class Nodo:

def \_\_init\_\_(self, nombre):

self.nombre = nombre

self.izquierda = None

self.derecha = None

class Arbol:

def \_\_init\_\_(self):

self.raiz = None

def vacio(self):

return self.raiz == None

def buscarNodo(self, nombre):

return self.\_buscarNodo(nombre, self.raiz)

def \_buscarNodo(self, nombre, nodo\_actual):

if nodo\_actual == None:

return None

elif nodo\_actual.nombre == nombre:

return nodo\_actual

elif nombre < nodo\_actual.nombre:

return self.\_buscarNodo(nombre, nodo\_actual.izquierda)

else:

return self.\_buscarNodo(nombre, nodo\_actual.derecha)

def Insertar(self, nombre):

if self.raiz == None:

self.raiz = Nodo(nombre)

else:

self.\_Insertar(nombre, self.raiz)

def \_Insertar(self, nombre, nodo\_actual):

if nombre < nodo\_actual.nombre:

if nodo\_actual.izquierda == None:

nodo\_actual.izquierda = Nodo(nombre)

else:

self.\_Insertar(nombre, nodo\_actual.izquierda)

else:

if nodo\_actual.derecha == None:

nodo\_actual.derecha = Nodo(nombre)

else:

self.\_Insertar(nombre, nodo\_actual.derecha)

def ImprimirArbol(self):

self.\_ImprimirArbol(self.raiz)

def \_ImprimirArbol(self, nodo\_actual):

if nodo\_actual != None:

self.\_ImprimirArbol(nodo\_actual.izquierda)

print(nodo\_actual.nombre)

self.\_ImprimirArbol(nodo\_actual.derecha)