

Assignment No. 9.

Title: To implement set as ADT.

Aim: To learn & implement set as ADT.

Problem statement: To create ADT that implement SET CONCEPT

(use array / LL)

a) add (new element) place value in set.

b) remove element

c) search element

d) size of set

e) intersection of two set

f) union of two set

g) difference of two set

h) subset.

Objective: i) To learn the SET as ADT & understand its different applications.

ii) To implement the concept SET using array or linked list.

Theory:

• SET: A set is an unordered collection of different element. A set can be written explicitly by listing its element using set brackets.

Set of all the numbers.

Representation of set:

i) Roaster form

ii) set-builder form.

i) Tabular form:- e.g. set of vowels
 $S = \{ 'a', 'e', 'i', 'o', 'u' \}$

ii) Set-builder form:-
e.g.:- $\{ x : P(x) \}$ is written as
 $A = \{ x : x \text{ is vowel in english} \}$

• Types of set:-

i) finite:- A set which contain definite number of element is called finite set.

ii) infinite:- A set which contain infinite number of element is called infinite set.
e.g. set of natural number.

• Operations on sets:-

i) Union:- It contain all elements from set 1 & set 2 excluding the duplicate data elements.

e.g.) Set 1 = $\{ 10, 20, 30, 40 \}$

Set 2 = $\{ 30, 40, 50, 60 \}$

Union = $\{ 10, 20, 30, 40, 50, 60 \}$

ii) Intersection:- It contain common element in both.

e.g. Set 1 = $\{ 10, 20, 30 \}$

Set 2 = $\{ 20, 30, 40, 50 \}$

data 1 + size =

True.

intersection = $\{20, 30\}$

3) difference :-

it return the element which in set one but not in set 2.

e.g. :- set 1 = $\{10, 20, 30\}$

set 2 = $\{30, 40, 50\}$

difference = $\{10, 20\}$

4) subset :-

it return true if data from set A & from B are same.

e.g. set 1 = $\{10, 20, 30\}$

set 2 = $\{10, 20, 30\}$

set 1 is subset of set 2.

Algorithms :-

i) To add an element :-

step 1.) check the size of set with max if it is true
return set is full

else

insert data into array

data[++size] = element

step 2.) return true.

step 3.) END

2) deletion:

step 1.) start

step 2.) if size of array is -1
print "underflow"

else

for (i := 0 to n)

if (data[i] == element)

perform shift to the last index

else

~~it~~ continue

step 3.) decrement size by 1.

step 4.) END

3) intersection:

step 1.) Initialize flag = 0

step 2.) create another empty set.

step 3.) while (set 1 is not empty)

while (set 2 is not empty)

if (data[set 1] == data[set 2])

insert data in set 3

break

end while

end while

step 4.) return set 3

step 5.) END

4) Union :-

Step 1.) initialize flag = 0.

Step 2.) create another empty set.

Step 3.) while (set 1 is not empty)

flag = 0

while (set 2 is not empty)

if (data[set 1] == data[set 2])

flag = 1

break

end if

end while

end while

Step 4.) if (flag == 1)

insert element element in set 3 ; go to 3 until 1 & 2 empty

Step 5.) END

5) Difference :-

Step 1.) initialize flag = 0

Step 2.) create another empty set.

Step 3.) while (set 1 is not empty)

flag = 0

while (set 2 is not empty)

if (data[set 1] == data[set 2])

flag = 1

break

end if

end while

if (flag == 1)

insert data[set 1] in set 3
end if
end while

step 4.) END

6) Search :-

step 1.) initialize flag to 0.

step 2.) Let element to be searched x .

step 3.) for ($i = 0$ to n)
if data[i] == x

flag = 1

break

end if

end for

step 4.) if (flag == 1)

found

else

not found

step 5.) stop.

7) subset :-

step 1.) initialize flag = 0

step 2.) while (Set 1 is not empty)

while (Set 2 is not empty)

if (data[i] == data[j])

flag = 1

break

end if
end while
end while

if (flag = 1)
~~if~~ flag++

step 3:) if (flag == size of set 1)
return true
else
return false.

step 4:) END

• Application & time complexity:-

i) many mathematical concepts can be defined through SETS.
ii) In structures like graphs, manifolds, rings & vector spaces, SETS are used.

iii) Theory of mathematical relation can be described in set

Union - intersection - difference - subsets $\Rightarrow O(n)$

Conclusion:-

• Set ADT are type of abstract data type that allow you to store a list of non-repeated value. unlike array set are unindexed & unordered. so through the assignment we have implemented SET as ADT.