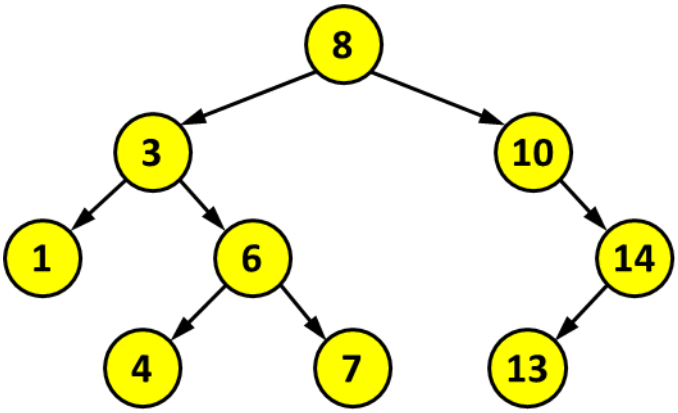
Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

15 junio 2017

IIC1103 – Introducción a la Programación – Sección 2 **Control 3**
(Tiempo: 30 minutos)

Se ha creado la clase **`Arbol`** para representar un **árbol binario de búsqueda** como el mostrado en la figura, en que cada nodo "hoja" contiene sólo un valor, y cada nodo "no hoja" contiene un valor, y un subárbol a su izquierda y/o un subárbol a su derecha. Además, se cumple que el subárbol izquierdo de cualquier nodo (si existe) contiene valores menores que el que contiene dicho nodo, y el subárbol derecho (si existe) contiene valores mayores.

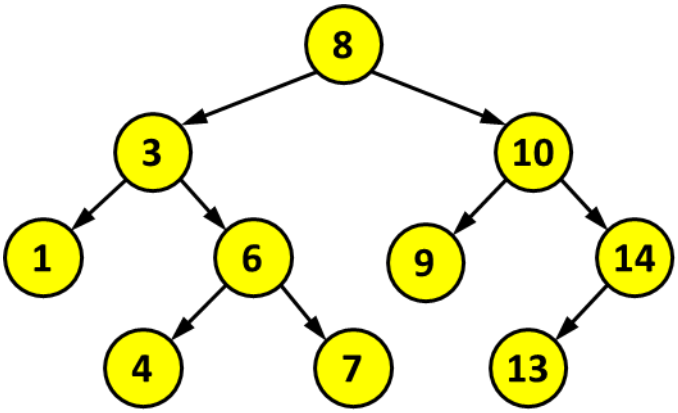


Se le pide implementar los métodos **recursivos** de la clase **`Arbol`**:

- **`buscar`**, que retorna **`True`** si un dato existe en el árbol, o **`False`**, en caso contrario, e
- **`insertar`**, que agrega un dato al árbol, manteniendo su orden.

Por ejemplo, la ejecución deseada del código siguiente es:

```
>>>
contenido del árbol luego de insertar: 9
 |- nodo: 8
  |- nodo: 3
   |- hoja: 1
   |- nodo: 6
    |- hoja: 4
    |- hoja: 7
  |- nodo: 10
   |- hoja: 9
   |- nodo: 14
    |- hoja: 13
    |- <>
Existe 10? True
Existe -1? False
Existe 11? False
Existe 99? False
Existe 9? True
```



Código a completar:

```
class Arbol:
  # constructor de clase Arbol
  def __init__(self, dato):
    self.esHoja = True    # es True si el nodo es una hoja
    self.valor = dato     # valor del nodo
    self.izquierda = None # árbol a la izquierda; objeto de clase Arbol
    self.derecha = None   # árbol de la derecha; objeto de clase Arbol

  # retorna True si un dato existe en el árbol, o False, si no
  def buscar(self, dato):
    # completar

  # agrega un dato al árbol, manteniendo su orden
  def insertar(self, dato):
    # completar

arbol_1 = Arbol(8)
for valor in [3, 10, 1, 6, 14, 4, 7, 13]:
    arbol_1.insertar(valor)

arbol_1.insertar(9)
print("contenido del árbol luego de insertar:", 9)
arbol_1.mostrar()

print("Existe 10?", arbol_1.buscar(10))
print("Existe -1?", arbol_1.buscar(-1))
print("Existe 11?", arbol_1.buscar(11))
print("Existe 99?", arbol_1.buscar(99))
print("Existe 9?", arbol_1.buscar(9))
```

Ayuda: para crear un árbol que es hoja y contiene un valor, por ejemplo 33, utilice **`Arbol(33)`**.

Nota: no se permiten preguntas, así es que haga los supuestos que estime conveniente, indicándolos en su hoja de respuesta.