Tecnología de la programación
Sesión 16

Objetivos de la sesión

1. Solucionar el ejercicio propuesto en la sesión anterior (eliminar todas las ocurrencias de un entero n de una lista de nodos).
2. Estudiar las transparencias restantes para acabar el tema

Guion

Ejercicio eliminar todas las ocurrencias de un entero n de una lista de nodos

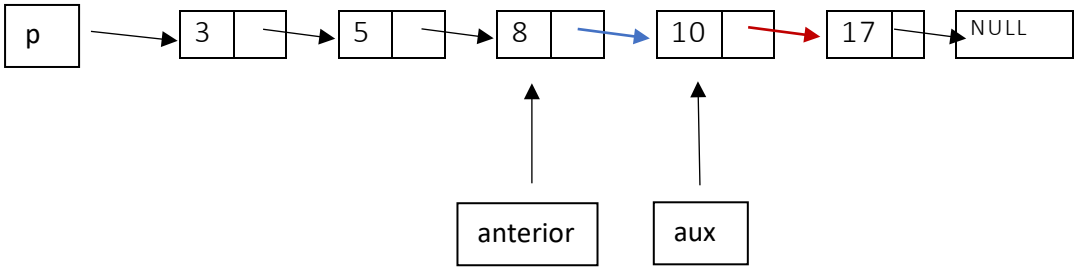
Hay varias formas de hacer el ejercicio. Una solución es la siguiente.

```
Acción eliminarTodasOcurrencias(E/S puntero a nodo p, Ent/ entero n)
Variables
      Puntero a Nodo aux, anterior
Principio
      mientras que p != NULL AND dest(p).dato==n
            aux = p
            p = dest(p).sig
            liberar(aux)
      fmq
      anterior = NULL
      aux = p
      mientras que aux != NULL
            si dest(aux).dato == n
                  dest(anterior).sig = dest(aux).sig
                  liberar(aux)
                  aux = dest(anterior).sig
            si_no
                  anterior = aux
                  aux = dest(aux).sig
            fsi
      fmq
Fin
```

Antes de nada, hay que darse cuenta de que el primer “mientras que” se encarga de eliminar al principio de la lista, en caso de que haya elementos a eliminar ahí (uno o varios seguidos).

Al igual que en el ejercicio de añadir en una lista ordenada, en este caso también necesitamos dos punteros. El puntero aux va avanzando, y cuando se encuentra con un nodo que contiene el elemento a eliminar, entonces

tenemos que hacer que el nodo anterior apunte al nodo siguiente de aux. Si no es igual, hay que ir avanzando. En la siguiente imagen se ve. Imaginaros que hay que eliminar el 10. Entonces cuando lleguemos ahí, lo que hay que hacer es que el Nodo que contiene el 8 (que está apuntado por el puntero anterior) debe apuntar ahora al nodo que contiene el 17. Eso se hace con la línea `dest(anterior).sig = dest(aux).sig`. De esa forma consigues que la flecha azul apunte a lo mismo que apunta la flecha roja.



Relación entre punteros y vectores

Estas transparencias son importantes para que os deis cuenta de que ambos conceptos están muy relacionados. Es más, los vectores pueden verse como punteros. Es importante que entendáis la aritmética de punteros: ¿Qué significa p+1? ¿Qué significa *(v+i)?

Ejercicios de la transparencia 63

Son sencillos, porque siguen el mismo esquema que los de listas enlazadas de nodos. De hecho, en el fondo son iguales.

```
Función longitud (cadena c) devuelve entero
{PRE: }
{POST: devuelve la longitud de c}
Variables
      puntero a Celda aux
      entero n
Principio
      n=0
      aux = c
      mientras que aux != NULL
            n = n+1
            aux = dest(aux).sig
      fmq
      devuelve n
Fin
```

```
acción añadir(E/S cadena, Ent/ carácter ch)
{PRE:}
{POST: c es la cadena obtenida de concatenar c con ch}
Variables
      puntero a Celda aux, nuevo
Principio
      nuevo = reservar(Celda)
      dest(nuevo).dato = ch
      dest(nuevo).sig = NULL
      si c==NULL entonces
            c=nuevo
      si_no
            aux = c
            mientras que dest(aux).sig != NULL
                  aux = dest(aux).sig
            fmq
            dest(aux).sig = nuevo
      fsi
fin

función esta? (cadena c, carácter ch) devuelve booleano
{PRE: }
{POST: devuelve VERDAD si ch está en c y falso en caso contrario}
Variables
      puntero a Celda aux
      booleano encontrado
Principio
      encontrado = Falso
      aux = p
      mientras que aux != NULL AND NOT encontrado hacer
            si dest(aux).dato == ch entonces
                  encontrado = Verdad
            fsi
            aux = dest(aux).sig
      fmq
      devuelve encontrado
Fin
```