We have a **TreeMap** in which we need to store the stock data. The key is the company's name, and value is the price of the stock of that company.

By default, when we store the Map's stock data, it will be stored in sorted order by key. But we need to store the data such that it is sorted by value. This means that the first element will be the company with the lowest stock price, and the last element will be the company with the highest stock price.

We know that while creating a **TreeMap** object we can provide a **Comparator** implementation that will be used to sort the elements. We can write a **Comparator** implementation so that it sorts the elements based on values instead of keys.

The below example shows how we can sort the elements on the basis of values. We have created a **Comparator** implementation that compares the values for the two keys passed to it.

> Please note that in the below implementation, we are returning 1 if both the values are the same. The reason for doing this is that if two values are the same, then the TreeMap will consider it as duplicate, and it will not insert the keys in the Map.

```
import java.util.Comparator;
import java.util.TreeMap;

public class TreeMapDemo {

    public static TreeMap<String, Integer> sortByValues(TreeMap<String, Integer> map) {

        Comparator<String> valueComparator = new Comparator<String>() {

            // return comparison results of values of two keys
            public int compare(String k1, String k2)
            {
                int comp = map.get(k1).compareTo(
                    map.get(k2));
                if (comp == 0)
                    return 1;
                else
                    return comp;
            }

        };

        TreeMap<String, Integer> mapSortedByValues = new TreeMap<>(valueComparator);

        mapSortedByValues.putAll(map);
        return mapSortedByValues;

    }
```

Run Save Reset

The above logic can be a bit simplified if we use the lambda expressions while creating Comparator as shown below.

```
import java.util.Comparator;
import java.util.TreeMap;

public class TreeMapDemo {

    public static TreeMap<String, Integer> sortByValues(TreeMap<String, Integer> map) {

        Comparator<String> valueComparator = (k1, k2) -> {

            int comp = map.get(k1).compareTo(map.get(k2));
            if (comp == 0)
                return 1;
            else
                return comp;
        };

        TreeMap<String, Integer> mapSortedByValues = new TreeMap<>(valueComparator);

        mapSortedByValues.putAll(map);
        return mapSortedByValues;

    }

    public static void main(String args[]) {

        TreeMap<String, Integer> map = new TreeMap<>();
        map.put("Oracle", 43);
        map.put("Microsoft", 56);
```

Run Save Reset