1. ADT – specification and interface

HMAP

Domain: HMP = {hmp I hmp is a hmap with elements key -> el, of type TKey -> TElem

Interface:

- init(hmp)

  DESCR

  Initialises a new empty hmap

  PRE

  True

  POST

  hmp is a valid hmap

- destroy(hmp)

  DESCR

  Destroy a hmap

  PRE

  True

  POST

  hmp was destroyed

- add(hmp, key, el)

  DESCR

  Adds a new element with a given key to the hmap

  PRE

  hmp is a valid Hmap, key is a valid TKey, el is a valid TElem

  POST

  HMP' = HMP + (key -> el)

- remove(hmp, key, el)

  DESCR

  Removes an element with a given key from the hmap

  PRE

  hmp is a valid HMap, key is a valid TKey, el is a valid TElem

  POST

  True if element was removed, False otherwise

- search(hmp, key)

  DESCR

  Searches an element with a given key in the map

  PRE

  hmp is a valid HMap, key is a valid TKey

  POST

  search <- the element if it is in the hmap, NULL otherwise

- size(hmp)

  DESCR

  Returns the number of key-value pairs from the hmap

  PRE

  hmp is a valid Hmap,

  POST

  An integer number is returned (representing the number of key-value pair from the hmap)

- values(hmp)

  DESCR

  Returns a bag with all the values from the hmap

  PRE

  hmp is a valid HMap

  POST

  keys <- B (which is a bag of all values from hmp)

HMAP ITERATOR

- init(hmp, it)

  DESCR

  Intialises the iterator

  PRE

  hmp is a valid HMAP

  POST

  IT is a valid iterator

- valid(it)

  DESCR

  Check if a given iterator is valid or not

  PRE

  POST

  valid <- True if it is a valid Iterator, False otherwise

- getCurrent(it)

  DESCR

  Gets the current element

  PRE

  it is a valid Iterator

  POST

  post <- the element from current position of the iterator

- next(it)

  DESCR

  Makes the 'iterator' to point to the next element

  PRE

  it is a valid Iterator

  POST

  Iterator will point to the next element from container

Representation:

HASH MAP

elems: DA<TElement>
next: DA<integer>
size: Integer
firstFree: Integer
hashFunction: TFunction

HASH MAP ITERATOR

hash_map: *Hash Map
currentPos: TPos

DA

cap: Integer
len: Integer
elems:*Telement[]

DA ITERATOR

da: *DA
currentPos: TPos

Statement of the problem

Johnie drew on a map N points (in a cartesian coordinate system). He now ask himself how many squares can he draw using those points (as the corners of the square).

(http://www.infoarena.ro/problema/patrate3)