

CS062

DATA STRUCTURES AND ADVANCED PROGRAMMING

29: Summary



Tom Yeh
he/him/his

Java Basics

- ▶ Chapter 1.1 (Pages 8–35).
- ▶ Chapter 1.2 (Pages 64–77, 84–88, 96–99, 107).
- ▶ Quick overview of Java tutorials.
 - ▶ <https://docs.oracle.com/javase/tutorial/java/>
- ▶ In general, review the basics of OOP and of Java so that you are comfortable reading and writing code.

Analysis of Algorithms

- ▶ Chapter 1.4 (Pages 172-205).
- ▶ Experimental analysis including doubling hypothesis.
- ▶ Mathematical analysis including reviewing (not memorizing) useful approximations of sums.
 - ▶ Use midterm review slides for practice.
- ▶ Order of growth classifications.
- ▶ Review of running time of operations on array lists, linked lists, stacks and queues.

ArrayLists

- ▶ Chapter 1.3 (Pages 136-137).
- ▶ Textbook API and code.
 - ▶ <https://github.com/pomonacs622021fa/LectureCode/blob/main/Lecture6/ArrayList.java>
- ▶ Java Oracle API <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
- ▶ Amortized and worst-case time analysis.

Singly Linked Lists

- ▶ Chapter 1.3 (Pages 142-146).
- ▶ Textbook API and code.
 - ▶ <https://github.com/pomonacs622021fa/LectureCode/blob/main/Lecture7/SinglyLinkedList.java>
- ▶ Worst-case time analysis for standard operations.

Doubly Linked Lists

- ▶ Chapter 1.3 (Pages 126-157).
- ▶ Textbook API and code.
 - ▶ <https://github.com/pomonacs622021fa/LectureCode/blob/main/Lecture8/DoublyLinkedList.java>
- ▶ Java Oracle API.
 - ▶ <https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>
- ▶ Worst-case time analysis for standard operations.

Stacks, Queues, and Iterators

- ▶ Chapter 1.3 (Pages 142-146).
- ▶ Textbook API and code for alternative implementations
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/ResizingArrayQueue.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/LinkedQueue.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/ResizingArrayStack.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/LinkedStack.java.html>
- ▶ Java Oracle Iterator and Iterable.
 - ▶ <https://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html>
 - ▶ <https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html>
- ▶ Worst-case time analysis for standard operations based on the underlying implementation

Sorting

- ▶ Chapter 2 (Pages 244-296).
- ▶ Selection sort.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Selection.java.html>
- ▶ Insertion sort.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Insertion.java.html>
- ▶ Mergesort.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Merge.java.html>
- ▶ Quicksort.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Quick.java.html>
- ▶ Know how to apply them, best and worst case running times, in-place or not, stability

Comparators

- ▶ Comparable vs Comparator interface.
 - ▶ <https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html>
 - ▶ <https://docs.oracle.com/javase/8/docs/api/java/util/Comparator.html>

Binary Trees

- ▶ Definitions, basic properties, and traversals.

Priority Queues

- ▶ Chapter 2.4 (Pages 308-325), 2.5 (336-344).
- ▶ Different implementations along with complexities.
- ▶ Binary Heaps and operations.
- ▶ Heapsort. Know how to apply, running time analysis.
- ▶ Textbook code.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/MaxPQ.java.html>

What you need to remember about sorting

	In place	Stable	Best	Average	Worst	Remarks
Selection	X		n^2	n^2	n^2	n exchanges
Insertion	X	X	n	n^2	n^2	Use for small arrays or partially ordered
Merge		X	$n \log n$	$n \log n$	$n \log n$	Guaranteed performance; stable
Quick	X		$n \log n$	$n \log n$	n^2	$n \log n$ probabilistic guarantee; fastest in practice
Heap	X		$n \log n$	$n \log n$	$n \log n$	$n \log n$ guarantee; in place

Symbol Tables and Binary Search

- ▶ Chapter 3.1 (Pages 362-386).
- ▶ Different implementations along with complexities.

Binary Search Trees

- ▶ Chapter 3.2 (Pages 396-414).
- ▶ Textbook code.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/BST.java.html>
- ▶ Addition, Search, Deletion.
- ▶ Ordered Operations.
- ▶ Cost of ordered symbol table operations depending on implementation.

2-3 Search Trees

- ▶ Chapter 3.3 (Pages 424-291).
- ▶ Definitions, Search, Insertion, Construction.
- ▶ Performance.

Left-leaning red-black trees

- ▶ Chapter 3.3 (Pages 292-447).
- ▶ Definitions, Operations, Insertion.
- ▶ Performance.

Hash tables

- ▶ Chapter 3.3 (Pages 458-477).
- ▶ Hashing, separate chaining, open addressing.
- ▶ Linear probing, quadratic probing.

Summary for symbol table operations

	Worst case			Average case		
	Search	Insert	Delete	Search	Insert	Delete
Sequential search (unordered list)	n	n	n	$n/2$	n	$n/2$
Binary search (ordered array)	$\log n$	n	n	$\log n$	$n/2$	$n/2$
BST	n	n	n	$1.39 \log n$	$1.39 \log n$	?
2-3 search tree	$c \log n$	$c \log n$	$c \log n$	$c \log n$	$c \log n$	$c \log n$
Red-black BSTs	$2 \log n$	$2 \log n$	$2 \log n$	$1 \log n$	$1 \log n$	$1 \log n$
Separate chaining	n	n	n	3 – 5	3 – 5	3 – 5
Linear probing	n	n	n	3 – 5	3 – 5	3 – 5

Undirected Graphs

- ▶ Chapter 4.1 (Pages 515-556).
- ▶ Definitions, representations, APIs.
- ▶ BFS.
- ▶ DFS.
- ▶ Connected Components.
- ▶ Textbook code.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Graph.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/DepthFirstSearch.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/BreadthFirstPaths.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/CC.java.html>

Directed Graphs

- ▶ Chapter 4.2 (Pages 566-594).
- ▶ Definitions, representations, APIs.
- ▶ BFS.
- ▶ DFS.
- ▶ Depth first orders.
- ▶ Topological sort
- ▶ Textbook code.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Digraph.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/DirectedDFS.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/DepthFirstOrder.java.html>

Shortest Paths

- ▶ Chapter 4.4 (Pages 638-676).
- ▶ Dijkstra's and Bellman-Ford's algorithms.
- ▶ Textbook code.
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/EdgeWeightedDigraph.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/DirectedEdge.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/DijkstraSP.java.html>
 - ▶ <https://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/BellmanFordSP.java.html>

Minimum Spanning Trees

- ▶ Chapter 4.4 (Pages 604-629).
- ▶ Kruskal's and Prim's algorithms.