Ejercicio 1.- Extender la especificación básica de los árboles binarios con las siguientes operaciones:

- `num_nodos: a_bin → natural`, para conseguir el número de nodos en total que hay en un árbol binario;
- `num_hojas: a_bin → natural`, para calcular la cantidad de hojas que tiene un árbol binario;
- `num_padre_de_2: a_bin → a_bin`, que obtiene cúantos nodos tienen dos hijos no vacíos.

Ejercicio 2.- Se dice que un árbol binario es "zurdo" en uno de estos tres casos:

- si es el árbol vacío; o
- si es una hoja; o
- si sus hijos izquierdo y derecho son los dos "zurdos" y el hijo izquierdo tiene más elementos que el hijo derecho.

Crear las operaciones necesarias para determinar si un árbol binario es "zurdo".

Ejercicio 3.- *(Examen del Grado en Ingeniería Informática, Enero 2011)*. Extender el TAD árboles binarios de naturales (solo hace falta mostrar la especificación de las operaciones básicas de los árboles binarios, pero si se utiliza alguna operación auxiliar hay que presentar su especificación y sus ecuaciones), añadiendo operaciones para:

- Obtener la suma de todos los elementos que sean números pares del árbol,
- Obtener la imagen especular de un árbol (reflejo respecto al eje vertical),
- Crear tres operaciones que generen una lista con los elementos del árbol recorrido en preorden, inorden y postorden,
- Comprobar si el árbol está ordenado en inorden, usando para ello únicamente operaciones de árboles (en concreto, no puede utilizarse el apartado anterior).

Ejercicio 4.- *(Examen del Grado en Ingeniería Informática, Enero 2012)*
Extender el TAD árboles binarios de naturales (solo hace falta mostrar la especificación de las operaciones básicas de los árboles binarios, pero si se utiliza alguna operación auxiliar hay que presentar su especificación y sus ecuaciones), añadiendo las operaciones siguientes:

- `igual_cantidad?: a_bin a_bin → bool`, detecta si dos árboles binarios tienen la misma cantidad de nodos;
- `igual_forma?: a_bin a_bin → bool`, que comprueba si dos árboles binarios tienen la misma forma;
- `suma_árboles: a_bin a_bin → bool`, que recibe dos árboles binarios de naturales y si tienen la misma forma genera el árbol resultante de sumar los valores de los nodos que ocupan el mismo lugar relativo en cada uno de ellos; y
- `cuenta_veces: a_bin a_bin → natural`, que recibe dos árboles binarios y devuelve la cantidad de veces que aparecen los nodos del primero en cualquier posición del segundo.

Ejercicio 5.- Se quiere hacer un recorrido de un árbol binario por niveles (el nivel *k* son todos los nodos que están a distancia *k* de la raíz del árbol). Se pide:

- `nivel_n: a_bin natural → lista`, que crea una lista con todos los nodos que se encuentren en el nivel indicado por el *natural* del segundo parámetro;
- `niveles_entre: a_bin natural natural → lista`, que crea una lista con todos los nodos que se encuentren entre los niveles indicados por los dos números naturales;
- `recorrer_niveles: a_bin → lista`, que crea una lista formada por todos los niveles del árbol binario.

Ejercicio 6.- Extender la especificación de los árboles generales vista en clase con las siguientes operaciones:

- `num_nodos: árbol → natural`, para calcular cuántos nodos hay en un árbol general;
- `num_hojas: árbol → natural`, para ver la cantidad total de hojas que tiene un árbol general;
- `max_hijos: árbol → natural`, que obtiene cuál es la mayor cantidad de hijos en un mismo nodo que hay en un árbol general.
- `reflejar: árbol → árbol`, que obtiene la imagen especular de un árbol;
- `frontera: árbol → lista`, que genera una lista formada por los elementos almacenados en las hojas del árbol, tomados de izquierda a derecha.

Ejercicio 7.- Extender la especificación de árbol general con las mismas operaciones del **Ejercicio 5**, aunque aplicadas sobre árboles generales en vez de árboles binarios.

Ejercicio 8.- *(Examen del Grado en Ingeniería Informática, Enero 2011)*
Llamaremos a un árbol general de naturales "maestro" si el valor de cada nodo es igual al número de hijos que tiene dicho nodo. Se pide:

- Especificar completamente el TAD árbol general,
- Comprobar si un árbol general es "maestro",
- Buscar el nodo con mayor valor de un árbol maestro (es decir, el que tenga más hijos).

Ejercicio 9.- *(Examen del Grado en Ingeniería Informática, Enero 2012)*
Llamaremos a un árbol general de naturales "creciente" si en cada nivel del árbol la cantidad de nodos que hay en ese nivel es igual al valor del nivel más uno; es decir, el nivel 0 tiene exactamente un nodo, el nivel 1 tiene exactamente dos nodos, el nivel *k* tiene exactamente *k* + 1 nodos. Se pide:

- Especificar completamente el TAD árbol general,
- Comprobar si un árbol general es "creciente",
- Buscar el nodo con mayor cantidad de hijos de un árbol creciente.

Ejercicio 10.- *(Examen de todos los Grados, Enero 2012)* Llamaremos a un árbol general de enteros "progresivo" si cada nodo con hijos tiene a éstos ordenados crecientemente por valor, es decir, si las raíces de cada árbol del bosque aparecen en orden creciente en dicho bosque. Se pide:

- Especificar completamente el TAD árbol general de enteros,
- Comprobar si un árbol general de enteros es "progresivo",
- Transformar un árbol general que no cumpla ser "progresivo" en el árbol "progresivo" equivalente reordenando los hijos en el bosque.