

Camino y posteriores

En el ensamblado `Weboo.Arboles.dll` está la clase `Arbol<T>` cuya interfaz pública se muestra a continuación

```
public class Arbol<T>
{
    public Arbol(T valor, params Arbol<T>[] hijos)
    public IEnumerable<Arbol<T>> Hijos { get ; }
    public Arbol<T> HijoNo(int i)
    public T Valor { get; }
    public int CantidadDeHijos { get ; }
    public bool EsHoja { get ; }
}
```

Complete la implementación de la clase en el proyecto que se le da para que se genere un ensamblado `CaminoPosteriores.dll` una clase

```
static class Respuesta {
    public static IEnumerable<T> Camino<T>(Arbol<T> a, T x) {
        //...
    }
    public static IEnumerable<T> Posteriores<T>(Arbol<T> a, T x) {
        //...
    }
}
```

El método estático `Camino` debe devolver un iterador que devuelva la secuencia de un camino que vaya de la raíz hasta un nodo con valor igual a `x`. El iterador respuesta puede ser vacío si no hay ningún tal camino. Para el árbol de la **Figura 1** si el valor de `x` fuese `100` el camino está formado por los valores de los nodos que se han coloreado en azul.

Por simplicidad asuma que no habrá valores repetidos en los árboles de prueba.

El método estático `Posteriores` debe devolver los valores de los nodos que se encuentran "a la derecha" del camino hasta `x`, para el ejemplo de la **Figura 1** sería los valores de los nodos que se han coloreado en rosado. No importa el orden en que se den los valores según su ubicación en el árbol.

Note que el iterador pudiera ser vacío si no hubiese nodos a la derecha del camino. Para el caso del ejemplo la respuesta a la evaluación de `Posteriores(árbol Figura 1, 44)` es un iterador vacío, no hay nodos a la derecha del camino `2, 24, 10, 11, 44` (**Figura 2**). De igual modo si el camino es de un solo nodo, caso en el que el valor al que se quiere llegar es el valor de la propia raíz `Posteriores(árbol Figura 1, 2)` entonces el iterador también es vacío (**Figura 3**).

Aunque el ejemplo se ha ilustrado con valores numéricos enteros y la herramienta auxiliar de ayuda `CreadorDeArbol.exe` se ha hecho para que usted cree árboles de enteros o string, usted debe dar su respuesta del modo más general posible para que trabaje para árboles con valores de cualquier tipo para los nodos. Para la evaluación se probará con tipos valores no solo enteros.

La herramienta visual [CreadorDeArbol.exe](#) que se adjunta con este documento le ayuda a construir árboles que pueda usar de ejemplos en sus pruebas. La herramienta le genera un texto con el código C# correspondiente para generar un árbol usando la definición de la clase `Arbol<T>`, copie ese texto y péguelo donde lo necesite dentro de su proyecto de prueba.

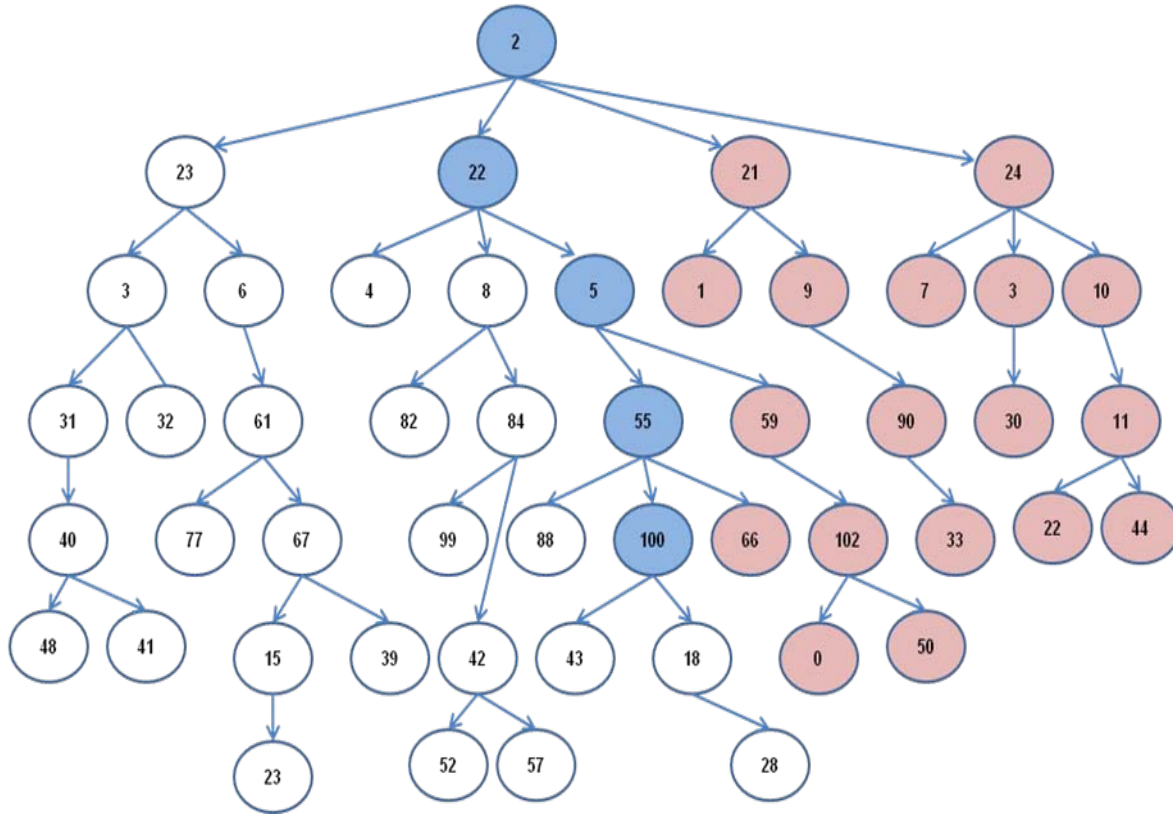


Figura 1. Los nodos sombreados en azul denotan un camino para llegar al nodo con valor 100, los nodos sombreados en rosado denotan los nodos posteriores al camino

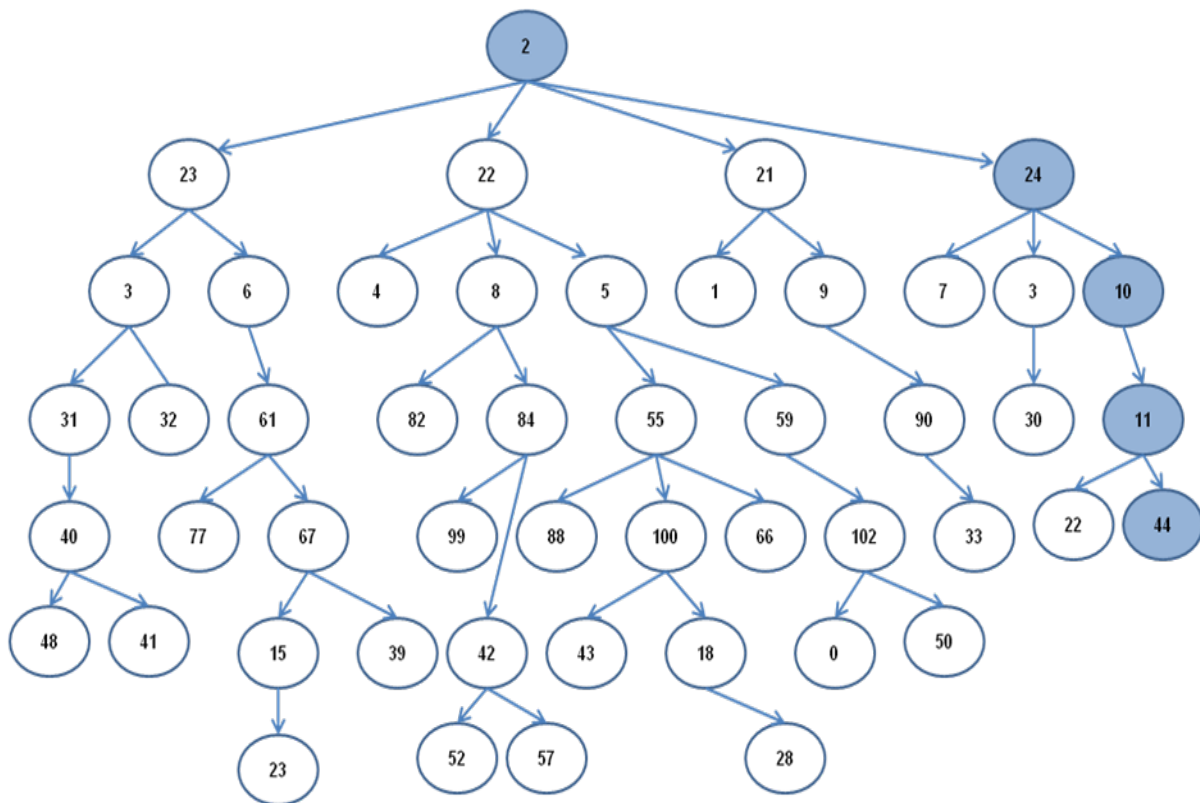


Figura 2. No hay nodos posteriores al camino hasta el nodo con valor 44

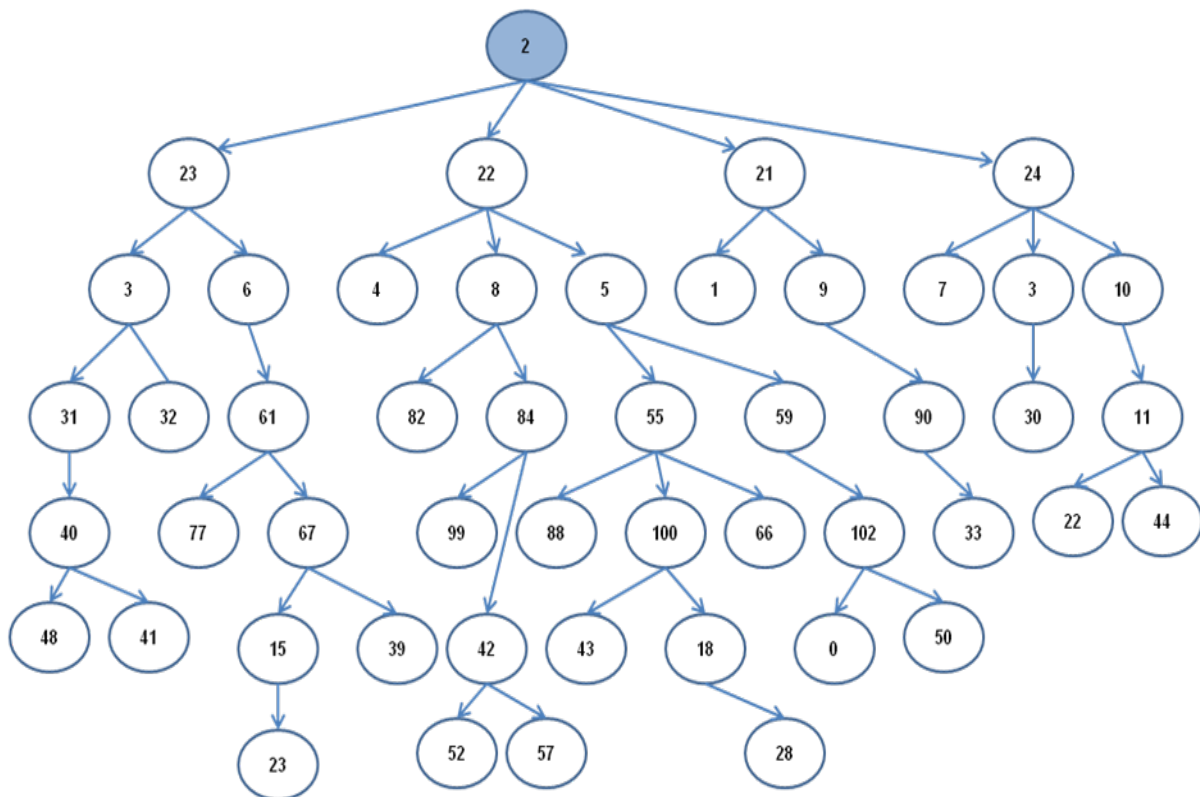


Figura 3. No hay nodos posteriores al camino formado por el único nodo con valor 2

NOTA

Si quiere construir directamente árboles de prueba, recuerde que al usar el especificador `params R[]` en el constructor, nos permite escribir `0` o más parámetros del tipo `R` por ejemplo

```
Arbol<int>a = new Arbol<int>(
    5,
    new Arbol<int>(30),
    new Arbol<int>(20),
    new Arbol<int>(50)
);
```