ENTREGA E5 (0,5 puntos)

Antes de empezar...

- Descargar el archivo **E5.zip** e importarlo a Eclipse.
- Renombrar el proyecto (tecla F2). Por ejemplo, para el grupo formado por Ana Pérez, Jon Azkue y Miren Landa el nuevo nombre sería: **E5_APerezJAzkueMLanda**.
- En el fichero **componentesGrupo**, escribir los nombres de los componentes del grupo.

Además, tened en cuenta que:

- Se valorará la eficiencia de las soluciones, es decir, además de que sean correctas, deben ser eficientes.
- Aparte de los casos de prueba que se os entregan, se espera que incorporéis casos de prueba adicionales y significativos.
- Para entregarlo, debéis exportar el proyecto y subirlo a eGela.

Fecha límite de entrega: 19/11/2023 a las 23:59

—

Ayuda para importar:

File -> Import... -> General -> Existing Projects into Workspace -> Select archive file (el .zip descargado) -> Finish

Ayuda para exportar:

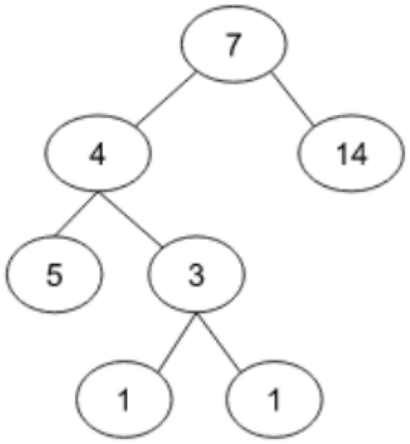
Pinchar en el proyecto. File -> Export... -> General -> Archive File -> (seleccionar las carpetas/archivos a exportar) -> To archive file (escribir una ruta y nombre para el nuevo archivo .zip) -> Finish

NOTA: A CONTINUACIÓN DEL ENUNCIADO PODÉIS ENCONTRAR LOS DIBUJOS DE LOS ÁRBOLES UTILIZADOS EN LOS CASOS DE PRUEBA QUE SE OS ENTREGAN, ASÍ COMO LOS RESULTADOS DE EJECUCIÓN ESPERADOS PARA DICHOS CASOS DE PRUEBA.

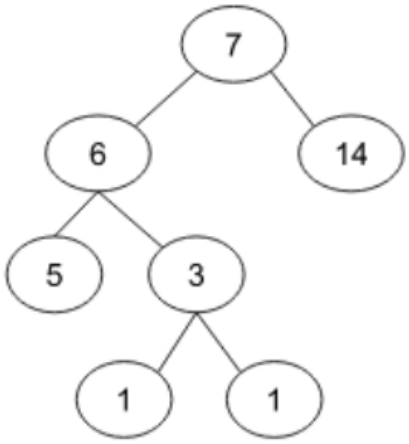
OJO, QUE LOS ÁRBOLES SON DIFERENTES PARA CADA EJERCICIO.

Ejercicio 1

Para un árbol binario de enteros positivos: implementa el método **esEquiponderado**. Este método devuelve true si para todos los nodos del árbol se cumple la siguiente condición: la suma de los valores del subárbol izquierdo es igual a la suma de los valores del subárbol derecho. En caso contrario, devolverá false. Suponemos que el árbol vacío es equiponderado.



esEquiponderado(): true



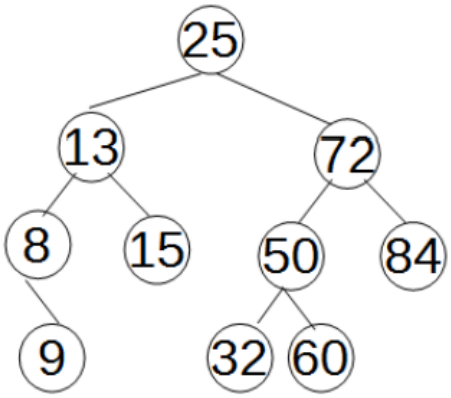
esEquiponderado(): false

Ejercicio 2

Para un **árbol binario de búsqueda (ABB)** que contiene enteros, implementar el método **listaNivelDeMayores**. Este método recibe como parámetro dos enteros *num* y *niv* y devuelve una lista con los elementos del árbol que están en el nivel *niv* y además son mayores que *num*. La lista debe estar ordenada ascendentemente. Si no hay ninguno, devuelve una lista vacía.

Hay que hacerlo de manera eficiente, es decir, no se debería, por ejemplo, visitar ramas que no son necesarias.

Por ejemplo, para el siguiente ABB, la llamada **listaNivelDeMayores(14, 2)** devolvería la lista <15,50,84>, puesto que son los elementos mayores que 14 que están en el nivel 2.

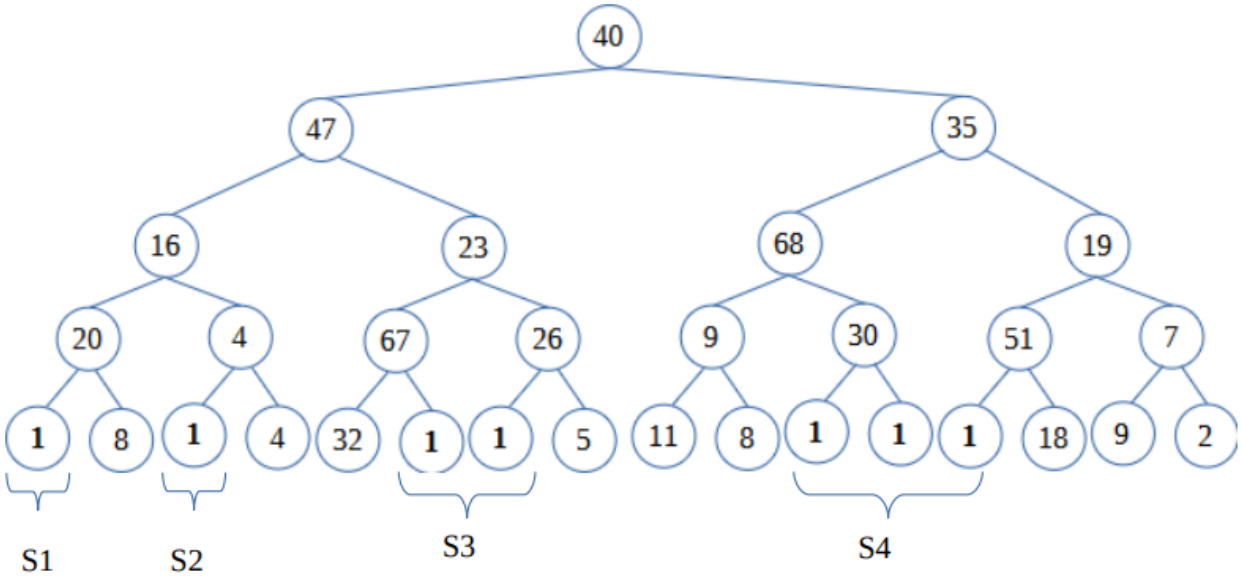


Ejercicio 3

Un árbol binario es perfecto si todos sus niveles están totalmente completos con nodos (es decir, tiene un nodo en el nivel 0, dos nodos en el nivel 1, cuatro nodos en el nivel 2, etc). Por tanto, en un árbol perfecto se cumple que todos los nodos internos tienen dos hijos y que todas las hojas están en el mismo nivel.

Dado un árbol binario perfecto de enteros, se pide implementar el método **numSecuencias**, que devuelve un entero que indica el nº de secuencias formadas por hojas con valor 1 que hay en el árbol.

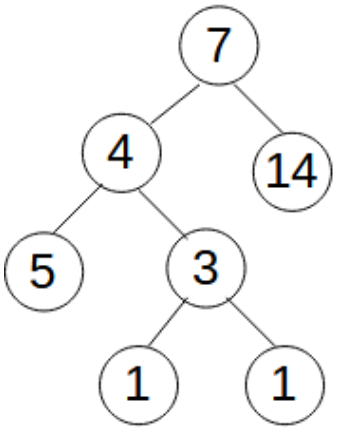
NOTA: No se permite extraer una lista con todas las hojas del árbol y luego determinar el nº de secuencias de la lista.



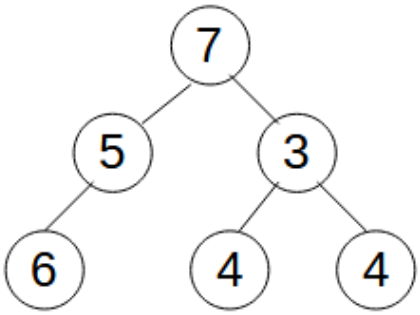
Para el árbol de ejemplo, el método devolvería que hay **4** secuencias (que serían las marcadas como S1, S2, S3 y S4). Fijaos por ejemplo, en la secuencia S3. Se ha formado porque la hoja más derecha del subárbol izquierdo del 23 tiene el valor 1 y la hoja más izquierda del subárbol derecho del 23 tiene el valor 1.

ÁRBOLES PARA EL EJERCICIO 1

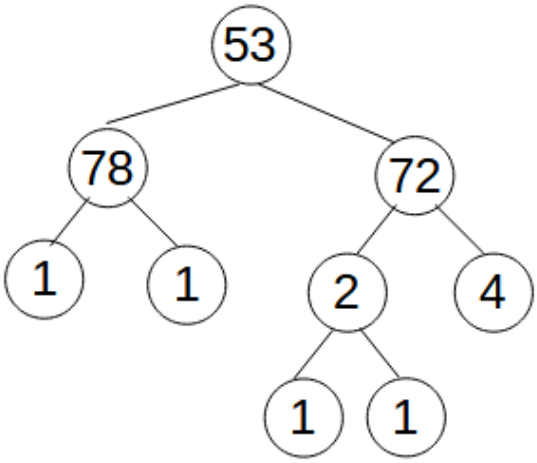
arbol_1.0.txt



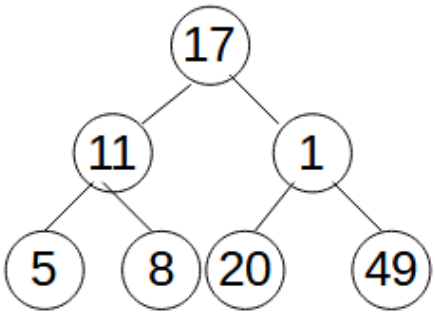
arbol_1.1.txt



arbol_1.2.txt



arbol_1.3.txt



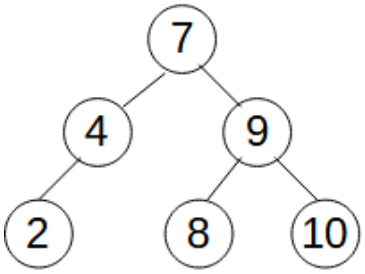
arbol_1.4.txt

(vacío)

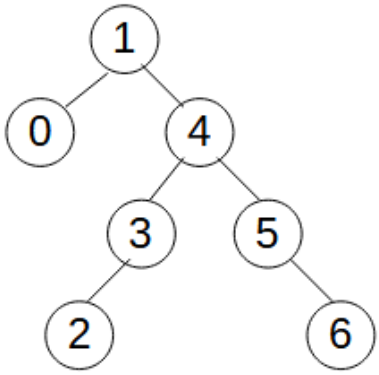
arbol_1.5.txt

ÁRBOLES PARA EL EJERCICIO 2

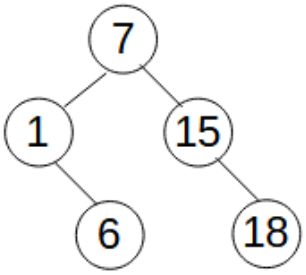
arbolABB_2.0.txt



arbolABB_2.1.txt



arbolABB_2.2.txt

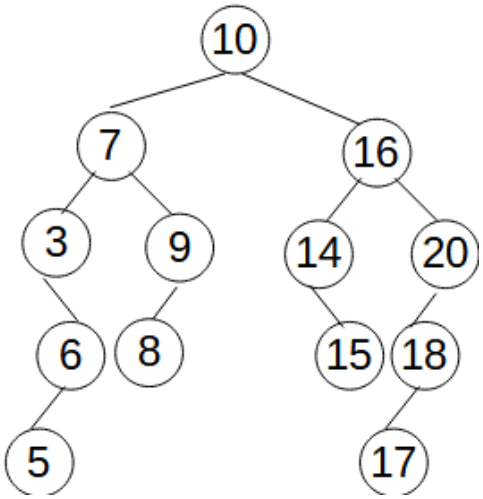


arbolABB_2.3.txt

(vacío)

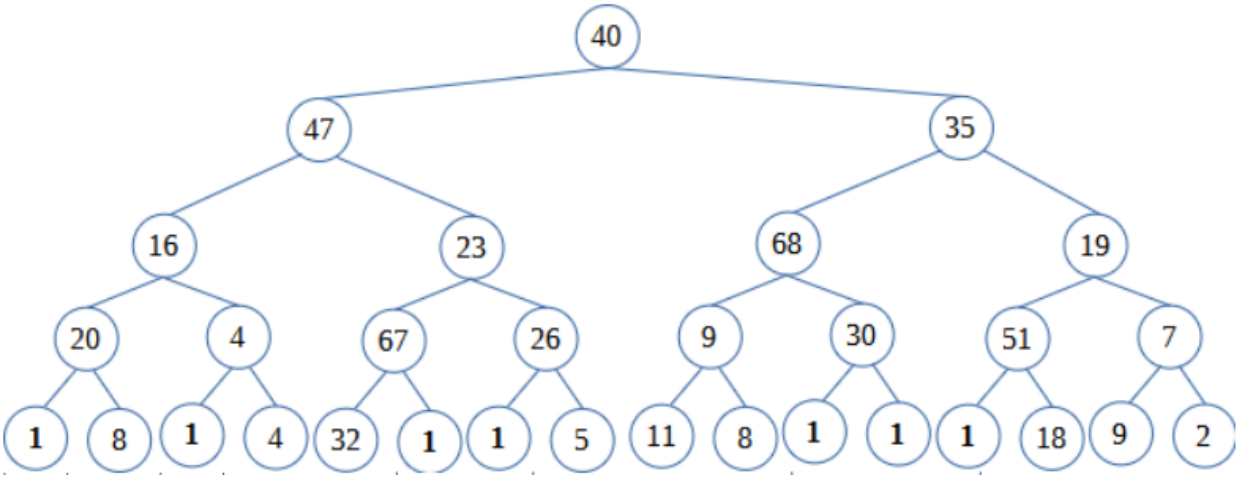
arbolABB_2.4.txt

arbolABB_2.5.txt



ÁRBOLES PARA EL EJERCICIO 3

arbolPerfecto_3.0.txt

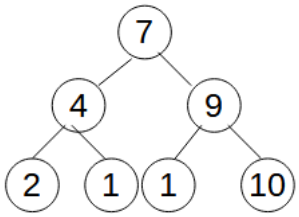


arbolPerfecto_3.1.txt

(vacío)

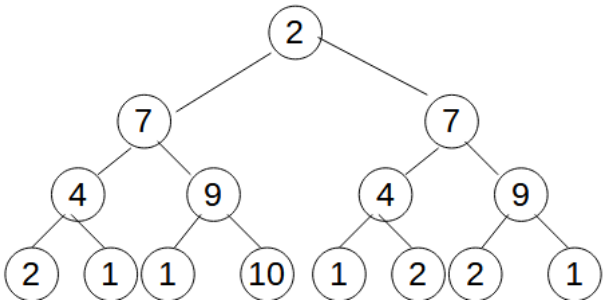
arbolPerfecto_3.2.txt

arbolPerfecto_3.3.txt



arbolPerfecto_3.4.txt

arbolPerfecto_3.5.txt



RESULTADOS DE LA EJECUCIÓN DE LOS CASOS DE PRUEBA PROPORCIONADOS

EJERCICIO 1

```
-------------
 ÁRBOL 0
-------------
esEquiponderado?: true

-------------
 ÁRBOL 1
-------------
esEquiponderado?: false

-------------
 ÁRBOL 2
-------------
esEquiponderado?: true

-------------
 ÁRBOL 3
-------------
esEquiponderado?: false

-------------
 ÁRBOL 4
-------------
esEquiponderado?: true

-------------
 ÁRBOL 5
-------------
esEquiponderado?: true
```

EJERCICIO 2

```
-------------
 ÁRBOL 0
-------------
Árbol: [ 7 [ 4 [ 2 ] * ] [ 9 [ 8 ] [ 10 ] ] ]
Mayores que 6 en nivel 2: [8, 10]
Mayores que 18 en nivel 2: []
Mayores que 5 en nivel 3: []
Mayores que 10 en nivel 4: []

-------------
 ÁRBOL 1
-------------
Árbol: [ 1 [ 0 ] [ 4 [ 3 [ 2 ] * ] [ 5 * [ 6 ] ] ] ]
Mayores que 6 en nivel 2: []
Mayores que 18 en nivel 2: []
Mayores que 5 en nivel 3: [6]
Mayores que 10 en nivel 4: []

-------------
 ÁRBOL 2
-------------
Árbol: [ 7 [ 1 * [ 6 ] ] [ 15 * [ 18 ] ] ]
Mayores que 6 en nivel 2: [18]
Mayores que 18 en nivel 2: []
Mayores que 5 en nivel 3: []
Mayores que 10 en nivel 4: []

-------------
 ÁRBOL 3
-------------
Árbol: *
Mayores que 6 en nivel 2: []
Mayores que 18 en nivel 2: []
Mayores que 5 en nivel 3: []
Mayores que 10 en nivel 4: []

-------------
 ÁRBOL 4
-------------
Árbol: [ 7 ]
Mayores que 6 en nivel 2: []
Mayores que 18 en nivel 2: []
Mayores que 5 en nivel 3: []
Mayores que 10 en nivel 4: []

-------------
 ÁRBOL 5
```

```
-------------
Árbol: [ 10 [ 7 [ 3 * [ 6 [ 5 ] * ] ] [ 9 [ 8 ] * ] ] [ 16 [ 14 * [ 15 ] ] [ 20
[ 18 [ 17 ] * ] * ] ] ]
Mayores que 6 en nivel 2: [9, 14, 20]
Mayores que 18 en nivel 2: [20]
Mayores que 5 en nivel 3: [6, 8, 15, 18]
Mayores que 10 en nivel 4: [17]
```

EJERCICIO 3

```
-------------
 ÁRBOL 0
-------------
numSecuencias: 4

-------------
 ÁRBOL 1
-------------
numSecuencias: 0

-------------
 ÁRBOL 2
-------------
numSecuencias: 0

-------------
 ÁRBOL 3
-------------
numSecuencias: 1

-------------
 ÁRBOL 4
-------------
numSecuencias: 1

-------------
 ÁRBOL 5
-------------
numSecuencias: 3
```