

Eliminar los elementos decrecientes de una lista enlazada doble

En este ejercicio se trata de practicar las listas enlazadas dobles, circulares y con nodo fantasma (`double_linked_list_ed`). Queremos una operación que elimine de la lista enlazada los elementos cuyo valor sea estrictamente menor que el anterior de la lista (teniendo en cuenta las eliminaciones que ya hayan ocurrido). Una vez finalizado el proceso de eliminación la lista resultante debe estar ordenada en orden creciente.

Requisitos de implementación.

En la resolución del problema, se extenderá *mediante herencia* la clase `double_linked_list_ed` con un método que elimine, de una lista genérica, todos los elementos cuyo valor sea menor que el del elemento inmediatamente anterior.

Al ser una lista doble, se implementará un operación `mostrarReves` que recorra la lista de derecha a izquierda. Esta operación se utilizará exclusivamente para probar que la lista enlazada es correcta antes de enviar el problema al juez. En la salida que pide el juez la lista se recorre de izquierda a derecha y por lo tanto no se utiliza esta operación.

El coste de la operación debe ser lineal con respecto al número de elementos en la lista. No pueden hacerse nuevos `news`, ni copiar el contenido de los nodos.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso se muestra en dos líneas, la primera contiene el número de elementos de la lista (un número mayor que 0). En la segunda se muestran los elementos de la lista, números enteros. La entrada termina cuando el número de elementos sea 0.

Salida

Para cada caso de prueba se escribirá en una línea la lista después de eliminar los elementos.

Entrada de ejemplo

```
6
1 2 4 3 6 5
4
21 10 11 22
3
1 2 3
3
3 2 1
3
3 2 3
0
```

Salida de ejemplo

```
1 2 4 6
21 22
1 2 3
3
3 3
```

Autor: Isabel Pita