202: Computer Science II

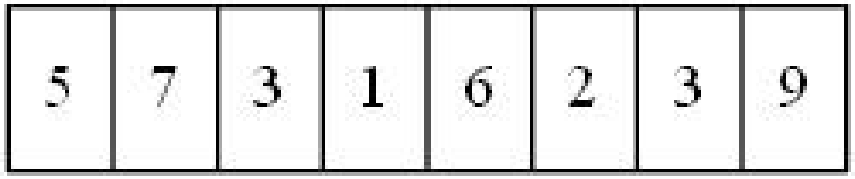
Northern Virginia Community College

Queues

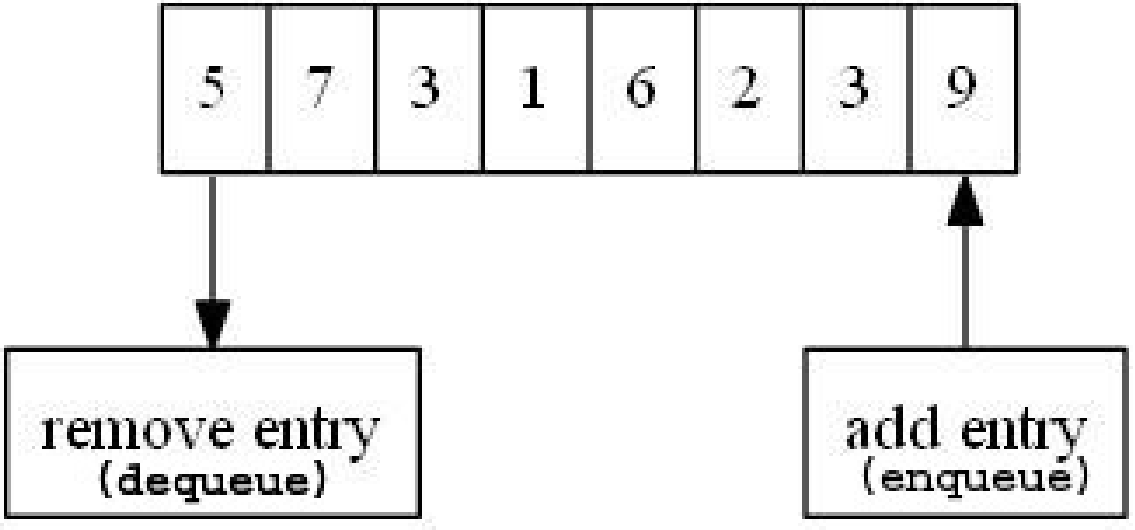
Cody Narber, M.S.
September 30, 2017

Data Structures - Queue

Queue: is a data structure that is expandable in size. Below shows a queue constructed with the data: 5,7,3,1,6,2,3,9



To modify the queue, you can only access/remove the first element and add new elements to the end. Known as **FIFO** (first in first out)



Data Structures - Queue

A **Queue**'s only restriction is that it has two methods:

- **enqueue** - adds an element to the back of the queue (cause a problem if bounded)
- **dequeue** - takes an element off the front of the queue (cause a problem if empty)

Using the above methods how can we?

- Access/Peek at an element
- Remove/insert an element
- Reverse a Queue

How long would those methods take (Big-O)?

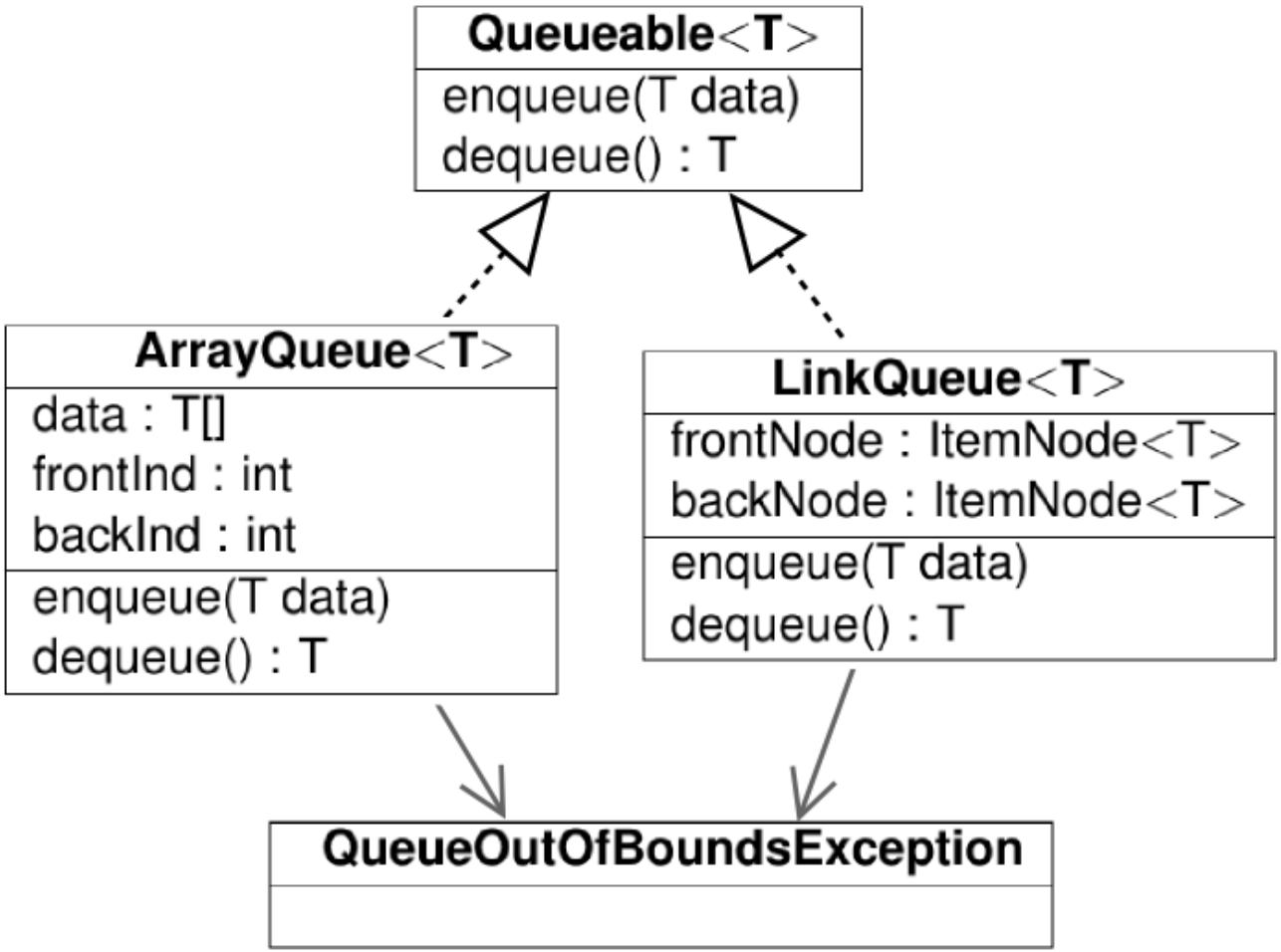
What would the result on the following queue be given the following operations?

- enqueue 5
- dequeue
- dequeue
- enqueue 3
- dequeue

9	2	4	1

Queue Framework

Let's create a set of queue classes:



Queue Applications

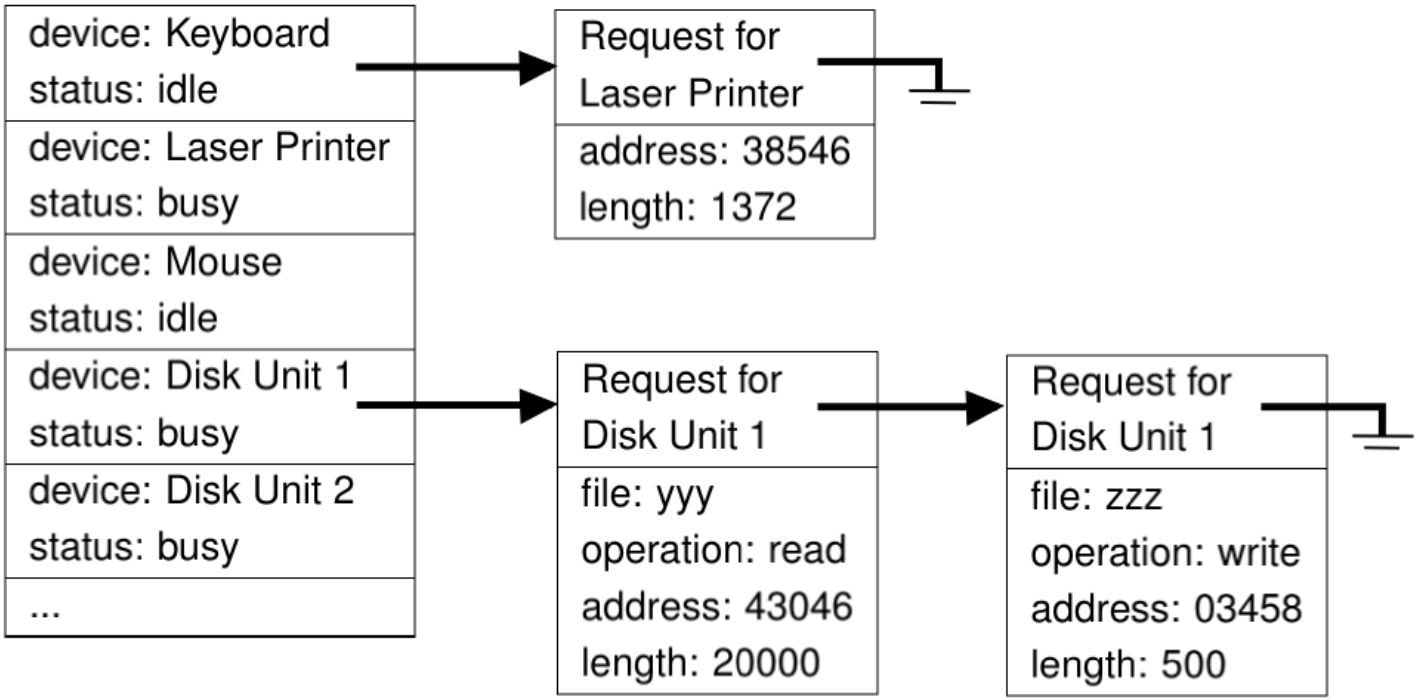
Queues are useful to organize ordering in order to be served/processed. Used extensively for systems/programs that involve buffers, and when resources are shared among consumers (task scheduling).

Example applications:

- **CPU Core thread assignment/processing**
- Hard disk scheduling
- Print Jobs
- **I/O Buffer**
- ...

I/O Device-Status Table

Device-Status Table is a table containing an entry for each device, where each entry indicates the device's type, address, and state (functioning/idle/busy). As multiple requests can be made to a device a **wait queue** will maintain a list of waiting requests when the device is busy.



Process

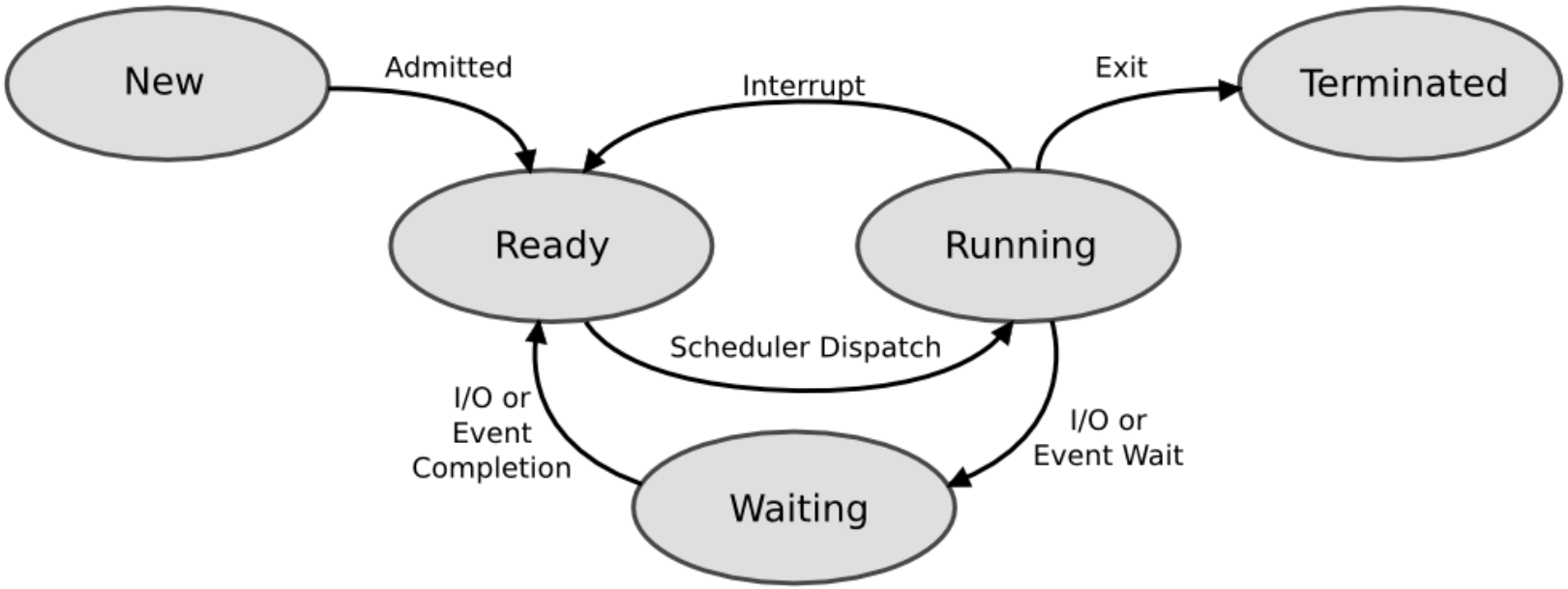
Process – is informally a program during execution (**Active**), however, a program (**Passive**) is more than just program code:

<table>
<tr><th>Process</th></tr>
<tr><td>Text Section (Instructions)</td></tr>
<tr><td>Program Counter</td></tr>
<tr><td>Process Stack:
<table>
<tr><th>Parameters</th><th>Return Addr</th><th>Local Variables</th></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
</table>
</td></tr>
<tr><td>Data Section (global variables)</td></tr>
<tr><td>Heap (Dynamically Allocated)</td></tr>
</table>

Process State

Process State – is the current activity of the Process.

- **New:** The process is being created
- **Running:** Instructions are being executed
- **Waiting:** The process is waiting for some event to occur
- **Ready:** The process is waiting to be assigned to a processor
- **Terminated:** The process has finished execution



Process Scheduler

In order to maximize CPU utilization, time sharing must switch the CPU/Core among processes so frequently so that users can interact with each program seemlessly. A **Process Scheduler** selects an available process for program execution on the CPU. As processes enter the system they will be added to a **Job Queue**, which consists of all processes on the system. The processes that are ready and waiting to execute are kept on the **Ready Queue**.