Учреждение образования

«Белорусский государственный технологический университет»

**Лабораторная работа №1**

Расчет и анализ параметров и информативных характеристик дискретных ИС

Выполнил:

Студент 3 курса 5 группы ФИТ

Скачко Илья Александрович

2024

**Задание 1.** Были использованы 2 алфавита – таджикский(кириллица) и мальтийский(латиница).

Для реализации приложения был использован язык программирования C#. Чтобы вычислить энтропию был создан отдельный метод, в котором была создана структура Dictionary для хранения символов алфавита и их количества. Далее проходя по этой структуре вычислялась вероятность появления каждого символа. Для вычисления была использована формула энтропии алфавита по К. Шеннону.

static double CalculateEntropy(string s)

{

Console.OutputEncoding = Encoding.UTF8;

var charFrequencies = new Dictionary<char, int>();

int totalCharacters = s.Length;

foreach (char c in s)

{

if (char.IsLetter(c) || char.IsDigit(c))

{

if (!charFrequencies.ContainsKey(c))

charFrequencies.Add(c, 1);

else

charFrequencies[c]++;

}

}

double entropy = default;

foreach (var с in charFrequencies)

{

double probability = (double)с.Value / totalCharacters;

entropy -= probability \* Math.Log(probability, 2);

}

return entropy;

}

Листинг 1.1 – Метод для вычисления энтропии алфавита

Были созданы два файла, в которых записан текст на соответствующем языке. Далее эти файлы считывались в программе и исходя из этого текста были произведены вычисления.

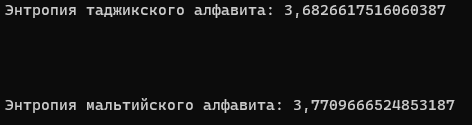


Рисунок 1.1 – Результаты вычисления энтропии алфавитов

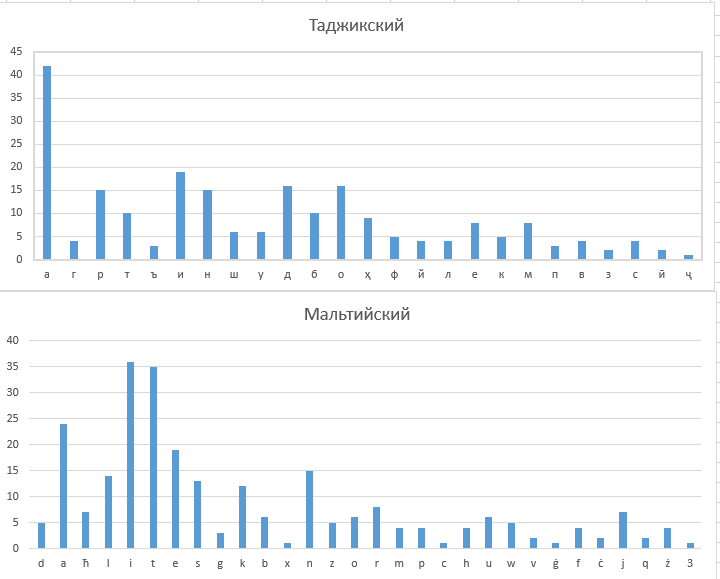


Рисунок 1.2 – Гистограммы частот появления символов

**Задание 2.** Для выполнения этого задания был создан ещё один файл в который были помещены данные с 2 файлов в бинарном виде. Далее в программе файл считывался и, используя тот же метод, вычислялась энтропия.



Рисунок 2.1 – Результат вычисления энтропии бинарного алфавита

**Задание 3.** Для вычисления количества информации используя собственное ФИО, каждая из энтропий были умножены на длину ФИО.

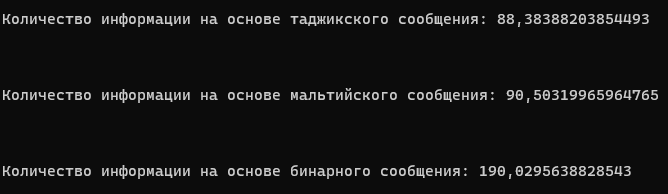


Рисунок 3.1 – Результат вычисления количества информации для каждой из энтропий

В нашем случае, полученные значения количества информации отражают количества информации, необходимое для передачи этой информации с использованием указанных алфавитов.

**Задание 4.** Для выполнения задания 3 с вероятностью ошибочной передачи единичного бита 0,1; 0,5; 1; был создан ещё один метод, который производит эти вычисления.

static double AmountInfoWithError(double entropy, double p, double q, int count, bool binary)

{

return entropy \* EffectiveEntropy(p, q, binary) \* count;

}

Листинг 4.1 – Метод для вычисления количества информации с вероятностью ошибочной передачи единичного бита

В этом метода вычисляется эффективная энтропия алфавита и далее умножается на длину ФИО и энтропию алфавита для получения количества информации.

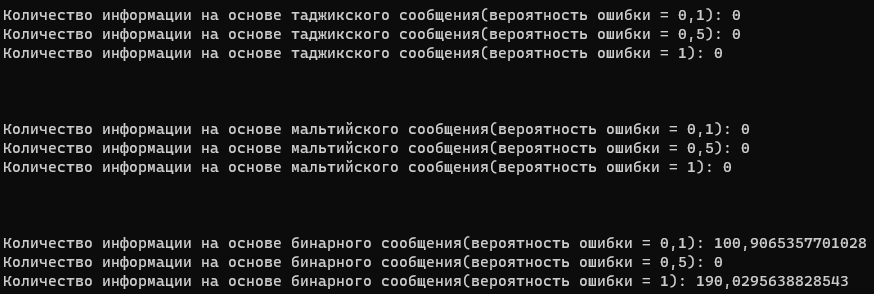


Рисунок 4.1 – Результат вычисления количества информации с вероятностью ошибочной передачи единичного бита

**Вывод:** Таким образом, чем выше вероятность ошибки, тем больше информации потребуется для надежной передачи сообщения, и в случае *P* *= 1.0*, передача становится невозможной, так как биты инвертируются. В небинарном алфавите при *P = 1* определить количество информации невозможно из-за того, что мощность алфавита больше 2.