Wenlan Tian

COP 3530

9/11/2014

Exercise1

Section 1087, MAEB211

Design an array-based, circular-buffer-based Queue, represented by a single array and two ints (that represent the head & tail indices)

* Answer these questions

- How to (easily) tell when the queue is empty?

Initialize head to -1 and tail to -1. When dequeue, consider a situation that head equals to tail (size ==1), after dequeue, reset head and tail to -1. So we can tell the queue is empty when head and tail equals to -1.

- How to (easily) tell when the queue is full?

Two possibilities:

(1) when head is smaller than tail -> when head ==0, and tail ==maxSize -1;

(2) when tail is smaller than head -> when head-tail =1;

- How to (easily) tell when the queue’s current size?

Two possibilities:

1. when head is smaller than tail -> tail –head +1
2. when tail is smaller than head -> maxSize-(head-tail-1);

- What is the process for enqueueing?

When the queue is empty, the first item is inserted into array[0], after that, when enqueue an item, if the queue is not full: If the next array index does not equal to the max size of the array, move the tail to the next array index; If the next array index equals to the max size of the array, enqueue to array[0], and change the tail to 0.

- What is the process for dequeueing?

When the queue is not empty, if the next array index does not equal to the max size of the array, dequeueing by move the head to next array index; if it equals to the max size of the array, move the head to array[0]. If current array size if 1, change head and tail to -1.

* Write pseudo-code for each function

Class Queue

head, tail, sizeN, array[maxSize]

fuction: size(); isEmpty();full();enqueue();dequeue();print

isEmpty()

if head and tail equals to -1

full()

if head equals 0 and tail equals maxSize-1, or head is one larger than tail

size()

one situation: head < tail, size = tail - head + 1

another situation: tail < head, size = maxSize –(head – tail -1)

enqueue()

if full, return;

if empty, head =0;

if tail < maxSize-1, insert item to array[tail+1], tail++;

if tail = maxSize+1,set tail to 0, insert item to array[0];

dequeue()

if empty, return;

if head == tail, set head and tail to -1;

if head < maxSize-1, head++;

if head = maxSize-1, set head to 0;

* Implement (and test) in C++