UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERIA INDUSTRIALA Y DE SISTEMAS
Área Académica de Sistemas y Telemática

CICLO: 2026-I
FECHA:25/03/2064

CURSO: ALGORITMIA Y ESTRUCTURA DE DATOS AVANZADA (SW-305)

PRIMERA PRACTICA CALIFICADA

NOTA:
1. Responder cualquier pregunta de forma continua y completa. No se calificará trozos de respuestas en diferentes partes.
2. No hacer borrones en el limpio. Se descontará por borrones. Use su hoja de borrador o liquid.
3. Sin copias, apuntes. Solo use lapicero de tinta azul o negro.
4. Sírvase facilitar su identificación presentando su carné.

1. Sergei ha diseñado una nueva línea de juguetes anidados: "babushkas generalizadas" (en adelante *BG*), inspirados en las tradicionales babushkas rusas (una babushkas es una muñeca que puede abrirse en dos mitades, de modo que se encuentra otra muñeca dentro, la misma que puede tener otra muñeca dentro y así seguir varias veces, hasta llegar a una muñeca final que no puede abrirse). En estas BG, cada juguete puede contener varios juguetes (no sólo 1) más pequeños dentro de él.
Vladimir ha desarrollado una notación para describir cómo deben construirse los juguetes anidados. Un juguete se representa con un número entero positivo, según su tamaño. Más concretamente: **si al abrir el juguete representado por m encontramos los juguetes representados por $n_1, n_2, \dots, n_r$, debe ser cierto que $n_1, n_2, \dots, n_r < m$ y además $\sum_{i=1}^{r} n_i < m$.** Y si esto es así, decimos que el juguete m contiene directamente a los juguetes $n_1, n_2, \dots, n_r$ (ojo: cualquier juguete que pueda estar contenidos en cualquiera de los juguetes $n_1, n_2, \dots, n_r$ no se consideran directamente contenido en el juguete m),.
Una babushka generalizada (*BG*) se denota con una secuencia no vacía de enteros no nulos de la forma:

$$a_1, a_2, \dots, a_N$$

tal que el juguete k se representa en la secuencia con dos enteros $-k$ y k, ubicándose el negativo antes que el positivo.
Por ejemplo, la secuencia:

-9 -7 -2 2 -3 -2 -1 1 2 3 7 9

representa una BG formada por seis juguetes, a saber, 1, 2 (dos veces), 3, 7 y 9. Obsérvese que el juguete 7 contiene directamente los juguetes 2 y 3. Obsérvese que la primera copia del juguete 2 se produce a la izquierda de la segunda y que la segunda copia contiene directamente al juguete 1. Sería erróneo entender que el primer -2 y el último 2 deberían estar emparejados.
Por otro lado, las siguientes secuencias no describen babushkas generalizadas:

-9 -7 -2 2 -3 -1 -2 2 1 3 7 9

porque el juguete 2 es mayor que el juguete 1 y no puede asignarse dentro de él.

-9 -7 -2 2 -3 -2 -1 1 2 3 7 -2 2 9

porque 7 y 2 no pueden asignarse juntos dentro de 9.
Tu tarea es escribir un programa que determine si una secuencia leída representa una babushkas generalizada válida.

Entrada
La primera línea constará de un entero "I" que indicará la cantidad de líneas a leer (I<20)
Cada caso de prueba es una línea de enteros no nulos de 3 cifras o menos.

Salida
Por cada línea, imprime ": -) babushkas!" si es válida o ":-(intenta de nuevo." si no lo es.

Ejemplo de Entrada
2
-9 -7 -2 2 -3 -2 -1 1 2 3 7 9
-9 -7 -2 2 -3 -1 -2 2 1 3 7 9

Ejemplo de Salida
:-) babushkas!
:-(intenta de nuevo.

(7 puntos)

2. Sea **DList** la clase que representa una lista doblemente enlazada e implementa las operaciones del tipo abstracto de datos Lista, estudiada en la asignatura. Esta clase sólo tiene como atributos la referencia al primer nodo (*_head*), al último nodo (*_tail*) y el tamaño de la lista (_size). Los nodos de **DList** son objetos de la clase **DNode**.
Sea **DList2** una subclase de **DList**. Dicha subclase debe proporcionar un método, **hollowMid(e),** que admite un parámetro **e** y que elimina los e elementos más cercanos al centro de la lista. El parámetro **e** debe ser siempre un número de la misma paridad que el tamaño de la lista (si el tamaño de la lista es par, **e** debe ser u número par, y si el tamaño es impar, **e** debe ser impar). Si esta condición no se cumple o **e** es mayor que el tamaño de la lista, el método no modifica la lista y muestra el mensaje **'e not valid'**. Este método debe tener la menor complejidad temporal que sea posible.
El método no devuelve nada, pero modifica la lista. El método no puede incluir llamadas a otros métodos de la clase DList o utilizar otras listas auxiliares (de ningún tipo). No puedes añadir nuevos atributos a las clases DNode y DList. Si necesitas más funciones para implementar **hollowMid(e),** estas deben ser implementadas en tu respuesta.
A continuacióse muestran algunos ejemplos de listas antes y después de invocar en ellas el método **hollowMid**(e):

LISTA ANTES	LLAMADA AL MÉTODO	LISTA DESPUÉS	MENSAJE
[1,2,3,4,5,6,7,8,9]	L.hollowMid(2)	[1,2,3,4,5,6,7,8,9]	'e not valid'
[1,2,3,4,5,6,7,8,9]	L.hollowMid(3)	[1,2,3,7,8,9]	
[1,2,3,4,5,6,7,8,9]	L.hollowMid(7)	[1,9]	
[1,2,3,4,5,6,7,8]	L.hollowMid(2)	[1,2,3,6,7,8]	
[1,2,3,4,5,6,7,8]	L.hollowMid(1)	[1,2,3,4,5,6,7,8]	'e not valid'
[1,2,3,4,5,6,7,8]	L.hollowMid(10)	[1,2,3,4,5,6,7,8]	'e not valid'

- ¿Cuál es la complejidad temporal de tu solución? Razona la respuesta y analiza también sobre el mejor y peor caso.

Para que la solución se considere correcta, la función propuesta debe ser correcta (resuelva el problema), robusta (no tenga errores y funcione para cualquier entrada) y eficiente en términos de complejidad temporal y espacial (evitar el uso de estructuras auxiliares como otras listas). Además, el código debe ser fácil de entender y mantener, y que esté refactorizado. **No está permitido el uso estructuras de la Clase Collection (java.util.Collections), deben implementar los TADs ustedes mismos.**

(6 puntos)

3. Utilizando la definición formal de notación Big O. demuestre la verdad o falsedad de las siguientes afirmaciones:

a) $f(n) = \frac{n^3}{\log_2 n}$ es $\mathcal{O}(n^3)$ para $n \geq 2$.

b) $f(n) = n^{\log_2 3}$ es $\mathcal{O}(n^2)$.

De las siguientes afirmaciones, indicar cuáles son ciertas y cuáles no. Justifique usando la definición formal, el criterio del límite o o propiedades de logaritmos/exponentes (polinomio característico), según corresponda.

a) $(\sqrt{n})^{\log_2 n} \in \mathcal{O}(n^2)$

b) Determinar si $(\ln n)^3 \in \mathcal{O}(n^{1/3})$

(3 puntos)

5. **Resolver la siguiente ecuación de recurrencia de forma exacta usando el método del polinomio característico. Determine la complejidad asintótica.**

$$T(n) = 3T(n-1) - 3T(n-2) + T(n-3), \quad \text{para } n \geq 3$$

con las condiciones iniciales: $T(0) = 1, T(1) = 2, T(2) = 4$. T(3) = 7

(2 puntos)