Algorithms and Data Structures

Stacks

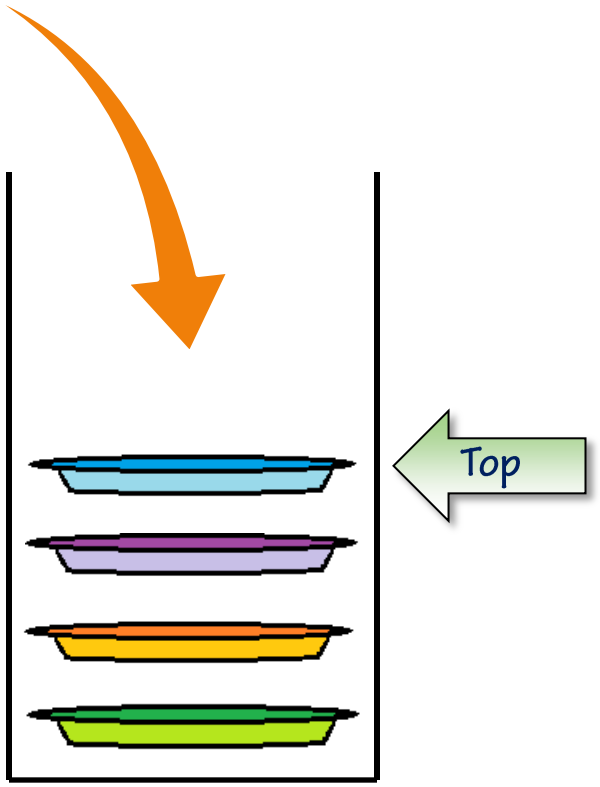
Robert Horvick

www.pluralsight.com

Outline

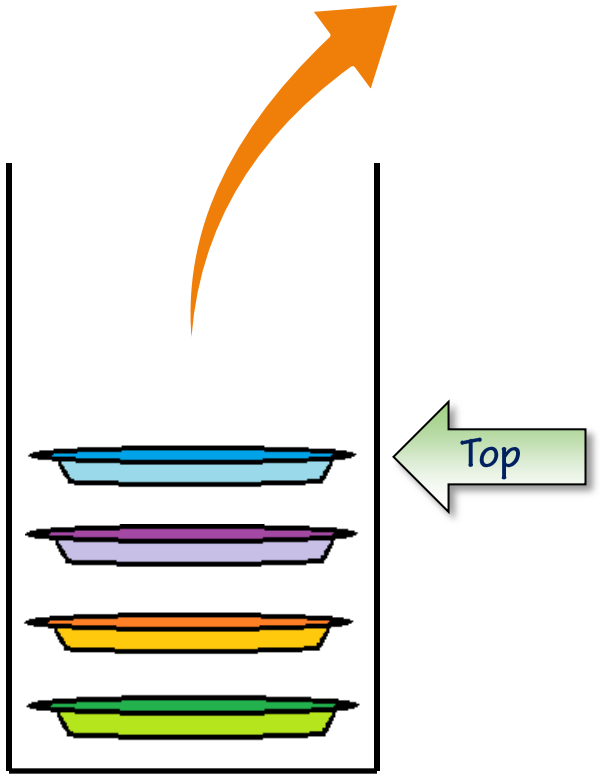
- Last In First Out (LIFO)
- Stack using a Linked List
- Stack using an Array
- Postfix Calculator
- Implementing “Undo”
- .NET and C++ implementations

Pushing Onto the Stack



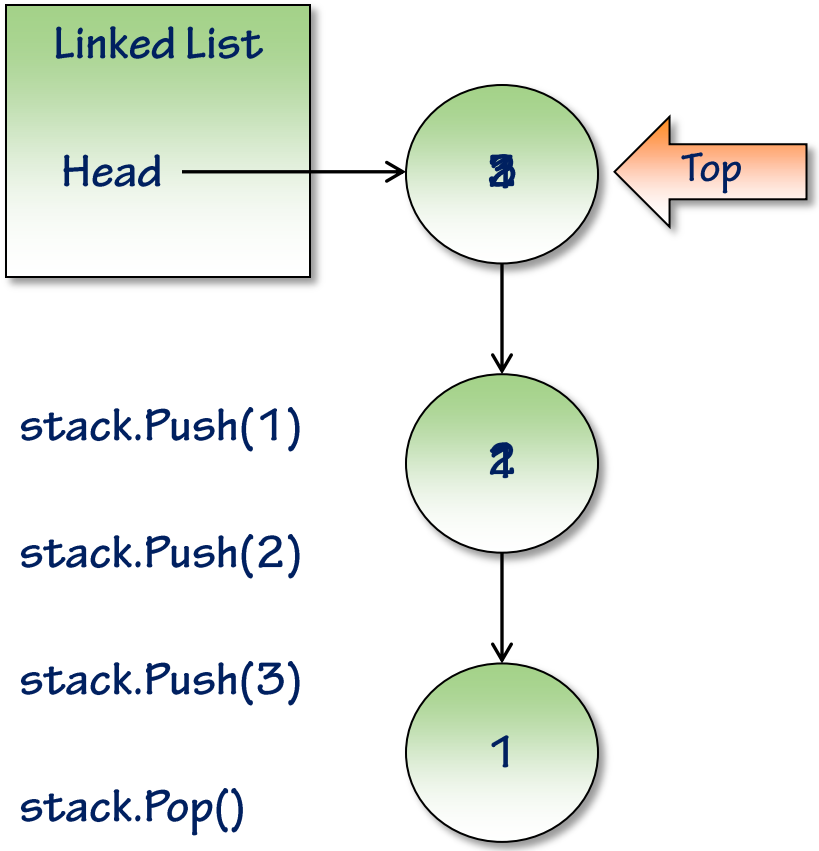
- The stack is empty
- A plate is “pushed” onto the stack
- It is now the top of the stack
- More plates are pushed on
  - This increases the stack “depth”
- The top plate can always be seen
  - This is known as “peeking”

Popping Off of the Stack



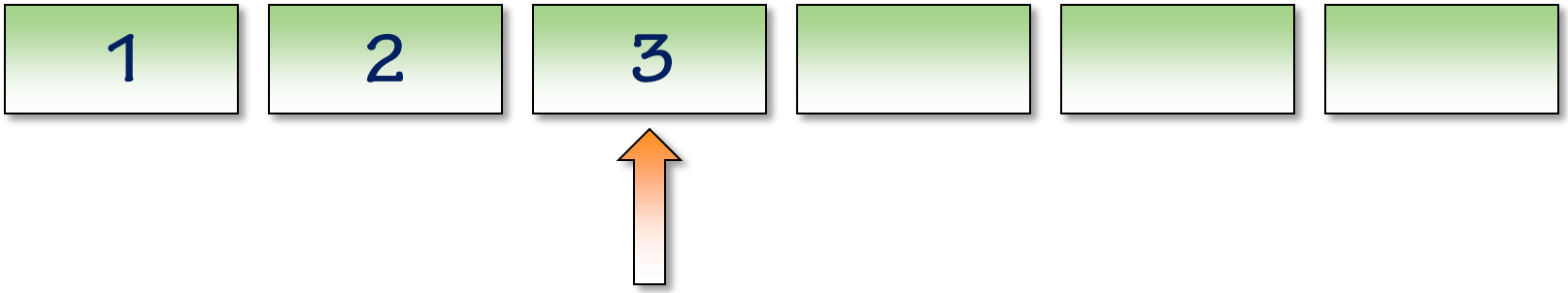
- **A plate is "popped" off the stack**
  - Last In First Out (LIFO)
- **More plates are popped**
  - Each reduces the stack depth
- **Eventually the stack is empty**

Using a Linked List



- **Pros**
  - No hard size (depth) limit
  - Easy to implement
    - No bounds checking
    - Empty list = Empty stack
- **Cons**
  - Memory allocation on push
  - Per-node memory overhead
  - Potential performance issues

Using an Array



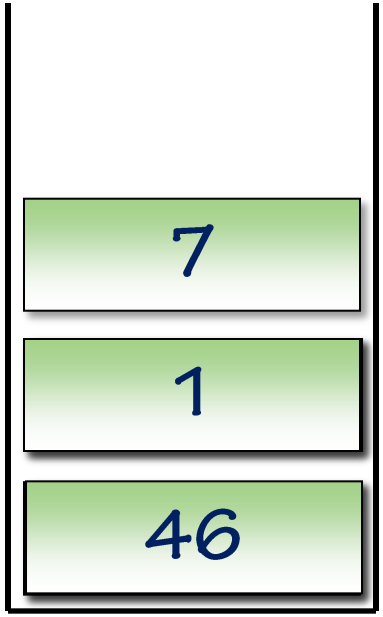
- Push 1
- Push 2
- Push 3
- Peek
- Pop
- Pop

Postfix Calculator

- Reverse Polish Notation
- Operator follows operands
  - Infix: 5 + 2
  - Postfix: 5 2 +
- Operation order can cause ambiguity
  - 5 + 6 * 7 – 1
- Postfix is unambiguous
  - 5 6 7 * + 1 -

Postfix Algorithm

5 6 7 * + 1 -




```
foreach token
   if token is integer
      push token
   else if token is operator
      pop right-side value
      pop left-side value
      evaluate operator
      push result
next
```

Modern Implementations

- C#
  - Stack<T>
  - Push, Peek, Pop
  - Items stored in array

```
Stack<int> values = new Stack<int>();

values.Push(10);
values.Push(20);

int twenty  = values.Pop();
int ten = values.Pop();
```

- C++
  - std::stack<T>
  - push, top, pop
  - Items stored in array

```
std::stack<int> values;

values.push(10);
values.push(20);

int twenty = values.top();
values.pop();

int ten = values.top();
values.pop();
```

Summary

- Last In First Out (LIFO) container
- Push on
- Pop off
- Peek at top
- A variety of backing stores
  - Linked List
  - Array
- Common feature of modern languages/platforms

References

- Stack<T> on MSDN
  - http://msdn.microsoft.com/en-us/library/3278tedw.aspx
- std::stack<T> on MSDN
  - http://msdn.microsoft.com/en-us/library/56fa1zk5(v=VS.100).aspx