Universidad Nacional Autónoma de México
Facultad de Ciencias
Estructuras de Datos 2022-2
Práctica 1

Pedro Ulises Cervantes González
confundeme@ciencias.unam.mx

Yessica Janeth Pablo Martínez,
yessica_j_pablo@ciencias.unam.mx

Jorge Macías Gómez
jorgemacias@ciencias.unam.mx

3 de Marzo del 2022

El código debe incluir comentarios y debe ser legible. Sigue las buenas prácticas al programar.
Únicamente envía los archivos con terminación .java.
No se reciben entregas fuera de Classroom.
El día limite es el jueves 10 de Marzo a la hora indicada en Classroom.

Actividades

1. Completa los métodos faltantes de la clase Lista:

   a) Reverse ->Invierte el orden de nuestra Lista.
   por ejemplo si tengo $1- > 2- > 4$ al invertirla tendré $4- > 2- > 1$
   Este método cambiara la lista original
   Tiempo: O(n)
   Espacio: O(1)

   b) toString .- Genera una representación en cadena de nuestra lista.

   c) Append .- Dado un ejemplar de nuestra clase lista como parámetro, lo anexaremos a nuestra lista.

   d) IndexOf .- Regresa un entero con la posición del elemento.
   Solo nos importara la primera aparición del elemento.
   Empieza a contar desde 0.
   Por ejemplo si tengo la lista $1- > 2- > 4- > 2$ y quiero saber la posición del 2. indexOf regresara 1.

   e) Insert .- Inserta un elemento en un índice explícito.
   Si el índice es menor que cero, el elemento se agrega al inicio de la lista.
   Si el índice es mayor o igual que el número de elementos en la lista, el elemento se agrega al fina de la misma.
   En otro caso, después de mandar llamar el método, el elemento tendrá el índice que se especifica en la lista.
   NO OLVIDES ACTUALIZAR EL CONTADOR DE LONGITUD.

   f) MezclaAlternada .- Dados 2 ejemplares de nuestra clase Lista A y B, necesitamos unirlas de manera alternada, es decir, a un elemento de A le seguirá un elemento de la lista B y viceversa.
   Por ejemplo, si tengo $1- > 2- > 4$ y quiero mezclar la lista $3- > 2- > 6$ mi lista final deberá ser $1- > 3- > 2- > 2- > 4- > 6$
   Este método cambiara la lista original
   Tiempo: $O(n + m)$.Donde n es el tamaño de la primer lista y m el tamaño de la segunda lista.

Espacio: O(1). Es decir, no ocupar más espacio (Por ejemplo, no crear más nodos.)

2. Completa los métodos de la clase practica1.java:

   a) AgregaOrdenado.- Dado un ejemplar de nuestra clase Lista ordenado necesitamos agregarle un elemento de manera ordenada. Es decir, si tengo $1- > 2- > 4$ y quiero agregar al 3 mi lista final deberá ser $1- > 2- > 3- > 4$
   Tiempo: O(n).
   Espacio: O(1). Es decir, no ocupar más espacio (Por ejemplo, no crear más listas.)

   b) Unión .- Dados dos ejemplares de nuestra clase Lista queremos obtener la unión de estos. Es decir, si tengo $1- > 2- > 4$ y quiero aplicarle unión a la lista $5- > 3- > 2$ mi lista final deberá ser $1- > 2- > 5- > 4- > 3$
   no hay duplicados, y no importara el orden.
   Tiempo: O(n * m). Donde n es el tamaño de la primer lista y m el tamaño de la segunda lista.
   Espacio: $O(n + m)$.

   c) Cómo mejorarías el tiempo de Unión escribe tu respuesta en el comentario del método.

   d) Intersección .- Dados dos ejemplares de nuestra clase Lista queremos obtener la intersección de estos. Es decir, si tengo $1- > 2- > 4$ y quiero aplicarle intersección a la lista $5- > 4- > 2$ mi lista final deberá ser $4- > 2$
   no hay duplicados, y no importara el orden.
   Tiempo: O(n * m). Donde n es el tamaño de la primer lista y m el tamaño de la segunda lista.
   Espacio: O(n). siendo n >m de otra forma sera O(m)

   e) Cómo mejorarías el tiempo de Intersección escribe tu respuesta en el comentario del método.

   NO puedes agregar más métodos públicos a la clase Lista.
   En el comentario de cada función debes explicar porque tú método cumple con las restricciones de complejidad.

¡Importante!
La intención de la tarea es que desarrolles tus habilidad de programación. Si tienes dudas de como escribir tus ideas en código no dudes en mandar un mensaje de Telegram al ayudante de laboratorio.
@Toaultor