Name: **GANESH SANJAY PANDHRE**

Roll Nos : **41** | Div: **D20B**

AIM: To implement and demonstrate the basic properties and operations of fuzzy sets using Python.

Experiment No: 7

Theory

A **fuzzy set** is a generalization of a classical set in which each element has a **degree of membership** ranging from 0 to 1, instead of having only full membership (1) or no membership (0). This approach allows modeling of **uncertainty**, **imprecision**, and **partial truth**.

Mathematically, a fuzzy set A in a universe of discourse X is represented as:

$$A = \{(x, \mu_A(x)) \mid x \in X,\ \mu_A(x) \in [0, 1]\}$$

Where:

- $x \rightarrow$ element of the universe X
- $\mu_A(x) \rightarrow$ membership function indicating the degree to which x belongs to A

Basic Operations on Fuzzy Sets

1. **Union** Combines two fuzzy sets A and B by taking the maximum membership value for each element:

$$\mu_{A \cup B}(x) = \max\left(\mu_A(x), \mu_B(x)\right)$$

2. **Intersection** Finds the common membership values between A and B:

$$\mu_{A \cap B}(x) = \min\left(\mu_A(x), \mu_B(x)\right)$$

3. **Complement** Represents the degree to which an element does **not** belong to A:

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

4. **Scalar Multiplication** Multiplies all membership values by a constant α (0 ≤ α ≤ 1):

$$\mu_{\alpha A}(x) = \alpha \cdot \mu_A(x)$$

5. **Fuzzy Sum** Two common forms are:

   - **Bounded Sum**:

$$\mu_{A \oplus B}(x) = \min\left(1,\ \mu_A(x) + \mu_B(x)\right)$$

   - **Probabilistic Sum**:

$$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$$

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4, 5]
```

```
A = [0.1, 0.4, 0.7, 0.9, 0.2]
B = [0.3, 0.6, 0.8, 0.5, 0.1]
```

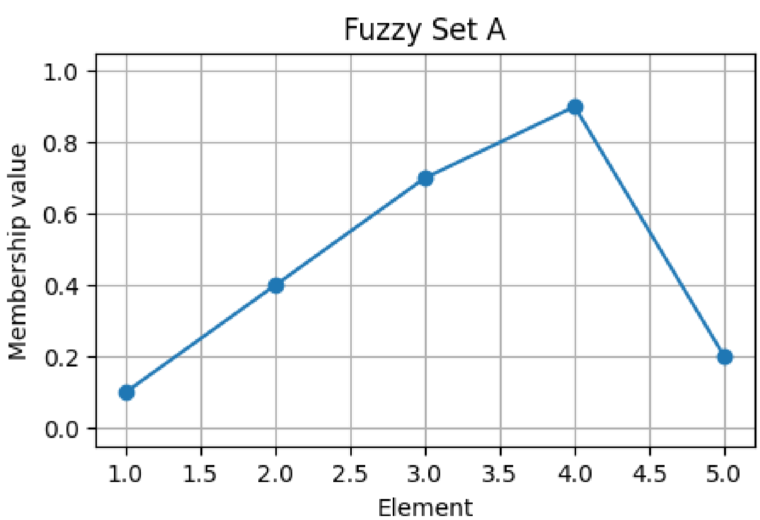
```
A_union_B        = [max(a, b) for a, b in zip(A, B)]      # Union
A_intersection_B = [min(a, b) for a, b in zip(A, B)]      # Intersection
A_complement     = [1 - a for a in A]                     # Complement of A
```

```
alpha = 0.6
alpha_A = [alpha * a for a in A]
```

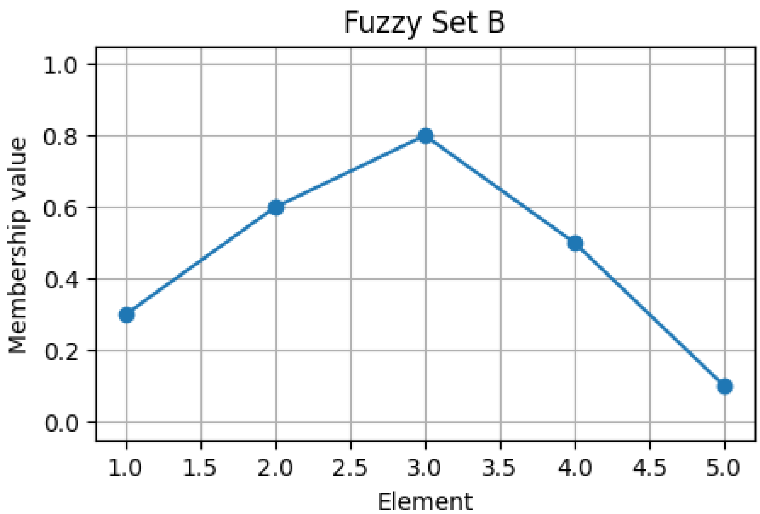
```
A_oplus_B_bounded = [min(1.0, a + b) for a, b in zip(A, B)]              # Bounded sum
A_oplus_B_prob    = [a + b - a*b for a, b in zip(A, B)]                  # Probabilistic sum
```

```
# --- Step 4: Plot helper ---
def plot_series(x, y, title):
    plt.figure(figsize=(5, 3))
    plt.plot(x, y, marker='o')
    plt.title(title)
    plt.xlabel('Element')
    plt.ylabel('Membership value')
    plt.ylim(-0.05, 1.05)
    plt.grid(True)
    plt.show()
```

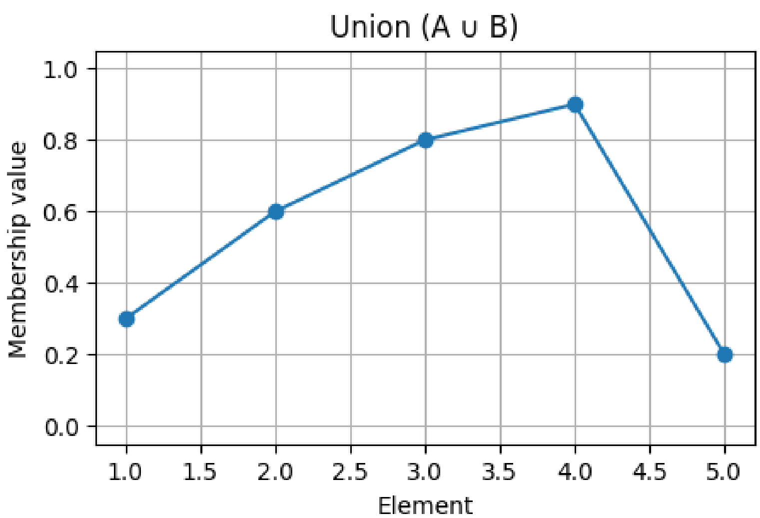
```
plot_series(x, A, "Fuzzy Set A")
```



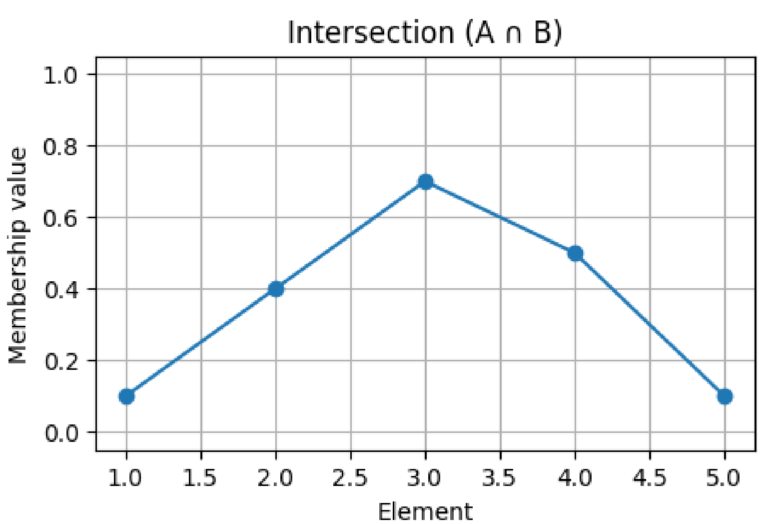

```
plot_series(x, B, "Fuzzy Set B")
```



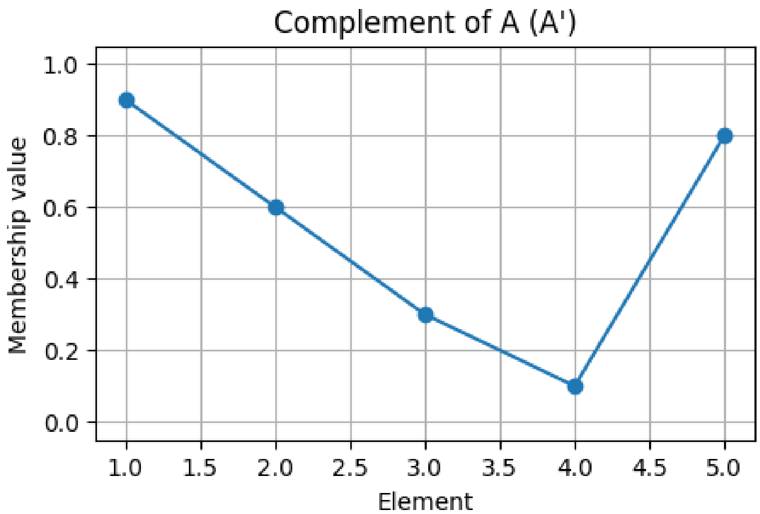

```
plot_series(x, A_union_B, "Union (A ∪ B)")
```



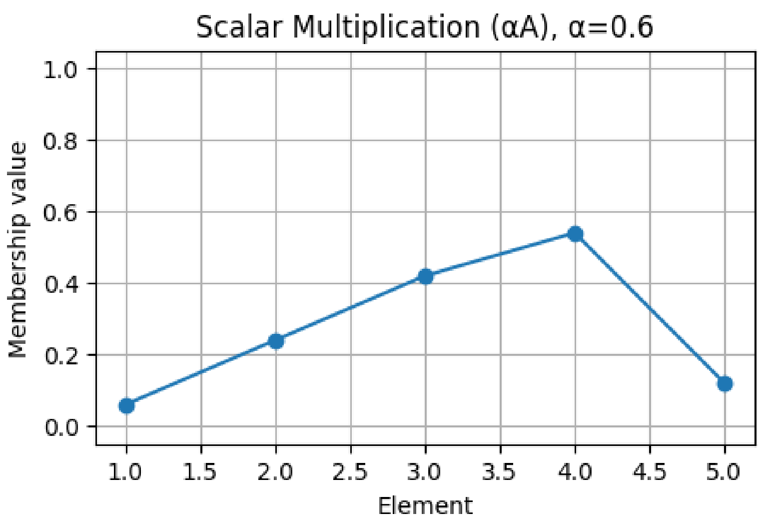

```
plot_series(x, A_intersection_B, "Intersection (A ∩ B)")
```



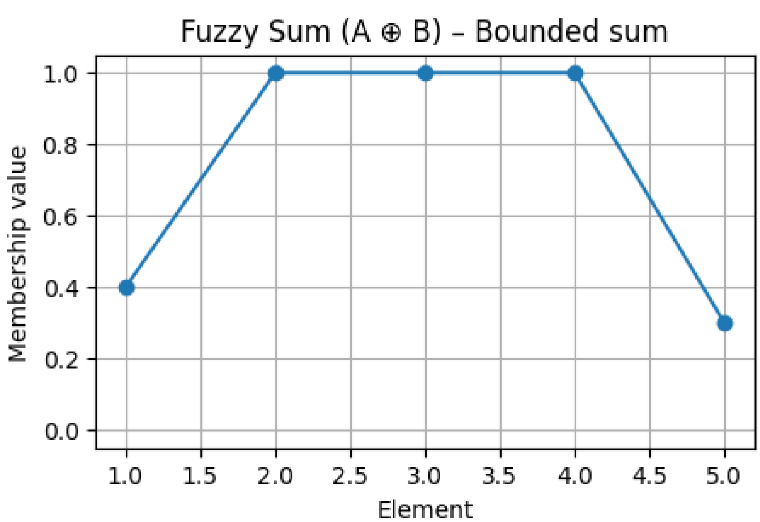

```
plot_series(x, A_complement, "Complement of A (A')")
```



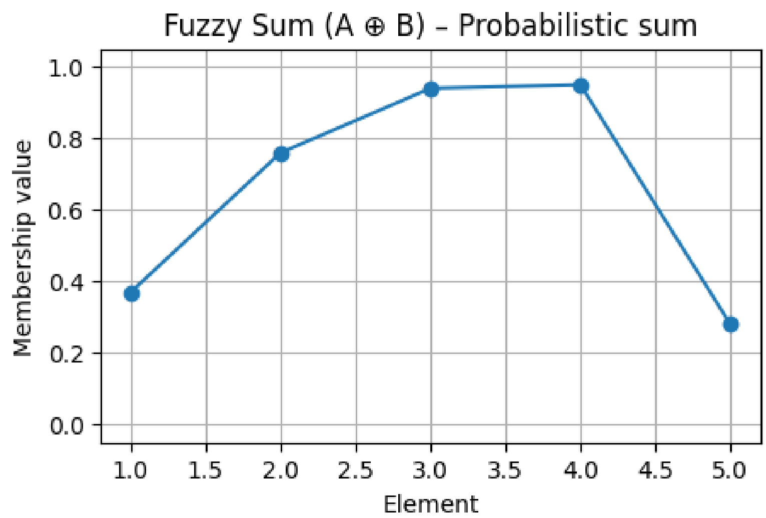

```
plot_series(x, alpha_A, f"Scalar Multiplication (αA), α={alpha}")
```




```
plot_series(x, A_oplus_B_bounded, "Fuzzy Sum (A ⊕ B) – Bounded sum")
```



```
plot_series(x, A_oplus_B_prob, "Fuzzy Sum (A ⊕ B) – Probabilistic sum")
```



Conclusion

In this experiment, we implemented the fundamental operations of fuzzy sets—union, intersection, complement, scalar multiplication, and fuzzy sum—using Python. The results verified the theoretical definitions, showing that:

- Union takes the maximum membership values.
- Intersection takes the minimum membership values.
- Complement inverts the membership degrees with respect to 1.
- Scalar multiplication scales membership values.
- Fuzzy sums combine sets while ensuring membership values remain within [0, 1].

These operations form the foundation for fuzzy logic systems and can be applied in fields like decision-making, control systems, and pattern recognition.
