

Disjoint-set Datastructures

* Set: A collection of distinct and well-defined elements.

$$\left. \begin{array}{l} S_1 = \{1, 2, 3, 4, 5\} \\ S_2 = \{a, b, c, d\} \end{array} \right\} \text{(Sets)}$$

* Disjoint-set: (Partitioned-set)

$$\left. \begin{array}{l} DS_1 = \{ \{1\}, \{2\}, \{3\}, \{4\}, \{5\} \} \\ DS_2 = \{ \{1, 2, 3\}, \{4, 5\} \} \\ DS_3 = \{ \{a, b, c\}, \{d\} \} \end{array} \right\} \text{(Disjoint sets)}$$

No elements are in common in the subsets of DS_1 , DS_2 and DS_3 .

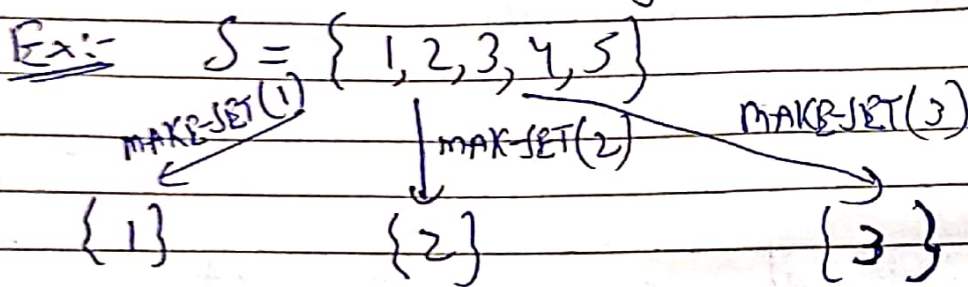
* Disjoint-set as a Datastructure:

A datastructure which can be viewed as a collection of sets, wherein no element appears in more than one set.

So, DS_1 , DS_2 and DS_3 are disjoint-sets as shown earlier.

* Operations supported by Disjoint-set data structure:

I) MAKE-SET(a) :- makes singleton sets, containing each element of a set exactly once.



Elements of a set are "isolated".

II) FIND-SET(a) :- finds representative (owner) of a set containing a.

Ex:- $S = \{1, \underline{2}, 3, 4, 5\}$
(2 is the representative of S)

So, $\left. \begin{array}{l} \text{FIND-SET}(3) = 2 \\ \text{FIND-SET}(5) = 2 \end{array} \right\} (\because 2 \text{ is representative of } S)$

Ex: $S_1 = \{ \{1, 2\}, \{3, 4\}, \{5\} \}$

$\text{FIND-SET}(3) = 3$

$\text{FIND-SET}(1) = 2$

$\text{FIND-SET}(5) = 5$

$\text{FIND-SET}(4) = 3$

III) UNION(a, b) : combines sets containing a and b and collects them into a new set. It means

that if a is present in set S_a and b is present in set S_b ; then this operation merges a and b into a single set (either S_a or S_b or maybe new set)

Generally S_a or S_b are taken and new set is not created.

Since FIND-SET and UNION operations are critical in disjoint-set datastructures, these structures are famously known as

"Union-Find" datastructures.