EADS LAB 2 DOCUMENTATION

Sai Koushik Neriyanuri (317100)

14th December 2022

This program is an implementation of a doubly-linked ring using templates. It consists of class Ring and class iterator defined inside class ring. Below, I've written down description of each method defined in the class.

Class Ring methods:

- Ring() – The constructor. It initialises the head pointer to nullptr
- ~Ring() – The destructor. It deletes all elements of the ring and initialises the head pointer to nullptr
- Ring(Ring& source) – The copy constructor. It copies all the contents of source to our object (providing that source is not empty) using operator=
- operator= - It initialises our object to source. If source is empty, it just returns our object as it was
- empty – Returns true if the ring is empty (if head pointer == nullptr)
- size – Returns the number of elements in the ring
- print – prints all the elements of the ring on the screen in a readable manner
- clear – removes all elements from the ring using popBack
- pushFront – Inserts an element with given key and info at the front of the ring
- popFront – Removes the first element of the ring and initialises head to the next element
- pushBack – Inserts an element with given key and info at the back of the ring
- popBack – Removes the last element of the list
- insertAfterSearchedKey – Inserts an element with given key and info after an element containing a specified key, but only if it's a specified number of occurrence of that key in the ring (starting from the head).

Edge cases:

  - ➢ If the ring is empty or the requested number of occurrences of searched key <= 0, the function returns false
  - ➢ If the searched key is not found even once, the function returns false
  - ➢ If the searched key is found, but not the desired number of times, the user is asked if he would like to insert the new key after the last occurrence of searched key. The decision is up to him

- removeSearchedKey – Removes an element containing a specified key from the ring, but only if it's a specified number of occurrence of that key in the ring (starting from the head).
  Edge cases:
  - ➢ If the ring is empty or the requested number of occurrences of searched key <= 0, the function returns false
  - ➢ If no occurrences of the searched key are found, the function returns false
  - ➢ If the searched key is found, but not the desired number of times, the user is asked if he would like to remove the element containing the last occurrence of the key. The decision is up to him
- insertAfterEachOccurrence – Inserts an element with given key and info after each element containing a specified key.
  Edge cases:
  - ➢ If the ring is empty, the function returns false
  - ➢ If no occurrences of the searched key are found, the function returns false
- removeEachOccurrence – Removes all elements which contain the specified key.
  Edge cases:
  - ➢ If the ring is empty, the function returns false
  - ➢ If head pointer needs to be removed, popFront is used. Then, the new head is checked and also removed using popFront if necessary and so on, until the ring is empty, or the new head doesn't need to be removed. In that case, the rest of the elements is checked as usual

  - ➢ If no occurrences of the searched key are found, the function returns false

- insertAfterNthElement – Inserts an element with given key and info after an element which is on a specified position in the ring (starting from the head).
  Edge cases:
  - ➢ If the ring is too small for the element on the specified position to exist, the function returns false
  - ➢ If the specified position is less than zero, the function returns false
  - ➢ If the specified position is equal to 0, the element is inserted at the front of the list using pushFront
- removeNthElement – Removes an element which is on a specified position in the ring (starting from the head).
  Edge cases:
  - ➢ If the ring is too small for the element on the specified position to exist, the function returns false
  - ➢ If the specified position is less than or equal to zero, the function returns false
  - ➢ If the specified position is equal to 1, the first element of the ring is removed using popFront
- adjustInfo – Searches for an element containing a specified key and alters info of this element, but only if it's a specified number of occurrence of that key in the ring (starting from the head).
  Edge cases:
  - ➢ If the ring is empty or the requested number of occurrences of searched key <= 0, the function returns false
  - ➢ If the searched key is not found even once, the function returns false
  - ➢ If the searched key is found, but not the desired number of times, the user is asked if he would like alter the info at the last occurrence of searched key. The decision is up to him

Class Iterator methods:

- Iterator() – The constructor of Iterator. Initialises iterator to nullptr

- ~Iterator() – The destructor of Iterator. Initialises iterator to nullptr
- pointIteratorToHead – Sets iterator to point to the head of the ring
- pointIteratorToRear – Sets iterator to point to the last element of the ring
- setIteratorToSearchedKey - Sets iterator to point to an element containing a specified key (to the first such element met, starting from the head)
- SetIteratorToNthNode - Sets iterator to point to an element on specified position (starting from the head)
- insertAfterIterator – Inserts an element containing given key and info after the element that the iterator points to
- removeAfterIterator – Removes an element after the element that the iterator points to
- copyElementsToRing – A method that is called in function split, but can also be used independently. It copies elements from one ring (starting from an element that the iterator points on) to another (to the back of this ring, using pushBack). The number of copied elements is also chosen by the user