Skill Development Lab-II (2018-2019)

BRACT's

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE – 48

An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune

SD(LP-II) ASSIGNMENT (S.Y.B. Tech. – DIV: C)

Name.: Prathamesh Gaikwad;
Roll no.: 223019;
GR. No.: 17u416; Batch: C1

Assignment 1:

Aim :

To create ADT that implement the set concept.

a . add (newelement) –place a value into the set.

b . remove (element).

c . contains (element) return true if element is in collection.

d . size() return number of values in collection.

e . intersection of two sets.

g . difference between two sets.

h . subset.

Objective :

We have to implement this using basic data structure.

Theory : In computer science , a set is an abstract data type that can store unique values , without anny particular order . It is a computer implementation of the mathematical concept of a finite set . some set data structures are designed for static orfrozen sets that do not change after they are constructed.

Applications :

- Hash function.
- Spelling checker.

ALGORITHM:

• FOR INTERSECTION:

Step 1: Take an empty set (intersection set) Step 2: pass each element of set-2 and the entire set-1 to the function member() Step 3: if it returns true, Add that element to the intersection set

• FOR UNION:

Step 1: Take an empty set (union set) Step 2: copy all the elements of set1 to this new set Step 3: traverse through the set2 and pass each element of set-2 along with the entire set-1 to the function member(), and if it returns false then add that specified element to the union set

• FOR CONTAINS:

Step1 : take the number as input which you want to search Step 2 : enter 1 for searching in set-1 or 2 for searching in set-2 Step 3 : initialise i=0 Step 4: traverse the set-1 or set-2 till the end depending on whether the input was 1 or 2 after passing the element and that set to the function member() Step 5: if element found then display element is present

• FOR SUBSET:

Step 1: Enter 1 if you want to check if set 2 is subset of 1, or enter 2 if you want to check if set-1 is subset of set-2.

Step 2: Depending on input we will traverse the set(which has to be the subset) until its end, by passing each element of this set and other set to the function member().

Step 3: If member() returns true, then continue else return false

Step 4: If false, then display “it is a subset” else display “it is not”

• FOR DIFFERENCE :

Step 1: Initialise the difference set to 0, difference set contains all the element. which are in set-1 but not in set-2

Step 2: Traverse the entire set-1 and, pass each element of this set and the set-2 to the function member()

Step 3: if it returns false then add this element to the difference set

• FOR REMOVE:

Step 1: enter 1 or 2 for removing element from set-1 or set-2 respectively Step 2: enter the index from which you want to remove the element Step 3: if entered position is less than the size of the set then move all the elements to their left from the position at which you want to remove the element and just decrease the size of the set else, entered position is equal to the size of the set then just decrease the size

• FOR SIZE:

step 1: show the 0th index of the set which contains the size of our set

Program :

```
#include<iostream>

using namespace std;

struct set

{
```

```
int s[10],size;
}s1,s2,s3,s4,s5;

void insert()
{int ch,key;
cout<<"\n Enter element to be insert : ";cin>>key;
cout<<"\n Which set : ";cin>>ch;
switch(ch)
{
case 1:s1.s[s1.size++]=key;break;
case 2:s2.s[s2.size++]=key;break;
default:cout<<"\nWrong set ";
}}
void remove()
{
int ch,key,i;
cout<<"\n Enter element to be remove : ";cin>>key;
cout<<"\n Which set : ";cin>>ch;
switch(ch)
{
case 1:for(i=0;i<10 && s1.s[i]!=key;i++)
     {}
```

```
      while(i<10)
      {s1.s[i]=s1.s[i+1];i++;}
      s1.size--;
      break;
case 2:for(i=0;i<10 && s1.s[i]!=key;i++){}
      while(i<10)
      {s2.s[i]=s2.s[i+1];i++;}
      s2.size--;
      break;
default:cout<<"\nWrong set ";
}}
void contain()
{
int ch,key,i,f=0;
cout<<"\n Enter element to be search : ";cin>>key;
cout<<"\n Which set : ";cin>>ch;
switch(ch)
{
case 1:for(i=0;i<10;i++)
      {
       if(s1.s[i]==key)
       {cout<<endl<<key<<" is present in set 1.";f=1;break;}
```

```
        }
       if(f==0)
       cout<<endl<<key<<" is not present in set 1.";
       break;
case 2:for(i=0;i<10;i++)
       {
        if(s2.s[i]==key)
        {cout<<endl<<key<<" is present in set 2.";f=1;break;}
         }
       if(f==0)
       cout<<endl<<key<<" is not present in set 2.";
       break;
default:cout<<"\nWrong set ";
}}
void size()
{
cout<<"\n size of set 1 : "<<s1.size;
cout<<"\n size of set 2 : "<<s2.size;
cout<<endl;
}
void printset()
{int i;
```

```
cout<<endl<<"set1 :";;
for(i=0;i<s1.size;i++)
cout<<" "<<s1.s[i];
cout<<endl<<"set2 :";;
for(i=0;i<s2.size;i++)
cout<<" "<<s2.s[i];
cout<<endl;
}
void intersect()
{
int i,j,f=0;
s3.size=0;
for(i=0;i<s1.size;i++)
{
for(j=0;j<s2.size;j++)
{
if(s1.s[i]==s2.s[j])
{s3.s[s3.size++]=s1.s[i];f=1;}
}}
cout<<"\n intersection of set : ";
printset();
cout<<"\n is ";
```

```
if(f==0)
cout<<"NULL. \n";
else
{
for(i=0;i<s3.size;i++)
cout<<" "<<s3.s[i];
cout<<endl;
}}
void uni()
{
int i,j,f;
s4.size=0;
for(i=0;i<s1.size;i++)
{s4.s[s4.size++]=s1.s[i];}

for(i=0;i<s2.size;i++)
{f=0;
for(j=0;j<s1.size;j++)
{
if(s2.s[i]==s1.s[j])
{f=1;}
}
```

```
if(f==0)
{s4.s[s4.size++]=s2.s[i];}
}
cout<<"\n union of set : ";
printset();
cout<<"\n is ";
for(i=0;i<s4.size;i++)
cout<<" "<<s4.s[i];
cout<<endl;
}
void difference()
{
int i,j,f;
cout<<"\n set1-set2 is ";
for(i=0;i<s1.size;i++)
{f=0;
for(j=0;j<s2.size;j++)
{
if(s1.s[i]==s2.s[j])
f=1;
}
if(f==0)
```

```
cout<<s1.s[i]<<" ";
}
cout<<endl;
cout<<"\n set2-set1 is ";
for(i=0;i<s2.size;i++)
{f=0;
for(j=0;j<s1.size;j++)
{
if(s2.s[i]==s1.s[j])
f=1;
}
if(f==0)
cout<<s2.s[i]<<" ";
}}
void subset()
{
int i,j,f,m;
for(i=0;i<s1.size;i++)
{f=0;
for(j=0;j<s2.size;j++)
{
if(s1.s[i]==s2.s[j])
```

```
{f=1;}
}
if(f==0)
{cout<<"\nset1 is not subset of set 2.\n";break;}
}
if(f==1)
{cout<<"\nset1 is subset of set 2.\n";}
for(i=0;i<s2.size;i++)
{f=0;
for(j=0;j<s1.size;j++)
{
if(s2.s[i]==s1.s[j])
{f=1;}
}
if(f==0)
{cout<<"\nset2 is not subset of set 1.\n";break;}
}
if(f==1)
{cout<<"\nset2 is subset of set 1.\n";}
}
int main()
{
```

```
int i,ch,m,n;
cout<<"How many elements you want to enter in set 1 : ";cin>>m;
s1.size=m;
cout<<" \n Enter elements of set 1 : ";
for(i=0;i<m;i++)
cin>>s1.s[i];
cout<<" \n How many elements you want to enter in set 2 : ";cin>>n;
s2.size=n;
cout<<" \n Enter elements of set 2 : ";
for(i=0;i<n;i++)
cin>>s2.s[i];
while(1)
{
cout<<"\n 1. Insert \n 2. Remove \n 3. Contains(search) \n 4. size \n 5. print set elements \n 6. intersection \n 7. union \n 8. difference \n 9. subset \n 10. exit \n Enter your choice : ";
cin>>ch;
if(ch==10)
break;
switch(ch)
{
case 1:insert();break;
```

```
case 2:remove();break;

case 3:contain();break;

case 4:size();break;

case 5:printset();break;

case 6:intersect();break;

case 7:uni();break;

case 8:difference();break;

case 9:subset();break;

default:cout<<"Wrong choice";

}}

return 0;

}
```

Output :

```
How many elements you want to enter in set 1 : 3

 Enter elements of set 1 : 1
2
3

 How many elements you want to enter in set 2 : 3

 Enter elements of set 2 : 4
5
6


 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 1

 Enter element to be insert : 7

 Which set : 1

 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 5

set1 : 1 2 3 7
set2 : 4 5 6

 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 2

 Enter element to be remove : 6
```

```
 Which set : 2

 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 5

set1 : 1 2 3 7
set2 : 4 5

 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 3

 Enter element to be search : 5

 Which set : 1

5 is not present in set 1.
 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 6

 intersection of set :
set1 : 1 2 3 7
set2 : 4 5

 is NULL.

 1. Insert
 2. Remove
 3. Contains(search)
 4. size
```

```
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 7

 union of set :
set1 : 1 2 3 7
set2 : 4 5

 is  1 2 3 7 4 5

 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 8

 set1-set2 is 1 2 3 7

 set2-set1 is 4 5
 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice : 9

set1 is not subset of set 2.

set2 is not subset of set 1.

 1. Insert
 2. Remove
 3. Contains(search)
 4. size
 5. print set elements
 6. intersection
 7. union
 8. difference
 9. subset
 10. exit
 Enter your choice :
```

Conclusion : Thus we have studied set theory using basic data structure.