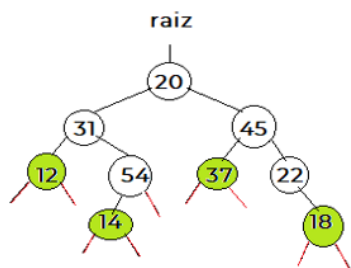


1. Método recursivo que recibe la raíz de un árbol binario y entonces muestra todas sus hojas y devuelve la cantidad de ellas que tienen contenido PAR.



Para este árbol el método **Mostraría 12 – 14 – 37- 18**

Además, **Devolvería 3** pues tres de las cuatro hojas son de contenido PAR

MET(R)

Si R es nulo entonces Devuelve 0	Si el método recibe un enlace R nulo debe devolver 0 (pues un Nodo que no existe no tiene hojas)
En caso contrario N = 0 Si R.izq es nulo y R.der es nulo entonces Muestra R.info Si R.info es PAR entonces N = 1 Fin Si Fin Si Devuelve N + MET(R.izq) + MET(R.der) Fin Si	Pero si recibe un enlace R que apunta a un Nodo Inicia en 0 el indicador de hoja con contenido PAR Si este nodo no tiene hijos para ningún lado es hoja Por lo tanto, MUESTRA su contenido Además, si su contenido es PAR entonces y Cambia el indicador a N = 1 Devuelve N más lo recibido al aplicar este mismo método a la izquierda y derecha del nodo actual
La primera parte mostrada arriba la tiene todo método recursivo . Es el CASO BASE , la situación en la que ya no se realizará un nuevo llamado al método. Pero lo que se escribe allí depende de si el método devuelve algo (por ejemplo, cantidad de nodos) o si el método solo realiza acciones (por ejemplo, mostrar el valor de cada nodo) En aquellos métodos que devuelven algo se usa Si R es nulo para devolver el valor que sea lógico devolver cuando el enlace recibido apunta a nada . En aquellos métodos que NO devuelven algo, que solo realizan acciones se usa Si R no es nulo , pues dichas acciones se pueden realizar solo si R apunta a un nodo que existe	

2. Método recursivo que recibe la raíz de un árbol binario y entonces devuelve verdadero si tiene más hojas para un lado de la raíz

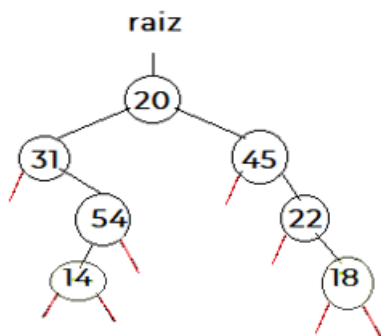
Para logra lo solicitado se usa siguiente método, **QUE NO ES RECURSIVO**

Si R es nulo entonces Devuelve Falso	Si recibe un R nulo entonces el árbol está vacío y obviamente NO tiene más hojas para un lado
En caso contrario HI = Hojas(R.izq) HD = Hojas(R.der) Si HI <> HD entonces Devuelve Verdadero Si No Devuelve Falso Fin Si Fin Si	Pero si recibe un enlace R que apunta a un Nodo Calcula cuántas hojas hay hacia la izquierda Calcula cuántas hojas hay hacia la derecha Si esas cantidades son distintas devuelve FALSO Si esas cantidades son distintas devuelve VERDADERO método a la izquierda y derecha del nodo actual

Pero ese método anterior hace llamados a este método **Hojas()**, el cual **ES RECURSIVO**

Si R es nulo entonces Devuelve 0	Si recibe un R nulo entonces el árbol está vacío y obviamente NO tiene hojas
En caso contrario Si R.izq es nulo y R.der es nulo entonces Devuelve 1 Si No Devuelve Hojas(R.izq) + Hojas(R.der) Fin Si Fin Si	Pero si el enlace R apunta a un Nodo que existe Entonces, si este nodo es hoja... Devuelve 1 (una hoja más) Pero si este nodo no es hoja... Devuelve la suma de las hojas que existen a lado y lado de él

3. Método recursivo que recibe la raíz de un árbol binario y entonces devuelve verdadero si ese árbol tiene solo dos ramas



Una rama es el recorrido desde el nodo raíz hasta una hoja. Entonces un árbol con dos ramas será como el mostrado en la imagen.

¿Cómo reconocerlo?

Porque, después del nodo raíz, cada nodo tiene un solo hijo (sin considerar las hojas, que no tiene hijos)

Si R es nulo entonces Devuelve Verdadero	Si R nulo árbol vacío, sin nada para lado y lado, es decir solo dos ramas nulas
En caso contrario Si R.izq NO es nulo y R.der NO es nulo entonces Devuelve Falso Si No Devuelve Met(R.izq) y Met(R.der) Fin Si Fin Si	Pero si R apunta a un Nodo que existe Si ese nodo tiene dos hijos entonces Devuelve Falso (nodo con dos ramas) Pero si no tiene dos hijos... Devuelve el resultado de la operación Lógica Y entre resultado a lado y lado
Es importante explicar la línea → Devuelve Met(R.izq) y Met(R.der) Si el nodo actual NO tiene dos hijos entonces podría tener NINGUNO o UNO . Si tiene NINGUNO devolverá verdadero Y verdadero , que es verdadero Si tiene solo UNO , con solo dos ramas, devolverá verdadero Y verdadero , que es verdadero Si tiene solo UNO , pero con varias ramas, devolverá falso Y verdadero , que es falso Este ejercicio resultó algo difícil de resolver en clase pues nos tomó tiempo darnos cuenta de que en un árbol vacío es VERDADERO que tiene solo dos ramas, aunque no existan. Para entender esto supongamos lo contrario, que es FALSO que un árbol vacío tiene solo dos ramas inexistentes, estaríamos diciendo que para un lado DE ESE NODO INEXISTENTE hay más nodos inexistentes que para el otro lado inexistente (esto si que sería absurdo)	

4. Método recursivo que recibe la raíz de un árbol binario y entonces muestra todos los nodos con un solo hijo (es decir que este método NO DEVUELVE algo SOLO REALIZA ALGO)

Met(R)

Si R NO es nulo entonces	Solo puede mostrar el nodo si el enlace R recibido apunta a un nodo que existe
Si R.izq NO es nulo y R.der es nulo entonces Muestra R.info Fin Si Si R.izq es nulo y R.der NO es nulo entonces Muestra R.info Fin Si Met(R.izq) Met(R.der) Fin Si	Si el Nodo solo tiene hijo izquierdo Entonces muestra este nodo Si el Nodo solo tiene hijo derecho Entonces muestra este nodo Aplica el método a la izquierda Aplica el método a la derecha

SUGERENCIAS

1. **Inventa como estos**, aunque parezcan ociosos (como, por ejemplo, tan ocioso como el siguiente: método que cuente los nodos de un árbol que son mayores que el nodo raíz)
2. **Intenta resolverlo**, sin importar cuanto te demores
Esa será la mejor preparación para el primer ejercicio de la segunda prueba, que se realizará de este viernes 2 de junio (**llega temprano**, se realizará al inicio de la clase)
3. Dale el planteamiento del ejercicio a **charGPT** y compara su respuesta con tu solución