Big O Cheat Sheet

Array Sorting Algorithms:

$O(1) < O(\log(n)) < O(n) < O(n \log(n)) < O(n^2) < O(2^n) < O(n!)$

	Best	Average	Worst
Quick Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$
Merge Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$
Timsort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$
Heap Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$
Bubble Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$
Insertion Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$
Selection Sort	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$
Tree Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$
Shell Sort	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$O(n(\log(n))^2)$
Bucket Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$
Radix Sort	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$
Counting Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$
Cubesort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$
Smooth Sort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$
Tournament Sort	-	$\Theta(n \log(n))$	$O(n \log(n))$
Stooge sort	-	-	$O(n^{\log 3 / \log 1.5})$
Gnome/Stupid sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$
Comb sort	$\Omega(n \log(n))$	$\Theta(n^2/p^2)$	$O(n^2)$
Odd – Even sort	$\Omega(n)$	-	$O(n^2)$

Data Structures :

Having same average and worst case:

	Access	Search	Insertion	Deletion
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))
Splay Tree	-	Θ(log(n))	Θ(log(n))	Θ(log(n))
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))

Having different average and worst case:

	Average				Worst			
	Access	Search	Insert	Delete	Access	Search	Insert	Delete
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O (n)	O (n)	O (n)	O (n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O (n)	O (n)	O (n)	O (n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O (n)	O (n)	O (n)	O (n)
Hash Table	-	Θ(1)	Θ(1)	Θ(1)	-	O (n)	O (n)	O (n)

Heap Data Structure:

S - Sorted, US - Unsorted , Binary Heap Heapify - O (n)

	Find Max	Extract Max	Increase Key	Insert	Delete	Merge
Linked List (S)	O (1)	O (1)	O (n)	O (n)	O (1)	O (m+n)
Linked List (US)	O (n)	O (n)	O (1)	O (1)	O (1)	O (1)
Binary Heap	O (1)	O (log (n))	O (log (n))	O (log (n))	O (log (n))	O (m+n)
Pairing Heap	O (1)	O (log (n))	O (log (n))	O (1)	O (log (n))	O (1)
Binomial Heap	O (1)	O (log (n))	O (log (n))	O (1)	O (log (n))	O (log (n))
Fibonacci Heap	O (1)	O (log (n))	O (1)	O (1)	O (log (n))	O (1)

Graph Data Structure

	Storage	Add Vertex	Add Edge	Remove Vertex	Remove Vertex	Query
Adjacency list	O (|V| + |E|)	O (1)	O (1)	O (|V| + |E|)	O (|E|)	O (|V|)
Incidence list	O (|V| + |E|)	O (1)	O (1)	O (|E|)	O (|E|)	O (|E|)
Adjacency matrix	O (|V|^{2})	O (|V|^{2})	O (1)	O (|V|^{2})	O (1)	O (1)
Incidence matrix	O (|V| . |E|)	O (|V| . |E|)	O (|V| . |E|)	O (|V| . |E|)	O (|V| . |E|)	O (|E|)

Searching Algorithms:

	Time Complexity
Linear Search	O (n)
Binary Search	O (log (n))
Jump Search	O (√ n)
Interpolation Search	O (log (log n))-Best | O (n)-Worst
Exponential Search	O (log (n))
Sequential search	O (n)
Depth-first search (DFS)	O (|V| + |E|)
Breadth-first search (BFS)	O (|V| + |E|)

Graph Algorithms

	Average	Worst
Dijkstra's algorithm	O (|E| log |V|)	O (|V|^{2})
A* search algorithm	O (|E|)	O (b^{d})
Prim's algorithm	O (|E| log |V|)	O (|V|^{2})
Bellman–Ford algorithm	O (|E| · |V|)	O (|E| · |V|)
Floyd-Warshall algorithm	O (|V|^{3})	O (|V|^{3})
Topological sort	O (|V| + |E|)	O (|V| + |E|)

Time Complexity for Java Collections

List: A list is an ordered collection of elements.

	Add	Remove	Get	Contains	Data Structure
ArrayList	O (1)	O (n)	O (1)	O (n)	Array
LinkedList	O (1)	O (1)	O (n)	O (n)	Linked List
CopyonWriteArrayList	O (n)	O (n)	O (1)	O (n)	Array

Set: A collection that contains no duplicate elements.

	Add	Contains	Next	Data Structure
HashSet	O(1)	O(1)	O(h/n)	Hash Table
LinkedHashSet	O(1)	O(1)	O(1)	Hash Table + Linked List
EnumSet	O(1)	O(1)	O(1)	Bit Vector
TreeSet	O(log n)	O(log n)	O(log n)	Red-black tree
CopyonWriteArraySet	O(n)	O(n)	O(1)	Array
ConcurrentSkipList	O(log n)	O(log n)	O(1)	Skip List

Queue: A collection designed for holding elements prior to processing.

	Offer	Peak	Poll	Size	Data Structure
PriorityQueue	O(log n)	O(1)	O(log n)	O(1)	Priority Heap
LinkedList	O(1)	O(1)	O(1)	O(1)	Array
ArrayDequeue	O(1)	O(1)	O(1)	O(1)	Linked List
ConcurrentLinkedQueue	O(1)	O(1)	O(1)	O(n)	Linked List
ArrayBlockingQueue	O(1)	O(1)	O(1)	O(1)	Array
PriorirityBlockingQueue	O(log n)	O(1)	O(log n)	O(1)	Priority Heap
SynchronousQueue	O(1)	O(1)	O(1)	O(1)	None
DelayQueue	O(log n)	O(1)	O(log n)	O(1)	Priority Heap
LinkedBlockingQueue	O(1)	O(1)	O(1)	O(1)	Linked List

Map: An object that maps keys to values.
A map cannot duplicate keys; each key can map to at most one value.

	Get	ContainsKey	Next	Data Structure
HashMap	O(1)	O(1)	O(h / n)	Hash Table
LinkedHashMap	O(1)	O(1)	O(1)	Hash Table + Linked List
IdentityHashMap	O(1)	O(1)	O(h / n)	Array
WeakHashMap	O(1)	O(1)	O(h / n)	Hash Table
EnumMap	O(1)	O(1)	O(1)	Array
TreeMap	O(log n)	O(log n)	O(log n)	Red-black tree
ConcurrentHashMap	O(1)	O(1)	O(h / n)	Hash Tables
ConcurrentSkipListMap	O(log n)	O(log n)	O(1)	Skip List