Arrays and Collections

- java.util.Arrays
- java.util.Collections

Arrays

java.util.Arrays

Returns a fixed-size list backed by the specified array.

static <T> List<T> asList(T... a)

Searches the specified array for the specified object using the binary search algorithm.

static int binarySearch(Object[] a, Object key)

Copies the specified array, truncating or padding with nulls so the copy has the specified length.

static <T> T[] copyOf(T[] original, int newLength)

Assigns the specified Object reference to each element of the specified array of Objects.

static void fill(Object[] a, Object val)

Sorts the specified array of objects into ascending order, according to the natural ordering of its elements.

static void sort(Object[] a)

java.util.Collections

Consists exclusively of static methods that operate on or return collections.

Contains polymorphic algorithms that operate on collections.

- Change the collection
- Search
- Immutable
- Change the order

Change the collection

Adds all of the specified elements to the specified collection

static <T> boolean addAll(Collection<? super T> c, T... elements)

Copies all of the elements from one list into another.

static <T> void copy(List<? super T> dest, List<? extends T> src)

Replaces all of the elements of the specified list with the specified element.

static <T> void fill(List<? super T> list, T obj)

Replaces all occurrences of one specified value in a list with another.

static <T> boolean replaceAll(List<T> list, T oldVal, T newVal)

Search

Returns the number of elements in the specified collection equal to the specified object.

static int frequency(Collection<?> c, Object o)

Returns the max/min element of the given collection, according to the order by the specified comparator.

static <T> T max/min(Collection<? extends T> coll, Comparator<? super T> comp)

Searches the specified list for the specified object using the binary search algorithm.

static <T> int binarySearch(List<? extends Comparable<? super T>> list, T key)

Returns true if the two specified collections have no elements in common.

static boolean disjoint(Collection<?> c1, Collection<?> c2)

Immutable

Returns an immutable list consisting of n copies of the specified object.

static <T> List<T> nCopies(int n, T o)

Returns an immutable list containing only the specified object.

static <T> List<T> singletonList(T o)

Change the order

Reverses the order of the elements in the specified list.

//This method runs in linear time.

static void reverse(List<?> list)

Rotates the elements in the specified list by the specified distance.

static void rotate(List<?> list, int distance)

Randomly permutes the specified list using a default source of randomness.

static void shuffle(List<?> list)

Sorts the specified list according to the order induced by the specified comparator.

//stable, adaptive, iterative mergesort

static <T> void sort(List<T> list, Comparator<? super T> c)