PARTE 2

60 minutos - con material manuscrito propio

EJERCICIO 1

Se puede definir un árbol binario como completo como aquél cuyos nodos internos tienen sus dos hijos no nulos. Desarrollar un algoritmo (**método de árbol** y **método de nodo**), y analizar el orden del tiempo de ejecución, que aplicado sobre un árbol binario devuelva **VERDADERO** si es completo, y **FALSO** en caso contrario.

EJERCICIO 2

Un instituto de enseñanza mantiene una lista con todos sus alumnos, que se insertan en la lista a medida que se matriculan. El tipo de lista usada es de encadenamiento simple identificado cada nodo por la cédula del alumno tomada como un número entero, sin puntos ni guiones y con el dígito verificador.

Cada alumno está representado mediante un identificador único (su CI, numérico, sin código de verificación), su apellido, carrera y un campo adicional "matriculado" que indica si está actualmente registrado en al menos una asignatura de esa carrera. Es habitual emitir un listado de todos los alumnos.

Tipo Alumno
- Cedula: un entero
- Apellido: una string
- Carrera: una string
- Matriculado: un boolean

Tipo Nodo Lista Alumno
- Etiqueta: un entero (la cédula del alumno)
- Alumno: un Tipo Alumno
- Siguiente: un Tipo Nodo Lista Alumno

Por otra parte, dado que es frecuente la consulta de los datos de alumnos que **de una cierta carrera que efectivamente están matriculados en alguna asignatura**, por **apellido**, se ha decidido crear también un árbol binario de búsqueda que permita un acceso eficiente para satisfacer esta consulta.

Tipo Nodo Árbol Binario de Búsqueda
- Etiqueta: una string (el apellido del alumno)
- Alumno: un Tipo Alumno
- Hijo Izquierdo: un Tipo Nodo Árbol Binario de Búsqueda
- Hijo Derecho: Tipo Nodo Árbol Binario de Búsqueda

Escribir un método (y analizar el orden del tiempo de ejecución) que, dada una lista simplemente encadenada de alumnos, genere un árbol binario de búsqueda por Apellido con todos los alumnos de cierta carrera que estén matriculados, de acuerdo a la siguiente firma:

TipoLista.indizarPorApellido(de Tipo Árbol Binario de Búsqueda elIndice; tipo string laCarrera)

Nota: se deben implementar **todos** los métodos que se invoquen.