

International University of Information Technology

Department of Computer Engineering

**Laborotoy Work №2**

Complete by a student of the group: Ospan Ramazan it2-2310

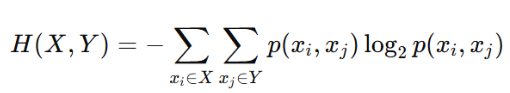
Checked by senior lector of the Department of Computer Engineering:

Daurenbaeva Nurkamilya A

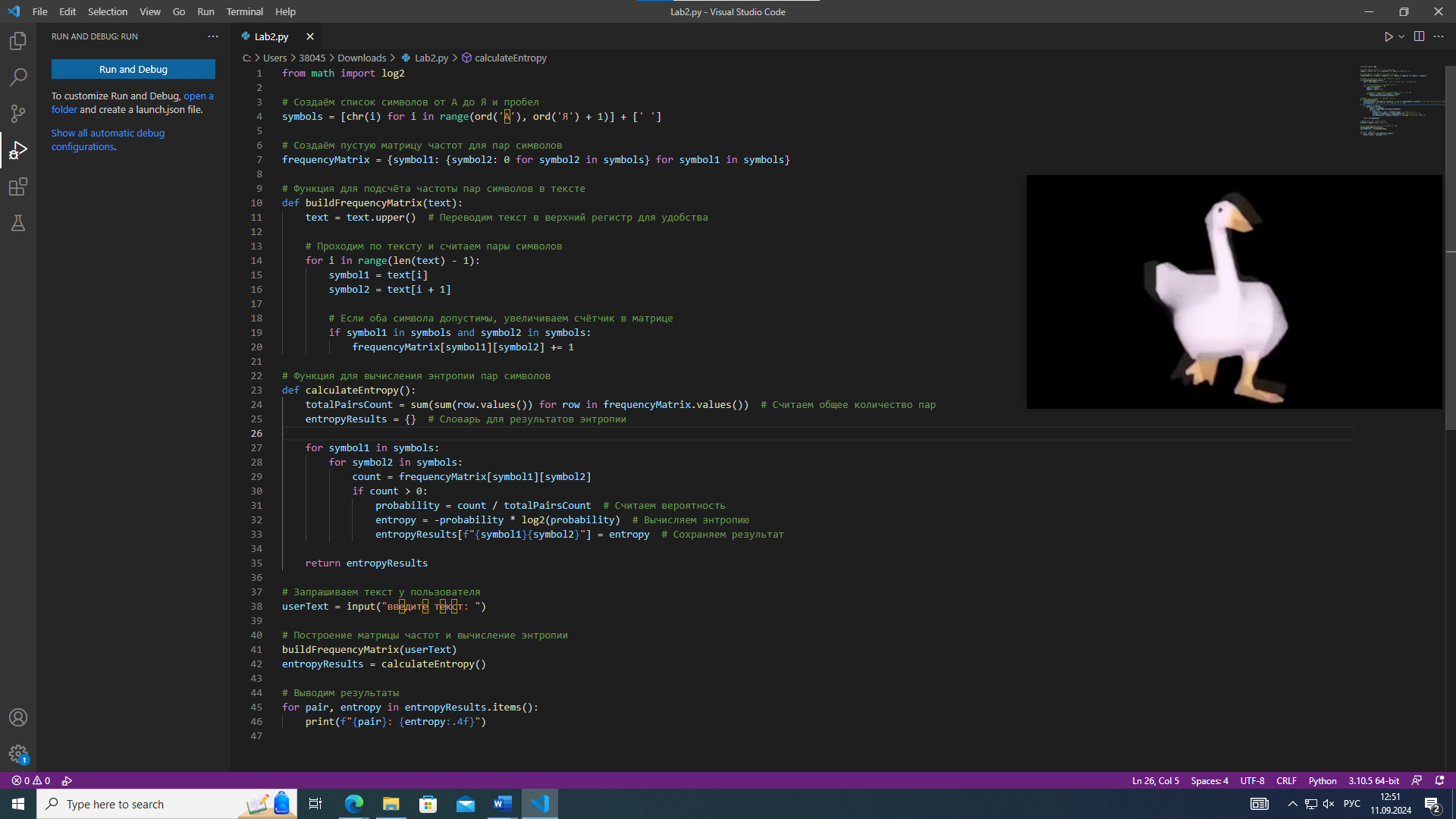
Almaty 2024

**Entropy for Two Objects** refers to the calculation of the information content associated with pairs of symbols or objects in a dataset. For a given set of symbols, the entropy of a pair (e.g., "AB" or "BA") is computed based on the frequency of occurrence of that pair in the data.

The entropy H(X, Y) of a pair of symbols X and Y can be calculated using the formula:

****

**Algorithm:**

****

**A screenshot of a computer

Description automatically generated**

from math import log2

# Создаём список символов от А до Я и пробел

symbols = [chr(i) for i in range(ord('А'), ord('Я') + 1)] + [' ']

# Создаём пустую матрицу частот для пар символов

frequencyMatrix = {symbol1: {symbol2: 0 for symbol2 in symbols} for symbol1 in symbols}

# Функция для подсчёта частоты пар символов в тексте

def buildFrequencyMatrix(text):

    text = text.upper()  # Переводим текст в верхний регистр для удобства

    # Проходим по тексту и считаем пары символов

    for i in range(len(text) - 1):

        symbol1 = text[i]

        symbol2 = text[i + 1]

        # Если оба символа допустимы, увеличиваем счётчик в матрице

        if symbol1 in symbols and symbol2 in symbols:

            frequencyMatrix[symbol1][symbol2] += 1

# Функция для вычисления энтропии пар символов

def calculateEntropy():

    totalPairsCount = sum(sum(row.values()) for row in frequencyMatrix.values())  # Считаем общее количество пар

    entropyResults = {}  # Словарь для результатов энтропии

    for symbol1 in symbols:

        for symbol2 in symbols:

            count = frequencyMatrix[symbol1][symbol2]

            if count > 0:

                probability = count / totalPairsCount  # Считаем вероятность

                entropy = -probability \* log2(probability)  # Вычисляем энтропию

                entropyResults[f"{symbol1}{symbol2}"] = entropy  # Сохраняем результат

    return entropyResults

# Запрашиваем текст у пользователя

userText = input("введите текст: ")

# Построение матрицы частот и вычисление энтропии

buildFrequencyMatrix(userText)

entropyResults = calculateEntropy()

# Выводим результаты

for pair, entropy in entropyResults.items():

    print(f"{pair}: {entropy:.4f}")

----------------------------------------------------------------------------------------------------------------



In this fragment, we create a symbols list that contains all Cyrillic characters from 'A' to 'Z' and a space. We use chr(i) to get the symbol by its code and ord('A') to get the symbol code 'A'. The list includes all the characters that may occur in the text



Here, a nested dictionary (frequency matrix) is created, where each symbol from symbols is a key for the internal dictionary, which also has all the symbols as keys. All values are initialized to zero, which means that pairs of characters have not yet been encountered in the text

A computer screen shot of a code

Description automatically generated

The buildFrequencyMatrix function performs the following actions: • Converts text to uppercase for consistency. • Runs through the text, analyzing pairs of characters (current and next). • If both symbols are in the symbols list, increments the counter in the frequency matrix for this pair.

A screen shot of a computer code

Description automatically generated

The calculate Entropy function performs the following actions: • Counts the total number of pairs of characters in the text. • Passes through all possible pairs of characters. • For each pair, if it occurs at least once, calculates its probability and entropy. • The results are stored in the entropyResults dictionary, where the key is a combination of characters and the value is their entropy.

A computer screen shot of text

Description automatically generated

In the main part of the program: • The user enters the text. • The buildFrequencyMatrix function is called to count the frequency of character pairs. • \* The calculateEntropy function is called to calculate entropy. • The results are displayed in the "pair of characters: entropy" format, rounded to four decimal places.