CPS 100 - Program Design and Analysis II
Dr. Susan Rodger
Section: Stacks and Queues (handout)

Read Chap 6.2-3, Chap 15, Chap 11

Data Structure

- representation of data
- operations on the data

Stack

(LIFO - Last In, First Out) data structure

Operations on a Stack

- Create an empty stack
- Destroy the stack
- void Push (Item)
- Item Top()
- void Pop()
-
-

Example using a stack:

Suppose you want to read in a file of words and write it to another file in reverse order

reverseword < file1 > file2

```
Stack WordStack;
// push words on stack
while (cin >> word)
{
    WordStack.Push(word);
}
// write words to file
while (
{



}
```

Example:

Are ()'s balanced?

```
((a)*((b+c)*d)+e)
(a+(b*d)
(b*d+e)(a+c))
)a+c(
```

Algorithm

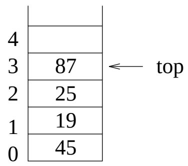
Ignore all chars that are not a '(' or ')'

Note indentation indicates body, didn't use {}'s to save space

```
while no more input
   x = next paren;
   if x=='('
      push(x)
   else if x==')'
      if (stack is not empty)
          y = top(); pop();
      else
          ERROR
```

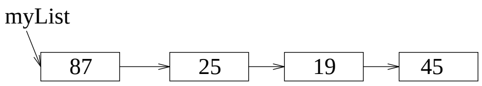
Implementation

1. Array/Vector



- IsEmpty
- Push(Item)
- Pop

2. Linked List




```
Node * myList;
```

- StackEmpty
- Push(Item)
- Pop()

See StackAr.cc and StackAr.h

Example: Evaluate Postfix

- Infix - operator in middle
  $3 * (2 + 4) - 6$
- Postfix - operator last
- Prefix - operator first

Example: Conversion: Infix to Postfix

$A + (B - C) + E * F$

Algorithm - indentation is body of loop and if, didn't use {} to save space

```
push (‘‘low’’) // has lower precedence than other operators
for each char ch do
   if (ch == operand) //  A,B,C,...
       print ch
   else if (ch == ’(’)
       push (ch)
   else if (ch == ’)’)
       y = pop();
       while (y != ’(’)   // print and pop all ops up to ’(’
          print y
          y = pop()
   else if (ch == op)
       y = pop()
       while (1)
           if (y >= preced op)    // pop all prev. ops with >= precedence
              print y
              y = pop()
           else
              push y
              push op
              break // exit this while loop
pop and print remaining ops on stack
```

Queue

(FIFO - First In First Out)

Operations on Queue

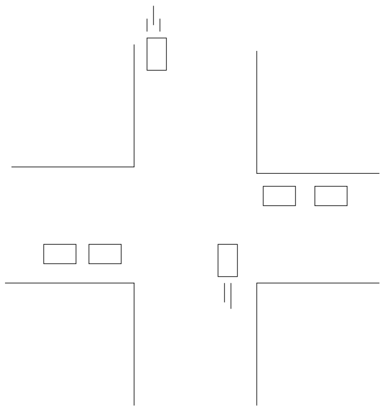
- Create();
- Destroy();
- Enqueue(Item);
- Item Front()
- void Dequeue()
- IsEmpty();

Example:

Queues are used in simulations

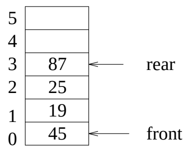
How long do cars wait at a traffic light?

How many queues do you need?

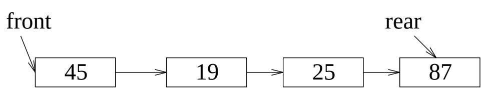


Implementation

- Array/Vector



- Linked list



See QueueAr.h and QueueAr.cc