

Algoritmos y Estructuras de Datos II

Recuperatorio del segundo parcial (extra) – 15 de diciembre de 2014

Aclaraciones

- El parcial es a **libro abierto**.
- Cada ejercicio debe entregarse **en hojas separadas**.
- Incluir en cada hoja el número de orden asignado, número de hoja, apellido y nombre.
- Al entregar el parcial, completar el resto de las columnas en la planilla.
- Cada ejercicio se calificará con **MB**, **B**, **R** o **M**, y podrá recuperarse independientemente de los demás. Para aprobar el parcial se puede tener hasta 1 (un) ejercicio **R** (regular), siempre que no se trate del ejercicio 1. Para más detalles, ver “Información sobre la cursada” en el sitio Web.

Ej. 1. Diseño

Debido a algunos incidentes ocurridos en la materia algoritmos y estructuras de datos II, el DC - FCEyN - UBA nos pide informatizar una pre-encuesta integrada con las famosas EADIS para que los docentes responsables de los parciales paguen en carne propia los daños provocados a los estudiantes cuando los exámenes no son apropiadas.

La etapa de especificación arrojó el siguiente TAD:

TAD PRE-ENCUESTA

observadores básicos

```

alumnos : pre-encuesta → conj (alumnos)
docentes : pre-encuesta → conj (docente)
exámenes : pre-encuesta → conj (examen)
responsable : examen  $e \times$  pre-encuesta  $p \rightarrow$  docente  $\{e \in \text{exámenes}(p)\}$ 
calificoADocente? : alumno  $a \times$  docente  $d \times$  pre-encuesta  $p \rightarrow$  bool  $\{a \in \text{alumnos}(p) \wedge d \in \text{docentes}(p)\}$ 
calificacionesDocente : docente  $d \times$  pre-encuesta  $p \rightarrow$  multiconj (nota)  $\{d \in \text{docentes}(p)\}$ 

```

generadores

```

crearEncuesta : conj(alumnos)  $\times$  conj(docente)  $c \times$  dicc(examen  $\times$  docente)  $d \rightarrow$  pre-encuesta  $\{(\forall e: \text{examen})(\text{definido?}(e, d) \Rightarrow \perp \text{obtener}(e, d) \in c)\}$ 
calificarDocente : alumno  $a \times$  docente  $d \times$  nota  $n \times$  pre-encuesta  $p \rightarrow$  pre-encuesta  $\{n > 0 \wedge (a \in \text{alumnos}(p) \wedge d \in \text{docentes}(p)) \wedge \neg \text{calificoADocente?}(a, d, p)\}$ 
reportarExamenInapropiado : alumno  $a \times$  examen  $e \times$  pre-encuesta  $p \rightarrow$  pre-encuesta  $\{a \in \text{alumnos}(p) \wedge e \in \text{exámenes}(p)\}$ 

```

otras operaciones

```

docentesNabo : pre-encuesta → conj (docente)

```

axiomas

```

 $(\forall d: \text{docente})(d \in \text{docentesNabo}(p) \iff \text{calificacionesDocente}(d, p) = \text{Ag}(0, \emptyset))$ 

calificacionesDocente(d, crearEncuesta(c, c', d))  $\equiv \emptyset$ 
calificacionesDocente(d, calificarDocente(a, d', n, p))  $\equiv$  if  $d = d' \wedge d \notin \text{docentesNabo}(p)$  then
    Ag(n, calificacionesDocente(d, p))
else
    calificacionesDocente(d, p)
fi

calificacionesDocente(d, reportarExamenInapropiado(a, e, p))  $\equiv$  if responsable(e, p) = d then
    Ag(0,  $\emptyset$ )
else
    calificacionesDocente(d, p)
fi

:

```

Fin TAD

TAD ALUMNO ES NAT, TAD DOCENTE ES STRING, TAD EXAMEN ES NAT, y TAD NOTA ES NAT.

Notar que los alumnos se identifican por número de libreta (Nat) y los docentes por nombre y apellido (String). Se sabe que el largo del nombre y apellido de los docentes está acotado por una constante.

Se nos solicita diseñar el TAD Pre-encuesta teniendo en cuenta los siguientes requerimientos:

- el costo temporal de las operaciones *calificarDocente*, *calificacionesDocente* y *docentesNabo* no puede depender de la cantidad de docentes, ni de la cantidad de alumnos (i.e. $O(1)$),
- la operación *responsable* debe tener complejidad temporal $O(\log(e))$, siendo e la cantidad de exámenes diferentes.

Se pide:

- (a) Diseñar una estructura para representar el TAD que cumpla con los órdenes de complejidad pedidos. Indicar en castellano el invariante de representación de la estructura propuesta, explicando para qué sirve cada parte, o utilizando nombres autoexplicativos.
- (b) Implementar los algoritmos correspondientes a las operaciones pedidas, respetando las complejidades temporales estipuladas.
- (c) Justificar claramente cómo y por qué los algoritmos, la estructura y los tipos soporte permiten satisfacer los requerimientos pedidos. No es necesario diseñar los módulos soporte, **pero sí describirlos, justificando por qué pueden (y cómo logran)** exportar los órdenes de complejidad que su diseño supone.

Ej. 2. Ordenamiento

Dados dos arreglos de números naturales sin repetidos, A y B , de tamaño n y m respectivamente, se desea encontrar un algoritmo que coloque en un tercer arreglo C la intersección ordenada entre ambos, cuya complejidad temporal en el peor caso sea $O(N \log(N))$, siendo N el máximo entre n y m .

- (a) Escriba un algoritmo para solucionar el problema modificando *Merge-sort*. Justifique la complejidad temporal del algoritmo propuesto.
- (b) Indique cómo otros algoritmos de ordenamiento y a qué costo podrían resolver el problema planteado. Mencione al menos 3 distintos.

Ej. 3. Técnicas algorítmicas

Una matriz A de tamaño $n \times n$, con $n = 2^k$, está *ordenada por cuadrantes* si la suma de los elementos del cuadrante 1 es menor a la del 2, la del 2 es menor a la del 3, y la del 3 es menor a la del 4 (con los cuadrantes ordenados en sentido horario, siendo el primero el que contiene a la casilla $(0,0)$), y a su vez cada uno de sus cuadrantes está ordenado por cuadrantes. Por ejemplo, la matriz siguiente se encuentra ordenada por cuadrantes.

$$\left(\begin{array}{cc|cc} 1 & 2 & 5 & 6 \\ 4 & 3 & 8 & 7 \\ \hline 13 & 14 & 9 & 10 \\ 16 & 15 & 12 & 11 \end{array} \right)$$

Suponga diseñado el módulo `Matriz(Nat)` con la operaciones $\bullet[\bullet,\bullet]$, $\#filas$ y $\#columnas$ en $O(1)$.

Se pide:

- (a) Escriba un algoritmo para determinar si una matriz está ordenada por cuadrantes, y calcule su complejidad temporal.
- (b) Justificar la complejidad temporal, que debe ser $O(n \times n)$ en peor caso.