

Sets, Multisets, and Multimaps

Syed Eqbal Alam

University of New Brunswick
Fredericton, New Brunswick, Canada

Overview

- 1 Sets
- 2 Multisets
- 3 Multimaps

Sets

- A set is an abstract data type.
- A set is a collection of elements that are not ordered; furthermore, it does not allow repeated elements.

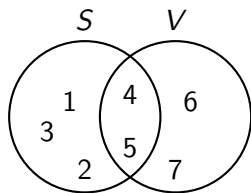
Example

Let us consider a course in which five students are enrolled, University IDs of all students can be stored in the set *Student*.

$Student = \{421113, 421110, 421115, 421122, 421118\}$.

Set operations

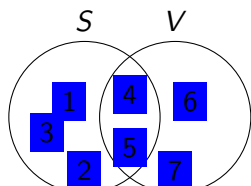
Let us consider two sets S and V . The set $S = \{1, 2, 3, 5, 4\}$ and the set $V = \{4, 5, 6, 7\}$.



A few basic set operations are:

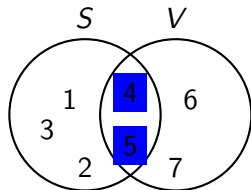
- (i) Union of two sets S and V : It is a set that contains all elements that are in the set S or the set V , without duplicate entries.

$$S \cup V = \{1, 2, 3, 4, 5, 7, 6\}.$$



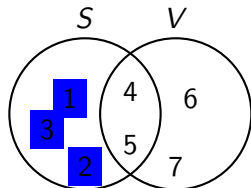
Set operations

- (ii) Intersection of two sets S and V : It is a set that contains all the common elements of the set S and the set V .



$$S \cap V = \{5, 4\}.$$

- (iii) Difference of two sets S and V : It is a set that contains all elements of the set S without the common elements of the set V .



$$S - V = \{1, 2, 3\}.$$

Java's Set interface

- (i) Set is an interface in Java, `java.util.Set`
- (ii) Some of the methods are:
 - `S.add(element)`: Adds the element to the set `S`.
 - `S.remove(element)`: Deletes the element from the set `S`.
 - `S.contains(element)`: Checks whether the element is there in the set `S` or not. Returns *true* or *false*.

Let `S` and `V` be two sets:

- `S.addAll(V)`: It adds all elements of the set `V` to the set `S`, without duplicate elements. It performs the union operation.
- `S.retainAll(V)`: It retains all common elements of the sets `V` and `S`. It performs the intersection operation.
- `S.removeAll(V)`: It removes common elements of the sets `V` and `S` from the set `S`. It performs the set difference operation.

Explore Oracle document on the Set interface at <https://docs.oracle.com/javase/tutorial/collections/interfaces/set.html>

Multisets

- It is an abstract data type.
- Multiset is similar to a set but it allows duplicate entries of an element.
- Multiset is also called a Bag.

Example

Let the multiset S be a collection of last names of four people T. Joy, S. Joy, U. James, and N. John, then $S = \{Joy, Joy, James, John\}$.

Multisets

Methods:

- `add(element)`: Adds an element to the multiset.
- `add(element, N)`: Adds N entries of the element to the multiset.
- `remove(element)`: Removes one occurrence of the element.
- `remove(element, N)`: Removes N occurrences of the element.
- `size()`: Returns the size of the multiset.

1. Explore Multiset interface at <https://guava.dev/releases/18.0/api/docs/com/google/common/collect/Multiset.html>.
2. You may also explore, <https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/util/Collection.html>.

Multimap

- Multimap is an abstract data type.
- It is similar to a map; however, it allows multiple values associated with a single key.

Example

Key 1 is associated with two values Value 1 and Value 2: (key1 , Value 1) and (key1 , Value 2).

Multimaps

A few of the basic methods are:

- `put(key, value)`: Adds the value without modifying the previous entry.
- `get(key)`: Returns all values associated with the key.
- `remove(key, value)`: Removes the value associated with the key.
- `size()`: returns the size of the multimap.

Explore <https://docs.oracle.com/middleware/11119/jdev/api-reference-esdk/oracle/javatools/util/MultiMap.html>.



Michael T. Goodrich and Roberto Tamassia and Michael H. Goldwasser, *Data Structures and Algorithms in Java*, 6th, 2014, Wiley.