CS 2100

Queues

Recap

- Reading due Fri (by 2pm)
  (Lists)
- Lab tomorrow: Zybooks on stacks
- Next HW: due on Fri. 22nd
  (HW4 - stacks)
  due via git (more this Fri.)
- HW3 - due this Sat. on ZyBooks
- First midterm: Week of Feb. 24th
  ↳ up through Queues/Stacks (+beginning of lists/vectors)

Last time: LIFO

Stacks: Last in, first out

Very simple – only 5 functions (+ only 1 way to alter):

top of stack: top
push
pop

2 versions:

Linked vs. Array

Why?? Useful → fast!

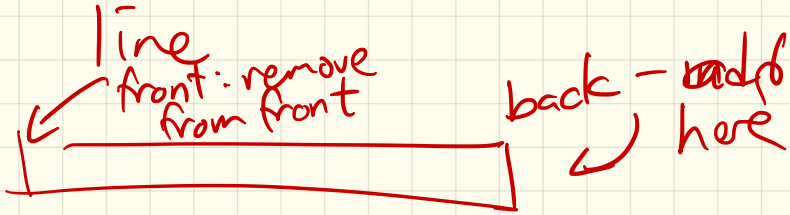
Runtime of stack operations

O(1) for all

(except housekeeping)

Today: Queues

British for what?



Key: first in, first out

FIFO

Again, will be light weight & simple.

Functions:

push

pop

front

empty

size

Behavior: (STL-style)

```
#include <queue>
using namespace std;
int main() {
    queue<float> myQ;
    myQ.push(10.2);
    myQ.push(16.5);
    myQ.push(2.6);
    cout << myQ.front() << endl;
    myQ.pop();
    cout << myQ.front() << endl;
}
```

myQ: 10.2, 16.5, 2.6
front
back

10.2

16.5

Setup + structure

This is also a simple data structure:

- limited functionality:

- but fast

O(1) for everything

Operations:

(see Cplusplus page)

Implementation

2 options:

1. Use linked list.

   For this version, I opted to rebuild, instead of using SLinkedList.

Also: helper functions!

private: copyFrom + deleteAll

Why?

Housekeeping:

Destructor: deleteAll

Copy constructor:
copyFrom(other)

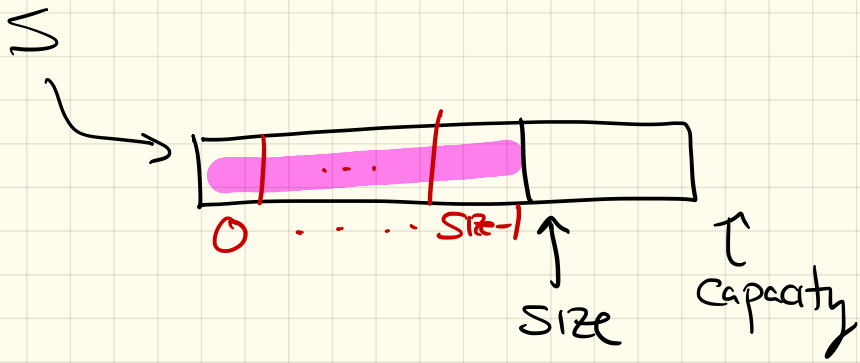
operator=:
deleteAll
copyFrom(other)

② Array version:

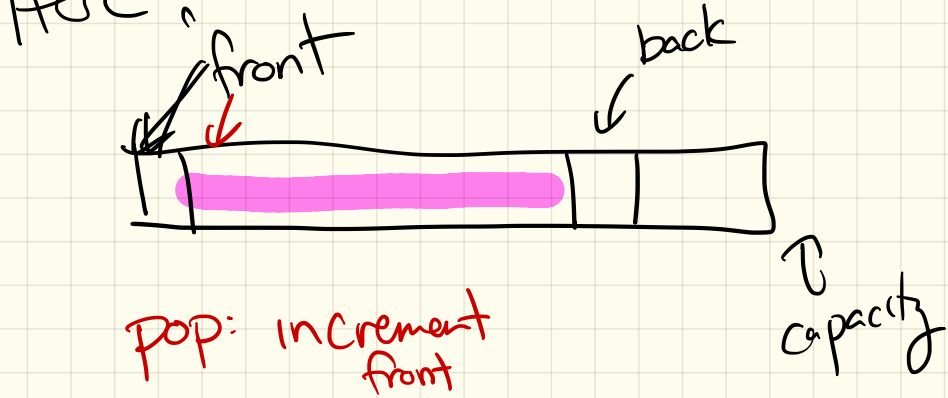
- private variables for stack were.

```
int size;
int capacity;
T* S; // pointer to array
```

Issue:



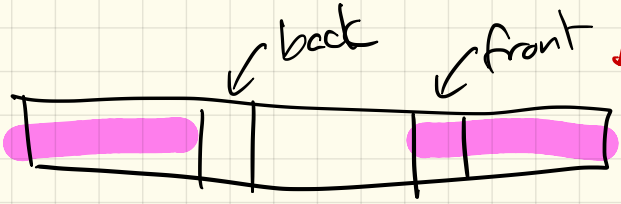
Here:



pop: increment front

Problem: more push/pops

even more...



push: back++

||

back=
(back+1)
%cap;

How to handle?

modulo ↔ remainder

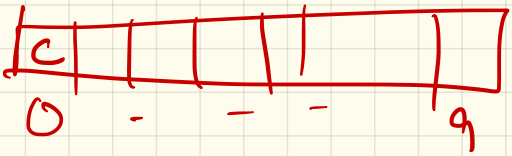
6 mod 4 = 2

18 mod 4 = 2

20 mod 4 = 0

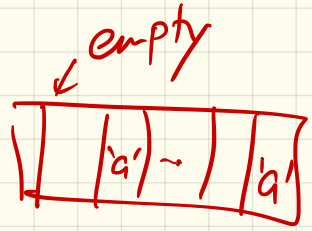
↔ %

cap = 10

Q: 

front = 0

back = 0



10

push('c')

Q[back] = 'c'

back ++;

back = 0

back = (back+1) % capacity