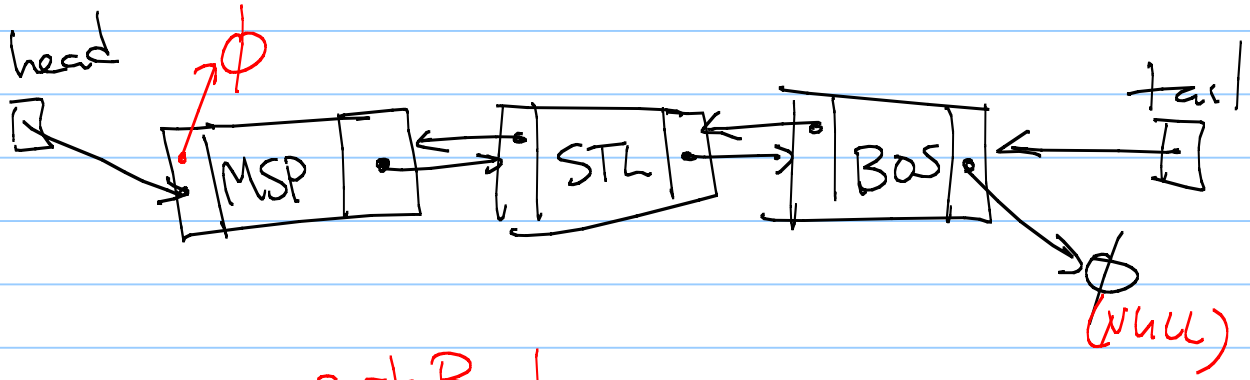
CS180 - Doubly Linked Lists + Stacks

Note Title 2/7/2011

Announcements

- Lab tomorrow
- Program will be posted in next 1-2 days
- First midterm in ~3 weeks.
- Book is available

Doubly Linked Lists



- insert Back
- delete Back

(see code on webpage)

LIFO - last in, first out

Stack: a way to store a list of data

Ex: Web browser: Store history

Hit back, goes to most recent page

Ex: Text editors: Store previously executed commands

Undo will revert to most recent "history"

The Stack Abstract Data Type (ADT)

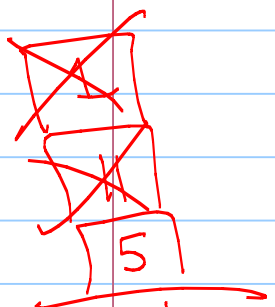
Supports 2 main functions:

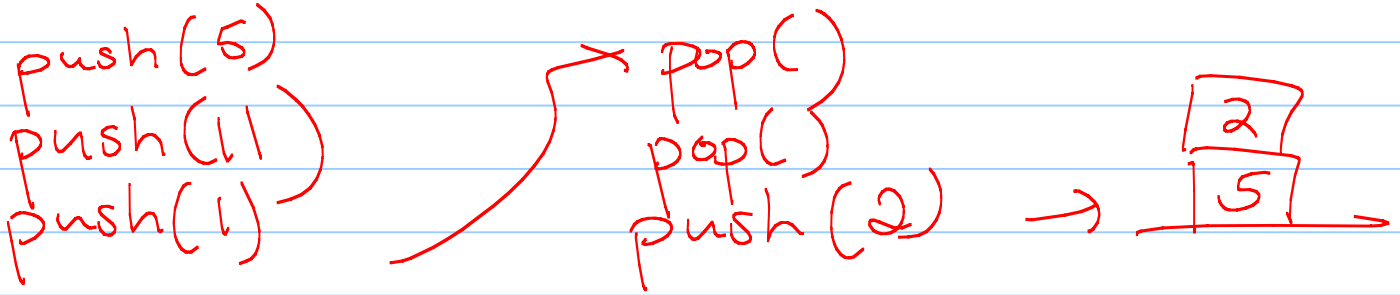
- push(o)

  Insert object o at top of stack

- pop()

  Remove top object from stack & return it





Additional behaviors

- size(): Return # of objects in the stack
- ~~isEmpty~~ empty(): Returns true if stack is empty, false otherwise
- top(): Returns top object on stack without removing it

Standard Template Library

Stacks are one of the built in class in the STL.

Functions: push, pop, top, size, & empty

Documentation is available online.

(We'll use this for lab soon...)

class Stack {
private:

Notice:

I haven't said what this is made with!

Ideas?

- Array-based stack
- Linked list stack

One complication: how should we return objects?

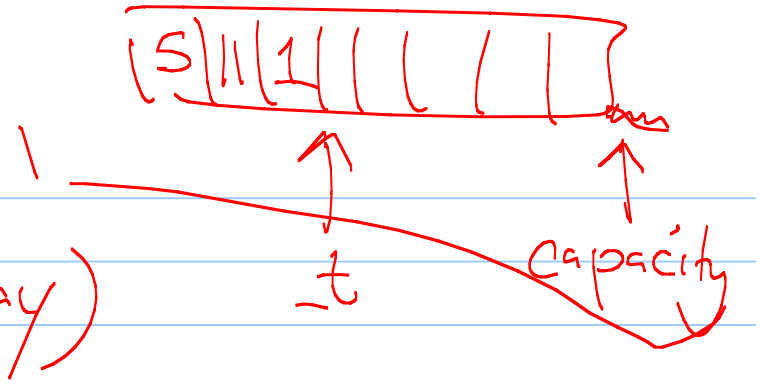
Should pop + top be different?

STL does not return a popped element.

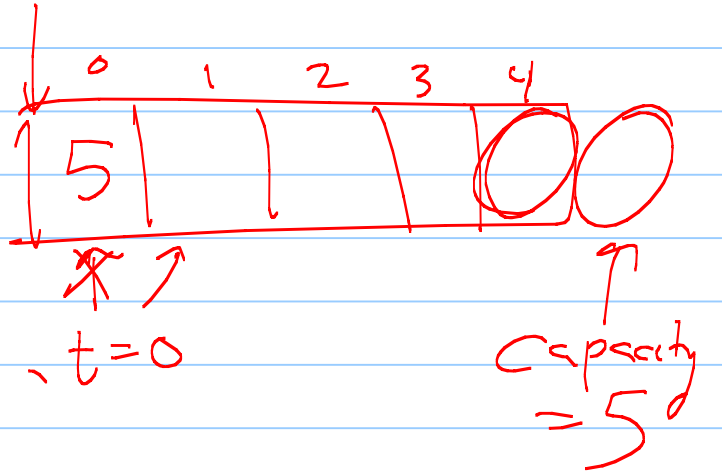
STL wins — pop will be void

Our interface

Private variables:




```
private:
    Object * _s;
    int      _capacity;
    int      _size;
```



Now – what is left??

- Constructor
- Destructor
- Copy Constructor
- Operator=

] deep copies

Another way

Linked Stacks

(We can use SLinkedList class!)