

Vitoria-Gasteiz, 21 de diciembre de 2011

Nombre y apellidos: _____ Nota: _____

- Duración: 1 hora 15 minutos
- No está permitido el uso de apuntes, libros o móviles
- **Todos los alumnos implicados en una copia de un ejercicio tendrán una nota final de 0.** El alumno es responsable de velar por su examen. Es decir **tanto el que copia como el que se deja copiar (ya sea de manera activa o pasiva) recibirán el mismo castigo sin que exista atenuante alguno**
- Revisión del examen: 9-1-2012, 12:00-13:00

2. Ejercicio (2 puntos)

Dada la siguiente especificación de las clases BTreeNode, BinTree y BinTreeItr:

```
public class BinTreeItr<T>{
    BinTree<T> bTree;
}

public class BinTree<T>{
    BTreeNode<T> root;
}

public class BTreeNode<T>{
    T content;
    BTreeNode<T> left;
    BTreeNode<T> right;
}
```

Nota: Todo método necesario tendrá que ser implementado.

Especificar, diseñar e implementar un método recursivo que dado un árbol binario de números enteros, que representan el coste de visitar ese nodo, calcule y devuelva cual es la longitud del camino que mayor coste supone recorrerla. En caso de que haya más de un camino con el mismo coste, entonces se devolverá el de mayor longitud.

int longitudCaminoCosteMaximo();

Un **camino** es la secuencia de arcos que une dos nodos de un árbol. Mientras, la **longitud** de un camino entre dos nodos es el número de arcos presentes en el camino. El coste de recorrerlo sería la suma del coste de visitar cada uno de los nodos que forman el camino. Cada nodo del árbol puede ser alcanzado (se llega a él) siguiendo un único camino que comienza en el nodo raíz. Su coste siempre es 0, mientras que visitar los **nodos derechos cuesta el doble** (de lo indicado) que visitar los nodos izquierdos.

Ejemplos:

