Assignment No - 2 (Group A)

Problem statement: To create ADT that implements the “set” concept

a) Add (new element) - place a value into a set, b) Remove (element) - remove the value
c) Contains (element) - return true if element is in a collection, d) Size () - return number of values in collection iterator() return an iterator used to loop over collection, e) Intersection of two sets, f) Union of two set g) Difference between two sets, h) Subset

Pre-requisite

Knowledge of Python programming

Knowledge of STL, set operations

Objective

To understand how Create, Display and perform various operations on set.

Input

Set A elements and Set B elements

Output

As per set operations

Software and Hardware requirements:-

1. **Operating system:** Linux- Ubuntu 16.04 to 17.10, or Windows 7 to 10,
2. **RAM-** 2GB RAM (4GB preferable)
3. You have to install **Python3** or higher version

Theory-

What is abstract data type?

An abstract data type is an abstraction of a data structure that provides only the interface to which the data structure must adhere. The interface does not give any specific details about something should be implemented or in what programming language.

Python Set

A Python set is the collection of the unordered items. Each element in the set must be unique, immutable, and the sets remove the duplicate elements. Sets are mutable which means we can modify it after its creation.

Set (): Creates a new set initialized to the empty set.

Length (): Returns the number of elements in the set, also known as the cardinality. Accessed using the len () function.

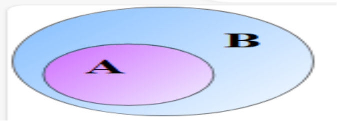
Contains (element): Determines if the given value is an element of the set and returns the appropriate Boolean value.

Add (element): Modifies the set by adding the given value or element to the set if the element is not already a member. If the element is not unique no action is taken and the operation is skipped.

Remove (element): Removes the given value from the set if the value is contained in the set and raises an exception otherwise.

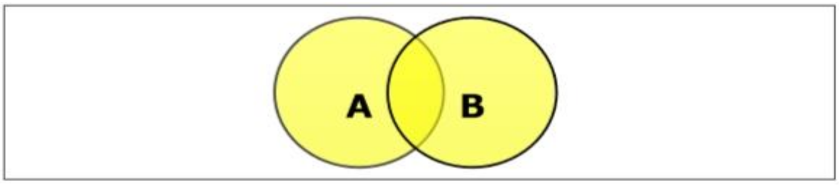
IsSubsetOf (setB): Determines if the set is a subset of another set and returns a Boolean value. For set A to be a Boolean value. For set A to be a subset of B , all elements in A must also be elements in B.

Example: Find all the subsets of set A = {1,2,3,4}
Solution: Given, A = {1,2,3,4}
Subsets of A are {},{1}, {2}, {3}, {4},{1,2}, {1,3}, {1,4}, {2,3},{2,4}, {3,4},{1,2,3}, {2,3,4}, {1,3,4}, {1,2,4} {1,2,3,4}



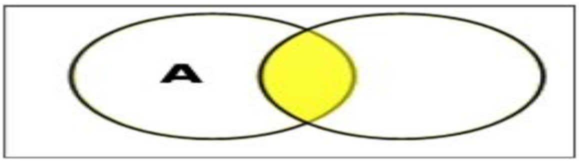
Union (set B): Creates and returns a new set that is the union of this set and set B. The new set created from the union of two sets. A and B, contains all elements in A plus those elements in B that are not in A Neither set A nor set B is modified by this operation.

Example − If A = { 10, 11, 12, 13 } and B = { 13, 14, 15 }, then A ∪ B = { 10, 11, 12, 13, 14, 15 }. (The common element occurs only once)



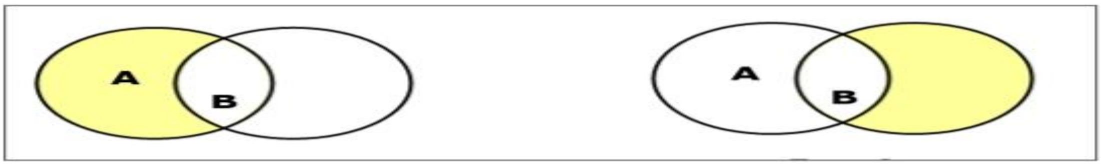
Intersect (setB): Creates and returns a new set that is the intersection of this set and setB . the intersection of sets A and B contains only those elements that are in both A and B. Neither set A nor set B is modified by this operation.

- Example − If A = { 11, 12, 13 } and B = { 13, 14, 15 }, then A ∩ B = { 13 }.



Difference (set B): Creates and returns a new set that is the difference of this and setB. The set difference, A-B, contains only those elements that are in A but not in B. Neither set A nor set B is modified by this operation.

- **Example −** If A = { 10, 11, 12, 13 } and B = { 13, 14, 15 }, then (A - B) = { 10, 11, 12 } and (B - A) = { 14, 15 }. Here, we can see (A - B) ≠ (B - A)



Iterator (): Creates and returns an iterator that can be used to iterator over the collection of items.

Algorithm:

1. To add elements in set:

i) Enter Element to add.

ii) Check element with all elements form i=0 to n

iii) if match found ask for another element.

iv) if no match found till last position then store element at last position.

2. To remove element from set

i) Ask element which is to be removed.

ii) if it is found at any position then shift all elements towards left by one position.

iii) If no match for element is found then display element not found message.

3. To check for contained element

i) Enter Element to check.

ii) Check element with all elements of set from position 0 to 1

iii) if match found display message stating element is found

iv) if no match found till last position then display not found message.

4. Intersection of sets

i) pick first element of set 1 compare with all elements of set2

ii) if match found at any position then put that element in intersection set.

iii) If match not found till end then ignore element and select next element in set 1

iv) Repeat steps i) , ii) and iii) for all elements of set 1

v) print intersection set.

5. Union of sets

i) Copy all elements of set 1 in union set

ii) Pick element of set2 compare with all elements of set1

iii) if match not found at any position then put that element in union set at last position.

iv) If match found at any position then ignore element and choose next element in set2

v) Repeat steps ii) , iii) and iv) for all elements of set 2

vi) print union set.

6. Difference of sets

i) Take empty set to collect difference (set1-set2)

ii) pick element of set1 compare with all elements of set2

iii) if match not found at any position then put that element in difference set.

iv) If match found at any position then ignore element and choose next element in set1

v) Repeat steps ii) , iii) and iv) for all elements of set 1

vi) print difference set.

7. Subset(Compare if set 2 is subset of set1)

i) pick up element of set 2 compare with all elements of set1

ii) if match found at any position then stop comparing and select next element in set 2

iii) If match not found till end then print message that set2 is not subset of set1

iv) Repeat steps i) , ii) and iii) for all elements of set 2

Conclusion:

We have studied in depth, the concept of ADT, function of sets, the implementation of set functions and working of set function.