado sin incluir. Si el elemento no existe, nos devolverá la lista original completa. Ejemplos:

$\text{Prefijo}([2,4,5,6,7,9],5) = [2,4]$

$\text{Prefijo}([2,4,5,6,7,9],2) = []$

$\text{Prefijo}([2,4,5,6,7,9],10) = [2,4,5,6,7,9]$

Se pide la especificación algebraica para dicha operación.

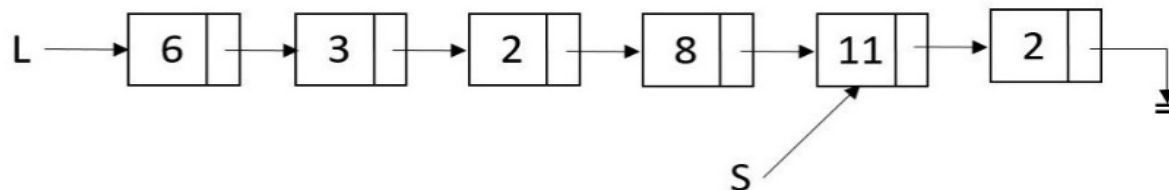
Ejercicio 9

Dada una lista definida de la siguiente forma:

```
TLista = ^TNode;  
TNode = RECORD  
    e: integer;  
    sgte: TLista;  
END;
```

Se pide implementar una operación que reciba una lista del tipo anterior y un número entero $N > 0$, y devuelva una variable de TLista cuya cabecera contenga la suma de los N primeros elementos de la lista original.

Si por ejemplo, se tiene la lista L de la siguiente figura y $N = 3$, se devolverá la variable S.



Restricción: debe hacerse en un solo recorrido de la lista original y se puede asumir que se está dentro de la unidad donde está definido el Tlista.

Ejercicios

10. Implementar una solución recursiva para las siguientes operaciones:

```
int LongitudRec (tLista l);
```

```
void borrarRecursivo(tLista *l, tElemento elem);
```

```
void Postfijo(tLista li, tElemento elem, tLista *lp);
```

11. Escribir una función recursiva NumPos que calcule el número de elementos positivos que hay en una lista enlazada simple apuntada por el puntero Lista. (Considérese que el campo Elemento de cada nodo es de tipo int).

Ejercicio 12

En la Universidad se reciben propuestas de proyectos para pedir una subvención para la realización de dicho proyecto. Cuando los proyectos llegan al servicio correspondiente, dichos proyectos se registran con un identificador formado por letras y números, el título del proyecto y el estado en el que se encuentra la propuesta. Este estado puede ser: E (En evaluación), A (Aceptado) o D (Denegado). Cuando se registra la petición de un nuevo proyecto se asigna el estado En evaluación. Ese estado se modifica posteriormente una vez decidida la concesión o no de la subvención.

Ejercicio 12 (cont.)

- a. Definir los tipos de datos para representar dicha información mediante una lista ordenada por el identificador del proyecto.
- b. Implementar una operación que devuelva el estado de un proyecto dado el identificador. Si el proyecto no existe devolver el código de estado X.
- c. Implementar una operación que inserte un nuevo proyecto en la estructura. La operación recibirá la información relativa al proyecto en una variable TipoProyecto que contenga toda la información del mismo.
- d. Implementar una operación que borre todos los proyectos de la estructura.