Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Ciencia de la Computación
Estructuras de Datos y Algoritmos II

Nombre y Apellido: Legajo:

Examen Parcial 1

1. El TAD MultiMap es una colección de pares (clave, valor), donde una clave puede estar asociada a muchos valores. Los siguientes son ejemplos de MultiMaps: $\{(1,a),(2,b),(1,c),(3,a)\}$, $\{(1,3),(1,4),(2,6),(2,7)\}$.

i) Completar la siguiente especificación del TAD MultiMap con la especificación de las operaciones *deleteAll* y *asMap*.

a) Nombre de tipo: MultiMap
Usa: Bool, Set, List

b) Operaciones:

$$
\begin{array}{ll}
\textit{tad MultiMap } (K : \textit{Ordered Set}, V : \textit{Set}) & \\
\textit{empty} & : \textit{MultiMap } K\ V \\
\textit{put} & : \textit{MultiMap } K\ V \to (K, V) \to \textit{MultiMap } K\ V \\
\textit{size} & : \textit{MultiMap } K\ V \to \textit{Int} \\
\textit{values} & : \textit{MultiMap } K\ V \to k \to \textit{List } V \quad \text{-- lista los valores asociados a una clave} \\
\textit{minKey} & : \textit{MultiMap } K\ V \to \textit{List } V \quad \text{-- lista los valores asociados a la menor clave} \\
\textit{delete} & : \textit{MultiMap } K\ V \to (K, V) \to \textit{MultiMap } K\ V \quad \text{-- elimina un par (k,v)} \\
\textit{deleteAll} & : \textit{MultiMap } K\ V \to K \to \textit{MultiMap } K\ V \quad \text{-- elimina todos los valores asociados a una clave} \\
\textit{asMap} & : \textit{MultiMap } K\ V \to \textit{Set } K\ (\textit{List } V) \quad \text{-- devuelve un conjunto con las claves} \\
 & \quad \text{-- y sus valores asociados}
\end{array}
$$

c) Especificación algebraica.

$$
\begin{aligned}
\textit{size empty} &= 0 \\
\textit{size } (\textit{put } m\ (a,b)) &= 1 + \textit{size } m \\
\textit{values empty } k &= \textit{nil} \\
\textit{values } (\textit{put } m\ (a,b))\ k &= \textbf{if } a \equiv k \textbf{ then } \textit{cons } b\ (\textit{values } m\ k) \textbf{ else } (\textit{values } m\ k) \\
\textit{delete empty } (a,b) &= \textit{empty} \\
\textit{delete } (\textit{put } m\ (c,d))\ (a,b) &= \textbf{if } a \equiv c \wedge d \equiv b \textbf{ then } m \textbf{ else } \textit{put } (\textit{delete } m\ (a,b))\ (c,d) \\
&\vdots
\end{aligned}
$$

ii) Para implementar las operaciones del TAD de manera eficiente se utilizará la estructura de datos *Pairing Heaps*. Un *pairing heap* es un heap vacío o un par formado por un elemento y una lista de pairing heaps. El invariante que debe cumplir es que el elemento de la raíz es menor o igual a todos los elementos de sus hijos.

Para implementar *pairing heap* en Haskell definimos el siguiente tipo de datos:

$$\textbf{data } \textit{PHeaps } a = \textit{Empty} \mid \textit{Root } a\ [\textit{PHeaps } a]$$

y lo utilizamos para definir el siguiente tipo de datos para multimaps:

$$\textbf{type } \textit{MultiMap } a\ b = \textit{PHeaps } (a, [b])$$

Implementar las operaciones *minKey* y *values* del TAD, definiendo en Haskell las funciones:

$$
\begin{aligned}
&\textit{values} :: \textit{Ord } a \Rightarrow a \to \textit{MultiMap } a\ b \to [b] \\
&\textit{minKey} :: \textit{Ord } a \Rightarrow \textit{MultiMap } a\ b \to [b]
\end{aligned}
$$

La función *minKey* debe ser de orden $O(1)$.