Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждения высшего образования

«Национальный исследовательский технологический университет «МИСиС»

Лабораторная работа №2.3

по дисциплине

«Основы теории информации»

на тему:

«Энтропия источника и энтропия сообщения»

Направление подготовки:

09.03.01 «Информатика и вычислительная техника»

Семестр 5

Выполнил: Проверил:

Попов Егор Куприянов Вячеслав Борисович Васильевич\_\_\_\_\_\_\_\_

(Ф.И.О. студента) (Ф.И.О преподавателя)

БИВТ-22-СП-4 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(№ группы) (оценка)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата сдачи) (дата проверки)

Подпись: \_\_\_\_\_\_\_\_\_\_ Подпись: \_\_\_\_\_\_\_\_\_\_

Москва – 2022

**Оглавление**

[Цель работы 3](#_Toc6351)

[1. Теоретическая часть 4](#_Toc6352)

[2. Условие задачи 5](#_Toc6353)

[3. Практическая часть 6](#_Toc6354)

[Заключение 9](#_Toc6355)

[Приложение 10](#_Toc6356)

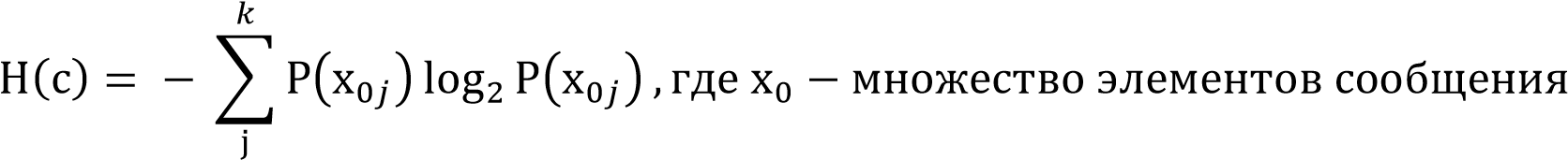
[Приложение 1: 10](#_Toc6357)

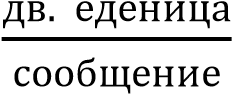
# Цель работы

Изучить тему «Энтропия источника и энтропия сообщения» и реализовать решение задачи №2.3 при помощи языка программирования Python 3.10 для закрепления материала.

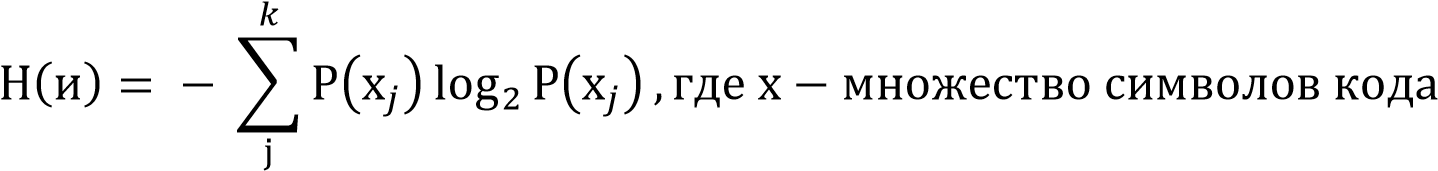
# Теоретическая часть

Энтропия сообщения – количество информации, содержащейся в любом осредненном сообщении.



Единица измерения энтропии сообщения: 

Энтропия источника – количество информации, содержащейся в одном символе сообщения.



дв. еденица

Единица измерения энтропии источника:

сообщение

Из приведенных формул видно, что энтропия источника зависит только от числа генерируемых элементов в используемом для передачи сообщения алфавите.

# Условие задачи

Пусть передается 16 равновероятных сообщений двоичным неизбыточным кодом. Сообщение отображается кодом: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111. Найти энтропию сообщения и энтропию источника.

# Практическая часть

Для реализации решения задачи потребуется функция логарифма, которую нужно подключить из библиотеки math командой from math import log. Данная функция

принимает 2 аргумента: аргумент логарифма, основание логарифма.

Код разбит на логические части, каждая из которых записана в отдельной функции.

Функция для поиска энтропии сообщения:

def EntropyMessage(countMessages):

p = 1 / countMessages result = 0

for i in range(0, countMessages):

result += p \* log(p, 2) result \*= -1 return result

Данная функция получает 1 аргумент: countMessages, хранящий в себе количество сообщений.

Так как события равновероятны, то сразу вычисляем вероятность одного события и записываем результат в переменную p.

Далее через цикл суммируем произведение вероятностей на логарифм от вероятности по основанию 2. После чего умножаем результат на -1 и возвращаем его.

Функция для поиска энтропии источника:

def EntropyOrigin(countSigns):

p = 1 / countSigns result = 0 for i in range(0, countSigns): result += p \* log(p, 2) result \*= -1 return result

Данная функция получает 1 аргумент – countSigns, хранящий в себе количество символов в алфавите кода в кодирования.

В данном случае алгоритм такой же, за исключением поиска вероятности. Так как в данном случае ищем вероятность появления символа алфавита, при условии, что все события равновероятны.

Ввод и валидация данных:

def Validation(): while True: try:

data = int(input('Введите целое число более 0: ')) if data > 0: return data

print('Введенное значение не больше 0') except:

print('Вы ввели не число') print('Попробуйте еще раз') Данная функция реализует ввод данных и их проверку на условия: 1. Введенное значение является целым числом;

2. Введенное значение больше 0.

Первое условие проверяется через конструкцию try – except, которая обрабатывает ошибки, не прекращая выполнения программы. То есть при вводе набора символом, который нельзя преобразовать в целочисленные тип данных, управляющий поток перейдет в блок except и выведет сообщение «Вы ввели не число». А при вводе целого числа не больше 0, программа выведет сообщение «Введенное значение не больше

0».

В конце, если значение не удовлетворило условиям выводится сообщение «Попробуйте еще раз».

Основная функция: def Main():

print('Поиск энтропии сообщения и энтропии источника')

# Проверка ввода для количества различных сообщений print('Введите количество различных возможных сообщений') countMessages = Validation()

print('Введите количество различных символов в кодировке') countSigns = Validation()

print('Энтропия сообщения', EntropyMessage(countMessages)) print('Энтропия источника', EntropyOrigin(countSigns)) wait = input("Нажмите Entre для заавершения")

Данная функция реализует в себе основную логику работы программы: получает необходимые значения через функцию Validation(), получает энтропию сообщения через функцию EntropyMessage(), получает энтропию источника через функцию EntropyOrigin().

Также выводит на экран сопровождающие сообщения для пользователя.

Последняя конструкция не является обязательной и нужна для возможности подключения данного python файла к другому. Программа начинает работу с этой конструкции.

if \_\_name\_\_ == '\_\_main\_\_':

Main()

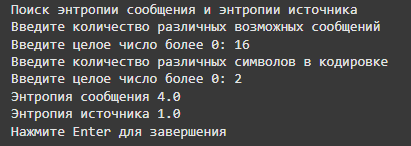
 Пример работы программы на значениях задачи представлен на рисунке 1.

Рисунок 1 – пример работы программы.

# Заключение

В ходе работы была изучена тема «Энтропия источника и энтропия сообщения», а также при помощи языка программирования Python 3.10 и библиотеки math было реализовано решение задачи №2.3 в общем виде (приложение 1) для закрепления материала.

# Приложение

## Приложение 1:

from math import log

# Функция для поиска энтропии сообщения

def EntropyMessage(countMessages):

    p = 1 / countMessages

    result = 0

    for i in range(0, countMessages):

        result += p \* log(p, 2)

    result \*= -1

    return result

# Функция для поиска энтропии источника

def EntropyOrigin(countSigns):

    p = 1 / countSigns

    result = 0

    for i in range(0, countSigns):

        result += p \* log(p, 2)

    result \*= -1

    return result

# Ввод и валидация данных

def Validation():

    while True:

        try:

            data = int(input('Введите целое число более 0: '))

            if data > 0:

                return data

            print('Введенное значение не больше 0')

        except:

            print('Вы ввели не число')

        print('Попробуйте еще раз')

def Main():

    print('Поиск энтропии сообщения и энтропии источника')

    # Проверка ввода для количества различных сообщений

    print('Введите количество различных возможных сообщений')

    countMessages = Validation()

    print('Введите количество различных символов в кодировке')

    countSigns = Validation()

    print('Энтропия сообщения', EntropyMessage(countMessages))

    print('Энтропия источника', EntropyOrigin(countSigns))

    wait = input("Нажмите Enter для завершения")

if \_\_name\_\_ == '\_\_main\_\_':

    Main()