Stacks and Queues
Stack

- LIFO: Last-in First-out

- Undo/Redo

- Back/Forward

- Function call/return

Stack

- Push

- Pop

- In/Out
Queue

- FIFO: First-in First-out

HPC

Jobs

Queue

NOW SERVING 6589

33
Queue and Stack ADT

Queue
- Enqueue
- Dequeue
- Front
- Size
- Empty?

Stack
- Push
- Pop
- Top
- Size
- Empty?
# Stack/Queue Implementation

## List
- Push_back
- Pop_back
- Push_front
- Pop_front
- Front
- Back
- Size
- Empty

## Queue
- Enqueue
- Dequeue
- Front
- Size
- Empty

## Stack
- Push
- Pop
- Top
- Size
- Empty
## Queue Implementation

<table>
<thead>
<tr>
<th>Queue</th>
<th>Array Impl.</th>
<th>Linked List Impl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>Dequeue</td>
<td>O(n)</td>
<td>O(1)</td>
</tr>
<tr>
<td>Front</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>Memory overhead</td>
<td>Small</td>
<td>Big</td>
</tr>
<tr>
<td>Random access</td>
<td>O(1)</td>
<td>O(n)</td>
</tr>
</tbody>
</table>
Circular Array Queue

Front

Back
Circular Array Queue

Enqueue

Front

Back
Circular Array Queue

Deque

Front

Back
Circular Linked List Queue
# Queue Implementation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>Dequeue</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>Front</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>Memory overhead</td>
<td>Small</td>
<td>Big</td>
</tr>
</tbody>
</table>
Lists, stacks, and queues are all implemented in STL.

In a *real* program, you would better use them; why?

For the sake of learning, you are not allowed to use STL during this class unless otherwise mentioned.
Stack Applications

- Expression evaluation

- Human-friendly infix expressions
  - The operator falls between the two operands
    \[ 3 + 2 \times 5 = 13 \]
  - Easier to read and understand
  - Can be easily broken into pieces

- Machine-friendly postfix expressions
  - The operator is placed after the two operands
    \[ 325 \times + = 13 \]
  - Easier to compute in one pass
  - No need for parentheses
Evaluate postfix expressions

- Infix: \((3 \times 5 + 4/2) \times 2 = 34\)
- Postfix: 3 5 × 4 2 / + 2 ×

Stack of operands
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×
Postfix Evaluation Example

```
3 5 × 4 2 / + 2 ×
```

Stack of operands

Push

3
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×

Stack of operands

Push

5
3

UCR
Postfix Evaluation Example

3 5 \times 4 2 / + 2 \times

Stack of operands

5

Pop

3

Stack of operands
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×

Stack of operands

5
3

Pop
Postfix Evaluation Example

Stack of operands
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×

Stack of operands

15

Push
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×

Push

Stack of operands

2
4
15
Postfix Evaluation Example

3  5  ×  4  2  /  +  2  ×

Stack of operands

2
4
15

Stack of operands
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×

Stack of operands

4 / 2 Push

15 Pop

Pop

Pop

Push
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×

Stack of operands

15 + 2

17 Push

Pop

Pop
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×

Stack of operands

Push

2
17

Stack of operands
Postfix Evaluation Example

3 5 × 4 2 / + 2

Stack of operands:

17

×

2

34

Push

Pop

Pop

×
Postfix Evaluation Example

3 5 × 4 2 / + 2 ×
Infix to Postfix Conversion

- Convert the input into a sequence of operators and operands
- Account for operator precedence
- Account for parentheses

Example

- **Infix (input):** \((3 \times 5 + 4/2) \times 2\)
- **Postfix (desired output):** \(35 \times 42/+2 \times\)
Example

Input: 
\[
( 3 \times 5 + 4 / 2 ) \times 2
\]

Output:

Stack of operators:
Example

Input

\[
(3 \times 5 + 4 / 2) \times 2
\]

Output

3

Stack of operators

(  )
Example

Input

\[(3 \times 5 + 4 \div 2) \times 2\]

Output

3

Stack of operators
Example

Input

\[(3 \times 5 + 4 / 2) \times 2\]

Output

35

Stack of operators
Example

Input

\[
(3 \times 5) + \frac{4}{2}) \times 2
\]

Output

35 ×

Stack of operators
Example

Input

\[(3 \times 5) + \left(\frac{4}{2}\right) \times 2\]

Output

35 \times 4

Stack of operators
Example

Input

\[
(3 \times 5 + 4) \div 2 \times 2
\]

Output

\[35 \times 4\]

Stack of operators

\[
(\quad) \quad \div \quad + \quad /\quad \times\quad 2
\]
Example

Input

\[ (3 \times 5 + 4) / 2 \times 2 \]

Output

\[ 35 \times 42 \]
Example

Input

\[(3 \times 5 + 4 \div 2) \times 2\]

Output

35 \times 42/+
Example

Input

\[(3 \times 5 + 4 / 2) \times 2\]

Output

35 \times 42 / +
Example

Input

\((3 \times 5 + 4 / 2) \times 2\)

Output

\[35 \times 42 / +2\]
Example

Input

\[(3 \times 5 + 4 / 2) \times 2\]

Output

35 \times 42 / +2 \times

Stack of operators