

Iterador a lo ancho

Dado un árbol definido por el tipo `Arbol<T>`

```
class Arbol<T>
{
    public T valor;
    public IEnumerable<Arbol<T>> Hijos;
    public Arbol(T valor, params Arbol<T>[] hijos)
    {
        //...
    }
}
```

y que se le da en el ensamblado `Arbol.dll` implemente el método de `IteradorAloAncho` de la interface `IIteradores` que se le da en el ensamblado `IIteradores.dll`.

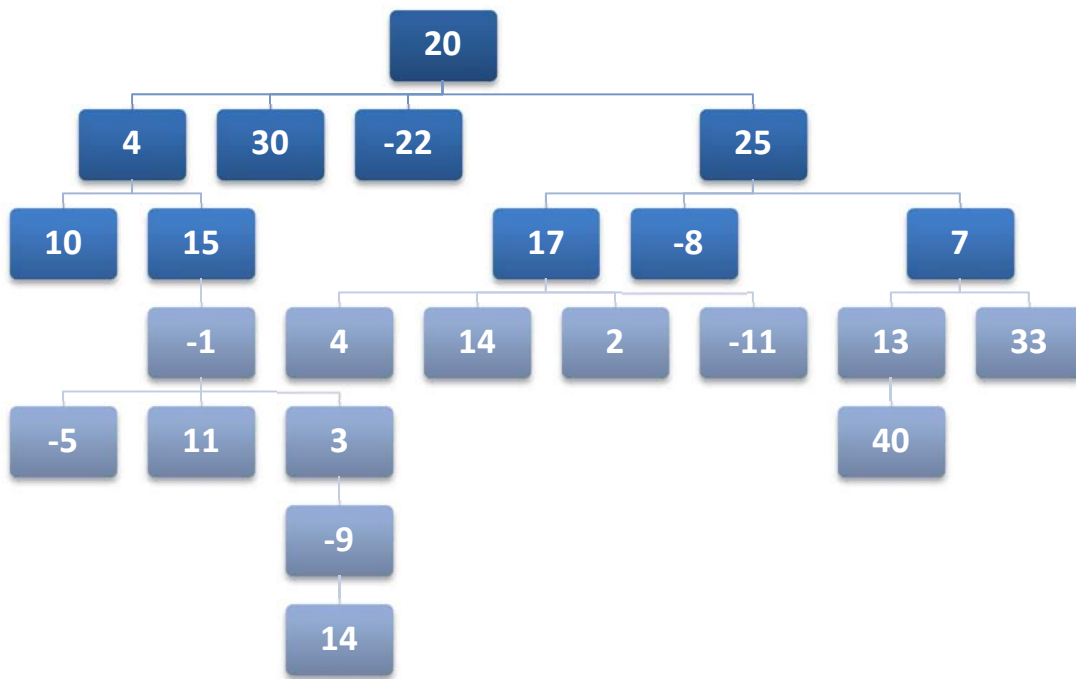
```
delegate R Func<T, R>(T x);

interface IIteradores<T>
{
    IEnumerable<IEnumerable<T>> IteradorAloAncho(Arbol<T> a,
                                                Func<T, bool> filtro);
}
```

El método debe devolver un `IEnumerable` de `IEnumerable<T>` donde cada `IEnumerable<T>` corresponde a los elementos `T` valores de los nodos de cada nivel de recorrido a lo ancho del árbol que se le da como parámetro y que hayan pasado el filtro, es decir que al aplicarle el delegado `filtro` evalúa `true`.

Para resolver el problema anterior usted deberá implementar en un proyecto de tipo *Class Library* llamado `ProblemaIIteradores` una clase de nombre `MiIterador<T>` que implemente la interfaz anterior.

Si `a` fuese el árbol que corresponde a la figura 1 entonces la ejecución de un código como el siguiente (suponga que ha llamado al método `IteradorAloAncho` indicando el filtro correspondiente para escoger los nodos que sean valores positivos)



```

IEnumerator<int> miIterador = new MiIterador<int>();
int nivel=0;
foreach (IEnumerable<int> nodos in
    miIterador.IteradorAloAncho(a, ... aquí iría el filtro ...))
{
    Console.WriteLine("Nivel {0}", nivel);
    foreach (int k in nodos)
        Console.Write("{0} ",k);
    Console.WriteLine();
    nivel++;
}

```

debe dar como resultado

```

Nivel 0
  20
Nivel 1
  4 30 25
Nivel 2
  10 15 17 7
Nivel 3
  4 14 2 13 33
Nivel 4
  11 3 40
Nivel 5
  14

```

Note que aunque el ejemplo se ha ilustrado con un árbol con nodos de valores de tipo `int` su respuesta debe valer para cualquiera sea el tipo concreto que se le dé al parámetro genérico `T`