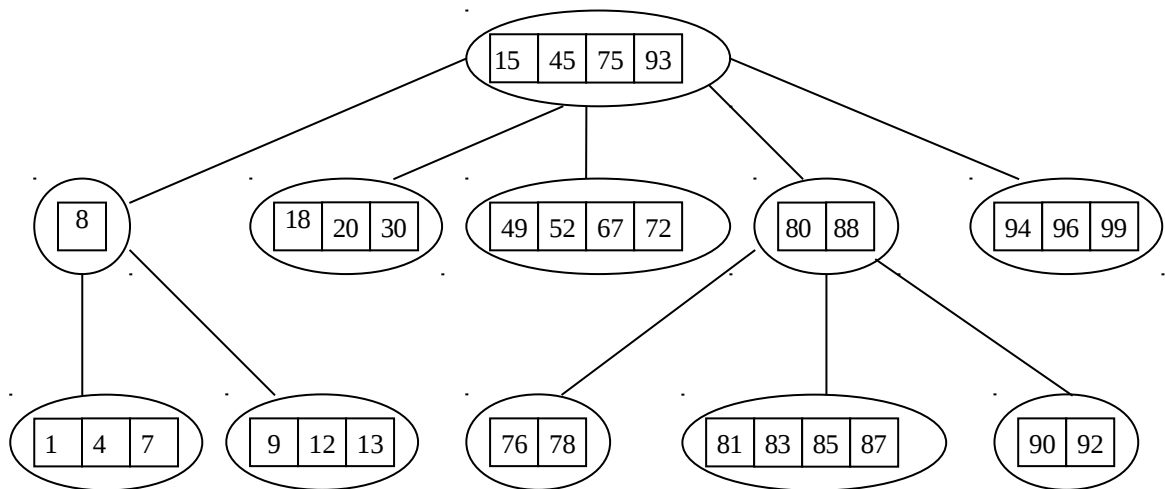


3. Búsqueda en árbol (1,5 puntos)

Tenemos un árbol N-ario de búsqueda, tal y como se muestra en la figura. Las restricciones son:

- Todos los valores del árbol son diferentes
- Cada nodo tiene N valores como máximo, ordenados ascendentemente
- Si un nodo tiene N valores a_1, a_2, \dots, a_n , entonces ese nodo tiene N+1 hijos, de tal manera que el hijo más a la izquierda contiene los valores menores que a_1 , el segundo subárbol contiene los valores entre a_1 y a_2, \dots y el último subárbol contiene los valores mayores que a_n .



Las declaraciones de datos son:

```
public class BinaryTreeNode<T> {
    T[] valores;
    BinaryTreeNode<T>[] hijos;
    // el tamaño es de la tabla de valores + 1
}

public class Arbol {

    BinaryTreeNode<Integer> root;

    public boolean esta(Integer elem)
        // post: el resultado es true si "elem" está en el árbol
        //      y false si no
}
}
```

Se pide:

- Implementar la función “esta”
- Calcular, de manera razonada, el coste del algoritmo implementado.