



Estructuras de Datos

ESTRUCTURAS NO LINEALES
ACTIVIDAD A-3.2: ÁRBOL ESPEJO

René Ornelis
Vacaciones de junio de 2024

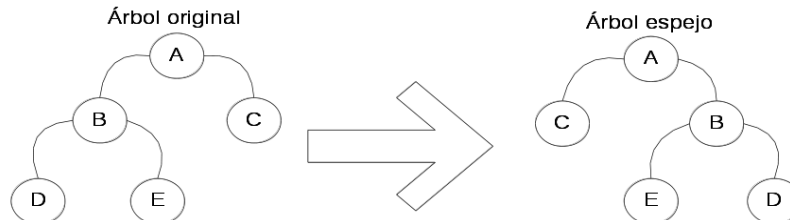
Árbol espejo

1 Objetivos

Los objetivos de esta actividad son que el estudiante sea capaz de resolver los problemas con los recorridos en un árbol binario.

2 Problema

Dada una clase `ArbolBinario` que no es un árbol de búsqueda y desconocemos cómo trabajan su método de inserción ni sus efectos en la posición de los nodos en el árbol y por lo tanto no se pueden utilizar; escriba método **espejo** (y cualquier método que utilice) que genere un **nuevo** árbol que sea una imagen espejo de dicho `ArbolBinario`. Es decir, la secuencia en orden del árbol espejo es la inversa de la secuencia en orden del árbol original. No debe cambiar ni los nodos, ni el árbol original. Este método debe ser de $O(n) = n$. Ejemplo:



Suponga que sólo existen las siguientes variables de cada clase:

<pre>template <class T> class ArbolBinario { public: ArbolBinario<T> *espejo(); private: NodoBinario<T> *raíz; }</pre>	<pre>template <class T> class NodoBinario { T valor; NodoBinario<T> *izq; NodoBinario<T> *der; }</pre>
---	--

Restricciones:

- No se permite agregar variables a las clases
- Cualquier método que invoque en su solución, debe desarrollarlo.
- No se permite cambiar la interfaz de los métodos público.
- No se permite agregar métodos públicos.
- No se permite cambiar la visibilidad de los miembros de la clase.

3 Tiempo de entrega

La entrega se debe realizar en la plataforma de la Facultad o, en caso de que esta no esté disponible, por correo electrónico, a más tardar el 18/junio a las 09:00. No se permitirá entregas posteriores al límite definido.

4 Entregables

Deberá entregar archivo ZIP con:

1. Archivo .h y .cpp compilables con la definición de clases y la implementación de método solicitado.
2. Archivo PDF con demostración formal de $O(n)$ de la solución presentada

5 Criterios de evaluación

Objetivo	Puntos	Detalle por evaluar
Eficacia de la solución	1.5	La solución propuesta funciona y crea el nuevo árbol
Integridad	1.5	La solución no modifica el árbol original ni utiliza ninguna estructura de datos adicional.
Determinar el orden de la solución	1	Debe demostrar formalmente, el orden del algoritmo equivalente $O(n)$
TOTAL	4	