Universidad de Granada
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIA
ETSII Informática y de Telecomunicación, C/ Periodista Daniel Saucedo Aranda s/n. 18071- Granada (España)

Estructuras de Datos
Curso 2017-2018. Convocatoria de Enero
Grado en Ingeniería Informática.
Doble Grado en Ingeniería Informática y Matemáticas

1. (0.5 puntos) Razonar la verdad o falsedad de las siguientes afirmaciones:

   (a) El orden en que las hojas se listan en los recorridos preorden, inorden y postorden de un árbol binario es el mismo en los tres casos.

   (b) Dado un árbol binario cuyas etiquetas están organizadas como un árbol binario de búsqueda, puedo recuperarlo a partir de su preorden.

   (c) Dado un árbol binario cuyas etiquetas están organizadas como un árbol parcialmente ordenado, puedo recuperarlo a partir de su postorden.

   (d) Es correcto en un esquema de hashing cerrado el uso como función hash de la función h(k)= [k + random(M)] % M, M primo y con random(M) una función que devuelve un número entero aleatorio entre 0 y M-1

   (e) Es correcto en un esquema de hashing doble el uso como función hash secundaria de la función h_0 (x)=[(B-1)-(x % B)] % B con B primo

2. (1.5 puntos) Supongamos que tenemos una clase **Liga** que almacena los resultados de enfrentamientos en una liga de baloncesto:

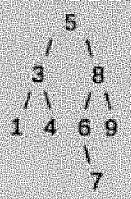
```
struct enfrentamiento{
  unsigned char eq1,eq2; // códigos de los equipos enfrentados
  unsigned int puntos_eq1, puntos_eq2; //puntos por cada equipo
};
class liga{
private:
  list<enfrentamiento> res;
...
};
```

   - Implementa un método que dado un código de equipo obtenga el número de enfrentamientos que ha ganado.
   - Implementa la clase iterator dentro de la clase Liga que permita recorrer los enfrentamientos en los que el resultado ha sido el empate. Implementar los métodos begin() y end().

3. (1 punto) Implementa una función **int orden (const list<int> &L);** que devuelva 1 si L está ordenada de forma ascendente de principio a fin, 2 si lo está de forma descendente y 0 si no está ordenada de ninguna forma.

Universidad de Granada
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIA
ETSII Informática y de Telecomunicación, C/ Periodista Daniel Saucedo Aranda s/n- 18071 - Granada (España)

4. (1 punto) Dado un árbol binario de búsqueda, implementa una función para imprimir las etiquetas de los nodos **en orden de mayor a menor profundidad**. Si tienen la misma profundidad pueden aparecer en cualquier orden. Ejemplo:



El resultado seria **7,1,4,6,9,3,8,5.**

5. (1 punto) Tenemos un contenedor de pares de elementos, {clave, bintree<int>} definida como:

```
template <typename T>
class contenedor {
    private:
      unordered_map<T, bintree<int> > datos;
    ..............
    ..............
}
```

Implementa un **iterador** que itere sobre los elementos que cumplan la propiedad de que la suma de los elementos del bintree<int> sea un número par. Debes implementar (aparte de las de la clase iterator) las funciones begin() y end().

6. (1 punto) Un "heap-doble" es una estructura jerárquica que tiene como propiedad fundamental que para cualquier nodo Z a profundidad **par** la clave almacenada en Z es **menor** que la del padre pero **mayor** que la del abuelo (cuando existen), y para cualquier nodo Z a profundidad **impar**, la clave almacenada en Z es **mayor** que la del padre pero **menor** que la del abuelo (cuando existen), siendo el árbol binario y estando las hojas empujadas a la izquierda. Diseña una función para insertar un nuevo nodo en la estructura y aplicarla a la construcción de un heap-doble con las claves {30, 25, 12, 16, 10, 15, 5, 18, 23, 32, 4, 17}.

Tiempo: 3 horas