Міністерство освіти та науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та

Комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи №1

“Побудова функцій приналежності нечіткої множини на основі експертної інформації”

Виконав:

Студент групи ФеІ-44

Сапанюк М.І.

Перевірила:

Притула М.

Львів 2022

**Мета:**

Ознайомитися з прямим методом побудови функцій приналежності нечіткої множини, який ґрунтується на статистичній обробці думок групи експертів.

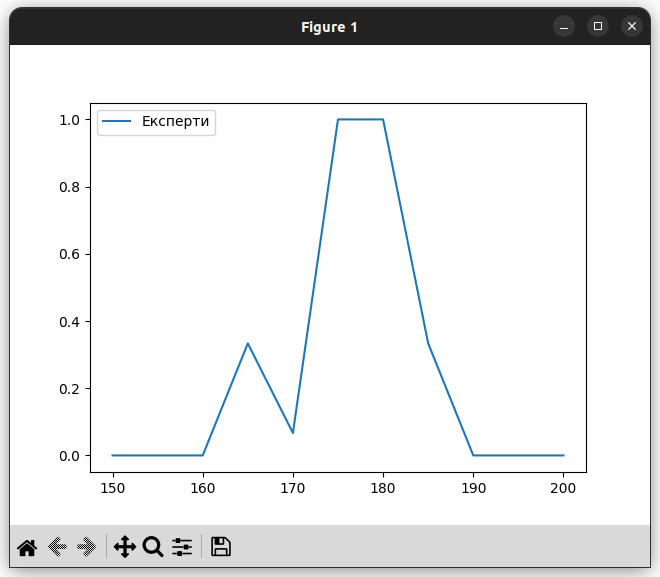
**Хід роботи:**

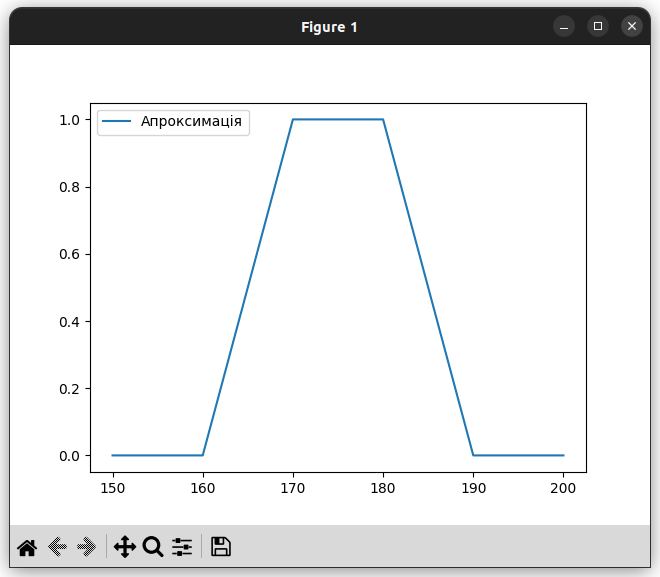
1. На основі опитування 5 або більше експертів задати нечітку множину «чоловік середнього зросту».

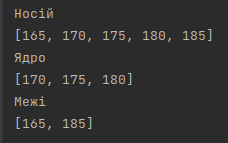
2. Створити програму, яка розраховує функцію приналежності на основі статистичної обробки думок групи експертів, будує її графік і апроксимує однією з П-подібних функцій.

3. Визначити носій нечіткої множини, її ядро та межі.

**Виконання завдання:**







**Висновок:**

Виконавши лабораторну роботу я ознайомився з нечіткими множинами

**Додаток:**

from matplotlib import pyplot as plt  
  
  
def aproc(x: int, a: int, b: int, c: int, d: int) -> float:  
 if x < a:  
 return 0  
 elif a <= x <= b:  
 return (x - a) / (b - a)  
 elif b <= x <= c:  
 return 1  
 elif c <= x <= d:  
 return (d - x) / (d - c)  
 else:  
 return 0  
  
  
def carrier(array\_label: list[int], array\_mu: list[float]):  
 temp: list[int] = []  
 for i in range(0, len(array\_mu)):  
 if array\_mu[i]:  
 temp.append(array\_label[i])  
 print("Носій")  
 print(temp)  
  
  
def core(array\_label: list[int], array\_mu: list[float]):  
 temp: list[int] = []  
 for i in range(0, len(array\_mu)):  
 if array\_mu[i] == 1:  
 temp.append(array\_label[i])  
 print("Ядро")  
 print(temp)  
  
  
def border(array\_label: list[int], array\_mu: list[float]):  
 temp: list[int] = []  
 for i in range(0, len(array\_mu)):  
 if 0 < array\_mu[i] < 1.0:  
 temp.append(array\_label[i])  
 print("Межі")  
 print(temp)  
  
  
a: int = 160  
b: int = 170  
c: int = 180  
d: int = 190  
array\_label: list[int] = [x for x in range(150, 205, 5)]  
array\_mu: list[float] = []  
array\_mu\_aprox: list[float] = []  
matrix: list[list[float]] = [  
 [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],  
 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],  
 [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],  
]  
size\_i: int = len(matrix)  
size\_j: int = len(matrix[0])  
for j in range(0, size\_j):  
 sup: float = 0  
 mu: float = 0  
 for i in range(0, size\_i):  
 mu += matrix[i][j]  
 if sup < matrix[i][j]:  
 sup = matrix[i][j]  
 if sup:  
 mu /= sup \* size\_i  
 array\_mu.append(mu)  
  
for x in array\_label:  
 array\_mu\_aprox.append(aproc(x, a, b, c, d))  
carrier(array\_label, array\_mu\_aprox)  
core(array\_label, array\_mu\_aprox)  
border(array\_label, array\_mu\_aprox)  
  
plt.plot(array\_label, array\_mu, label='Експерти')  
plt.legend(loc=2)  
plt.show()  
plt.plot(array\_label, array\_mu\_aprox, label='Апроксимація')  
plt.legend(loc=2)  
plt.show()