Thursday, April 5

Warmup
Evaluate the following expression using a stack that may contain real numbers. You MUST show your work using a stack.

\[ 3 \times 2 \times 7 - 4 \times 5 + 4 / 2 \]

Answer is 10, but 7 and 6 are also on the stack
Adding another * before 5 would get 42 and 10 on the stack

Queues (chapter 7)
classic data structure accessed from two ends
first-in, first-out - FIFO
Java (and C++) have a structure called a Deque (p. 399 - optional)
add and remove from either end
we are not using that

Queue operations:
add (E item) adds an item to the rear
(traditionally called enqueue)
E remove () removes and returns the front item
(traditionally called dequeue)
boolean isEmpty () whether queue is empty
int size () how many items it contains
E peek () value of the first entry

Uses:
- copying a word or series in order received
- simulating real-time activities (traffic intersections, waiting in line)
- input/output buffering (hold info in a buffer/queue until it's needed)
- processes in computer
- some card games (war)
- palindromes (p. 366)

A man, a plan, a canal – Panama!
Able was I ere I saw Elba
(text claims analysis of genetic material as a use of palindromes)
convert to lower case if an alphabetic character

// pseudo-code - not quite Java
while (ch is input) {
  if (Character.isLetter(ch)) { // Java uses Unicode
    ch = Character.toLowerCase(ch);
    s.push(ch);
    q.add(ch);
  }
}
while (!q.isEmpty( )) {
  while (!q.isEmpty( ) && ((s.pop( )).equals(q.remove( )))) {
    // sCh = s.pop( );
    // qCh = q.remove( );
    // stop if sCh != qCh
  }
// can use stack to check balance of parentheses
Implementation:

array

Queue implemented with an array
pop at data[first]
push at data[next(last)]
next(i) gives next index wrapping around to beginning

linked list

Queue implemented with a linked list
pop at front
push at rear