

Below are the Big O performance of common functions of different Java Collections.

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

Set	Add	Remove	Contains	Next	Size	Data Structure
HashSet	$O(1)$	$O(1)$	$O(1)$	$O(h/n)$	$O(1)$	Hash Table
LinkedHashSet	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	Hash Table +
Linked List						
EnumSet	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	Bit Vector
TreeSet	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(1)$	Red-black tree
CopyOnWriteArraySet	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	Array
ConcurrentSkipListSet	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(1)$	$O(n)$	Skip List

Queue	Offer	Peak	Poll	Remove	Size	Data Structure
PriorityQueue	$O(\log n)$	$O(1)$	$O(\log n)$	$O(n)$	$O(1)$	Priority Heap
LinkedList	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	Array
ArrayDeque	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(1)$	Linked List
ConcurrentLinkedQueue	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	Linked List
ArrayBlockingQueue	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(1)$	Array
PriorityBlockingQueue	$O(\log n)$	$O(1)$	$O(\log n)$	$O(n)$	$O(1)$	Priority Heap
SynchronousQueue	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(1)$	None!
DelayQueue	$O(\log n)$	$O(1)$	$O(\log n)$	$O(n)$	$O(1)$	Priority Heap
LinkedBlockingQueue	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(1)$	Linked List

Map	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