Announcements

- Program due Sat.
- Next HW up tomorrow, due next Friday
- Midterm 1 will be the following week (Feb. 28 - March 4)
- Office hours: after class today and tomorrow 2-3pm

(let me know if you plan to come!)
SLinkedList (const SLinkedList & other) {
    SNode<Obj> * current, prev, temp;
    temp = other._head;
    _head = NULL;
    while (temp != NULL) {
        current = new SNode<Obj>;
        current->_elem = temp->_elem;
        if (_head == NULL)
            _head = current;
        else
            prev->_next = current;
        temp = temp->_next;
        prev = current;
        prev->_next = NULL;
    }
}
Doubly Linked Lists

[Diagram of a doubly linked list node with fields `prev`, `elem`, and `next`]

Que we

A line

First in, first out

(A list, with specific functionality)
Alright – let’s think about the setup:

```cpp
template <typename Object>
class Queue {
public:
    int size() const;
    bool isEmpty() const;
    const Object& top() const;
    void enqueue(Object obj);
    Object dequeue();
};
```
How to implement?

- linked version
  (doubly linked list, or circularly linked list)

- array

Private data:

- Object * - O
- int i
- int size
- int capacity

(in constructor, we'll use new to make an array)
Wrapping Around:

\[ b = (f + \text{size}) \mod \text{capacity} \]
Two options:

- A lot of if statements

- Modular arithmetic: remainders

\[ 4 \mod 3 = 1 \]
\[ 56 \mod 6 = 2 \]
\[ 3 \mod 4 = 3 \]
Pseudocode

isEmpty():
    return (size == 0)

size():
    return -size
Constructor:

Array Queue(int cap = 1000) {
    -size = 0;
    -capacity = cap;
    -f = @0;
    -Q = new Object [capacity];
}

    top ()
        return Q[f]
void push (Object & element) { 
    if (_size == _capacity) 
        throw runtime_error(...)
    _Q[(f + _size) % _capacity] = element;
    _size++;
}
void pop()
{
  if (_size == 0)
    throw error

  f = (f + 1) % capacity;

  _size = --f;

  Object top() {
    return *f;
  }
}
Housekeeping Functions

- Copy Constructor
- Destructor
- Operator = (basically same as Array Start)
Actual code
(on webpage or in text)