

Examen Extraordinario de Programación Curso 2013-2014

Buscando al padre para reconstruir un árbol

NOTA: Si usted está leyendo este documento sin haber extraído el compactado que descargó del sitio, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios **no se guarden**. Si usted comete este error y entrega una solución vacía, **no tendrá oportunidad de reclamar**.

Un árbol puede ser construido a partir de los valores de sus nodos y sabiendo quién es su nodo padre. De este modo teniendo dos arrays, uno con los valores de cada nodo y el otro array con la posición del padre para cada valor nodo del primer array podemos entonces reconstruir el árbol.

En la Figura 1 se muestran un tal par de arrays de valores y un array indicando la posición del padre para cada valor. Se puede inferir, por ejemplo, que el padre del nodo valores[2] que tiene valor 7 es el nodo que está en la posición posicióndeelpadre[2] es decir en la posición 3. Luego tomando valores[3], se llega a que es el 20. Note también que el elemento en la posición 3 que tiene valor 20 no tiene padre dado que posicióndeelpadre[3] es -1. Esto significa que el 20 es el elemento raíz del árbol. En la **Error! Reference source not found.** se muestra el árbol resultante para el Ejemplo 1.

Ejemplo 1

	0	1	2	3	4
valores	3	11	7	20	5
posición del padre	2	3	3	-1	3

Figura 1. Valores y padres, ejemplo 1

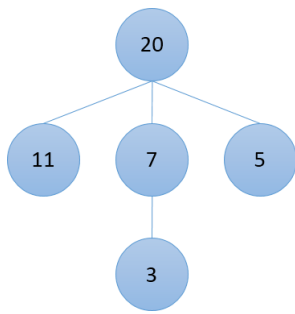
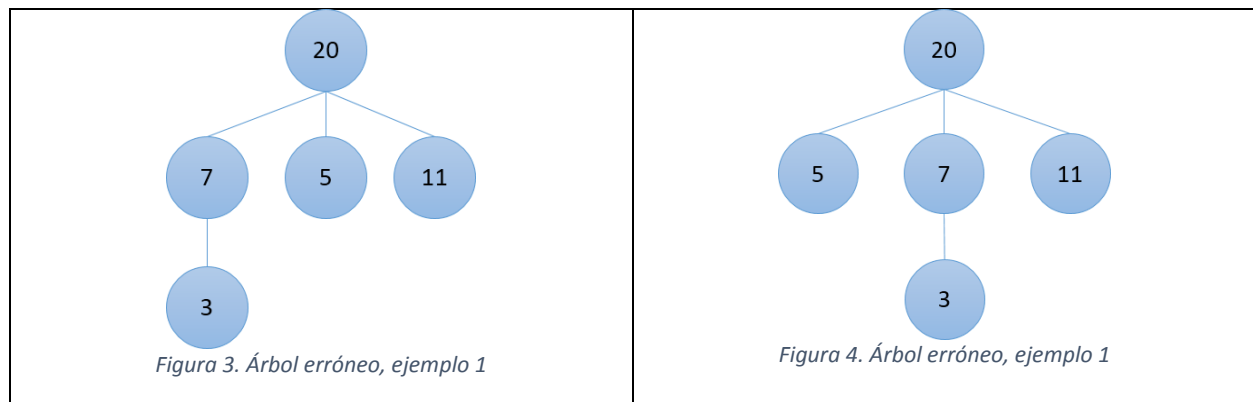


Figura 2. Árbol, ejemplo 1

Note que los árboles de las Figura 3 y Figura 4 aunque tiene la misma relación de paternidad entre sus nodos que el árbol de la Figura 2, no son una solución válida porque los hermanos deben estar en el mismo orden (de izquierda a derecha) en que aparecen en el array de valores.



Ejemplo 2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
valores	9	29	2	19	31	7	23	45	3	37	11	51	3	9
posición del padre	3	3	10	-1	3	7	7	0	4	1	9	10	7	4

Figura 5. Valores y padres, ejemplo 2

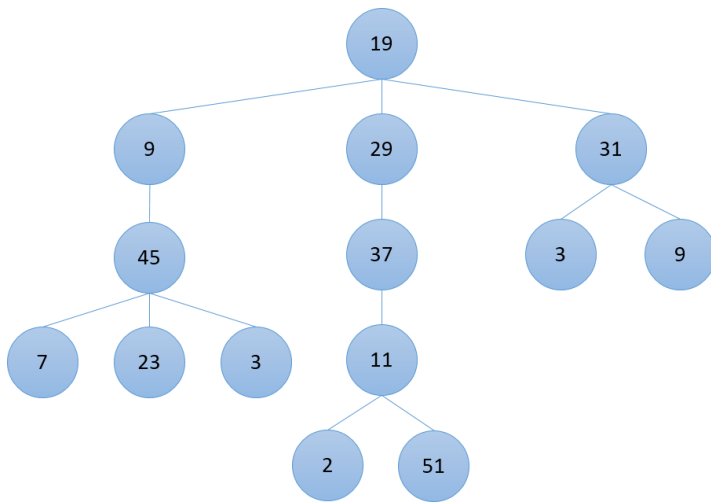


Figura 6. Árbol, ejemplo 2

Note como en el caso del Ejemplo 2, el árbol contiene valores repetidos como el 3 y el 9 lo cual no impide su reconstrucción (Figura 6).

Ejemplo 3

	0	1	2	3	4	5	6	7	8	9	10	11
<i>valores</i>	12	7	8	9	9	11	7	1	2	3	2	1
<i>posición del padre</i>	7	9	3	9	3	9	0	9	0	11	0	-1

Figura 7. Valores y padres, ejemplo 3

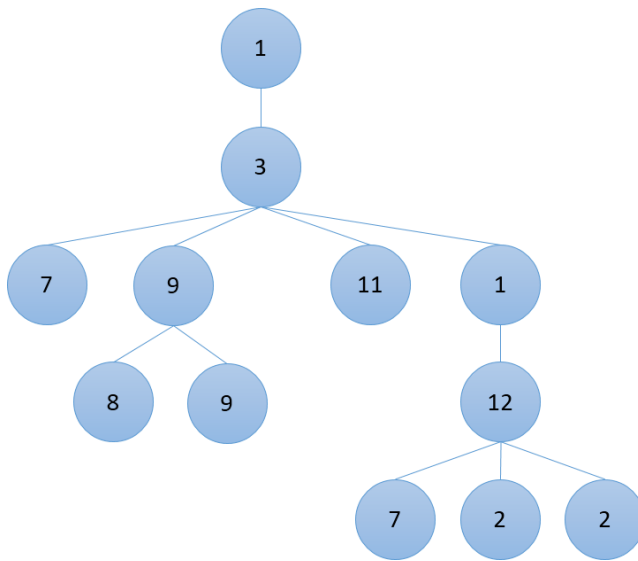


Figura 8. Árbol, ejemplo 3

Usted deberá implementar el método `Reconstruye` de la clase `BuscandoPadre`. Dicho método podrá ser encontrado en la plantilla de código que le es provista junto con este documento.

```
namespace Weboo.Examen.Extraordinario
{
    public class BuscandoPadre
    {
        public static Arbol<T> Reconstruye<T>(T[] valores, int[] indicePadres)
        {
            throw new NotImplementedException("Su código debe ir aquí");
            //En lugar de disparar excepción coloque aquí su implementación
        }
    }
}
```