

# 堆栈、队列 (Stack、Queue)

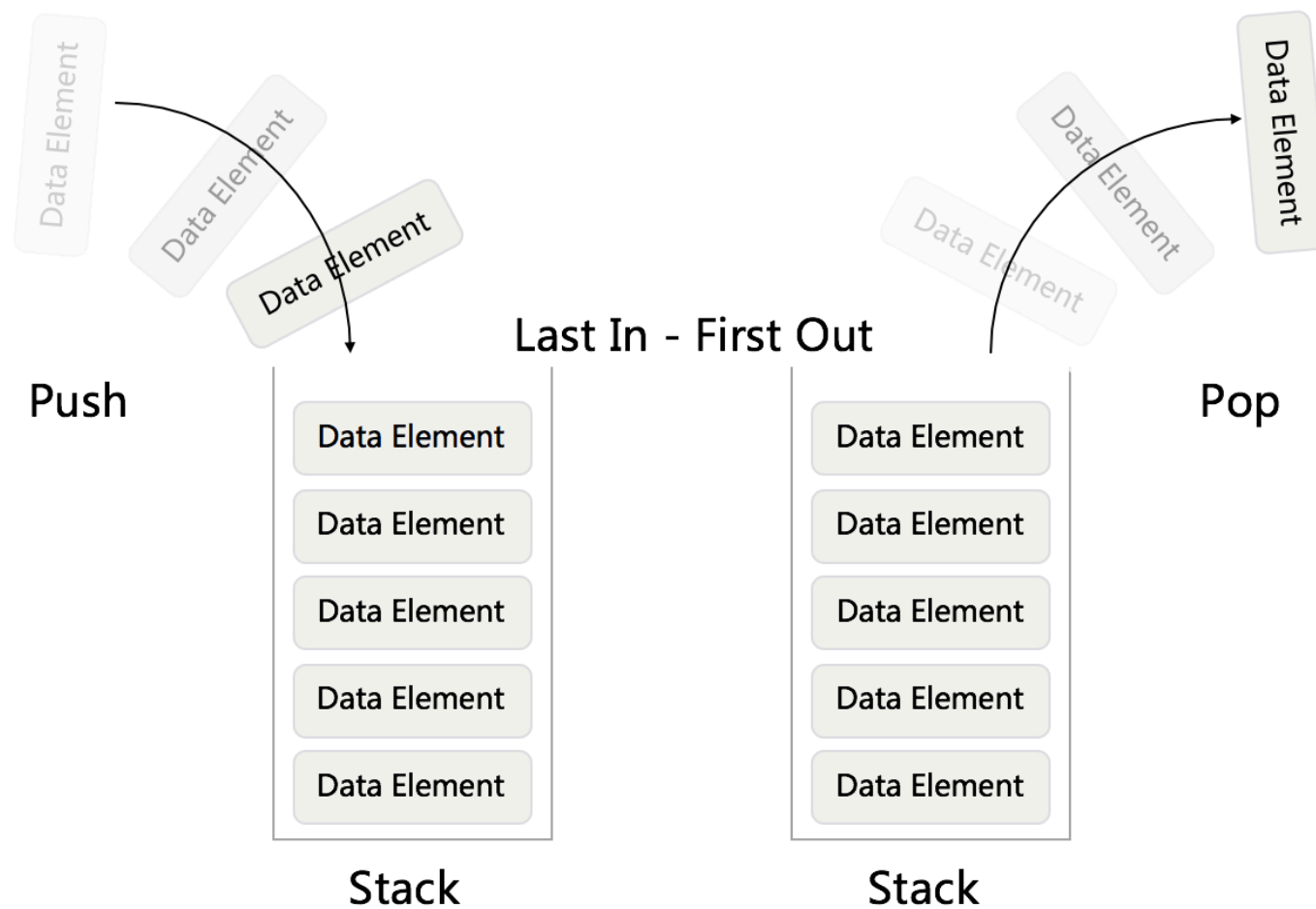
# 本节内容

1. Stack - First In First Out (FIFO)
  - Array or Linked List
2. Queue - First In Last Out (FILO)
  - Array or Linked List

# Stack

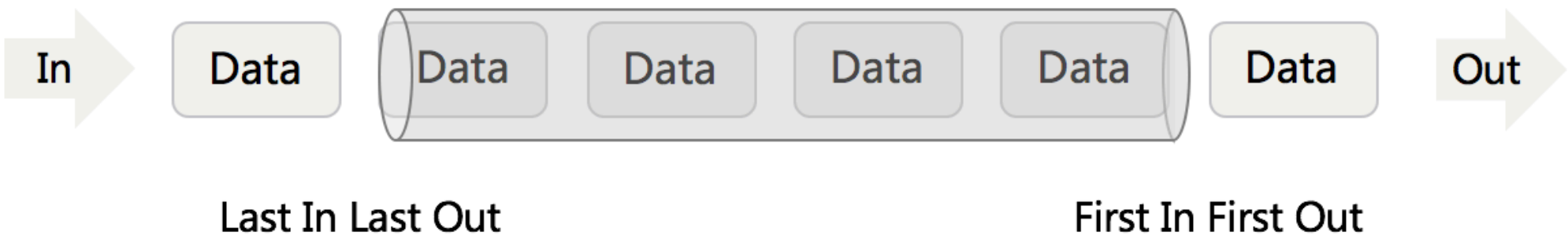


# Stack



# Queue





Queue



# Common Data Structure Operations

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
<u>Array</u>	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>Stack</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Queue</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Singly-Linked List</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Doubly-Linked List</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Skip List</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n \log(n))$
<u>Hash Table</u>	N/A	$\theta(1)$	$\theta(1)$	$\theta(1)$	N/A	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>Binary Search Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>Cartesian Tree</u>	N/A	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	N/A	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>B-Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>Red-Black Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>Splay Tree</u>	N/A	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	N/A	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>AVL Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>KD Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$

图片来源: <http://www.bigocheatsheet.com/>

# Big O Cheat Sheet

## LEGEND

TIME Complexity  VS.  SPACE Complexity

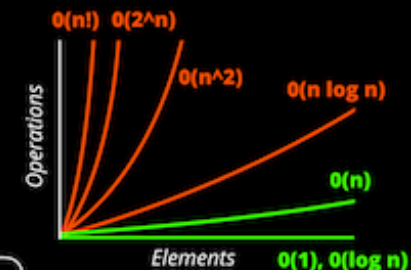
 Good  Fair  Bad

 Good  Fair  Bad

## <BIG-O-CHEATSHEET>



[www.bigocheatsheet.com](http://www.bigocheatsheet.com)



### DATA STRUCTURE

Operations

### ARRAY SORTING

Algorithms

 DATA Structure

 TIME Complexity

 SPACE Complexity

 ARRAY Algorithms

 TIME Complexity

 SPACE Complexity

Average

Worst

Best

Average

Worst

Worst

 Access

 Search

 Insertion

 Deletion

 Access

 Search

 Insertion

 Deletion

Array



$\Theta(1)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

Stack



$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

Queue



$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

Singly-Linked List



$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

Doubly-Linked List



$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

$\Theta(n)$

$\Theta(1)$

$\Theta(1)$

$\Theta(n)$

Skip List



$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n \log(n))$

Hash Table



N/A

$\Theta(1)$

$\Theta(1)$

$\Theta(1)$

N/A

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

Binary Search Tree



$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

Cartesian Tree



N/A

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

N/A

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

B-Tree



$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(n)$

Red-Black Tree



$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(n)$

Splay Tree



N/A

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

N/A

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(n)$

AVL Tree



$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(n)$

KD Tree



$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(\log(n))$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

$\Theta(n)$

Quicksort



$\Omega(n \log(n))$

$\Theta(n \log(n))$

$\Theta(n^2)$

$\Theta(\log(n))$

Mergesort



$\Omega(n \log(n))$

$\Theta(n \log(n))$

$\Theta(n \log(n))$

$\Theta(n)$

Timsort



$\Omega(n)$

$\Theta(n \log(n))$

$\Theta(n \log(n))$

$\Theta(1)$

Heapsort



$\Omega(n \log(n))$

$\Theta(n \log(n))$

$\Theta(n \log(n))$

$\Theta(1)$

Bubble Sort



$\Omega(n)$

$\Theta(n^2)$

$\Theta(n^2)$

$\Theta(1)$

Insertion Sort



$\Omega(n)$

$\Theta(n^2)$

$\Theta(n^2)$

$\Theta(1)$

Selection Sort



$\Omega(n^2)$

$\Theta(n^2)$

$\Theta(n^2)$

$\Theta(1)$

Tree Sort



$\Omega(n \log(n))$

$\Theta(n \log(n))$

$\Theta(n^2)$

$\Theta(n)$

Shell Sort



$\Omega(n \log(n))$

$\Theta(n(\log(n))^2)$

$\Theta(n(\log(n))^2)$

$\Theta(1)$

Bucket Sort



$\Omega(n+k)$

$\Theta(n+k)$

$\Theta(n^2)$

$\Theta(n)$

Radix Sort



$\Omega(nk)$

$\Theta(nk)$

$\Theta(nk)$

$\Theta(n+k)$

Counting Sort



$\Omega(n+k)$

$\Theta(n+k)$

$\$



# 实战题目

1. <https://leetcode.com/problems/implement-queue-using-stacks/solution/>
2. <https://leetcode.com/problems/implement-stack-using-queues/description/>
3. <https://leetcode.com/problems/valid-parentheses/description/>