Hash table implementation of a dictionary

Variables

hash-table: array of dictionaries[1] of size N
hash-function: function from keys to $[0, \ldots, N-1]$
size: integer
inv: the dictionaries *hash-table*$[0], \ldots,$ *hash-table*$[N-1]$ contain the items of the dictionary; the items in the dictionary *hash-table*$[i]$ all have a key k with *hash-function*$(k) = i$; *size* is the size of the dictionary

Initialization

for $i = 0, \ldots, N-1$
    hash-table$[i] \leftarrow$ empty dictionary
size $\leftarrow 0$

Algorithms

size():
  output: size of dictionary
return *size*

isEmpty():
  output: dictionary is empty?
return (*size* $= 0$)

findElement(*key*):
  input: key to be searched for
  output: element of item with *key* in dictionary; NO-SUCH-KEY if no such item exists
hash-value $\leftarrow$ *hash-function*(*key*)
return result of findElement(*key*) applied to the dictionary *hash-table*[*hash-value*]

insertItem(*key*, *element*):
  input: item to be inserted
  postcondition: item (*key*, *element*) has been inserted into dictionary
hash-value $\leftarrow$ *hash-function*(*key*)
apply insertItem(*key*, *element*) to the dictionary *hash-table*[*hash-value*]

remove(*key*):
  input: key to be searched for
  output: element of item with *key* in dictionary; NO-SUCH-KEY if no such item exists
  postcondition: item has been removed from dictionary
hash-value $\leftarrow$ *hash-function*(*key*)
return result of remove(*key*) applied to the dictionary *hash-table*[*hash-value*]

[1] These dictionaries are implemented by some data structure, for example a sorted sequence.