



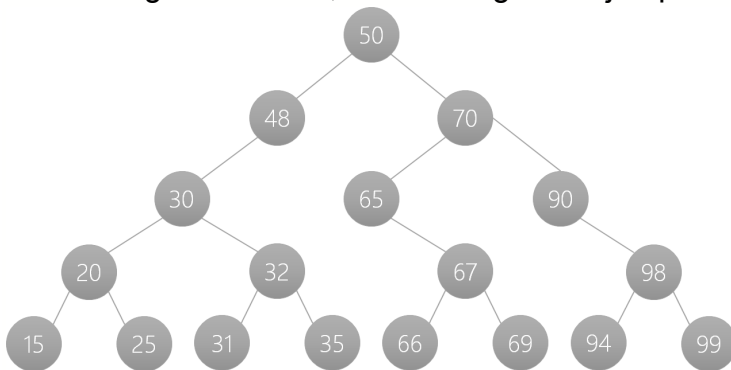
Convocatoria Extraordinaria. 26 Junio 2020

Problema 3 (20 puntos):

Sea MyBST la clase que implementa un árbol binario de búsqueda en Python (es una versión reducida que sólo incluye los métodos necesarios para este problema). En un árbol binario, el **mínimo ascendiente común** (*lowest common ancestor*) de dos nodos n y m es el nodo que contiene a los nodos n , m y a todos sus descendientes.

Implementa un método función, **lwc**, que reciba dos enteros a y b , encuentre sus nodos y obtenga el mínimo ascendiente común de ambos nodos. El método debe devolver el elemento asociado de dicho nodo ascendente. Si a ó b , no existen, el método debe devolver None. Si el árbol está vacío, el método debe devolver None. Pista: La solución más fácil sigue un enfoque recursivo. Puedes utilizar una función auxiliar.

Dado el siguiente árbol, veamos algunos ejemplos:



```
- tree.lwc(48,70)=50  
- tree.lwc(30,70)=50  
- tree.lwc(30,65)=50  
- tree.lwc(20,32)=30  
- tree.lwc(67,90)=70  
- tree.lwc(69,99)=70  
- tree.lwc(31,94)=50  
- tree.lwc(15,25)=20  
- tree.lwc(15,35)=30  
- tree.lwc(15,32)=30  
- tree.lwc(20,30)=30  
- tree.lwc(67,70)=70  
- tree.lwc(30,35)=30
```