

V. Queues

[Most of the details about queues are left for to read about and work out in Lab 6.]

Def. As a data structure, a **queue** is an ordered collection of data items with the property that items can be removed only at one end, called the *front* of the queue, and items can be added only at the other end, called the *back* of the queue. Basic operations are:

construct: Create an empty queue
empty: Check if a queue is empty
addQ: Add a value at the back of the queue
front: Retrieve the value at the front of the queue
removeQ: Remove the value at the front of the queue

Whereas a stack is a Last-In-First-Out (LIFO) structure, a queue is a _____ or _____ structure.

2. Examples:

a. I/O buffers: queues, scrolls, dequeues

Æ From a file: (*queue*) p. 128

Æ Interactively: (*scroll* — _____) p. 219

Æ Screen handling: (*deque* — _____) p. 219

b. Scheduling queues in a multi-user computer system:

Æ Printer queue: When files are submitted to a printer, they are placed in the printer queue. The printer software executes an algorithm something like:

```

for (;;)
{
    while (printerQueue.empty())
        sleep 1;
    printFile = printerQueue.removeQ();
    Print(printFile);
}
  
```

Æ Other Queues:

Resident queue: _____

Ready queue: _____

Suspended queue: _____

c. CPU Scheduling: Probably uses a priority queue: _____

(Usually a new item is inserted behind those with the same priority.)