

## Arboles Mutantes

Vamos a definir una operación en los arboles binarios que llamaremos “mutación”. Esta operación consiste en tomar un nodo del árbol e intercambiar los hijos, el hijo izquierdo pasa a ser hijo derecho y el hijo derecho pasa a ser hijo izquierdo. En la figura 1 y 2 se muestra un ejemplo de la aplicación de la mutación en el nodo de color gris.

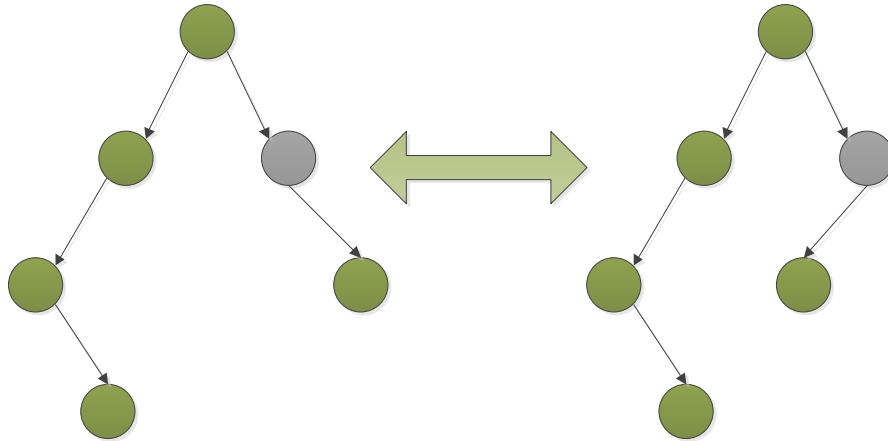


Figura 1 Mutación del árbol aplicada al nodo gris.

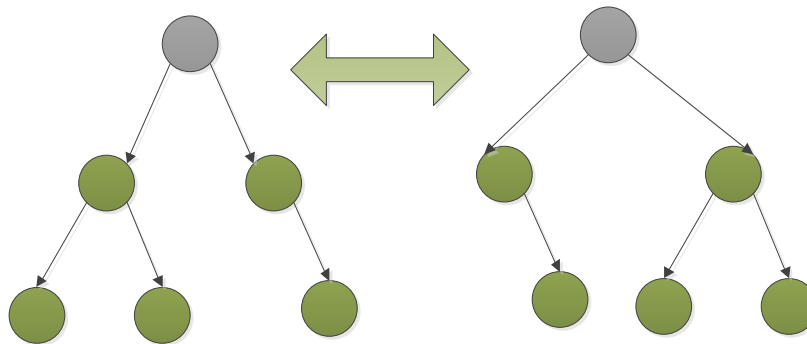


Figura2 Mutación del árbol aplicada al nodo gris.

Ahora si definimos la igualdad para los arboles binarios A y B de la siguiente manera:

El árbol A es igual al árbol B cuando partiendo de las raíces de los dos árboles son iguales recursivamente los hijos izquierdo y derecho.

Como se puede notar en las figuras 1 y 2 si se aplica la mutación al árbol de la izquierda en el nodo gris se obtiene el árbol de la derecha y si se aplica sobre al árbol de la derecha se obtiene el de la izquierda.

Si se intenta aplicar la operación a un nodo que sea hoja (no posee ningún hijo) no ocurre nada y esto **NO** se consideraría una mutación

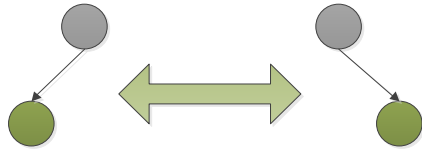


Figura 3 Mutación aplicada al nodo de color gris con un solo hijo.

Se quiere resolver el problema de dados dos árboles A y B ¿cuál es la menor cantidad de mutaciones que hay que aplicar para a partir del árbol A llegar al árbol B?

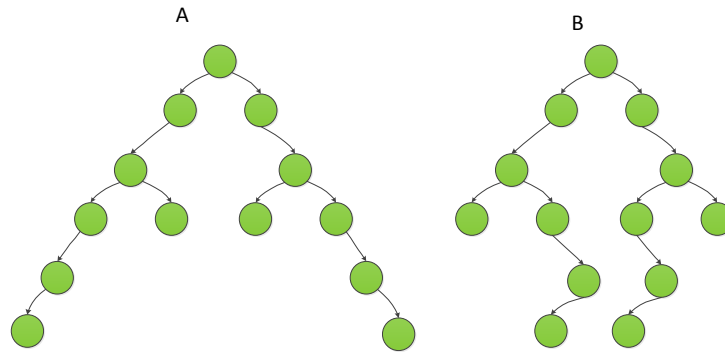


Figura 5 Árboles binarios A y B

Dado el ejemplo de la Figura 4, se puede llegar del árbol A al árbol B con un mínimo de cuatro mutaciones tal y como se muestra en la Figura 5. En la figura están coloreados de amarillo los nodos que se seleccionan para aplicar la mutación para obtener el próximo árbol.

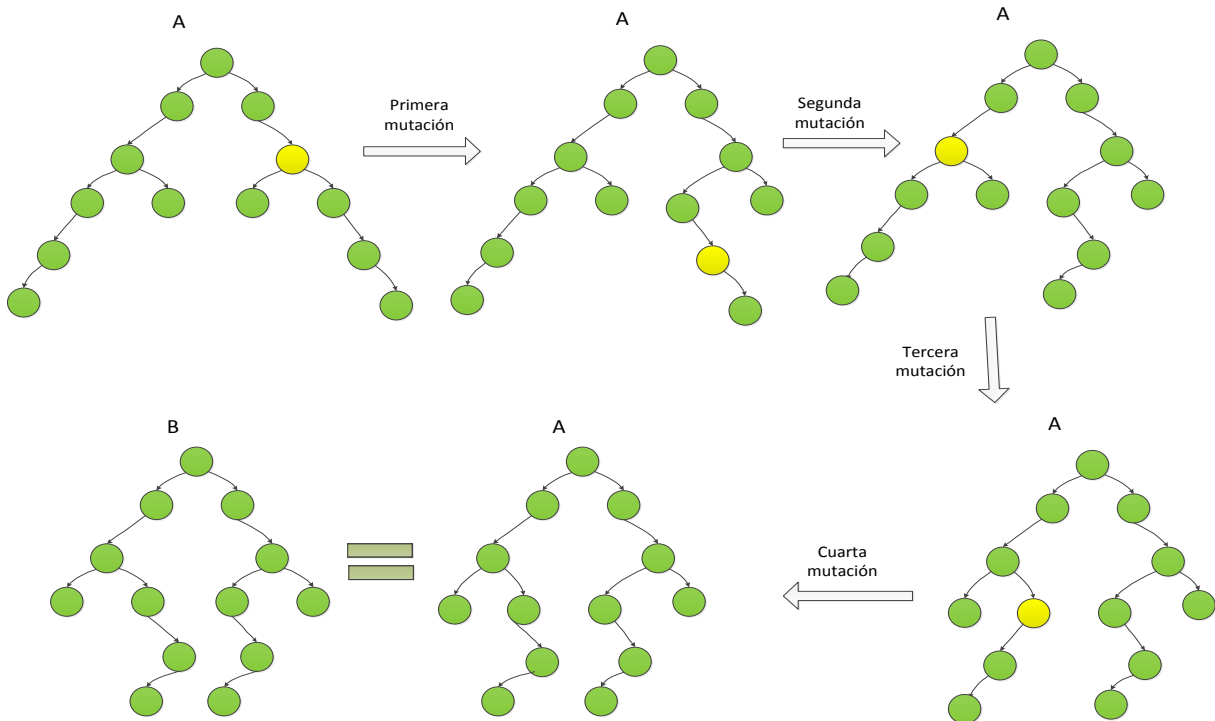


Figura 5 Mediante cuatro mutaciones se transforma el árbol A en el árbol B.

Se debe de notar que se pudieran hacer muchas otras mutaciones para transformar el árbol A en el B, pero la cantidad mínima para convertir el árbol A en el árbol B son cuatro.

Se debe de recalcar que en este problema los valores de los nodos del árbol son irrelevantes.

Siendo la clase `ArbolBinario` del Listado 1 que se encuentra en el ensamblado `Weboo.ArbolesMutables`.

```
/// Clase que representa un Árbol Binario.
public class ArbolBinario
{
    /// Hijo izquierdo del Árbol
    public ArbolBinario Izquierdo { get{...} set{...} }
    /// Hijo derecho del Arbol
    public ArbolBinario Derecho { get{...} set{...} }
    /// Propiedad que permite saber si es hoja o no del Árbol.
    /// Si no tiene ni hijo izquierdo ni derecho es una hoja.
    public bool EsHoja
    {
        get{return Izquierdo == null && Derecho == null; }
    }
    /// Constructor por defecto.
    public ArbolBinario()
    {
        Izquierdo = null;
        Derecho = null;
    }
    /// Constructor que tiene como parámetros los hijos izquierdo
    /// y derecho.
    public ArbolBinario(ArbolBinario izquierdo, ArbolBinario derecho)
    {
        Izquierdo = izquierdo;
        Derecho = derecho;
    }
}
```

Listado 1 Código de la clase `ArbolBinario`

Se debe de implementar el método: MinimaMutacion

```
static int MinimaMutacion(ArbolBinario A, ArbolBinario B)
{
    // Implementación de su algoritmo.....
}
```

que retorne la menor cantidad de mutaciones que se deben de aplicar para convertir el árbol binario del parámetro A al árbol binario del parámetro B. Para ello haga uso de las plantillas que se le brinda.

## Nota

- Considere que a los árboles que se pasan como valores a los parámetros siempre se pueden transformar uno en el otro aplicando mutaciones salvo que se pase un valor `null`.
- Si alguno de los parámetros se le pasa `null` se debe de lanzar una excepción `ArgumentException`.
- Debe de usar las plantillas que se adjuntan con este documento para implementar su solución.