

Retos con Listas Enlazadas

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
Algoritmos y Estructuras de Datos I (CE 1103)
II Semestre 2024

OBJETIVOS

GENERAL

- Resolver problemas conocidos sobre listas enlazadas

ESPECÍFICOS

- Familiarizarse con problemas conocidos de listas enlazadas en C#
- Implementar pruebas unitarias mediante MSTest
- Investigar y utilizar excepciones

REQUERIMIENTOS

Primeramente, deberá investigar qué son pruebas unitarias y como se utiliza MSTest para implementarlas en C# .NET. Una vez teniendo esto claro, deberá resolver los siguientes tres problemas de listas enlazadas y construir pruebas unitarias con los casos de prueba para demostrar que la solución es correcta.

Definición de la interface IList:

- InsertInOrder(int value): void
- DeleteFirst(): int
- DeleteLast(): int
- DeleteValue(int value): bool
- GetMiddle(): int
- MergeSorted(IList listA, IList listB, SortDirection direction): void

PROBLEMA #1: MEZCLAR EN ORDEN

Dadas dos listas doblemente enlazadas en orden (siempre ascendente), el método *MergeSorted(listA, listB, direction)* mezcla listB en listA en orden descendente o

ascendente según lo indicado por el argumento direction (deberá crear un Enum llamado SortDirection).

Por ejemplo:

ListA	ListB	SortType	Resultado (en ListA)
Null	Cualquier valor	Cualquier valor	Exception
Cualquier valor	Null	Cualquier valor	Exception
0,2,6,10,25	3,7,11,40,50	Asc	0,2,3,6,7,10,11,25,40,50
10,15	9,40,50	Desc	50,40,15,10,9
Vacia	9,40,50	Desc	50,40,9
10,15	Vacia	Asc	10,15

PROBLEMA #2: INVERTIR LISTA

Dada una lista doble sin ningún orden en particular, el método Invert(list) modifica list retornando la inversión de dicha lista. **No se permite crear una nueva lista.**

Por ejemplo:

List	Resultado (en List)
Null	Exception
Vacia	Vacia
1,0,30,50,2	2,50,30,0,1
2	2

PROBLEMA #3: OBTENER EL ELEMENTO CENTRAL

Construya un método InsertInOrder (int value) que recibe un entero y lo inserta en el orden ascendente que le corresponde en la lista. Asimismo, deberá crear un método llamado GetMiddle que retorna el elemento en la posición de la lista sin tener que recorrer la lista para encontrarlo, es decir, obtener el elemento central siempre conlleva un solo acceso a memoria. Esto significa que su implementación de lista, deberá llevar el rastro de cual el elemento central al insertar o al eliminar.

List	Middle
Null	Exception
Vacia	Exception
1	1
1,2	2

0,1,2	1
0,1,2,3	2

ASPECTOS OPERATIVOS

- El trabajo se realizará de forma **individual utilizando .NET 8 y C# 12**
- El uso de Git y Github es obligatorio
- La fecha de entrega será según lo especificado en el TEC Digital. Se entrega en el TEC digital, un archivo .txt dentro del cual hay un URL del repositorio donde se encuentra el código.
- Para los tres problemas, puede utilizar una sola clase llamada ListaDoble que implemente la interface Lista definida por usted.