Stacks

Comp Sci 1575 Data Structures

Missouri S&T | Computer Science

Definitions
Operations
Stack ADT
Array-based stack
Stack overflow
Link-based stack
Variations
Applications
Example problem

Stacks are stacked!

The stack is one of the most fundamental data structures in computer science.

Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Stacks

- List-like structure in which elements may be added or removed from only one end
- Limited to insert and remove elements at the "top" of the stack.
- Also known as **LIFO**:
  Last In First Out

Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Operations

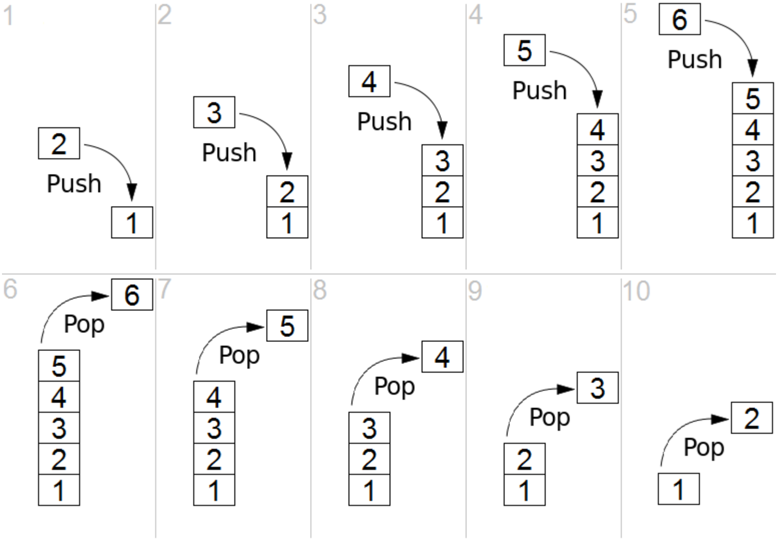
```
push(S, x)
Stack after adding x onto stack S.
Example:
push(<1,2,3,4,5>, 9) = <9, 1, 2, 3, 4, 5>;

pop(S)
Stack after removing the top of stack S.
Example:
pop(<1,2,3,4,5>) = <2, 3, 4, 5>;

peek(S)
Element at the top of the stack S
08e_StacksExample:
top(<1,2,3,4,5>) = 1;
```

Stacks are sweet!

Push

Pop

Push and pop


1
2
Push
1
2
3
Push
2
1
3
4
Push
3
2
1
4
5
Push
4
3
2
1
5
6
Push
5
4
3
2
1
6
6
Pop
5
4
3
2
1
7
5
Pop
4
3
2
1
8
4
Pop
3
2
1
9
3
Pop
2
1
10
2
Pop
1

Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Stack ADT and abstract class

Check out the code
stack.h

Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Array-based stack

Design questions:

- Which end should be the top?
- What are the costs of the operations?
- Check out the code: astack.h
- Word reversal example

Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Stack overflow and underflow

- Pushing to a full stack: **Stack overflow**
- Popping from an empty stack: **Stack underflow**

What are some solutions to this problem?
Exploits?

Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Link-based stack

- What are the costs of the operations?
- How do space requirements compare to the array-based stack implementation?
- Which end should be the top if it is singly linked?
- Check out the code: lstack.h

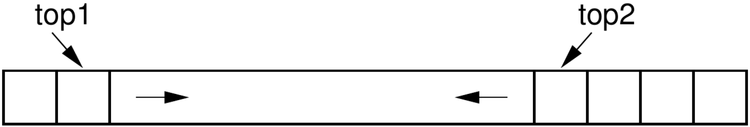
Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Variations

Multiple lists with inversely related sizes enable efficiency boost:

- Freelist node cache (recall overloaded new and delete) was actually a linked list stack
- Array-based list can also capitalize on this phenomenon by placing two stacks in one array:



Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Applications

- Recognizing non-regular expressions, like “balanced parenthesis” for orders of operations
- LIFO caches are very efficient
- Implementing recursive processes
- Backtracking
- Pre-order or Post-order traversals of Trees or Graphs.
- Evaluation of expressions in post-fix form
- Compilation and Translation.
- Expression evaluation and syntax parsing
- Runtime memory management

Outline

1. Definitions
   - Operations
2. Stack ADT
3. Array-based stack
   - Stack overflow
4. Link-based stack
5. Variations
6. Applications
   - Example problem

Example

Check out the code: Main_lstack_balance.cpp

- Balancing brackets