Name: PLOSCAR SERGIU RAZVAN

Group: 812

Task: 30

Project Stage
Data Structures and Algorithms

Contents

1. Task

ADT Sorted Multi Map - implementation on a hash table, collision resolution by separate chaining

2. ADT Specification

- SMM= {smm| smm is a Sorted Multimap with pairs TKey, TValue, where we can define a relation R on the set of all possible keys}

- The general elements of the container are pairs of TKey, TValue
  The interface for TKey contains the following operations:

  - assignment ($k_1 \leftarrow k_2$)
    pre: $k_1, k_2 \in$ TKey
    post: $k_1' = k_2$

  - equality ($k_1 = k_2$)
    pre: $k_1, k_2 \in$ TKey
    post:

$$equal \begin{cases} True, if \ \ k_1 = k_2 \\ False, otherwise \end{cases}$$

  The interface for TValue contains the following operations:

  - assignment ($v_1 \leftarrow v_2$)
    pre: $v_1, v_2 \in$ TValue
    post: $v_1' = v_2$

  - equality ($v_1 = v_2$)
    pre: $v_1, v_2 \in$ TValue
    post:

$$equal \begin{cases} True, if \ v_1 = v_2 \\ False, otherwise \end{cases}$$

- Iterator = { it | it – iterator over Sorted Multimap }

3. ADT Interface

- Sorted MultiMap

  - init (smm, R)
    pre: R – relation on the set of all possible keys
    post: smm ∈ SMM, smm = ø

  - destroy (smm)
    pre: smm ∈ SMM
    post: smm was destroyed (allocated memory was freed)

  - add (smm, k, v)
    pre: smm ∈ SMM, k ∈ TKey, v ∈ TValue
    post: the pair <k,v> was added into smm

  - remove (smm, k, v)
    pre: smm ∈ SMM, k ∈ TKey, v ∈ TValue
    post: the pair <k,v> was deleted from smm (if it was in it)

  - search (smm, k, l)
    pre: smm ∈ SMM, k ∈ TKey, l ∈ L
    post:
    $\begin{cases} true\ and\ l\ is\ the\ list\ of\ values\ associated\ with\ c, if\ c\ is\ in\ smm \\ false\ and\ l = \emptyset\ otherwise \end{cases}$

  - iterator (smm, it)
    pre: smm ∈ SMM
    post: it ∈ Iterator, it is an iterator over smm

- Iterator

  - init (it, smm)
    pre: smm ∈ SMM
    post: it ∈ Iterator, it – iterator over smm pointing to first key

- next (it)
  pre: it ∈ Iterator, it is a valid iterator
  post: it' – pointing to the next element

- valid (it)
  pre: it ∈ Iterator
  post: $valid(it) = \begin{cases} True\ if\ it\ valid \\ False, otherwise \end{cases}$

- getCurrent (it, k)
  pre: it ∈ Iterator
  post: k ∈ TKey, k – the current key pointed by it

4. ADT Representation

- Node:
  - key: TKey
  - next: ↑Node

- HashTable:
  - T: ↑Node[]
  - m: Integer
  - h: TFunction

- Iterator:
  - smm : ↑SMM
  - currentPos: Integer

5. Problem Statement

You are put in charge of administrating an email database where each user owning an account can receive messages. You must make sure that all incoming messages reach their destination and that for any account all received messages are stored.

6. Problem Explanation

The hash function will associate the emails (pairs of email address and message – a string) to the corresponding position in the database (the hash table). Because an address can obviously receive more than one email, all emails received by an address will be stored in a linked list (separate chaining).