

Estremeciendo listas

El objetivo de este control es extender el TAD Lista implementado con una lista doblemente enlazada.

“Estremecer” una lista consiste en colocar todos los elementos que aparecen en posiciones pares, por orden de aparición, seguidos de todos los que aparecen en posiciones impares, por orden inverso de aparición (primero el último, después el penúltimo, etc.).

Por ejemplo, el resultado de “estremecer” la lista

0 1 2 3 4 5 6 7

es

0 2 4 6 7 5 3 1

(IMPORTANTE: consideramos que las posiciones comienzan en 0, es decir, la posición del primer elemento es la 0, la del segundo elemento es la 1, etc.)

Tu trabajo es añadir una nueva operación mutadora `void estremece();` a la implementación del TAD Lista basada en nodos doblemente enlazados, que “estremezca” la lista, y determinar justificadamente su complejidad.

Dicha operación no puede invocar, ni directa ni indirectamente, operaciones de manejo de memoria dinámica (`new`, `delete`) ni tampoco puede realizar asignaciones a los contenidos de los nodos (realizar copias de los valores de los elementos almacenados en la lista). Aparte de implementar esta operación, debes indicar la complejidad de la misma.

IMPORTANTE: Debes usar el archivo `lista.h` que se adjunta con este enunciado y no el visto en clase. En el archivo `lista.h` adjunto a este enunciado, a las operaciones del TAD vistas en clase se han añadido dos operaciones para imprimir la lista que son usadas por el archivo `main.cpp`.

Trabajo a realizar

Se proporciona el archivo `main.cpp` en el que se implementa toda la lógica de entrada / salida necesaria. NO MODIFIQUES NADA EN ÉL. Además, se proporciona un archivo `leeme.txt` que incluye información sobre el formato de la entrada / salida, así como un ejemplo de entrada junto con la salida producida.

En el archivo `lista.h` proporcionado tienes que completar la implementación de la operación pedida (el prototipo aparece al principio de la parte pública, ¡búscalo!) e indicar la complejidad de la misma.