SET, UNORDERED SET & MULTISET

- Map store (key, values) while set store only keys.

Set is collection of elements/key.

(i) SET / ordered SET

Set stores the unique elements. In set values/elements are stored in sorted form.

```
// Decleration of set
    set < string > s;

// Insertion of value in set
    s.insert ("abc"); // T.C. = O(log n)
    s.insert ("zsdf");
    s.insert ("pqr");

// searching or geting value from set.
    auto it = s.find ("abc"); // T.C. = O(log n)
```

↳ this find f^n will return the itrator of the value if exist otherwise return s.end();

```
// print entire set
void print (set<string> &s)
{
    // way-1
    for (string value : s)
    {
        cout << value << endl;
    }

    // way-2
    for (auto it = s.begin(); it != s.end(); it++)
    {
        cout << (*it) << endl;
    }
}

// Erase value from set
auto it = s.find ("abc");
if (it != s.end()){
// way-1 → s.erase (it);
                    ↑ iterator
{
// way-2
s.erase ("abc")
          ↑ value.
```

Imp.

The internal implementation of sets is done by using Red-Black Tree.

Red-Black Trees are self balancing trees.

(ii) Unordered Set.

unordered_set <string> s;

They also store the unique values but order is not maintied. & all the time complexity are O(1).

The internal implemention of unordered set is done by hash tables.

imp → As like in unordered map we can't use complex data structure into them

like unordered_map < pair<int, int>, int> ✗

same we use only primitive data type in unorded_set.

(iii) Multiset

multiset <string> s.

In multiset we can insert multiple values of same data like "abc" & "abc" can be inserted two times.

And all other T.C. are same O(log n).

Also store values into sorted order.

// find fn

auto it = S.find ("abc"); // This find fn will return the itrator of the first value if duplicats are presents.

S.erase (it); → This erase fn will erase only that "abc" pointing by it itrator.

// erase fn

S.erase ("abc"); // This erase fn will delete all the "abc" in multiset

Imp जब erase fn में value pass करते है तो internaly वो उस iterator को find करता है & तब तक करता रहता है जब तक iterator milna बंद ना हो जाये।