Queue

Arnab Ganguly, Assistant Professor
Department of Computer Science, University of Wisconsin – Whitewater
Data Structures (CS 223)

1 Queue

A queue is a collection of items (integers, char, float, etc.) which allows the following two operations:

- $enqueue(i)$: adds an item i to the collection, and
- $dequeue()$: returns and removes the OLDEST item, provided the queue is not empty

Besides the above two operations, a queue may also support the following additional operations:

- $peek()$: returns the oldest item from the collection without removing it,
- $size()$: returns the number of items in the queue, and

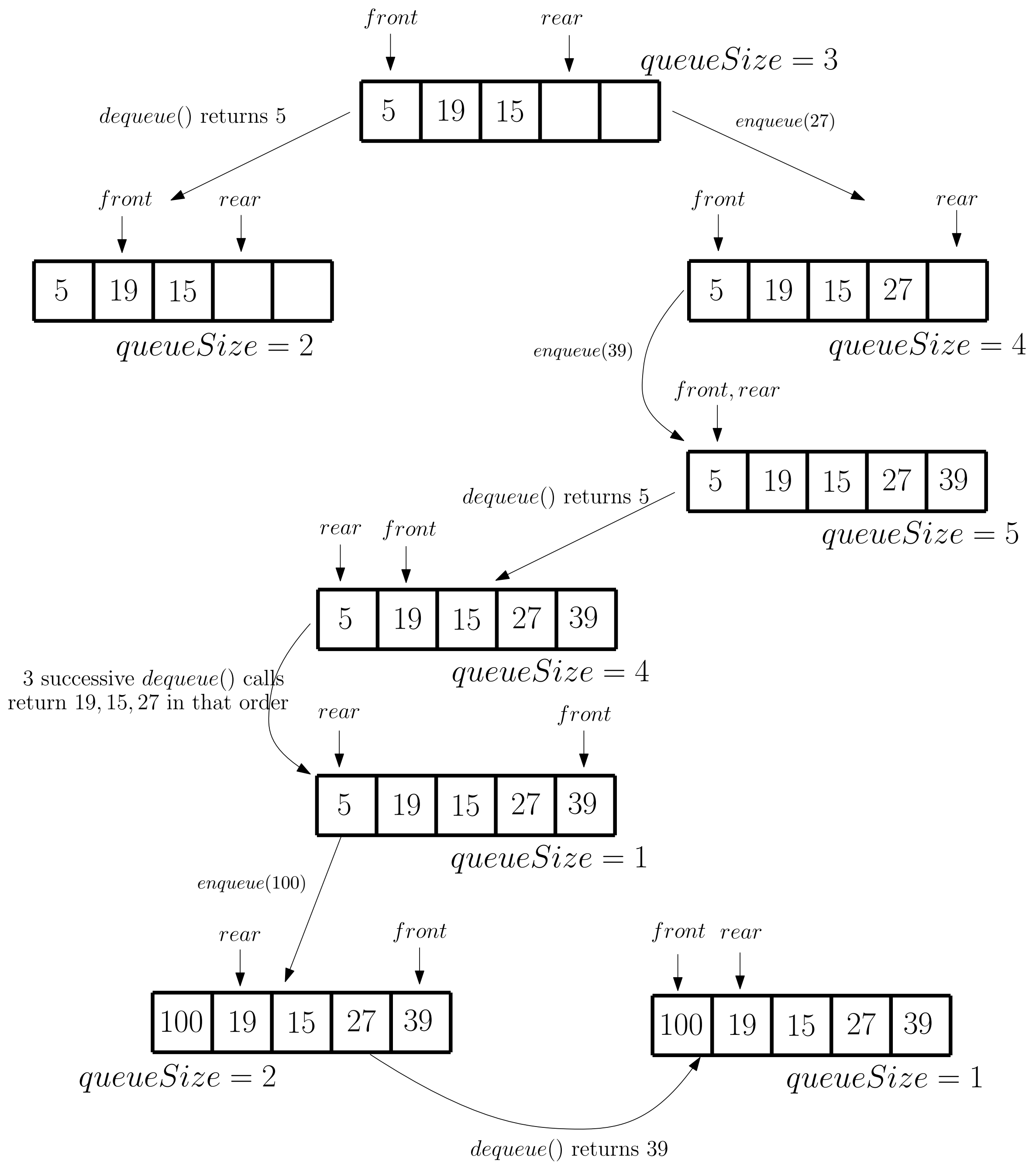

Figure 1: Queue: array that simulates it, $front$, and $currentSize$ variables. In this example, we have used $MAX\_SIZE = 5$.

Algorithm 1 Implementation of a Queue

```
int $MAX\_SIZE = 5, queueSize = 0, front = 0, rear = 0$;
int $queueArray[MAX\_SIZE]$; // an array which simulates the queue

function ENQUEUE(int val)
    if ($queueSize == MAX\_SIZE$) then
        "Cannot enqueue! Queue is full.";
    else
        $queueArray[rear++] = val$;
        $queueSize++$;
        if ($rear == MAX\_SIZE$) then
            $rear = 0$;

function DEQUEUE()
    if ($queueSize == 0$) then
        "Cannot dequeue! Queue is empty.";
    else
        int $value = queueArray[front++]$;
        $queueSize--$;
        if ($front == MAX\_SIZE$) then
            $front = 0$;
        return $value$;

function PEEK()
    if ($queueSize == 0$) then
        "Cannot peek! Queue is empty.";
    else
        return $queueArray[front]$;

function SIZE()
    return $queueSize$;
```