**Big-O of Java Collection Hierarchy**

The book [Java Generics and Collections](http://oreilly.com/catalog/9780596527754/) has this information (pages: 188, 211, 222, 240).

List implementations:

get add contains next remove(0) iterator.remove

ArrayList O(1) O(1) O(n) O(1) O(n) O(n)

LinkedList O(n) O(1) O(n) O(1) O(1) O(1)

CopyOnWrite-ArrayList O(1) O(n) O(n) O(1) O(n) O(n)

Set implementations:

add contains next notes

HashSet O(1) O(1) O(h/n) h is the table capacity

LinkedHashSet O(1) O(1) O(1)

CopyOnWriteArraySet O(n) O(n) O(1)

EnumSet O(1) O(1) O(1)

TreeSet O(log n) O(log n) O(log n)

ConcurrentSkipListSet O(log n) O(log n) O(1)

Map implementations:

get containsKey next Notes

HashMap O(1) O(1) O(h/n) h is the table capacity

LinkedHashMap O(1) O(1) O(1)

IdentityHashMap O(1) O(1) O(h/n) h is the table capacity

EnumMap O(1) O(1) O(1)

TreeMap O(log n) O(log n) O(log n)

ConcurrentHashMap O(1) O(1) O(h/n) h is the table capacity

ConcurrentSkipListMap O(log n) O(log n) O(1)

Queue implementations:

offer peek poll size

PriorityQueue O(log n) O(1) O(log n) O(1)

ConcurrentLinkedQueue O(1) O(1) O(1) O(n)

ArrayBlockingQueue O(1) O(1) O(1) O(1)

LinkedBlockingQueue O(1) O(1) O(1) O(1)

PriorityBlockingQueue O(log n) O(1) O(log n) O(1)

DelayQueue O(log n) O(1) O(log n) O(1)

LinkedList O(1) O(1) O(1) O(1)

ArrayDeque O(1) O(1) O(1) O(1)

LinkedBlockingDeque O(1) O(1) O(1) O(1)

The bottom of the javadoc for the [java.util](http://java.sun.com/javase/6/docs/api/java/util/package-summary.html) package contains some good links:

* [Collections Overview](http://java.sun.com/javase/6/docs/technotes/guides/collections/overview.html) has a nice summary table.
* [Annotated Outline](http://java.sun.com/javase/6/docs/technotes/guides/collections/reference.html) lists all of the implementations on one page.

Note: You have to specify for which case scenario are those figures, for example, delete from Arraylist could take O(n), if you delete element in middle or end of array.