Bag

Documentation

Made by: Illia Takhtamyshev

Contents

Task 3
Bag type 3
Set of values 3
Operations 4
1. insertElem 4
2. removeElem 4
3. getFreq 5
4. getOcc1Els 5
Representation 6
Implementation 6
1. insertElem 6
2. removeElem 7
3. getFreq 8
4. getOcc1Els 8
Testing 9
1) Creating bag 9
2) insertElem 9
3) removeElem 9
4) getFreq 9
5) getOcc1Els 9

Task

Implement the bag type for integers. Represent the bag as a sequence of (element, frequency) pairs. Implement as methods:

1. inserting an element
2. removing an element
3. returning the frequency of an element
4. returning the number of elements which occur only once in the bag
5. printing the bag

Bag type

Set of values

Bag

$Bag.elems \in BagElem$
$BagElem = \{elem \in \mathbb{Z} \land freq \in \mathbb{N}\}$
$Bag.size \geq 0$
$Bag.occ1Els \geq 0$

Operations

1. **insertElem**

   Insert an element into the Bag. If the element is in the Bag and its frequency is 1, then decrease occ1Els by 1 or if its frequency is not 1, do nothing. After that increase element's frequency by 1. If the element is not in the Bag, create new pair (elem, freq) with frequency one, add it in the Bag and increase by 1 both size and occ1Els.

   Formally:

   A = (el : $\mathbb{Z}$, bag : Bag, isInBag : $\mathbb{L}$, occ1Els : $\mathbb{N}$, size : $\mathbb{N}$)
   Pre = (el = el' $\wedge$ bag = bag' $\wedge$ occ1Els = occ1Els' $\wedge$ size = size')
   Post = (el = el' $\wedge$
   (bag, isInBag, occ1Els, size) = $\text{SEARCH}_{i=1..size}$ (bag.elems[i].elem = el))

2. **removeElem**

   Remove an element from the Bag. If the element is in the Bag and its frequency is 1, then decrease both size and occ1Els by 1 and after that remove the element. If the element is in the Bag and its frequency is not 1, then decrease the frequency of the element and if after decreasing its frequency is equal to 1, then increase occ1Els by 1.

   Formally:

   A = (el : $\mathbb{Z}$, bag : Bag, isInBag : $\mathbb{L}$, occ1Els : $\mathbb{N}$, size : $\mathbb{N}$)
   Pre = (el = el' $\wedge$ bag = bag' $\wedge$ occ1Els = occ1Els' $\wedge$ size = size')
   Post = (el = el' $\wedge$
   (bag, isInBag, occ1Els, size) = $\text{SEARCH}_{i=1..size}$ (bag.elems[i].elem = el))

3. **getFreq**

Search for the element in the Bag and return its frequency.

Formally:

A = (el : $\mathbb{Z}$, bag : Bag, isInBag : $\mathbb{L}$, occ1Els : $\mathbb{N}$, size : $\mathbb{N}$, frequency : $\mathbb{N}$)
Pre = (el = el' ∧ bag = bag' ∧ occ1Els = occ1Els' ∧ size = size')
Post = (Pre ∧
(isInBag, frequency) = $\text{SEARCH}_{i=1..size}$ (bag.elems[i].elem = el))

4. **getOcc1Els**

Return the number of elements which occur only once
(frequency is 1) in the Bag.

Formally:

A = (bag : Bag, occ1Els : $\mathbb{N}$)
Pre = (bag = bag' ∧ occ1Els = occ1Els′)
Post = (Pre)

Representation

Bag = rec (elems, size, occ1Els)

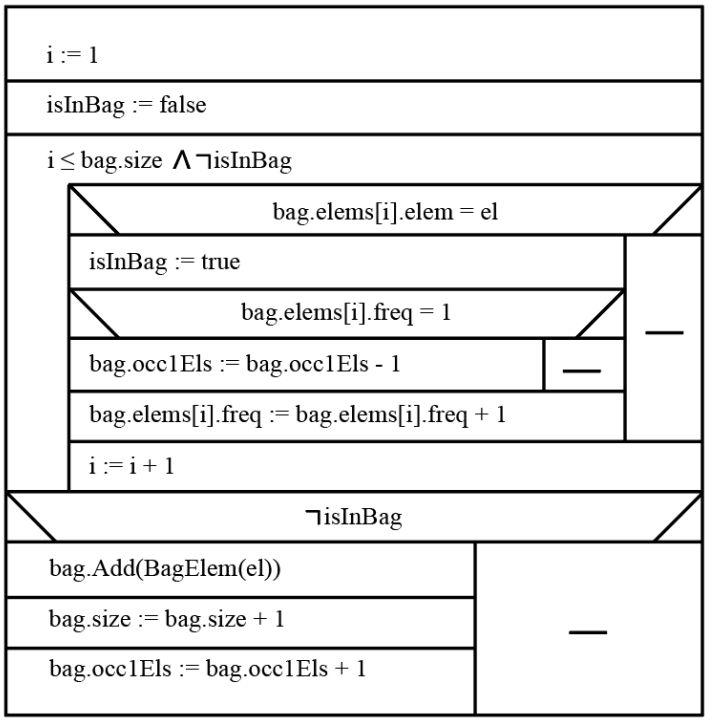
elems = list BagElem

BagElem = (elem : $\mathbb{Z} \wedge$ freq : $\mathbb{N}$)

Implementation

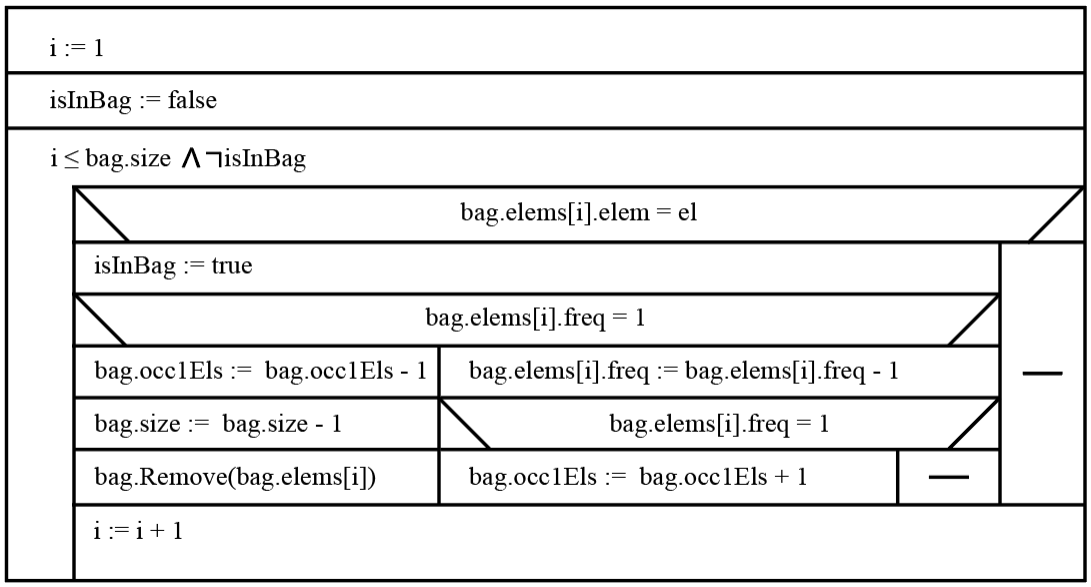
1. **insertElem**

   Insert an element into the Bag. If the element is in the Bag and its frequency is 1, then decrease occ1Els by 1 or if its frequency is not 1, do nothing. After that increase element's frequency by 1. If the element is not in the Bag, create new pair (elem, freq) with frequency one, add it in the Bag and increase by 1 both size and occ1Els.



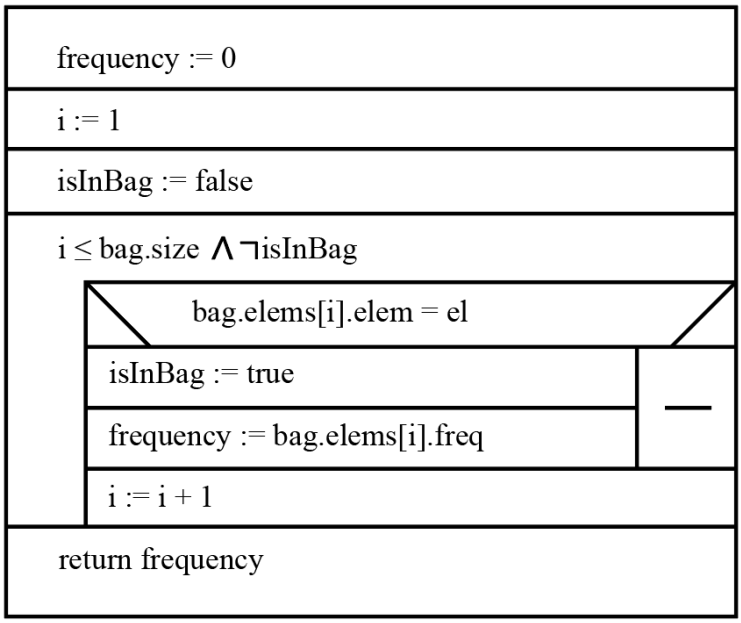
2. removeElem

Remove an element from the Bag. If the element is in the Bag and its frequency is 1, then decrease both size and occ1Els by 1 and after that remove the element. If the element is in the Bag and its frequency is not 1, then decrease the frequency of the element and if after decreasing its frequency is equal to 1, then increase occ1Els by 1.



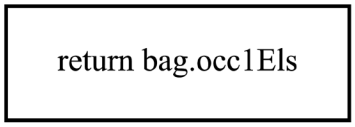
3. getFreq

Search for the element in the Bag and return its frequency.



4. getOcc1Els

Return the number of elements which occur only once (frequency is 1) in the Bag.



Testing

Testing the methods of the Bag

1) **Creating bag**
   a) Check that the size of the bag is not equal to 0

2) **insertElem**
   a) Insert elements (one time each) and check the size of the bag (expected size is 3)
   b) Insert element multiple times and check its frequency (expected frequency is 3)

3) **removeElem**
   a) Insert elements, then remove one of the elements one time and check its frequency (expected frequency is 1)
   b) Insert elements, then remove one that does not exist (expected error "cantFindElem")

4) **getFreq**
   a) Insert elements, get the frequency of one of the inserted elements (expected frequency is 3)
   b) Insert elements, get the frequency of element which is not in the bag (expected frequency is 0)

5) **getOcc1Els**
   a) Insert and remove elements, then get an amount of the elements which occur only once (expected return is 2)
   b) Insert and remove elements to get two times each element, then get an amount of the elements which occur only once (expected return is 0)