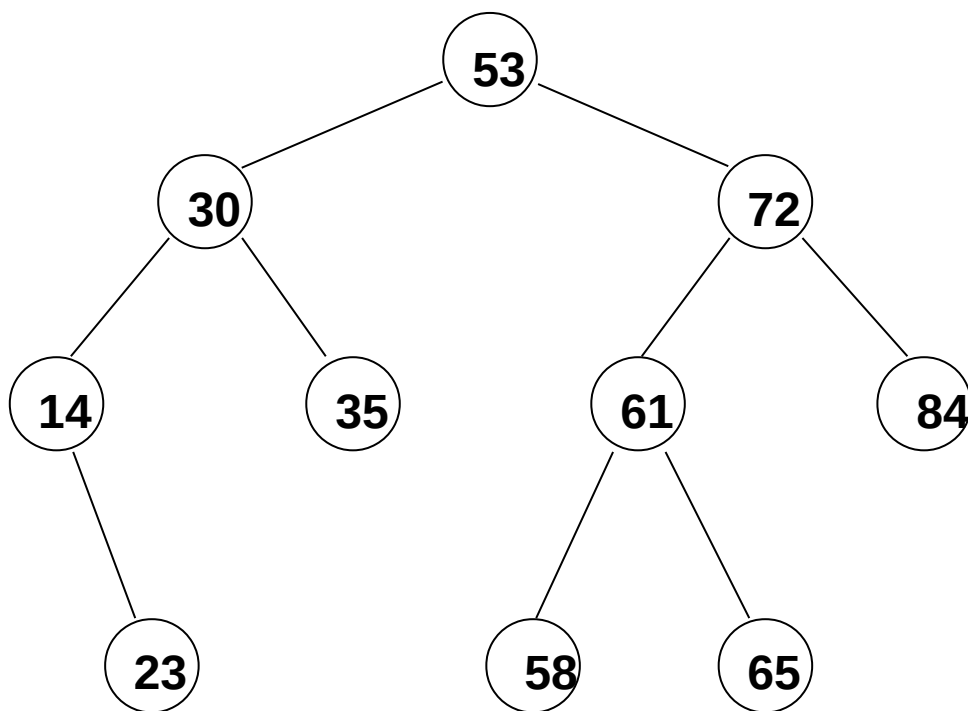


EJERCICIO 3.- Añadir a ABB e imprimir valores en un intervalo dado (1,5 puntos)

Tenemos un **árbol binario de búsqueda**, en el que cada nodo, además de un valor, tiene un campo que indica el número de nodos del subárbol enraizado en el nodo (ver clase `BinaryTreeNode` abajo).

Queremos obtener dos algoritmos:

- `add(v)` añade el elemento `v` al árbol, actualizando también el campo `numberOfNodes`
- `printIntervalo(a, b)` imprimirá los valores del árbol entre `a` y `b` (inclusive)



```
public class BinaryTreeNode<T> {
    T data;
    BinaryTreeNode<T> left, right;
    int numberOfNodes;
}

public class ABB {
    BinaryTreeNode<Integer> root;

    public void add(int v)
        // Precondición: v no está en este árbol.
        // Descripción: Se añade v al árbol.
        // Importante: se actualizará el campo numberOfNodes de todos los nodos afectados
        // por el cambio.

    public void printIntervalo(int a, int b)
        // Descripción: Imprime en la salida estándar los elementos del conjunto
        // mayores o iguales que a y menores o iguales que b, en orden creciente.
        // Importante: implementar recursivamente, evitando procesar nodos innecesariamente.
}
```

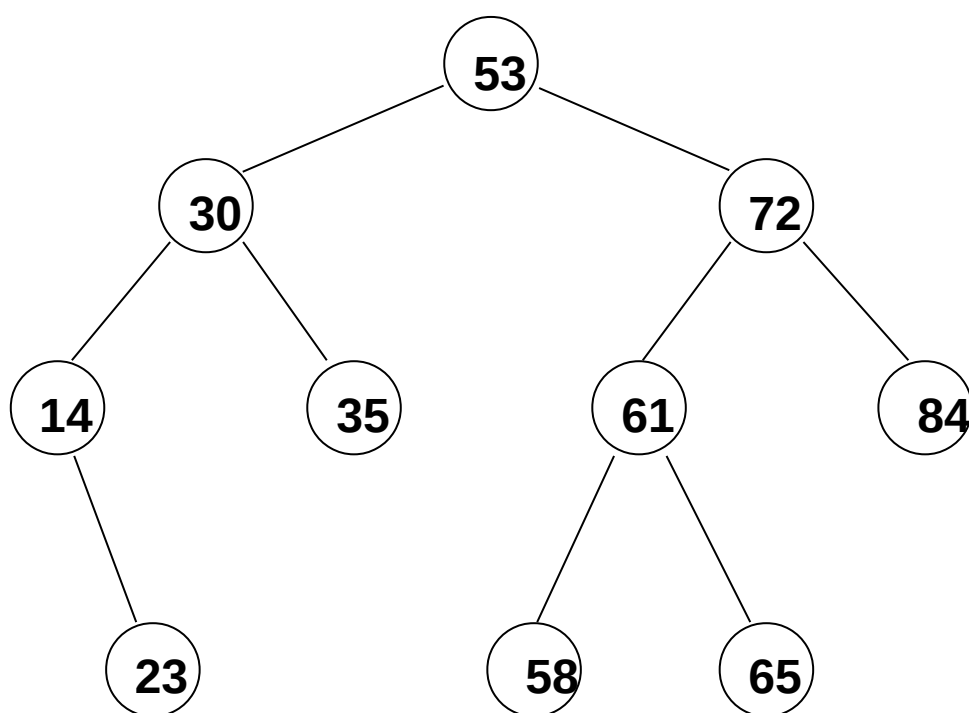
- Implementar los métodos `add(int v)` y `printIntervalo(int a, int b)`
- Calcular el coste de los algoritmos, **de manera razonada**

ARIKETA 3.- BZB batean balioa gehitu eta tarteko balioak idatzia (1,5 puntu)

Bilaketa-zuhaitz bitar bat dugu, adabegiko balioaz gain, adabegi bakoitzak atributu gehigarri bat du azpizuhaitzak duen nodo kopurua adierazten duena (ikus BinaryTreeNode azpian).

Bi algoritmo hauek lortu nahi ditugu:

- add(v) metodoak v elementua zuhaitzean gehituko du, *numberOfNodes* eremua eguneratuz
- printTartea(a, b) metodoak a eta b tarteko balioak (a eta b barne) idatziko ditu, goranzko ordenan



```
public class BinaryTreeNode<T> {
    T data;
    BinaryTreeNode<T> left, right;
    int numberOfNodes;
}

public class BZB {
    BinaryTreeNode<Integer> root;

    public void add(int v)
        // Aurrebaldintza: v ez dago zuhaitzean.
        // Postbaldintza: v zuhaitzean txertatu da
        // Oharra: gogoratu txertaketaren ondorioz zenbait adabegiren numberOfNodes
        //          atributua eguneratu beharko dela.

    public void printTartea(int a, int b)
        // Azalpena: Idatzi irteera estandarrean a eta b tarteko balioa duten nodoen balioak
        //          (a eta b barne) goranzko ordenan.
        // Oharra: Errekurtsiboki implementatu eta soilik behar diren nodoak tratatu.
```

- a) add(int v) eta printTartea(int a, int b) metodoak implementatu
- b) Algoritmoaren kostuak kalkulatu, **modu arrazoituan**