Universidad del Istmo de Guatemala
Facultad de Ingenieria
Ing. en Sistemas
Informatica 2
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

Hoja de trabajo #7

Fecha de entrega: 12 de Abril, 2018 - 11:59pm

Instrucciones: Realizar cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Hoja de trabajo 7". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex. Los ejercicios de programación deben ser colocados en una carpeta llamada "Programas", la cual debe colocarse dentro de la carpeta correspondiente a esta hoja de trabajo.

Iniciación

Crear una solución llamada *BinaryTree*. Dentro de esta solución crear:

- un proyecto llamado *BinaryTree* de tipo console
- un proyecto llamado *BinaryTreeTests* de tipo xunit

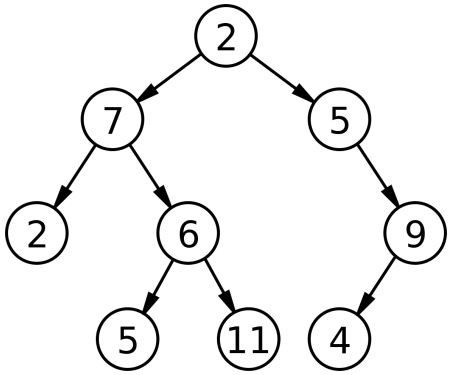
Arbol binario mutable (10%)

Agregar la clase e interfaz *IBinTree* y *BinaryTree* que se encuentra en los ejemplos de esta carpeta "Semana 10" al proyecto *BinaryTree*.

Modificar la interfaz *IBinTree* (y la clase *BinaryTree*) de tal forma que su sus propiedades *Derecho*, *Izquierdo* y *Valor* sean modificables utilizando la palabra reservada `set`.

Metodo *ToArray* (10%)

Agregar a la interfaz *IBinTree* (y a la clase *BinaryTree*) un metodo llamado *ToArray* de tipo: `ToArray` : `void` → `int`[ ]. Este metodo debe recorrer todos los nodos del arbol (recursivamente) y construir un arreglo de tal forma que primero se agregan todos los numeros que se encuentran en el lado *Izquierdo* del nodo, luego el numero de su propiedad *Valor* y finalmente todos los numeros en el lado derecho. Por ejemplo, si se tiene el siguiente arbol binario:



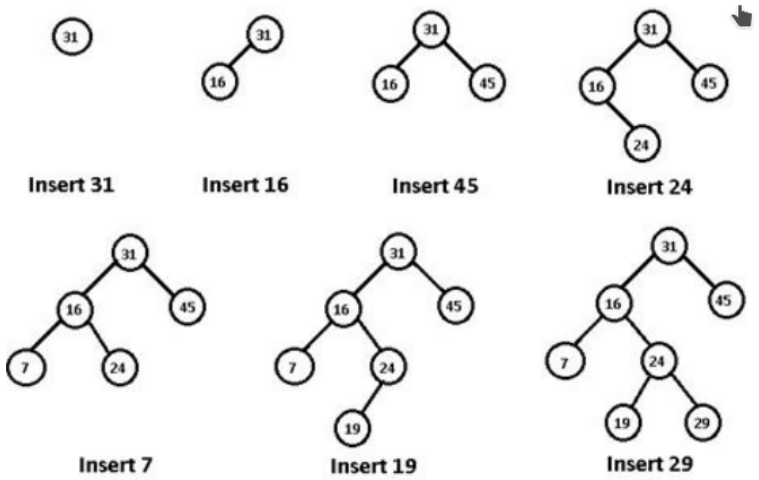
El arreglo retornado por *ToArray* seria: [2,7,5,6,11,2,5,4,9].

Metodo *Insert* (40%)

Agregar a la interfaz *IBinTree* (y a la clase *BinaryTree*) un metodo llamado *Insert* de tipo `Insert : int` $\rightarrow$ `void`. Este metodo debe agregar un numero llamado n al arbol binario de la segun las siguientes reglas:

- Si $n \leq$ `Valor`:
  - Si `Izquierdo` $\equiv$ `null` entonces crear un *BinaryTree* con el valor n y asignarlo a la propiedad *Izquierdo* del arbol.
  - De lo contrario, llamar recursivamente al metodo *Insert* de *Izquierdo* con el valor n.
- Si $n >$ `Valor`:
  - Si `Derecho` $\equiv$ `null` entonces crear un *BinaryTree* con el valor n y asignarlo a la propiedad *Derecho* del arbol.
  - De lo contrario, llamar recursivamente al metodo *Insert* de *Derecho* con el valor n

La siguiente imagen muestra como es el comportamiento de *Insert*:



Prueba unitaria para *Insert* (40%)

Crear una pureba unitaria para el metodo *Insert* y colocarla en el proyecto "BinaryTreeTests". Este debe verificar que *Insert* funciona correctamente con un conjunto de numeros de su elección. *Pista:* Puede

utilizar el método *ToArray* para verificar que el método funciona correctamente ya que el arreglo retornado por *ToArray* debe estar ordenado. En otras palabras, el valor en el indice i debe ser menor o igual al valor en $i + 1$.