**DEERWALK INSTITUTE OF TECHNOLOGY**

**Tribhuvan University**

**Faculties of Computer Science**

****

**Bachelors of Science in Computer Science and Information Technology (BSc. CSIT)**

**Course: Discrete Structures (CSC160)**

**Class of 2024/Semester: 2**

**A Lab report on:**

**Set Operation**

Submitted to:

Binod Niraula

Department of Computer Science

Submitted by:

Name: Sharad Sharma

Roll: 1037

Submission Date: 2022/01/03

# Background/Theory:

Set operations refers to the operations performed between the sets and their respective elements. Two, or more, sets can be combined in many different ways. For instance, starting with the set of mathematics majors at your school and the set of computer science majors at your school, we can form the set of students who are mathematics majors or computer science majors, the set of students who are joint majors in mathematics and computer science, the set of all students not majoring in mathematics, and so on.

Set operations are equivalent to fundamental operations on integers. Sets in mathematics deal with a finite collection of items, which can be numbers, alphabets, or any real-world object. Sometimes it is necessary to establish the link between two or more sets. The idea of set operations is introduced.

Set union, set intersection, set complement, and set difference are the four primary set operations. In this article, we will learn about the different set operations, notations for describing sets, how to operate on sets, and how to use them in practice.

A set is defined as a group of items. Each item in a set is referred to as an 'Element.' There are three ways to represent a set. There are three of them: statement form, roster form, and set builder form. Set operations are operations that are performed on two or more sets in order to establish a relationship between them. Set operations are classified into four types:

1. Sets' union
2. Sets intersecting
3. a set's complement

Distinction between sets/Relative Complement:

Before we go into the various set operations, let's review the notion of Venn diagrams, which is essential for understanding set operations. A Venn diagram is a logical diagram that depicts the potential relationship between several finite sets.

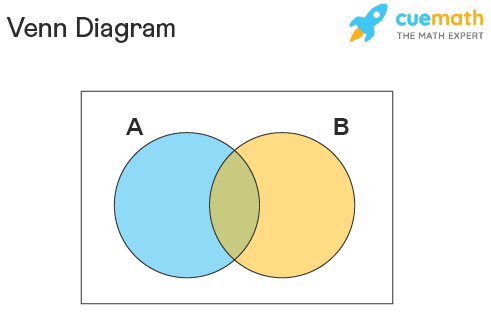


Figure 1: Venn-Diagram

# Algorithm:

Step 1: Take the input sets A, B and U

Step 2: Read the process desired by user.

Step 3: If process is union

Check the repeating elements and remove them from the concatenation of the two arrays.

If process is intersection

Check the common elements and store the elements in new set AnB.

If process is difference

Write down the elements of the first set until elements get repeated and remove the element from the set and store in new set A-B or B-A

Step 4: Print the desired set operation

Step 5: Stop

# Source Code:

a = input(" enter number or characters separated by space: ")

b = input(" enter number or characters separated by space: ")

universal = input(" enter number or characters separated by space: ")

a = set(a.split(" "))

b = set(b.split(" "))

universal = set(b.split(" "))

print("A = ", a)

print("B =", b)

print("U =", universal)

# performing set operations

print(" Union operation: A ∪ B = ", a | b)

print(" Intersection operation: A ∩ B", a & b)

print(" Compliment opreation: A', B', (A ∪ B)', (A ∩ B)': ",

      universal - a, universal - b, (universal-(a | b)), (universal-(a & b)))

print(" Difference opreation: A - B, B - A: ", a - b, b - a)

# Conclusion:

In this way, we performed set operations in the high-level programming languages by using arrays and library functions.