Examen Final de PRED - Junio de 2006

Pregunta 1 (8 puntos)

Una Plataforma de TV ha decidido ofrecer un servicio denominado TV_a_la_Carta. El servicio TV_a_la_Carta se caracteriza por el hecho de que sus abonados pueden contratar programas en lugar de canales para verlos a una hora determinada. Con esta facilidad, cada abonado puede crear su propia programación. En concreto, con este servicio están disponibles para los abonados las siguientes operaciones:

- `AltaProg` que, dado un abonado, una hora de emisión y un programa, lo incluye entre los programas contratados por el abonado. Si el abonado no esta en el sistema, lo incluye. Si el programa no existe o ya había sido contratado por el abonado a esa hora de emisiónno realiza ninguna modificación en la estructura de datos.
- `BajaProg` que, dado un abonado y un programa elimina el programa de los contratados por el abonado. Si el abonado o el programa no existen, no se modifica la estructura de datos.
- `ListarProgContratados` que, dado un abonado, lista en orden alfabético de programas todos los programas que tiene contratados junto con sus horas de emisión.

Se ha decidido implementar la abstracción TV_a_la_Carta utilizando una tablas de hash con encadenamiento para los abonados. En concreto, en la tabla de hash sobre el DNI de los abonados, vía hashing y recorriendo la lista de sinónimos correspondientes accederíamos a un objeto que contiene el nombre del abonado, su dni, y del que colgaría una lista de los programas que tiene contratados, ordenada en orden alfabético de programas. Adicionalmente, se supone que se tiene una tabla, tprog: **tabla** [1.. `MAX_P`] **de** Programa, que contiene la información sobre todos los programas existentes y que está ordenada en orden alfabético de programas.

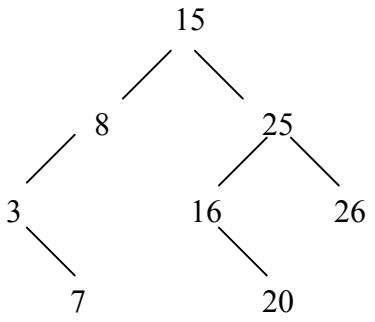
```
clase Abonados
      nombre: String;
      dni: String;
      lista_prog: ^Prog_contrat; /* lista de programas contratados */
      sig_abo: ^Abonado;     /* siguiente sinónimo */
f_clase

clase Prog_contrat
      p: Programa;
      hora_emision: entero; /* entre 1 y 168 (horas de una semana)*/
      sig_prog: ^Prog_contrat; /* siguiente programa contratado */
f_clase

clase TV_a_la_Carta

      t_prog: tabla[1..MAX_P] de Programa;
      t_abonados: tabla[1..MAX_ABO] de ^Abonados;

      privada función h_abo(k:String) retorna i: entero
      Pre: cierto
      Post: : 1 ≤ i ≤ MAX_ABO;
      /* función de hashing sobre abonados*/

f_clase
```

Se pide:

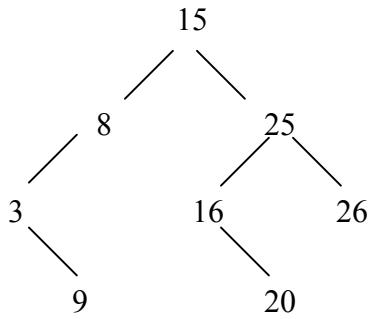
a) Especificar la implementación de la abstracción `TV_a_la_Carta`. Es decir, definir el invariante y la equivalencia de la representación.
b) Implementar la operación `AltaProg`. Justificar la corrección de la implementación de esta operación.
c) Supongamos que se desea añadir una operación que, para cada hora de emisión, nos escriba la lista de pares <programa, abonado> que incluya todos los programas contratados por los abonados que han de ser emitidos a esa hora, Esta lista ha de estar ordenada por orden alfabético de programas y, para cada programa, por orden del dni de los abonados. Se supone que las horas de emisión están entre 1 y 168 (número de horas de una semana). Explicar como se debería modificar la estructura de datos para implementar eficientemente esta nueva operación, describiendo también como habría que modificar el resto de las operaciones de la estructura.

Pregunta 2 (2 puntos)

Un árbol binario de búsqueda es un árbol binario con la propiedad de que, para cualquier subárbol, todos los valores que están en el hijo izquierdo del subárbol son menores o iguales que el valor que está en la raíz del subárbol y todos los que están en el hijo derecho son mayores estrictos que el que está en la raíz. Por ejemplo, dados los árboles



(a)



(b)

tenemos que el árbol (a) es un árbol binario de búsqueda. En cambio, el árbol (b) no lo es ya que el valor 9 está en el hijo izquierdo del subárbol que tiene el valor 8 en la raíz.

Se pide escribir el algoritmo que escribe la lista ordenada de los valores guardados en un árbol binario de búsqueda.