**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

­­

­­­­­

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление информатика и вычислительная техника

Отчет

по лабораторной работе №2

по дисциплине

«ТЕОРИЯ ИНФОРМАЦИИ»

**Кодирование алгоритмом Шеннона – Фано**

Выполнил:

Студент группы 8В32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Г. Усачев

Проверил:

Ассистент ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Кузнецов

Томск 2024

# Цель работы

Разработка кодирования алгоритмом Шеннона – Фано.

# Задание 1

Используя алгоритм Шеннона-Фано, провести эффективное кодирование ансамбля из восьми знаков с заданными вероятностями zi. Для построенного эффективного кода определить энтропию, среднюю длину кодовой комбинации, избыточность.



Рисунок 1 – Вариант задания

# Демонстрация построчной отладки кода

В файле «shannonFano.h» создаём структуру «Symbol», в котором создаём структуру для хранения данных о ансамбле символов.

struct Symbol

{

    char symbol;

    double probability;

    std::string code;

    Symbol (char symbol, double probability): symbol(symbol), probability(probability){};

};

Листинг 1 – Программа «shannonFano.h»

На рисунке 2 представлена реализация кода Шеннона – Фано.

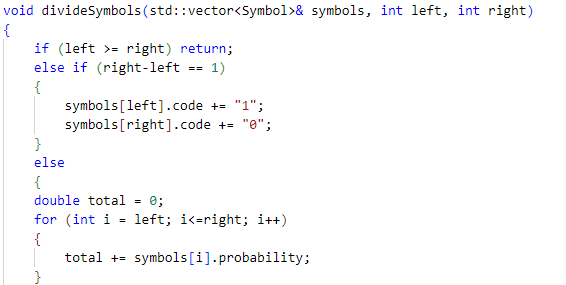


Рисунок 2 – Построение кода Шеннона – Фано.

В листинге 2 представлено определение энтропии и средней длины кодовой комбинации.

// Функция определения энтропии

double entropy(const std::vector<Symbol>& symbols)

{

    double entropy = 0;

    for (const auto& symbol : symbols){

        entropy += symbol.probability \* log2f(symbol.probability);

    }

    entropy \*= -1;

    return entropy;

}

// функция для определения средней длины комбинации

double averageLengthCombination(const std::vector<Symbol>& symbols)

{

    double averageLengthCombination = 0;

    for (const auto& symbol : symbols){

        averageLengthCombination += symbol.probability \* symbol.code.length();

    }

    return averageLengthCombination;

}

Листинг 2 – Программа «shannonFano.cpp»

На рисунке 3 представлен форматированный вывод результата работы программы.

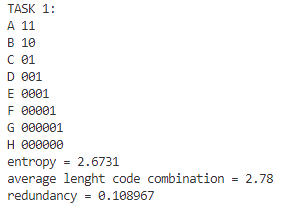




Рисунок 3 – Вывод результата работы программы

# Задание 2

При помощи алгоритмов Шеннона-Фано и Хаффмана построить эффективный код для ансамбля (см. варианты) с использованием блоков по два или три знака (см. варианты). Вычислить среднюю длину кодовой комбинации.



Рисунок 4 – Вариант задания

# Демонстрация построчной отладки кода

В файле «main.cpp» создаём ансамбль с использование блоков по три знака (листинг 3).

std::vector<std::string> symbol(20);

    char ci = 65;

    int i = 0;

    for (const auto pair1 : taskVar)

    {

        for (const auto pair2 : taskVar)

        {

            for (const auto pair3 : taskVar)

            {

                fanoCode.push\_back({ci, pair1.second\*pair2.second\*pair3.second});

                huffmanCode.push\_back({ci, pair1.second\*pair2.second\*pair3.second});

                symbol[i] = pair1.first+pair2.first+pair3.first;

                ci+=1;

                i+=1;

            }

        }

    }

Листинг 3 – Программа «main.cpp»

Так как задание повторяется, то используется код из прошлого задания и первой лабораторной.

На рисунке 5 представлен форматированный вывод результата работы программы.

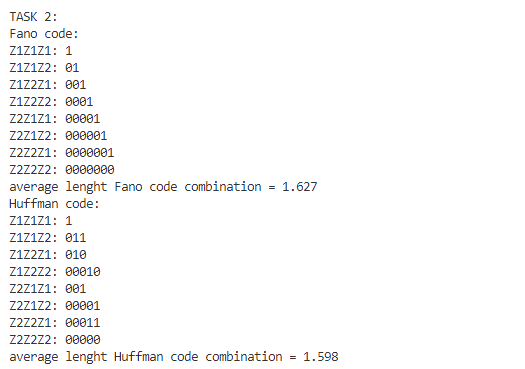




Рисунок 5 – Вывод результата работы программы

# Задание 3

3.1. Записать в текстовый файл произвольный текст на английском языке в кодировке ASCII.

3.2. Разработать программу, открывающую составленный текстовый файл на чтение и читающую все содержимое файла.

3.3. Найти оптимальное количество символов для блока многознаковой комбинации постоянной длины, учитывая только реальные частоты встречаемости в тексте (таких многознаковых комбинаций, скорее всего, будет значимо меньше, чем всевозможных комбинаций). Рассматривать многознаковые комбинации не более 5 символов в блоке.

3.4. Составить по одному ансамблю каждой длины блока и найти вероятности появления блоков в ансамбле. Получить и сравнить энтропию источника, избыточность для всех ансамблей.

# Демонстрация построчной отладки кода

В файле «main.cpp» считываем в строку данные с текстового файла.

На рисунке 6 представлен подсчёт символов блока многозначной комбинации.

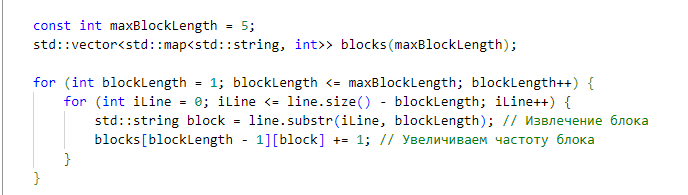


Рисунок 6 – Подсчёт символов.

В листинге 4 представлено определение энтропии и средней длины кодовой комбинации.

for (int i = 0; i < maxBlockLength; i++)

    {

        double entropy = 0;

        for (const auto pair : blocks[i]) {

            double probability = static\_cast<double>(pair.second) / total[i];

            entropy += probability \* log2(probability);

        }

        entropy \*= -1;

        std::cout << "entropy blok length " << i+1 << " = " << entropy << std::endl;

        double maxBits = log2(blocks[i].size());

        double redundancy = 1- (entropy/maxBits);

        std::cout << "redundancy blok length " << i+1 << " = " << redundancy << "\n" << std::endl;

    }

Листинг 4 – Программа «task.cpp»

На рисунке 7 представлен форматированный вывод результата работы программы.

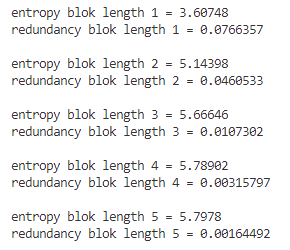




Рисунок 3 – Вывод результата работы программы

**Выводы**

В результате лабораторной работы получены навыки сборки проекта на языке программирования С++ с применением программного средства автоматизации сборки программного обеспечения CMake и компилятора GCC в редакторе кода Visual Studio Code. Разработана программа кодирования алгоритмом Шеннона – Фано. Написана и отлажена программа.

# Приложение 1 – Код программы

https://github.com/SergeyUsachev021/lab2\_T\_inf\_2.git