Algoritmos y Estructuras de Datos II
Recuperatorio del segundo parcial – 1/12/2007

Aclaraciones

- El parcial **NO** es a libro abierto.
- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Incluya el número de orden asignado (léalo cuando circule la planilla), apellido y nombre en cada hoja.
- Al entregar el parcial complete los datos faltantes en la planilla.
- Cada ejercicio se calificará con B, R ó M. Una B no significa que el ejercicio está "perfecto", sino que cumple con los requisitos necesarios para aprobar. En los parciales promocionados se asignará una nota numérica más precisa a cada ejercicio.
- Para aprobar el parcial debe obtenerse B en ambos ejercicios. Un parcial se considera promocionado si está aprobado y su puntaje es 70 o superior.

Ej. 1. Diseño (50 puntos)

La empresa de telefonía española TeleExpoliación S.A. nos encargó la implementación de un software de registro de clientes que debe ser implementado siguiendo con fidelidad la siguiente especificación:

TAD TeleExpoliación

géneros teleExpo

generadores

crear	:		$\longrightarrow$	teleExpo	
agCliente	:	teleExpo t × nombre n × nat t'	$\longrightarrow$	teleExpo	$\neg\text{cliente}(t,n) \wedge (\forall n' : \text{nombre})(\text{cliente}(t,n') \Rightarrow \neg(t' = \text{telefono}(t,n')))$
llamar	:	teleExpo t × nombre n × nat	$\longrightarrow$	teleExpo	$\text{cliente}(t,n)$
crearGrupo	:	teleExpo × nombre	$\longrightarrow$	teleExpo	
agrupar	:	teleExpo t × nombre n × nat n'	$\longrightarrow$	teleExpo	$\text{cliente}(t,n) \wedge (n' \leq \text{grupos}(t) \wedge_{\mathsf{L}} \neg(n \in \text{integrantes}(t,n')))$

observadores básicos

cliente	:	teleExpo × nombre	$\longrightarrow$	bool	
consumo	:	teleExpo t × nombre n	$\longrightarrow$	nat	$\text{cliente}(t,n)$
telefono	:	teleExpo t × nombre n	$\longrightarrow$	teleExpo	$\text{cliente}(t,n)$
grupos	:	teleExpo	$\longrightarrow$	nat	
integrantes	:	teleExpo t × nat n	$\longrightarrow$	conj(nombre)	$n \leq \text{grupos}(t)$

otras operaciones

consumoDelGrupo	:	teleExpo t × nat n	$\longrightarrow$	nat	$n \leq \text{grupos}(t)$

axiomas .

...

Fin TAD

TAD Nombre **ES** tupla<NombrePila: String, Apellido: String>

Además, sabiendo que la longitud de los apellidos no está acotada, que la de los nombres sí lo está y que el tamaño de los grupos es de a lo sumo 5 integrantes, se nos imponen las siguiente restricciones sobre los órdenes de complejidad:

- **telefono** debe tener complejidad temporal $O(l \times log\ a)$,
- **consumoDelGrupo** debe tener complejidad temporal $O(l \times log\ a)$,

donde l es una cota para la longitud de los apellidos y a es la cantidad de apellidos de los clientes de la empresa.

i) Describir la estructura a utilizar, documentando claramente cómo la estructura resuelve el problema y cómo cumple con los requerimientos de eficiencia. El diseño debe incluir sólo la estructura de nivel superior. De ser necesario para justificar los órdenes de complejidad, describa las estructuras soporte. Si alguna de las estructuras utilizadas requiere que sus elementos posean alguna función especial (por ejemplo, comparación o función de hash) deberá escribirla.
ii) Escribir una versión en lenguaje imperativo del algoritmo **consumoDelGrupo**, indicando la complejidad de cada uno de los pasos.
iii) ¿Cómo cambia la solución propuesta (estructura, algoritmos y órdenes de complejidad) si los grupos se identificaran con el apellido y nombre de uno de los miembros?

Ej. 2. Ordenamiento (25 puntos)

Sea $f : R \to R$ una función que se sabe que tiene una única raiz en R_+ (los reales positivos). Escriba un algoritmo que retorne $\langle a, b\rangle$ tal que $b = a + 1$ y, o bien $f(a) > 0$ y $f(b) < 0$, o bien $f(a) < 0$ y $f(b) > 0$. Se pide además que el algoritmo tenga orden de complejidad $O(log\ b)$

Ej. 3. Eliminación de la recursión (25 puntos)

Dada la función $f(n) : secu(nat) -> tupla < natxnat >$ definida como sigue:

$$\begin{aligned} f(<>) &= \langle 1, 0\rangle \\ f(a.s) &= \langle a * \pi_1(f(s)), a + \pi_2(f(s))\rangle \end{aligned}$$

a) Indique qué tipo de recursión tiene la función f.
b) Aplicar la técnica de inmersión+plegado/desplegado para obtener una función que pueda ser transformada algorítmicamente a una forma iterativa. Indique claramente la signatura y la axiomatización completa de todas las funciones que introduzca.
c) A partir del resultado anterior genere un algoritmo imperativo con la versión iterativa de la función f.