Stacks and Queues

Abstract Data Type

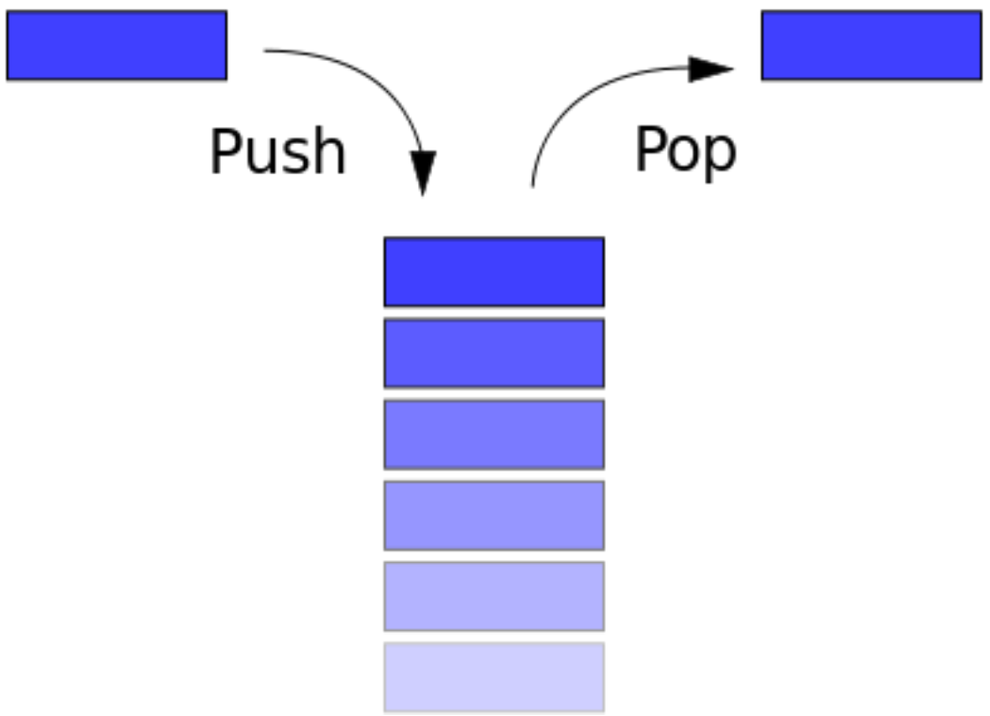
an interface to a data structure

Stack

http://media4.onsugar.com/files/2014/07/02/568/n/1922398/f6cbdbf0582dca43_stack.xxxlarge.jpg

Stack Interface

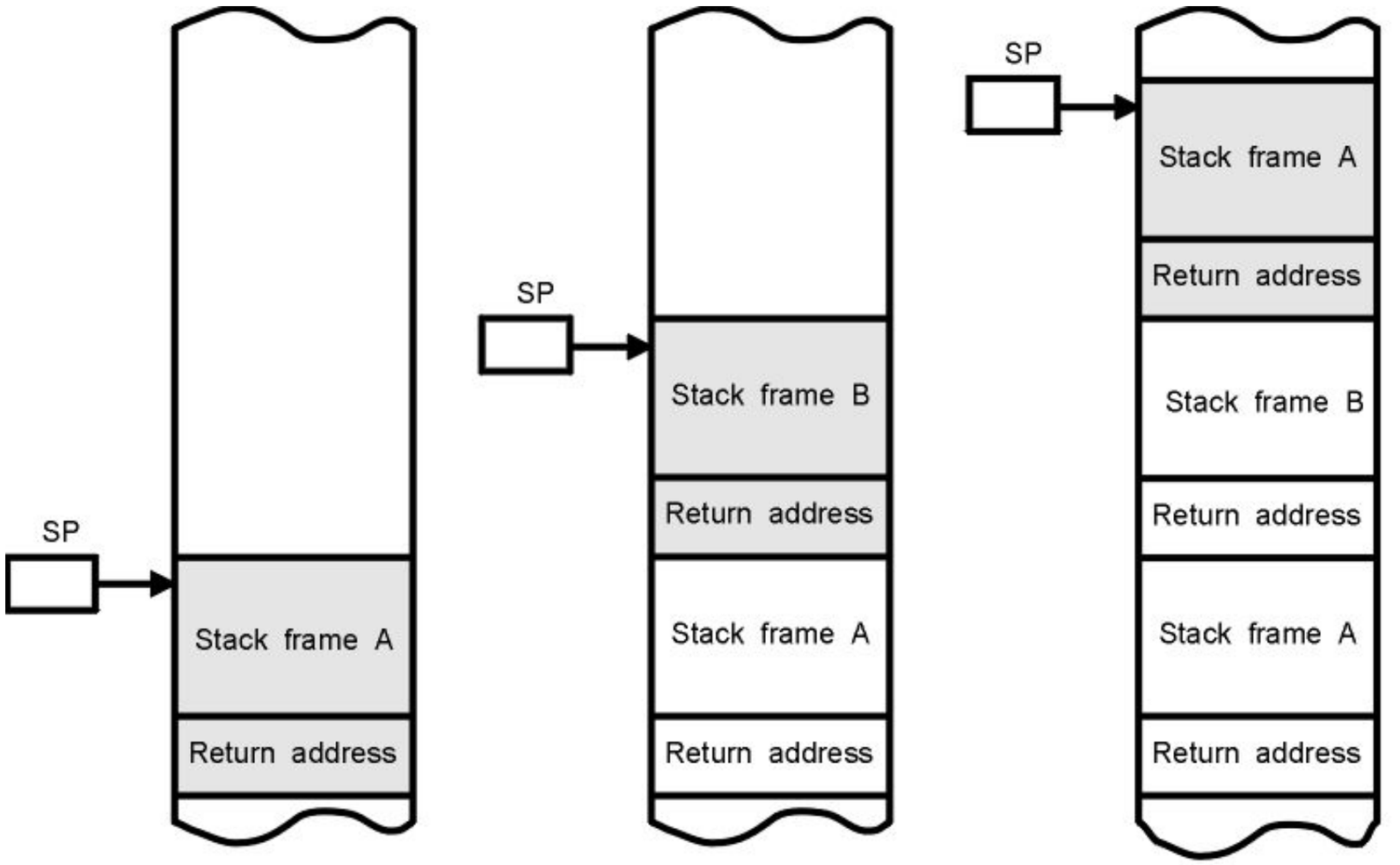
push, pop, (and sometimes peek)



http://en.wikipedia.org/wiki/Stack_%28abstract_data_type%29#mediaviewer/File:Data_stack.svg

Use Cases

- scoping
- reverse
- syntax checking
- recursion



a. The state of the stack during subroutine A

b. The state of the stack during subroutine B

c. The state of the stack during a second call to subroutine A

https://tiagodev.files.wordpress.com/2013/03/figure2.jpg

Reverse

https://s-media-cache-ak0.pinimg.com/736x/5f/8a/8d/5f8a8df98b7d04fe4e3d6d6c8902926d.jpg

Try Me

reverse an array using another array
(as though it were a stack)

```
function reverse(array) {
  var stack = [];
  for(var i = 0; i < array.length; i++) {
    stack.push(array[i]);
  }
  var j = 0;
  while(stack.length){
    array[j++] = stack.pop();
  }
  return array;
}
```

Information Hiding

http://www.hdwallpapers.in/walls/sleepy_kitten-wide.jpg

Restricting the Interface with composition

```
function Stack() {
  var array = [];

  this.push = array.push.bind(array);
  this.pop = array.pop.bind(array);
  this.peek = function () {
    return array[array.length-1];
  };
}
```

Syntax Checking

http://erinhunter.katecary.co.uk/wp-content/uploads/2013/05/cat-on-typewriter.jpg

Try Me

reverse an array using a Stack

```
function reverse(array) {
  var stack = new Stack();
  for(var i = 0; i < array.length; i++) {
    stack.push(array[i]);
  }
  var j = 0;
  while(stack.peek()){
    array[j++] = stack.pop();
  }
  return array;
}
```

Try Me

check for matching brackets

```
function checkBrackets(string) {
  var leftBrackets = /[\[{\(]/;
  var rightBrackets = /[\]}\)]/;
  var rightToLeft = {
    "]":"[",
    ")":"(",
    "}":"{"
  };
  var bracketStack = new Stack();
  var current;
  for(var i = 0; i < string.length; i++) {
    current = string.charAt(i);

    if(current.match(leftBrackets)){
      bracketStack.push(current);
    }

    if (current.match(rightBrackets)) {
      if(rightToLeft[current] !== bracketStack.pop())
        return false;
    }
  }

  return true;
}
```

Queue

http://www.backyardchickencoops.com.au/wp-content/uploads/2013/11/chickens_chicks_baby.jpg

Queue Interface

enqueue, dequeue (and, sometimes, hasNext)

Use Cases

- scheduling
- communication

  (events, messages)

Try Me

Create a constructor for a Queue class. Implement it in a way that hides the underlying data structure (the array).

```
function Queue() {
  var array = [];

  this.queue = array.push.bind(array);
  this.dequeue = array.shift.bind(array);
  this.hasNext = function () {
    return !!(array.length);
  };
}
```

Try Me

now try to make dequeue O(1)

```
function Queue() {
  var array = [];
  var front = 0;

  this.queue = array.push.bind(array);
  this.dequeue = function () {
    var next = array[front];
    array[front] = null;
    front++;
    return next;
  }
  this.hasNext = function () {
    return !!(array.length);
  };
}
```