

Set

- ★^A contains(Object o) : boolean - Set - 20%
- ★^A iterator() : Iterator - Set - 6%
- ★^A addAll(Collection c) : boolean - Set - 2%
- ★^A isEmpty() : boolean - Set - 2%
- ★^A size() : int - Set - 0.86%
- ★^A toArray(Object[] a) : Object[] - Set - 0.29%
- ★^A clear() : void - Set - 0.28%
- ★^A add(Object e) : boolean - Set - used
- ★^A containsAll(Collection c) : boolean - Set
- ★^A equals(Object o) : boolean - Set
- ^D forEach(Consumer action) : void - Iterable
- getClass() : Class<?> - Object
- ★^A hashCode() : int - Set
- notify() : void - Object

- notifyAll() : void - Object
- ^D parallelStream() : Stream - Collection
- ★^A remove(Object o) : boolean - Set
- ★^A removeAll(Collection c) : boolean - Set
- ^D removeIf(Predicate filter) : boolean - Collection
- ★^A retainAll(Collection c) : boolean - Set
- ^D splitter() : Splitter - Set
- ^D stream() : Stream - Collection
- ★^A toArray() : Object[] - Set
- toString() : String - Object
- wait() : void - Object
- wait(long timeout) : void - Object
- wait(long timeout, int nanos) : void - Object

SortedSet

- ★^A `iterator() : Iterator` - Set - 21%
- ★^A `clear() : void` - Set - 9%
- ★^A `remove(Object o) : boolean` - Set - 7%
- ★^A `size() : int` - Set - 7%
- ★^A `first() : Object` - SortedSet - 6%
- ★^A `contains(Object o) : boolean` - Set - 4%
- ★^A `isEmpty() : boolean` - Set - 2%
- ★^A `add(Object e) : boolean` - Set - used
- ★^A `addAll(Collection c) : boolean` - Set
- ★^A `comparator() : Comparator` - SortedSet
- ★^A `containsAll(Collection c) : boolean` - Set
- ★^A `equals(Object o) : boolean` - Set
- ^D `forEach(Consumer action) : void` - Iterable
- `getClass() : Class<?>` - Object
- ★^A `hashCode() : int` - Set
- ★^A `headSet(Object toElement) : SortedSet` - SortedSet
- ★^A `last() : Object` - SortedSet
- `notify() : void` - Object

- `notifyAll() : void` - Object
- ^D `parallelStream() : Stream` - Collection
- ★^A `removeAll(Collection c) : boolean` - Set
- ^D `removeIf(Predicate filter) : boolean` - Collection
- ★^A `retainAll(Collection c) : boolean` - Set
- ^D `splitter() : Splitter` - SortedSet
- ^D `stream() : Stream` - Collection
- ★^A `subSet(Object fromElement, Object toElement) : SortedSet` - SortedSet
- ★^A `tailSet(Object fromElement) : SortedSet` - SortedSet
- ★^A `toArray() : Object[]` - Set
- ★^A `toArray(Object[] a) : Object[]` - Set
- `toString() : String` - Object
- `wait() : void` - Object
- `wait(long timeout) : void` - Object
- `wait(long timeout, int nanos) : void` - Object

NavigableSet

★ ^A headSet(Object toElement, boolean inclusive) : NavigableSet - NavigableSet - 0.01%	
★ ^A subSet(Object fromElement, boolean fromInclusive, Object toElement, boolean toInclusive) : NavigableSet - NavigableSet	
★ ^A tailSet(Object fromElement, boolean inclusive) : NavigableSet - NavigableSet	★ ^A iterator() : Iterator - NavigableSet
★ ^A add(Object e) : boolean - Set - used	★ ^A last() : Object - SortedSet
★ ^A addAll(Collection c) : boolean - Set	★ ^A lower(Object e) : Object - NavigableSet
★ ^A ceiling(Object e) : Object - NavigableSet	● notify() : void - Object
★ ^A clear() : void - Set	● notifyAll() : void - Object
★ ^A comparator() : Comparator - SortedSet	● ^D parallelStream() : Stream - Collection
★ ^A contains(Object o) : boolean - Set	★ ^A pollFirst() : Object - NavigableSet
★ ^A containsAll(Collection c) : boolean - Set	★ ^A pollLast() : Object - NavigableSet
★ ^A descendingIterator() : Iterator - NavigableSet	★ ^A remove(Object o) : boolean - Set
★ ^A descendingSet() : NavigableSet - NavigableSet	★ ^A removeAll(Collection c) : boolean - Set
★ ^A equals(Object o) : boolean - Set	● ^D removeIf(Predicate filter) : boolean - Collection
★ ^A first() : Object - SortedSet	★ ^A retainAll(Collection c) : boolean - Set
★ ^A floor(Object e) : Object - NavigableSet	★ ^A size() : int - Set
● ^D forEach(Consumer action) : void - Iterable	● ^D spliterator() : Spliterator - SortedSet
● getClass() : Class<?> - Object	● ^D stream() : Stream - Collection
★ ^A hashCode() : int - Set	★ ^A subSet(Object fromElement, Object toElement) : SortedSet - NavigableSet
★ ^A headSet(Object toElement) : SortedSet - NavigableSet	★ ^A tailSet(Object fromElement) : SortedSet - NavigableSet
★ ^A higher(Object e) : Object - NavigableSet	★ ^A toArray() : Object[] - Set
★ ^A isEmpty() : boolean - Set	★ ^A toArray(Object[] a) : Object[] - Set
	● toString() : String - Object
	● wait() : void - Object

Map

- 🟢 hashCode() : int - Map
- 🟢 isEmpty() : boolean - Map
- 🟡 merge(Object key, Object value, BiFunction remappingFunction) : Object - Map
- 🟢 notify() : void - Object
- 🟢 notifyAll() : void - Object
- 🟢 putAll(Map m) : void - Map
- 🟡 putIfAbsent(Object key, Object value) : Object - Map
- 🟡 remove(Object key, Object value) : boolean - Map
- 🟡 replace(Object key, Object value) : Object - Map
- 🟡 replace(Object key, Object oldValue, Object newValue) : boolean - Map
- 🟡 replaceAll(BiFunction function) : void - Map
- 🟢 size() : int - Map
- 🟢 toString() : String - Object
- 🟢 wait() : void - Object
- 🟢 wait(long timeout) : void - Object
- 🟢 wait(long timeout, int nanos) : void - Object

- 🟡 get(Object key) : Object - Map - 11%
- 🟡 put(Object key, Object value) : Object - Map - 7%
- 🟡 containsKey(Object key) : boolean - Map - 2%
- 🟡 remove(Object key) : Object - Map - 2%
- 🟡 keySet() : Set - Map - 1%
- 🟡 values() : Collection - Map - 1%
- 🟢 clear() : void - Map
- 🟡 compute(Object key, BiFunction remappingFunction) : Object - Map
- 🟡 computeIfAbsent(Object key, Function mappingFunction) : Object - Map
- 🟡 computeIfPresent(Object key, BiFunction remappingFunction) : Object - Map
- 🟢 containsValue(Object value) : boolean - Map
- 🟢 entrySet() : Set - Map
- 🟢 equals(Object o) : boolean - Map
- 🟡 forEach(BiConsumer action) : void - Map
- 🟢 getClass() : Class<?> - Object
- 🟡 getOrDefault(Object key, Object defaultValue) : Object - Map

SortedMap

🌟 ^A get(Object key) : Object - Map - 6%	🌟 ^A headMap(Object toKey) : SortedMap - SortedMap
🌟 ^A put(Object key, Object value) : Object - Map - 3%	🌟 ^A isEmpty() : boolean - Map
🌟 ^A entrySet() : Set - SortedMap - 2%	🌟 ^A lastKey() : Object - SortedMap
🌟 ^A keySet() : Set - SortedMap - 2%	🌟 ^D merge(Object key, Object value, BiFunction remappingFunction) : Object - Map
🌟 ^A size() : int - Map - 2%	🟢 notify() : void - Object
🌟 ^A values() : Collection - SortedMap - 2%	🟢 notifyAll() : void - Object
🌟 ^A clear() : void - Map	🌟 ^A putAll(Map m) : void - Map
🌟 ^A comparator() : Comparator - SortedMap	🌟 ^D putIfAbsent(Object key, Object value) : Object - Map
🌟 ^D compute(Object key, BiFunction remappingFunction) : Object - Map	🌟 ^A remove(Object key) : Object - Map
🌟 ^D computeIfAbsent(Object key, Function mappingFunction) : Object - Map	🌟 ^D remove(Object key, Object value) : boolean - Map
🌟 ^D computeIfPresent(Object key, BiFunction remappingFunction) : Object - Map	🌟 ^D replace(Object key, Object value) : Object - Map
🌟 ^A containsKey(Object key) : boolean - Map	🌟 ^D replace(Object key, Object oldValue, Object newValue) : boolean - Map
🌟 ^A containsValue(Object value) : boolean - Map	🌟 ^D replaceAll(BiFunction function) : void - Map
🌟 ^A equals(Object o) : boolean - Map	🌟 ^A subMap(Object fromKey, Object toKey) : SortedMap - SortedMap
🌟 ^A firstKey() : Object - SortedMap	🌟 ^A tailMap(Object fromKey) : SortedMap - SortedMap
🌟 ^D forEach(BiConsumer action) : void - Map	🟢 toString() : String - Object
🟢 getClass() : Class<?> - Object	🟢 wait() : void - Object
🌟 ^D getOrDefault(Object key, Object defaultValue) : Object - Map	🟢 wait(long timeout) : void - Object
🌟 ^A hashCode() : int - Map	🟢 wait(long timeout, int nanos) : void - Object

NavigableMap

- 🚀 lowerEntry(Object key) : Entry - NavigableMap - 8%
- 🚀 floorEntry(Object key) : Entry - NavigableMap - 6%
- 🚀 put(Object key, Object value) : Object - Map - 6%
- 🚀 subMap(Object fromKey, Object toKey) : SortedMap - NavigableMap - 6%
- 🚀 remove(Object key) : Object - Map - 5%
- 🚀 tailMap(Object fromKey, boolean inclusive) : NavigableMap - NavigableMap
- 🚀 ceilingEntry(Object key) : Entry - NavigableMap
- 🚀 ceilingKey(Object key) : Object - NavigableMap
- 🚀 clear() : void - Map
- 🚀 comparator() : Comparator - SortedMap
- 📦 compute(Object key, BiFunction remappingFunction) : Object - Map
- 📦 computeIfAbsent(Object key, Function mappingFunction) : Object - Map
- 📦 computeIfPresent(Object key, BiFunction remappingFunction) : Object - Map
- 🚀 containsKey(Object key) : boolean - Map
- 🚀 containsValue(Object value) : boolean - Map
- 🚀 descendingKeySet() : NavigableSet - NavigableMap
- 🚀 descendingMap() : NavigableMap - NavigableMap
- 🚀 entrySet() : Set - SortedMap
- 🚀 equals(Object o) : boolean - Map
- 🚀 firstEntry() : Entry - NavigableMap
- 🚀 firstKey() : Object - SortedMap
- 🚀 floorKey(Object key) : Object - NavigableMap
- 📦 forEach(BiConsumer action) : void - Map
- 🚀 get(Object key) : Object - Map
- 📦 getClass() : Class<?> - Object
- 📦 getOrDefault(Object key, Object defaultValue) : Object - Map
- 🚀 hashCode() : int - Map

- 🚀 headMap(Object toKey) : SortedMap - NavigableMap
- 🚀 headMap(Object toKey, boolean inclusive) : NavigableMap - NavigableMap
- 🚀 higherEntry(Object key) : Entry - NavigableMap
- 🚀 higherKey(Object key) : Object - NavigableMap
- 🚀 isEmpty() : boolean - Map
- 🚀 keySet() : Set - SortedMap
- 🚀 lastEntry() : Entry - NavigableMap
- 🚀 lastKey() : Object - SortedMap
- 🚀 lowerKey(Object key) : Object - NavigableMap
- 📦 merge(Object key, Object value, BiFunction remappingFunction) : Object - Map
- 🚀 navigableKeySet() : NavigableSet - NavigableMap
- 📦 notify() : void - Object
- 📦 notifyAll() : void - Object
- 🚀 pollFirstEntry() : Entry - NavigableMap
- 🚀 pollLastEntry() : Entry - NavigableMap
- 🚀 putAll(Map m) : void - Map
- 📦 putIfAbsent(Object key, Object value) : Object - Map
- 📦 remove(Object key, Object value) : boolean - Map
- 📦 replace(Object key, Object value) : Object - Map
- 📦 replace(Object key, Object oldValue, Object newValue) : boolean - Map
- 📦 replaceAll(BiFunction function) : void - Map
- 🚀 size() : int - Map
- 🚀 subMap(Object fromKey, boolean fromInclusive, Object toKey, boolean toInclusive) : NavigableMap
- 🚀 tailMap(Object fromKey) : SortedMap - NavigableMap
- 📦 toString() : String - Object
- 🚀 values() : Collection - SortedMap
- 📦 wait() : void - Object