

## Examen Final de PRED (15 de junio de 2007)

1. (8 puntos) Se desea implementar la estructura de datos Info\_aerea que guarda información sobre vuelos entre ciudades. En concreto, se desea que la estructura tenga la siguiente operación de consulta:

- vuelos?, que dadas dos ciudades, Ciudad1 y Ciudad2, nos diga todos los vuelos que hay de Ciudad1 a Ciudad2 ordenados por la hora de salida del vuelo y, en caso que la hora de salida coincida, por el número de vuelo.

Se ha decidido implementar la abstracción Info\_aerea utilizando una tabla con hash abierto (en el libro de Ricardo Peña se denominaría tabla dispersa cerrada) sobre las parejas de ciudades. En concreto, dadas las ciudades Ciudad1 y Ciudad2, vía hashing encontraríamos una posición de la tabla donde guardaríamos la información asociada a Ciudad1 y Ciudad2. De esa posición debería de colgar la lista encadenada de vuelos que van de Ciudad1 a Ciudad2 ordenada según las horas de salida y, en caso de coincidencia, por orden del número de vuelos.

Es decir, tendríamos (entre otras) las siguientes declaraciones de clases:

```
clase vuelo /* información sobre un vuelo */
```

```
    num_vuelo: entero;  
    compañía: string; /* compañía aérea */  
    origen: string; /* ciudad de donde sale el vuelo */  
    destino: string; /* ciudad a la que llega el vuelo */  
    hora: entero; /* hora de salida */
```

```
f_clase
```

```
clase pos_hash /*posiciones de la tabla hash*/
```

```
    origen: string; /* ciudad origen */  
    destino: string; /* ciudad de destino */  
    vuelos: ^nodo_vuelos; /* puntero a la lista de vuelos que conecta la ciudad origen con la  
                           ciudad destino */
```

```
f_clase
```

```
clase nodo_vuelos /*nodos de las listas ordenadas de vuelos entre dos ciudades*/
```

```
    num_vuelo: entero;  
    compañía: string; /* compañía aérea */  
    hora: entero; /* hora de salida */  
    sig_vuelo: ^nodo_vuelos; /* puntero al siguiente vuelo en la lista */
```

```
f_clase
```

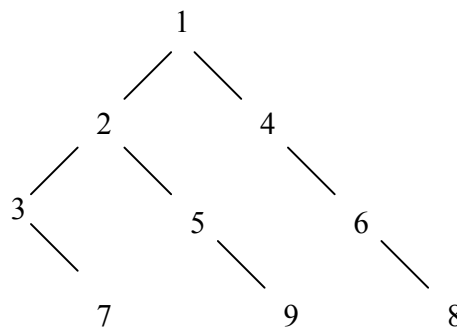
se pide:

a) Especificar la implementación de esta estructura de datos. Es decir, definir el invariante y la equivalencia de la representación.

b) Implementar la operación *añadir\_vuelo* que, dado un vuelo V nos lo añade a la estructura de datos. Si ya existiera un vuelo entre las dos ciudades de V que saliera exactamente a la misma hora y de la misma compañía aérea entonces lo único que hay que hacer es modificar el número de vuelo (substituyendo el número anterior por el de V). Se supone que la función de hash (llamada h) ya está implementada y se puede usar para hacer las inserciones en la tabla hash. Justificar la corrección de la operación.

c) Supongamos que queremos tener otra operación de consulta que, dada una compañía aérea, nos escriba la lista de todos los vuelos de esa compañía aérea, ordenados, en primer lugar, por la ciudad de origen del vuelo, en segundo lugar por la ciudad de destino del vuelo (en caso de que la ciudad de origen coincida) y, en tercer lugar, por la hora de salida del vuelo (en caso de que también coincidan las ciudades de destino). Decid (y justificad) cómo modificaríamos la estructura de datos para hacer que esta operación se pueda implementar eficientemente. Se supone que hay 100 compañías aéreas distintas.

2. (2 puntos) Dado el siguiente árbol binario



decid cuales de las siguientes secuencias de nodos corresponden a uno de los recorridos habituales (preorden, inorden, postorden o niveles):

- a) 1 2 3 4 5 6 7 8 9
- b) 1 2 3 7 5 9 4 6 8
- c) 1 4 6 8 2 5 9 3 7
- d) 3 2 7 5 9 1 4 6 8
- e) 7 9 8 3 5 6 2 4 1
- f) 3 7 2 5 9 1 4 6 8