RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	CSC201A
Course Name	Discrete Mathematics-1
Programme	B. Tech.
Department	Computer Science and Engineering
Faculty	Engineering and Technology

Name of the Student	Deepak R
Reg. No	18ETCS002041
Semester/Year	3rd/2019
Course Leader/s	Dr Sahana P. Shankar

Declaration Sheet			
Student Name	Deepak R		
Reg. No	18ETCS002041		
Programme	B. Tech.	Semester/Year	3rd/2019
Course Code	CSC201A		
Course Title	Discrete Mathematics-1		
Course Date		to	
Course Leader	Dr Sahana P. Shankar		

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	24-10-2019
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Engineering and Technology			
Ramaiah University of Applied Sciences			
Department	Computer Science and Engineering	Programme	B. Tech.
Semester/Batch	3rd/2019		
Course Code	CSC201A	Course Title	Discrete Mathematics-1
Course Leader(s)	Ms Sahana P. Shankar and Ms.Supriya		

Questions	Marking Scheme		Marks		
			Max Marks	First Examiner Marks	Moderator
1					
	1.1	Development of a Python program with comments to create sets and perform the specified operations	5		
	1.2	Illustration using Venn diagrams	2		
		Question 1 Max Marks	**7**		
2					
	2.1	Justification of the statement with appropriate reasoning	2		
	2.2	Solution to the example problem	1		
		Question 2 Max Marks	**3**		
		Total Assignment Marks	**10**		

Course Marks Tabulation				
Question	**First Examiner**	**Remarks**	**Moderato r**	**Remarks**
1				
2				
Marks (Max 10)				
Signature of First Examiner				**Signature of Moderator**

Please note:

1. Documental evidence for all the components/parts of the assessment such as the reports, photographs, laboratory exam / tool tests are required to be attached to the assignment report in a proper order.
2. The First Examiner is required to mark the comments in RED ink and the Second Examiner's comments should be in GREEN ink.
3. The marks for all the questions of the assignment have to be written only in the **Component – CET B: Assignment** table.
4. If the variation between the marks awarded by the first examiner and the second examiner lies within +/- 3 marks, then the marks allotted by the first examiner is considered to be final. If the variation is more than +/- 3 marks then both the examiners should resolve the issue in consultation with the Chairman BoE.

Assignment

Instructions to students:

1. The assignment consists of **2** questions.
2. Maximum marks is **10**.
3. The assignment has to be neatly word processed as per the prescribed format.
4. The maximum number of pages should be restricted to **5**.
5. The printed assignment must be submitted to the course leader.
6. **Submission Date: 04/11/2019**
7. **Submission after the due date is not permitted.**

8. **IMPORTANT**: It is essential that all the sources used in preparation of the assignment must be suitably referenced in the text.
9. Marks will be awarded only to the sections and subsections clearly indicated as per the problem statement/exercise/question

Preamble

In this Course, the principles, concepts and applications of logic and discrete mathematical structures. Set theory, relations, functions, ordering, induction and modular integer arithmetic are covered. Theory and application of Propositional, Predicate and Hoare Logics for verification of computing systems are discussed. Abstract algebraic structures of Boolean algebras, lattices, groups, rings and fields are taught along with their computer science and engineering applications. This Assignment is designed to evaluate the student's learning outcomes pertinent to the Course.

Question 1 **(7 Marks)**

Sets are one of the basic building blocks for the types of objects considered in discrete mathematics. All programming languages have set operations. Many different systems of axioms have been used to develop set theory. Boolean algebra is used extensively in the design of digital electronic circuitry, for example in, calculators and personal computers. Set theory provides the basis of topology, the study of sets together with the properties of various collections of subsets. In this context, the student is required to develop a program to create and populate two sets A and B and perform various set operations on them such as union, intersection, complement and difference. The student is also required to illustrate the results using Venn diagrams with the help of software. The effort needs to be documented along the following lines:

1.1 Development of a Python program with comments to create sets and perform the specified operations

1.2 Illustration using Venn diagrams

Solution

```
#creating an empty list for universel set
universelset=[]
#number of elements as input for universel set
n1=int(input("enter number of elements in universel set:"))
#iterating till the range
for i in range(0,n1):
    eleu1=(input())
    universelset.append(eleu1)#adding the element to universel set
print("the universel set is",set(universelset))

#creating an empty list for set1
set1=[]
#number of elements as input for set1
n1=int(input("enter number of elements in set1:"))
#iterating till the range
for i in range(0,n1):
    ele1=(input())
    set1.append(ele1)#adding the element to set1
print("the set1 is",set(set1))
#creating an empty list for set2
set2=[]
#number of elements as input for set2
n2=(input("enter number of elements in set2:"))
#iterating till the range
for i in range(0,n2):
    ele2=int(input())
    set2.append(ele2)#adding the element to set2
print("the set2 is",set(set2))
print("the universel set is",set(universelset))
print("the set1 is",set(set1))
#union of two sets set1 and set2
union=set1+set2
print("union of two sets set1 and set2",set(union))
```

```python
#intersection of two sets set1 and set2
intersection=[]
for num in set1:
    for item in set2:
        if num==item:
            intersection.append(num)
print("intersection of two sets set1 and set2 is ",set(intersection))
#difference of two given sets
#set1-set2
res1=[]
for value1 in set1:
    if not value1  in  set2:
        res1.append(value1)
print("diference of set1-set2 is",set(res1))
#set2-set1
res2=[]
for value2 in set2:
    if not value2  in  set1:
        res2.append(value2)
print("diference of set2-set1 is", set(res2))
#complement of set1
res3=[]
for value3 in universelset:
    if not value3  in  set1:
        res3.append(value3)
print("complement of set1 is ",set(res3))
#complement of set2
res4=[]
for value4 in universelset:
    if not value4  in  set2:
        res4.append(value4)
print("complement of set2 is ",set(res4))
print("program done by Deepak R")
```

Fig 1 program to perform various set operations on set1 and set 2 such as union, intersection, complement and difference.

TESTCASE1

```
Debug I/O    Python Shell
Commands execute without debug.  Use arrow keys for history.

    Python Type "help", "copyright", "credits" or "license" for more information.
>>> [evaluate untitled-3.py]
    enter number of elements in universel set:9
    0
    1
    2
    3
    5
    6
    7
    8
    9
    the universel set is {0, 1, 2, 3, 5, 6, 7, 8, 9}
    enter number of elements in set1:7
    1
    3
    5
    9
    7
    0
    2
    the set1 is {0, 1, 2, 3, 5, 7, 9}
    enter number of elements in set2:5
    7
    0
    2
    6
    8
    the set2 is {0, 2, 6, 7, 8}
    the universel set is {0, 1, 2, 3, 5, 6, 7, 8, 9}
    the set1 is {0, 1, 2, 3, 5, 7, 9}
    union of two sets set1 and set2 {0, 1, 2, 3, 5, 6, 7, 8, 9}
    intersection of two sets set1 and set2 is  {0, 2, 7}
    diference of set1-set2 is {1, 3, 5, 9}
    diference of set2-set1 is {8, 6}
    complement of set1 is  {8, 6}
    complement of set2 is  {1, 3, 5, 9}
    program done by Deepak R
>>> |
```

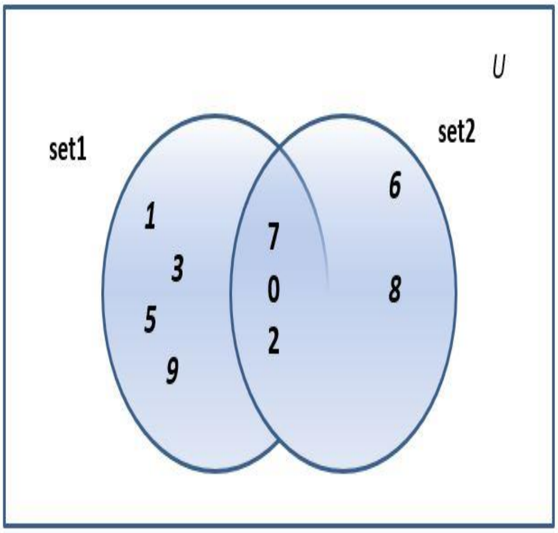
TESTCASE2

```
Debug I/O    Python Shell
Commands execute without debug.  Use arrow keys for history.

    3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)]
    Python Type "help", "copyright", "credits" or "license" for more information.
>>> [evaluate untitled-1.py]
    enter number of elements in universel set:5
    a
    b
    c
    d
    3
    the universel set is {'a', '3', 'd', 'c', 'b'}
    enter number of elements in set1:3
    a
    3
    d
    the set1 is {'a', '3', 'd'}
    enter number of elements in set2:4
    a
    b
    d
    c
    the set2 is {'a', 'c', 'b', 'd'}
    the universel set is {'a', '3', 'd', 'c', 'b'}
    the set1 is {'a', '3', 'd'}
    Union of two sets set1 and set2 is: {'a', '3', 'd', 'c', 'b'}
    Intersection of two sets set1 and set2 is : {'a', 'd'}
    Difference of set1-set2 is: {'3'}
    Difference of set1-set2 is: {'c', 'b'}
    complement of set1 is : {'c', 'b'}
    complement of set2 is : {'3'}
    program done by Deepak R
>>> |
```

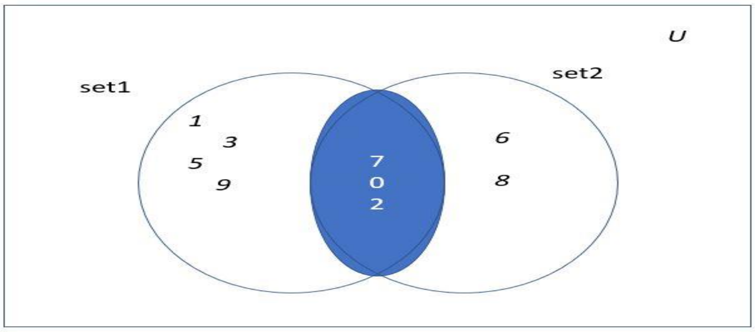
Solution for 1.2

Union set operation



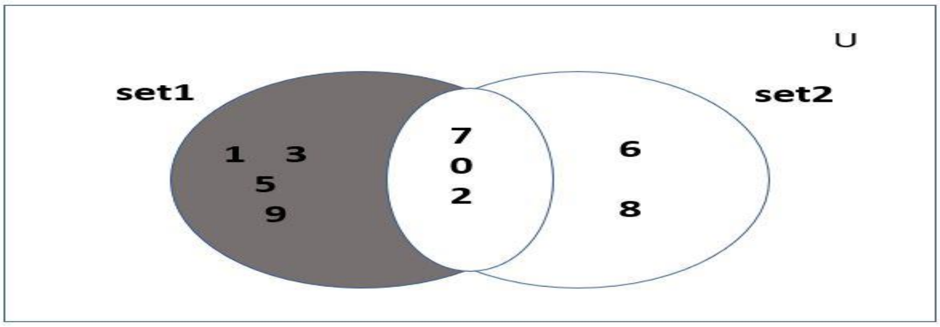
The symbol ∪ is employed to denote the **union** of two **sets**. Thus, the **set1** ∪ set2 is defined as the set that consists of all elements belonging to either set1 or set2 (or both).In above example union is {1,3,5,9,7,0,2,6,8}.

Intersection



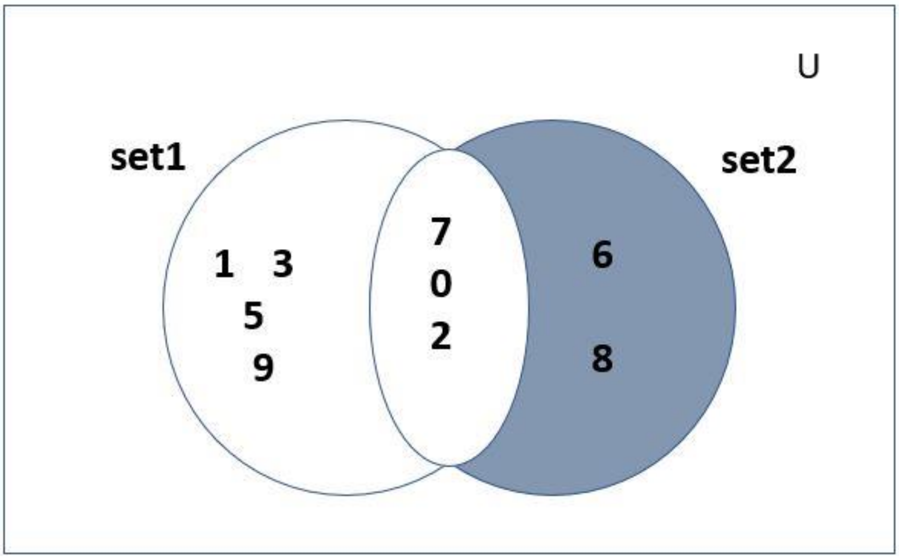
The intersection operation is denoted by the symbol ∩. The set1 ∩ *set2*—is defined as the set composed of all elements that belong to both *set1* and *set2*.
In above example intersection is {7,0,2}.

Difference set1-set2



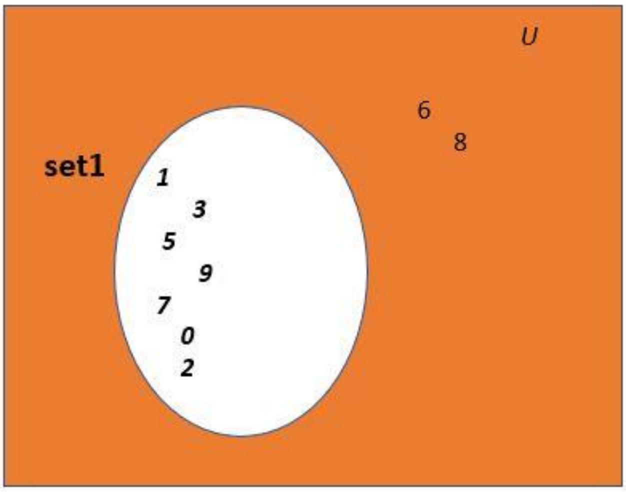
Difference between sets is denoted by 'set1 – set2', is the set containing elements of set1 but not in set2. i.e all elements of set1 except the element of set2. In above example Difference set1-set2 is {1,3,5,9}.

Difference set2-set1



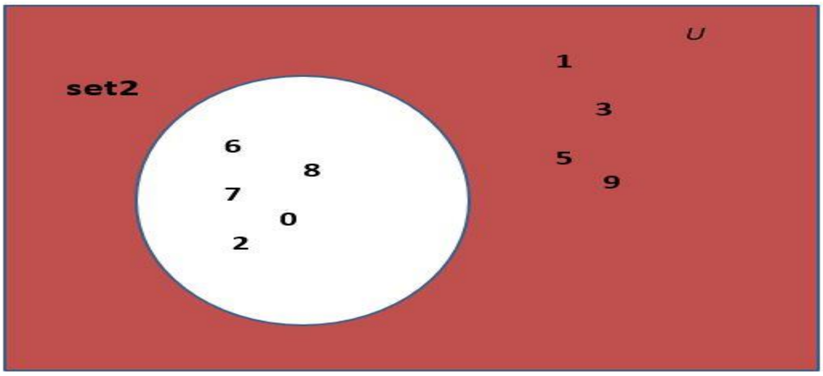
Difference between sets is denoted by 'set2 – set1', is the set containing elements of set2 but not in set1. i.e all elements of set2 except the element of set1. In above example Difference set2-set1 is {6,8}.

Complement of set1



The complement of set1, denoted by (set1)' , is the set of all elements in the universal set that are not in set1. It is denoted by (**set1)'.** In above example Complement of set1 is {6,8}.

Complement of set2



The complement of set2, denoted by (set2)' , is the set of all elements in the universal set that are not in set2. It is denoted by (**set2)'.** In above example Complement of set2 is {1,3,5,9}.

Question 2 **(3 Marks)**

"Mathematical Induction can be applied to prove that the cardinality of a power set is 2n, where n is the cardinality of the set". State whether the statement is true or false with justification. If True, prove with an example.

The effort needs to be documented along the following lines:

2.1 Justification of the statement with appropriate reasoning

2.2 Solution to the example problem

Solution

Yes "Mathematical Induction can be applied to prove that the cardinality of a power set is 2n, where n is the cardinality of the set".

The objects involved in this claim are sets. To apply induction to facts that aren't about the integers, we need to find a way to use the integers to organize our objects. In this case, we'll organize our sets by their cardinality.

The proposition P(n) for our induction is then "For any set S containing n elements, S has 2^n subsets." Notice that each P(k) is a claim about a whole family of sets,

Proof: We'll prove this for all sets S, by induction on the cardinality of the set.

Base: Suppose that S is a set that contain no elements. Then S is the empty set, which has one subset, i.e. itself. Putting zero into our formula, we **get $2^0 = 1$ which is correct**.

Induction: Suppose that our claim is true for all sets of k elements, where k is some non-negative integer. We need to show that it is true for all sets of k + 1 elements.

Suppose that S is a set containing k + 1 elements. Since k is non-negative, $k + 1 \geq 1$, so S must contain at least one element. Let's pick a random element a in S. Let $T = S - \{a\}$.

If B is a subset of S, either B contains a or B doesn't contain a. The subsets of S not containing a are exactly the subsets of T. The subsets of S containing a are exactly the subsets of T, with a added to each one. So S has twice as many subsets as T.

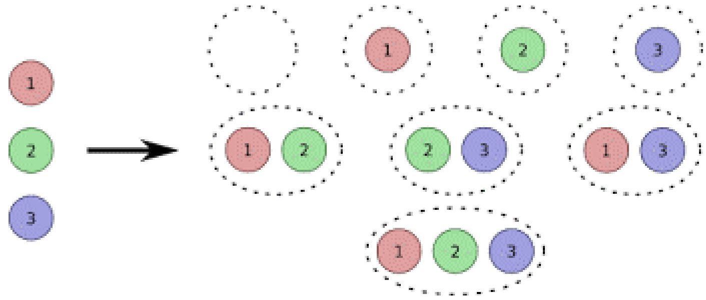
By the induction hypothesis, T has 2^k subsets. **So S has $2 \cdot 2^k = 2^{k+1}$** subsets, which is what we needed to show.

In the inductive step, we need to show that our claim is true for all sets of k + 1 elements. Because we are proving a universal statement, 7 we need to pick a representative element of the right type. This is the set S that we choose in the second paragraph of the inductive step.

Example

Z = {1, 2, 3}

n(Z) = 3 i.e. the number of elements of Z is 3.



The subsets of Z are { },{1},{2},{3},{1,2},{2,3},{1,3},{1,2,3}

$|P(Z)| = 2^n = 2^3 = 8$

P(Z)={ { }, {1}, {2}, {3}, {1, 2}, {2, 3}, {1, 3}, {1, 2, 3} } i.e. a set of all the subsets of Z.

So cardinality of a power set is 2^n, where n is the cardinality of the set is true

Bibliography

Discrete Mathematics and Its Applications by Kenneth Rosen