

## Quinto Appello - 31 Gennaio 2013

Cognome:

Nome:

Matricola:

**Attenzione:**

Si consegna solo se la soluzione compila perfettamente e l'esecuzione rispetta i test forniti.

1. Sia  $Q$  una coda a priorità. Scrivere un metodo **static**  $\langle K, V \rangle \text{ Map} \langle K, V \rangle$  **minimiPQ** (**PriorityQueue** $\langle K, V \rangle Q$ ) che restituisce una mappa  $M$  costruita nel seguente modo: per ogni entry in  $Q$  di chiave  $k$ , la mappa  $M$  contiene una sola entry di chiave  $k$  il cui value è il minimo tra tutti i value delle entry che in  $Q$  hanno chiave  $k$ . Ad esempio, se  $Q = \{(Antonio, 70), (Antonio, 1), (Enrico, 1), (Marta, 21), (Enrico, 9), (Giovanni, 1), (Filippo, 1)\}$ , allora  $M = \{(Antonio, 1), (Enrico, 1), (Marta, 21), (Giovanni, 1), (Filippo, 1)\}$ .

Il contenuto di  $Q$  può anche essere perso.

*Questo esercizio va svolto all'interno della classe di test del pacchetto `priorityqueue`. La correttezza della propria soluzione può essere testata all'interno della stessa classe.*

2. Si implementi con una lista di nodi un ADT `BTContainer` che sia un contenitore di alberi binari. Il contenitore dovrà avere, oltre a `size()` e `isEmpty()`, i seguenti metodi:
- (a) `BinaryTree<E> insert(E elem)`: crea un albero binario avente l'elemento `elem` nella radice e lo inserisce nel contenitore.
  - (b) `Position<E> insertLeft(BinaryTree<E> T, Position<E> v, E elem )`: aggiunge un figlio sinistro, avente `elem` come elemento, al nodo `v` dell'albero binario `T`. Se non è possibile aggiungerlo, il metodo deve restituire `null`.
  - (c) `Position<E> insertRight(BinaryTree<E> T, Position<E> v, E elem )`: aggiunge un figlio destro, avente `elem` come elemento, al nodo `v` dell'albero binario `T`. Se non è possibile aggiungerlo, il metodo deve restituire `null`.
  - (d) `BinaryTree<E> remove()`: rimuove l'albero binario di altezza massima.
  - (e) `BinaryTree<E> remove(BinaryTree<E> T)`: rimuove l'albero binario `T` dal contenitore. Lancia un'eccezione nel caso `T` non sia nel contenitore.

**Attenzione: i metodi (d) e (e) dovranno avere complessità  $O(1)$ .**

*Questo esercizio va svolto all'interno del pacchetto `btcontainer` che contiene anche l'interfaccia dell'ADT. La correttezza della propria soluzione può essere testata utilizzando il file di test presente nello stesso pacchetto.*