→ SETS

→ Initializing a set
→ Set is same as map, remove all values from a map and keep only keys. It's a set now.

```
SET<STRING> S;
```

→ Inserting values in set

```
S.INSERT("abc"); // O(log(n))
S.INSERT("def");
S.INSERT("ghi");
S.INSERT("xyz");
```

→ If we try to add the same value which already exists in set, it won't get added as set stores unique values only.

```
S.INSERT("xyz");
```

→ Printing size of a set

```
COUT << S.SIZE() << "\n";
```

→ Printing set

```
FOR (AUTO &x : S)
{
    COUT << x << "\n";
}
```

→ FIND() function takes a value and returns an iterator corresponding to it. If there is no value corresponding to it, it returns

END() iterator.

```
AUTO IT = S.FIND("XYZK");   O(log(n))
IF (IT == S.END())
{
    COUT << "NO VALUE" << "\n";
}
ELSE
{
    COUT << *IT << "\n";
}
```

→ ERASE() function takes a value or it iterator and removes that value if that exist.

```
S.ERASE("GHI");
```

→ CLEAR() clears the whole set

```
S.CLEAR();
```

NOTE: Set stores unique value only, and are stored in a sorted manner.