

Table of Contents

- 1 Sorted Set
- 2 Usage
 - 2.1 Create
 - 2.1.1 Container Operations
 - 2.1.2 Set Operations
 - 2.1.3 Stringify

Sorted Set

A mutable set backed by a red-black tree.

Usage

Create

You can create an empty SortedSet or a SortedSet from other containers.

```
1
2  test {
3      let _set1 : @sorted_set SortedSet[Int] = @sorted_set new()
4      let _set2 = @sorted_set singleton(1)
5      let _set3 = @sorted_set from_array([1])
6
7  }
```

Container Operations

Add an element to the SortedSet in place.

```
1
2  test {
3      let set4 = @sorted_set from_array([1, 2, 3, 4])
4      set4 add(5)
5      let set6 = @sorted_set from_array([1, 2, 3, 4, 5])
6      assert_eq(set6 to_array(), [1, 2, 3, 4, 5])
7  }
```

Remove an element from the SortedSet in place.

```
1
2  test {
3      let set = @sorted_set from_array([3, 8, 1])
4      set remove(8)
5      let set7 = @sorted_set from_array([1, 3])
6      assert_eq(set7 to_array(), [1, 3])
7  }
```

Whether an element is in the set.

```
1
2  test {
3      let set = @sorted_set from_array([1, 2, 3, 4])
4      assert_eq(set contains(1), true)
5      assert_eq(set contains(5), false)
6  }
```

Iterates over the elements in the set.

```

1
2  test {
3      let arr = []
4      @sorted_set from_array([1, 2, 3, 4]) each(v => arr push(v))
5      assert_eq(arr, [1, 2, 3, 4])
6  }

```

Get the size of the set.

```

1
2  test {
3      let set = @sorted_set from_array([1, 2, 3, 4])
4      assert_eq(set size(), 4)
5  }

```

Whether the set is empty.

```

1
2  test {
3      let set : @sorted_set SortedSet[Int] = @sorted_set new()
4      assert_eq(set is_empty(), true)
5  }

```

Set Operations

Union, intersection and difference of two sets. They return a new set that does not overlap with the original sets in memory.

```

1
2  test {
3      let set1 = @sorted_set from_array([3, 4, 5])
4      let set2 = @sorted_set from_array([4, 5, 6])
5      let set3 = set1 union(set2)
6      assert_eq(set3 to_array(), [3, 4, 5, 6])
7      let set4 = set1 intersection(set2)
8      assert_eq(set4 to_array(), [4, 5])
9      let set5 = set1 difference(set2)
10     assert_eq(set5 to_array(), [3])
11 }

```

Determine the inclusion and separation relationship between two sets.

```

1
2  test {
3      let set1 = @sorted_set from_array([1, 2, 3])
4      let set2 = @sorted_set from_array([7, 2, 9, 4, 5, 6, 3, 8, 1])
5      assert_eq(set1 subset(set2), true)
6      let set3 = @sorted_set from_array([4, 5, 6])
7      assert_eq(set1 disjoint(set3), true)
8  }

```

Stringify

SortedSet implements `to_string` (i.e. Show trait), which allows you to directly output it.

```
1
2  test {
3    let set = @sorted_set from_array([1, 2, 3])
4    assert_eq(set to_string(), "@sorted_set.from_array([1, 2, 3])")
5  }
```