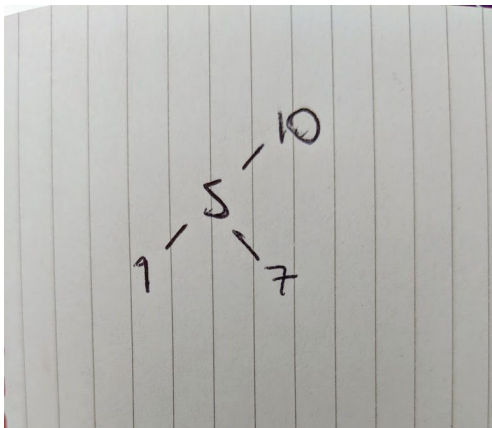


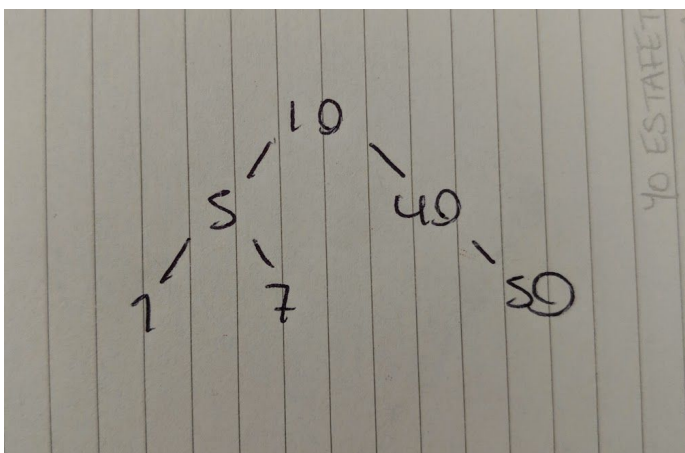
**PreOrder:** 10 5 1 7 40 50

La secuencia de un Preorder traversal es: Root, Left, Right.

Dado esto podemos observar que el 10 es la raíz del árbol. Después, la secuencia imprime todos los nodos a la izquierda de este. La secuencia imprime 5 y repite la secuencia de PreOrder (root, left, right). Por lo cual Imprime 1, este siendo el hijo izq de 5 ya que no sale del rango  $[-\infty, 10]$  y después 7, este siendo el hijo derecho de 5 que está en el rango  $[5, 10]$ .



Después, podemos ver que el siguiente número es 40. El 40 es más grande que el 7, pero también es más grande que la raíz, esto quiere decir que es el comienzo de la secuencia de los hijos derechos de la raíz. El nodo 40 vuelve a llamar a PreOrder, por lo cual imprime el hijo izq y derecho recursivamente. Como podemos ver, el 50 es el siguiente hijo, pero debido a que no está en el rango  $[10, 40]$  significa que es el hijo derecho de 40.



En este caso, sólo puede haber un sólo BST debido a que se establece la raíz desde el principio y con esta los rangos para los hijos izq  $(-\infty, 10)$  y derechos  $(10, \infty)$ .

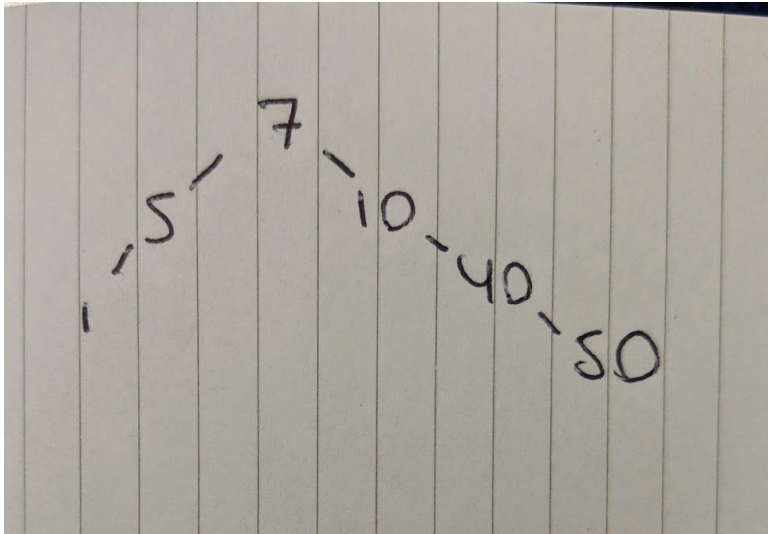
Código para PreOrder:

```
public void preOrder(Node node){  
  
    if(node != null){  
        visit(node);  
        preOrder(node.left);  
        preOrder(node.right);  
    }  
  
}
```

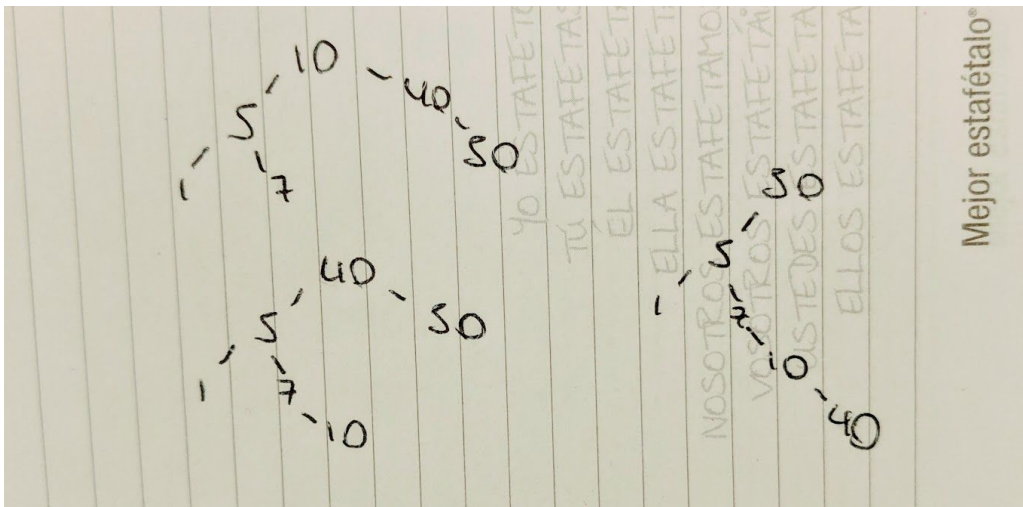
**InOrder:** 5 1 7 10 40 50

La secuencia de un Inorder traversal es: Left, Root, Right.

Esta secuencia se puede dar a la creación de diferentes BST. Primero, tenemos nuestro caso base que es 5, este siendo el hijo izq de la raíz. Luego tenemos el 1, este es menor que el 5 entonces sólo puede ser el hijo izq de 5. El nodo 7 es mayor que 5, esto nos da 2 posibilidades: o es el hijo derecho de 5 o es la raíz. Para este ejemplo, digamos que es la raíz. Esto nos lleva a el nodo 10, el cual es mayor que 7, por lo tanto es hijo derecho. De aquí sólo queda ver que 40 y 50 no caben en el rango  $[7, 10]$  y por lo tanto son hijos derechos. Este árbol quedaría algo así:



Sin embargo, podemos ir al caso en donde 7 es hijo derecho de 5. Esto da las posibilidades a todos los números siguientes de que puedan ser o hijos derechos o raíces. Esas posibilidades se expresarían así:



Código para InOrder:

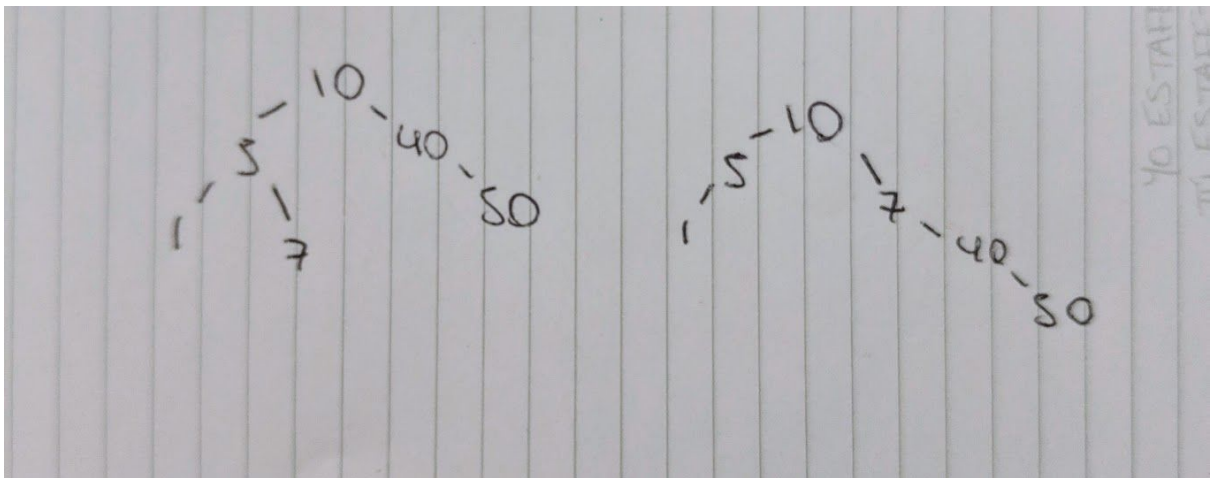
```
public void inOrder(Node node){  
  
    if(node != null){  
        inOrder(node.left);  
        visit(node);  
        inOrder(node.right);  
    }  
}
```

```
}
```

### PostOrder: 5 1 7 40 50 10

La secuencia de un Postorder traversal es: Left, Right, Root.

Para esta secuencia comenzamos con el caso base 5, siendo el hijo izq de la raíz. El siguiente nodo es 1, el cual es menor que 5, por lo cual es hijo izq de 5. El nodo 7 es mayor a 5, por lo cual tiene dos posibilidades, o es hijo derecho de 5 o es el nodo derecho de la raíz. De aquí se pueden crear 2 BST, los cuales son los siguientes.



Igual se pueden tomar los siguientes números de la secuencia y tener las mismas posibilidades, sin embargo, todo depende de el último número de la secuencia, este siendo la raíz.

Código para PostOrder

```
public void postOrder(Node node){

    if(node != null){
        postOrder(node.left);
        postOrder(node.right);
        visit(node);
    }

}
```