Documentation

16. *ADT Bag – implementation on a hash table, collision resolution by coalesced chaining*

Name: *Todoran Ana-Corina*

Group: *917*

Contents

1) ***ADT Specification:***

- **Bag** = { b | b – bag with elements of type TElement }

- **TElement** -> the general element in containers

  The interface of TElem contains the following operations:

  - assignment ($e_1 \leftarrow e_2$)
    - **pre**: $e_1, e_2 \in$ TElem
    - **post**: $e'_1 = e_2$
  - equality test ($e_1 = e_2$)
    - **pre**: $e_1, e_2 \in$ TElem
    - **post**:

$$equal = \begin{cases} \text{True, if e1 = e2} \\ \\ \text{False, otherwise} \end{cases}$$

- **Iterator =** { it | it – iterator over Bag }

2) *ADT Interface:*

a. *Bag:*

- ***init( ):***
  - **pre**: True
  - **post**: b ∈ Bag
  - **throws**: - (None)
- ***destroy(b):***
  - **pre**: b ∈ Bag
  - **post**: b is destroyed
  - **throws**: - (None)
- ***add(b, e):***
  - **pre**: b ∈ Bag **,** e ∈ TElement
  - **post**: $b' \in Bag, b' = b + \{ e \}$
  - **throws**: - (None)
- ***remove(b, e):***
  - **pre**: b ∈ Bag **,** e ∈ TElement
  - **post**: $b' \in Bag, b' = b - \{ e \}$
  - **throws**: - (None)
- ***size(b):***
  - **pre**: b ∈ Bag
  - **post**: size = the number of elements in b
  - **throws**: - (None)
- ***search(b, e):***
  - **pre**: b ∈ Bag **,** e ∈ TElement
  - **post**:

$$search = \begin{cases} \text{True, if e is in b} \\ \\ \text{False, otherwise} \end{cases}$$

  - **throws**: - (None)
- ***resize(b):***
  - **pre**: b ∈ Bag
  - **post**: $b' \in Bag$, rehash(b)
  - **throws**: - (None)

- ***rehash(b):***
  - **pre**: b ∈ Bag
  - **post**: b′ - bag, m′ = 2 * m
  - **throws**: - (None)
  -
- ***iterator(b, it):***
  - **pre**: b ∈ Bag
  - **post**: it ∈ Iterator, it – iterator over b
  - **throws**: - (None)

b. Iteraror:

- ***init(b):***
  - **pre**: b ∈ Bag
  - **post**: it ∈ Iterator , it – iterator over b pointing to ”first element”
  - **throws**: - (None)
- ***next(it):***
  - **pre**: it ∈ Iterator, it is a valid iterator
  - **post**: it′ - pointing to the next element
  - **throws**: - (None)
- ***valid(it):***
  - **pre**: it ∈ Iterator
  - **post**:

$$valid(it) = \begin{cases} \text{True, if it valid} \\ \\ \text{False, otherwise} \end{cases}$$

  - **throws**: - (None)
- ***getCurrent(it, e):***
  - **pre**: it ∈ Iterator
  - **post**: e ∈ TElement, e – the current element pointed by it
  - **throws**: - (None)
- ***begin(it, b):***
  - **pre**: b ∈ Bag
  - **post**: it ∈ Iterator , it – iterator over b pointing to first element
  - **throws**: - (None)

- ***end(it, b):***
  - **pre**: b ∈ Bag
  - **post**: it ∈ Iterator , it – iterator over b pointing to last element
  - **throws**: - (None)

3) *ADT Representation:*

a. ***Bag (Implemented on a hash table, collision resolution by coalesced chaining)***:

- length : Integer
- m (capacity) : Integer
- T : TElement[]
- next : Integer[]
- firstFree : Integer
- h : Tfunction

b. ***Iterator***:

- b : ↑Bag
- currentPos : ↑Node

4) *Problem Statement:*

Having a password database, memorised by it's password code, check if a given password exists. Password format consisting in numbers (each number has to have at least 4 digits and maximum 10). A password is not necessarly unique, so for the given password if it exists show also the number of apparitions.