Problem-1

Returns and removes the entry associated with the specified key from a TreeMap  
Eg:TreeMap: {One=1, Three=3, Two=2}  
Removed value = 2  
Is the entry {Three=3} removed? True  
Updated TreeMap: {One=1}

Code-

package pkg;

import java.util.TreeMap;

public class TreeMapEx {

public static void main(String[] args) {

// Create and insert values in the TreeMap

TreeMap<String, Integer> treeMap = new TreeMap<>();

treeMap.put("One", 1);

treeMap.put("Three", 3);

treeMap.put("Two", 2);

// Display the original TreeMap

System.out.println("Original TreeMap: " + treeMap);

// Remove the entry with key "Two"

Integer removedValue = treeMap.remove("Two");

// Display the removed value

System.out.println("Removed value = " + removedValue);

// Check if the entry with key "Three" is removed

boolean isThreeRemoved = !treeMap.containsKey("Three");

System.out.println("Is the entry {Three=3} removed? " + isThreeRemoved);

// Display the updated TreeMap

System.out.println("Updated TreeMap: " + treeMap);

}

}

Output-

Original TreeMap: {One=1, Three=3, Two=2}

Removed value = 2

Is the entry {Three=3} removed? false

Updated TreeMap: {One=1, Three=3}

Problem-2

Navigate over the elements of the treemap.(All methods)

Code-

package pkg;

import java.util.\*;

public class TreeMapNavigation {

public static void main(String[] args) {

// Create the TreeMap

TreeMap<String, Integer> treeMap = new TreeMap<>();

treeMap.put("One", 1);

treeMap.put("Two", 2);

treeMap.put("Three", 3);

treeMap.put("Four", 4);

treeMap.put("Five", 5);

// 1. Iterate over keys using keySet()

System.out.println("Iterating over keys using keySet():");

for (String key : treeMap.keySet()) {

System.out.println("Key: " + key + ", Value: " + treeMap.get(key));

}

// 2. Iterate over entries using entrySet()

System.out.println("\nIterating over entries using entrySet():");

for (Map.Entry<String, Integer> entry : treeMap.entrySet()) {

System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());

}

// 3. Iterate over values using values()

System.out.println("\nIterating over values using values():");

for (Integer value : treeMap.values()) {

System.out.println("Value: " + value);

}

// 4. Navigating using NavigableMap methods

System.out.println("\nNavigating with NavigableMap methods:");

// First key

System.out.println("First Key: " + treeMap.firstKey());

// Last key

System.out.println("Last Key: " + treeMap.lastKey());

// Higher than "Three" key

System.out.println("Higher than 'Three' key: " + treeMap.higherKey("Three"));

// Lower than "Three" key

System.out.println("Lower than 'Three' key: " + treeMap.lowerKey("Three"));

// Floor key of "Three" (greatest key <= "Three")

System.out.println("Floor key of 'Three': " + treeMap.floorKey("Three"));

// Ceiling key of "Three" (smallest key >= "Three")

System.out.println("Ceiling key of 'Three': " + treeMap.ceilingKey("Three"));

// Submap from "Two" to "Four" (inclusive)

try {

System.out.println("\nSubMap from 'Two' to 'Four':");

SortedMap<String, Integer> subMap = treeMap.subMap("Two", true, "Four", true);

for (Map.Entry<String, Integer> entry : subMap.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

} catch (IllegalArgumentException e) {

System.out.println("Error: " + e.getMessage());

}

// NavigableMap view (Descending order)

System.out.println("\nDescending order of TreeMap:");

NavigableMap<String, Integer> descendingMap = treeMap.descendingMap();

for (Map.Entry<String, Integer> entry : descendingMap.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

// TailMap from "Three" (inclusive)

System.out.println("\nTailMap from 'Three' (inclusive):");

SortedMap<String, Integer> tailMap = treeMap.tailMap("Three", true);

for (Map.Entry<String, Integer> entry : tailMap.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

// HeadMap up to "Three" (exclusive)

System.out.println("\nHeadMap up to 'Three' (exclusive):");

SortedMap<String, Integer> headMap = treeMap.headMap("Three", false);

for (Map.Entry<String, Integer> entry : headMap.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

}

}

Output-

Iterating over keys using keySet():

Key: Five, Value: 5

Key: Four, Value: 4

Key: One, Value: 1

Key: Three, Value: 3

Key: Two, Value: 2

Iterating over entries using entrySet():

Key: Five, Value: 5

Key: Four, Value: 4

Key: One, Value: 1

Key: Three, Value: 3

Key: Two, Value: 2

Iterating over values using values():

Value: 5

Value: 4

Value: 1

Value: 3

Value: 2

Navigating with NavigableMap methods:

First Key: Five

Last Key: Two

Higher than 'Three' key: Two

Lower than 'Three' key: One

Floor key of 'Three': Three

Ceiling key of 'Three': Three

SubMap from 'Two' to 'Four':

Error: fromKey > toKey

Descending order of TreeMap:

Two: 2

Three: 3

One: 1

Four: 4

Five: 5

TailMap from 'Three' (inclusive):

Three: 3

Two: 2

HeadMap up to 'Three' (exclusive):

Five: 5

Four: 4

One: 1