



Actividad 4. Manejo de Árboles.

Objetivos:

- Utilización de árboles binarios de búsqueda.
- Elaboración y prueba de algoritmos básicos sobre árboles (recorridos, inserción y eliminación).

Desarrollo de la práctica.

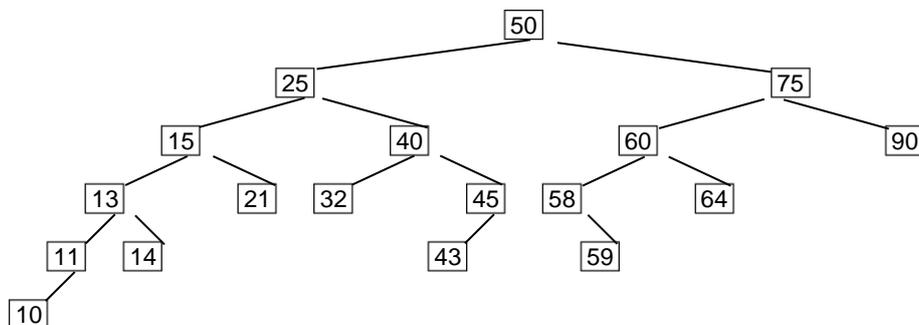
Preparación del entorno de trabajo.

- Recupere el *workbench* de la actividad práctica anterior y cree en él un nuevo proyecto de nombre *Actividad4*.
- Descargue de Moodle los ficheros que puedan resultarle de utilidad.
- Descargue, también, el archivo de texto *arbol.txt*.

Ejercicios.

1. Construcción de un árbol.

Codificar una función *void* en C++ que permita crear un árbol binario de búsqueda de números enteros y cargarlo con los datos contenidos en un fichero de texto (*arbol.txt*). La figura siguiente muestra el modelo de árbol que deberá construir.

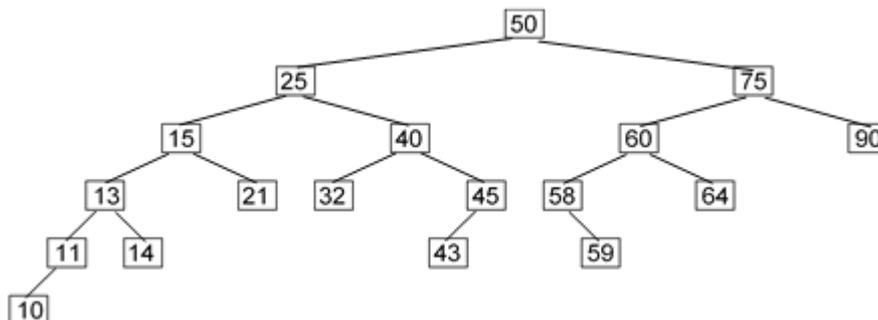


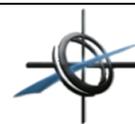
2. Recorrido sin modificar la estructura.

Codificar una función entera en C++ que, recibiendo como argumento un árbol, devuelva como resultado la suma de las claves impares contenidas en nodos con un hijo.

EJEMPLO:

Dado el siguiente árbol, las claves que cumplen ambas condiciones son 45 y 11, luego el resultado será 56.





3. Eliminación. Ejercicio propuesto en examen (julio 2016).

Dada la siguiente declaración de Árbol Binario de Búsqueda:

```
class NodoArbol {  
public:  
    int clave;  
    NodoArbol *iz, *de;  
}  
  
typedef NodoArbol *pNodoArbol;  
class Arbol {  
public:  
    pNodoArbol raiz;  
}
```

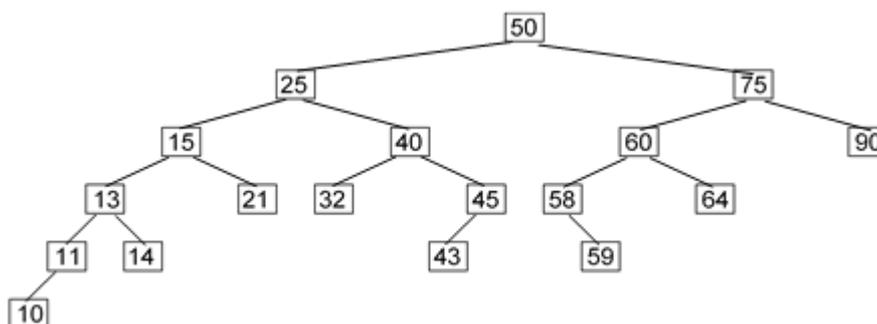
SE PIDE codificar un método booleano en C++ que, recibiendo como argumentos un *árbol binario de búsqueda* del tipo anterior y una clave, sustituya dicha clave por la clave menor de sus descendientes por la derecha. El método devolverá *false* si la clave no está o la clave no tiene descendientes por la derecha y *true* en caso de ser posible la sustitución.

OBSERVACIONES:

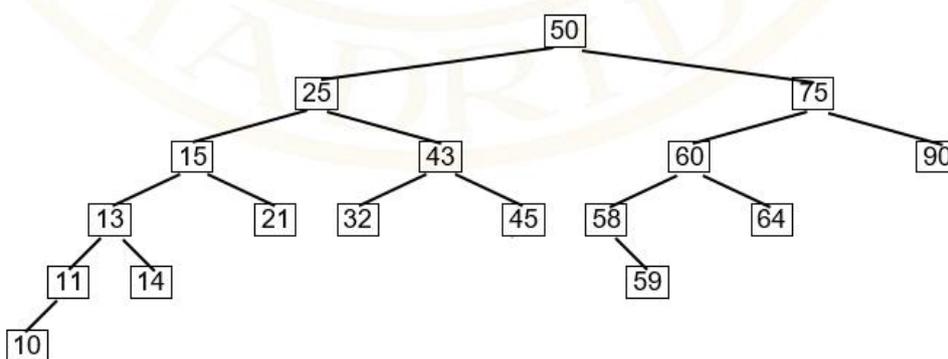
- El nodo cuya clave es la menor de los descendientes por la derecha debe ser eliminado.
- Se penalizarán las soluciones que no liberen la memoria.

EJEMPLO:

Dado el siguiente árbol y la clave 40,



el árbol devuelto sería:



NORMAS DE OBLIGADO CUMPLIMIENTO (su no observación supondría la calificación de 0 puntos):

- No se permite la utilización de ninguna estructura de datos auxiliar.
- Sólo se permite visitar cada nodo de árbol una única vez.
- Una vez finalizado el proceso, el resto del árbol deberá mantener intacta tanto su estructura como la información contenida

Al finalizar asegúrese de tener su *workbench (TallerED)* disponible para posteriores sesiones prácticas y en cualquier equipo que tenga instalado Eclipse.