CP 4: Árbol binario de búsqueda

1. Se tiene el siguiente algoritmo que devuelve una lista con los elementos de un árbol recorridos en entre-orden.

```
List InOrder(TreeNode t)
{
   if (t is empty)
      return {};
   return InOrder(t.LeftChild) + { t.Key } + InOrder(t.RightChild)
}
```

   a. Demuestre que si **L** = **InOrder(t)**, entonces L contiene todos los elementos de t ordenados de menor a mayor.

2. Implemente un algoritmo que recorra un árbol en entre-orden y que no haga uso de la recursividad (ni la simule mediante la utilización de una pila). ¿Cuál es el tiempo de ejecución del algoritmo? ¿Por qué?

3. Demuestre que si en un árbol binario de búsqueda un nodo tiene dos hijos entonces su sucesor y su antecesor son descendientes de dicho nodo.
   a. Demuestre que además su sucesor no tiene hijo izquierdo y su antecesor no tiene hijo derecho.

4. Sea T un árbol binario cuyas llaves son todas distintas. Sea x un nodo hoja y sea y su padre. Demuestre que $y.Key$ es o el menor elemento de T que es mayor que $x.Key$ o el mayor elemento de T que es menor que $x.Key$.

5. Se puede ordenar un conjunto de n números insertándolos primero en un árbol binario de búsqueda y después recorriendo este árbol en entre-orden. ¿Cuáles son los casos peor y mejor para este algoritmo de ordenación? ¿En qué orden funcionaría el algoritmo para esos casos?

6. ¿Es conmutativa la operación de eliminación en un árbol binario de búsqueda? Decimos que la eliminación es conmutativa si al eliminar de un árbol binario x primero y luego y se obtiene el mismo árbol que al eliminar y primero y luego x. Justifique su respuesta.

7. Suponga que tiene un árbol T en el que en cada nodo x, en lugar de tener el atributo x.Parent que apunta al padre de x se tiene un atributo x.Successor, apuntando al sucesor de x en T. Implemente las operaciones SEARCH, INSERT y DELETE en un árbol binario con dicha representación de manera que el tiempo de ejecución de las mismas siga siendo $O(h)$, donde h es la altura del árbol. (HINT: Implemente un método que devuelve el padre de un nodo)