

Национальный исследовательский университет
«Высшая школа экономики»
Факультет компьютерных наук
Департамент Программной инженерии

***Отчет по контрольному домашнему заданию
по дисциплине «Алгоритмы и структуры данных»***

Тема работы: Разработка программы, реализующей алгоритмы сжатия данных без потерь, их применение и проведение анализа полученных результатов.

Выполнил(а): студентка группы БПИ184(1)
_____ **Коннова М.Д.**
e-mail адрес: mdkonnova@edu.hse.ru

СОДЕРЖАНИЕ

1.	<i>ПОСТАНОВКА ЗАДАЧИ</i>	3
2.	<i>ОПИСАНИЕ АЛГОРИТМОВ И ИСПОЛЬЗОВАННЫХ СТРУКТУР ДАННЫХ</i>	4
3.	<i>ОПИСАНИЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ</i>	6
3.1.	Алгоритмы кодирования и декодирования.....	6
3.1.1.	Класс ShannonFanoCoder	6
3.1.2.	Класс LZ77Coder.....	7
3.1.3.	Класс LZWCoder.....	7
3.2.	Упаковка и распаковка данных	8
3.3.	Файловый ввод и вывод	9
4.	<i>ПЛАН ЭКСПЕРИМЕНТА</i>	11
5.	<i>ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ ЭКСПЕРИМЕНТА</i>	12
6.	<i>ДИАГРАММА КЛАССОВ</i>	14
7.	<i>ИСПОЛЬЗОВАННЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТА</i>	15
8.	<i>РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ</i>	16
9.	<i>СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ</i>	23
10.	<i>ВЫВОДЫ И ЗАКЛЮЧЕНИЕ</i>	24
11.	<i>ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ</i>	25
	<i>ПРИЛОЖЕНИЕ А</i>	26

1. ПОСТАНОВКА ЗАДАЧИ

1. Краткая постановка задачи

Разработать на языке C++ программу, реализующую алгоритмы Шеннона-Фано, LZ77 и LZW сжатия данных без потерь, получение архивного файла из исходного и разархивированного файла из архивного. Провести сравнительный анализ эффективности и времени работы каждого алгоритма.

2. Выполненные пункты

- реализованы сжатие и распаковка методом Шеннона-Фано
- реализованы сжатие и распаковки методом LZ77
- реализованы сжатие и распаковка методов LZW
- проведен вычислительный эксперимент
- построены таблицы и графики, оформлен отчет.

3. Невыполненные пункты

Отсутствуют.

2. ОПИСАНИЕ АЛГОРИТМОВ И ИСПОЛЬЗОВАННЫХ СТРУКТУР ДАННЫХ

- *Алгоритм Шеннона-Фано*

Основной принцип алгоритма в том, что он использует коды переменной длины: часто встречающийся символ кодируется кодом меньшей длины, редко встречающийся – кодом большей длины. Также соблюдается условие: никакое кодовое слово не является префиксом любого другого, что позволяет однозначно декодировать последовательность.

Этапы работы алгоритма следующие:

1. Во входной последовательности подсчитывается, сколько раз в ней встречается каждый символ. Будем называть это число вероятностью. Далее символы выписываются в порядке убывания вероятности.
2. Все символы делятся на две части так, чтобы суммарные вероятности обеих частей были максимально близки друг к другу.
3. В префиксном коде для первой части алфавита присваивается цифра 0, второй части – 1.
4. Полученные части рекурсивно делятся – алгоритм повторяется снова, пока не будет получена часть, содержащая только один символ.

Для декодирования, чтобы исключить просмотр всех существующих кодов на каждом шаге, используется префиксное дерево, построенное на основе всех кодов. На ребрах дерева записаны двоичные символы '0' или '1', а в листьях – символы, которые кодируются данной последовательностью. Путь от корня до символа и является кодом этого символа.

- *Алгоритм LZ77*

Идея, лежащая в основе LZ77, заключается в замене повторений на ссылки на позиции в тексте, где такие подстроки уже встречались. Информацию о повторении можно закодировать парой чисел — смещением назад от текущей позиции и длиной совпадающей подстроки. В целом, такие ссылки кодируются тройками из трех элементов: сдвига, длины и первого найденного после совпадающего фрагмента символа. Если совпадение найти не удалось, то очевидно, что сдвиг и длина в тройке равны 0.

Для эффективного поиска повторов применяется метод «скользящего окна» — совпадения ищутся не на всем обработанном префиксе, а в небольшом окне, состоящем из последних обработанных символов. Совокупность символов, входящих в это окно, называется словарем. Длина словаря задается заранее. Таким образом, при больших объемах ввода алгоритм тратит меньше времени за счет того, что просматривается не вся исходная строка. Также возможна ситуация, когда алгоритм забегаает вперед по анализируемой строке, и находит префикс, конец которого расположен позже текущего

символа. То, насколько далеко вперед можно просмотреть текущую строку задается другим ограничением – размером буфера предпросмотра.

- *Алгоритм LZW*

Особенность алгоритма в том, что в процессе итерирования по последовательности символов создается словарь фраз: последовательностям символов ставится некий номер, причем у каждой уникальной последовательности номер различен. Изначально словарь инициализируется однобуквенными словами. По мере итерирования алгоритм просматривает текст символ за символом слева направо. При чтении алгоритмом очередного символа в данной позиции находится строка W максимальной длины, совпадающая с какой-то фразой из словаря. Затем эта фраза подается на выход, а строка WK , где K — это символ, следующий за W во входном сообщении, вносится в словарь в качестве новой фразы и ей присваивается следующий доступный номер (так как W выбрана жадно, WK ещё не содержится в словаре). Символ K используется в качестве начала следующей фразы.

Для ускорения поиска наибольшего префикса, начинающегося в некоторой позиции исходной последовательности, в словаре, словарь представляется как префиксное дерево, при чем в каждой вершине записан номер, а на ребрах символы – тогда слово, которое получается при спуске от корня к этой вершине записано в словаре под этим номером.

3. ОПИСАНИЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ

В коде каждый алгоритм реализован в своем классе и входными данными для функции кодирования является объект `CharSequence`. `CharSequence` – это алиас для динамического массива с символами. Выходные данные для каждого алгоритма различаются. Непосредственно сразу запись результатов в память не происходит – это сделано для того, чтобы вынести функционал работы с файловой системой, а также с сжатием/распаковкой в отдельные классы, следуя принципу единственной ответственности.

Все классы, отвечающие за кодирование данных, а также за упаковку и распаковку результатов, покрыты тестами нескольких видов. В первых (файл *coders_test.cpp*) тестируется кодирование на небольших тестах, которые не сложно проверить руками. В тестах файла *packing_test.cpp* тестируется корректность записи результатов в файл – после считывания данные приводятся в вид, который может быть использован для декодирования, и сравниваются с теми, что были получены изначально. В файле *special_tests.cpp* расположены тесты на конкретные типы файлов (.jpg, .docx, .pptx, .exe, .bmp, .txt), исходные файлы расположены в папке *files/small-test-data-source*. Помимо побайтового сравнения результатов, как это было в предыдущем файле, декодированные файлы записываются с новыми именами в папку *files/small-test-data-results*, где их можно открыть и проверить на корректность. Для реализации тестов используется библиотека для модульного тестирования Google C++ Testing Framework.

Рассмотрим описание идей, которые заложены в реализации конструктора / метода кодирования / метода декодирования в каждом из трех классов, отвечающих за алгоритмы Шеннона-Фано, LZ77 и LZW соответственно, а также особенности упаковки и распаковки.

3.1. Алгоритмы кодирования и декодирования

3.1.1. Класс `ShannonFanoCoder`

3.1.1.1. Конструктор

На вход принимает последовательность символов, на основании которой для каждого символа будет получен битовый код. Сначала для каждого символа подсчитывается, сколько раз он встретился в исходной последовательности. Эти данные записываются в массив **`numberOfMatches`**, а сами символы – в **`values`**. Так, количество раз, которое встретился символ **`values[index]`**, записано в **`numberOfMatches[index]`**.

3.1.1.2. Кодирование

В начале строится дерево, в листьях записаны символы – такие вершины являются концами кодов, а на ребрах – 0 или 1. Лучшее разбиение выбирается посредством перебора всех возможных разделений. Вершина представлена структурой **`Node`**: так как дерево бинарное, у нее есть ссылки на левого/правого ребенка и ассоциированное с ней значение (в случае, если лист).

Затем из дерева формируется **`std::map`**, где ключом выступает символ, а значением – его код, **`std::map`** используется для кодирования переданной

последовательности символов. Результатом работы является словарь, код каждого символа из словаря и код полученный в результате зашифровки. Эти значения передаются при помощи структуры-обертки **Result**.

3.1.1.3. Декодирование

На вход функция декодирования принимает последовательность 0 и 1 - закодированные данные, и дерево с кодами символов, в листьях которого записаны символы, а на ребрах – 0 или 1. По закодированной последовательности двигается итератор одновременно со спуском по дереву. Как только в дереве встречается лист, найденный символ из листа записывается в ответ, а указатель на лист перезаписывается указателем на вершину дерева и поиск продолжается.

3.1.2. Класс LZ77Coder

3.1.2.1. Конструктор

На вход принимает размер словаря и размер всего окна, равного сумме размера словаря и размера буфера.

3.1.2.2. Кодирование

На вход подается последовательность символов. Результатом кодирования данных является набор троек, каждая из которых содержит сдвиг, длину совпадающего префикса и новый символ соответственно. Тройки представляются объектами структуры **Triple**, содержащей поля `offset` – сдвиг, `length` – длину и `character` – следующий символ. Во время прохода по кодируемой последовательности поддерживается четыре указателя – на конец/начало словаря и на конец/начало буфера предпросмотра. На каждом шаге итератор пробегает все индексы, которые покрывает окно словаря, с этого индекса ищется максимальный общий префикс. В конце лучшая длина присваивается тройке, корректно вычитывается отступ и в тройку записывается следующий новый символ.

3.1.2.3. Декодирование

Результатом является последовательность символов, на вход принимается набор троек. Декодирования реализовано итеративно: сначала к ответу записываются символы, расположенные левее, начиная со сдвига текущей тройки, указанной длины, затем новый символ.

3.1.3. Класс LZWCoder

3.1.3.1. Конструктор

Используется конструктор по умолчанию.

3.1.3.2. Кодирование

На каждой итерации по переданному набору символов с текущей позиции начинается поиск максимального совпадающего со словом из словаря префикса. Словарь хранится как префиксное дерево. Для его представления реализовано два

класса: **Node**, который является вершиной, (у каждой вершины есть поле `index` – ее номер в словаре, поле `character` – символ, который с ней ассоциирован, и `next` – словарь ссылок на детей), и **Tree**, являющийся оберткой над корнем и хранящий число слов в дереве. Размер алфавита равен 256, поэтому операции поиска/вставки в дерево работают за $O(n)$, где n – текущее количество слов в словаре.

Результат работы записывается в структуру **Result**, содержащую начальный словарь (только односимвольные слова) и полученную последовательность кодов.

3.1.3.3. Декодирование

На вход подается объект структуры **Result**, в котором записан начальный словарь (содержащий только односимвольные слова), который позже будет расширен, и набор кодов.

Особого внимания заслуживает случай, когда в массиве кодов встречается значение, которое еще не было получено. Значит этот объект является следующим, который будет добавлен в словарь. Достаточно хранить его начало, чтобы однозначно записать новое слово в словарь: последним элементом нового слова будет его первый символ. Словарь хранится при помощи **std::map**, где ключом является целочисленный номер слова, а значением – само слово как набор символов.

3.2. Упаковка и распаковка данных

Таблица 1 показывает, какие методы были реализованы в классах **Packer** (отвечает за упаковку данных) и **Unpacker** (отвечает за распаковку данных) для работы с каждым из трех алгоритмов. Каждый класс на вход принимает имя файла, с которым будет работать. Так как вывод каждого из них отличается, то принцип упаковки тоже различен. Все константы вынесены в файл *declarations.cpp*. При упаковке данные не пишутся сразу в файл, и когда речь идет о том, что записывается очередной символ, подразумевается сначала запись байтов в массив байтов `CharSequence`, который после передается объекту класса **Converter** для непосредственной записи в файл. Распаковщики же на вход принимают последовательность байт, которая была получена при считывании файла объектом класса **Converter**.

Название класса	Shannon-Fano	LZ77	LZW
<i>Packer</i>	Первой записывается длина словаря (8 бит). Затем каждый символ словаря следующим образом: сам символ (8 бит), длина его кода (8 бит) и сам код указанной	Важными параметрами, используемыми при записи, являются длина словаря и длина буфера. Сдвиг может принимать значения из отрезка $[0, \text{длина словаря}]$, однако случай, когда сдвиг равен нулю особый: длина тогда тоже всегда равна 0. Поэтому этот случай записываем как нулевой сдвиг и нулевая длина. Получается, что на запись сдвига тратится $\log_2(\text{длина словаря})$. На	Сначала записывается длина словаря, состоящего из однобуквенных слов (8 бит). Затем записывается каждый символ (на каждый тратится по 8 бит). Порядок записи важен, запись происходит

	длины. После этого записывается длина полученного кода (32 бита) и сам код этой длины.	запись длины тратится $\log_2(\text{длина буфера}) + 1$, если длина буфера является степенью двух, и $\log_2(\text{длина буфера})$ округленный вверх иначе. Символ записываются 8 битами.	в порядке возрастания номера символа в словаре. Затем записывается размер массива с кодами (32 бита) и сами коды (на каждый код – 32 бита).
<i>Unpacker</i>	Распаковка происходит аналогично упаковке: сначала однобайтовый размер словаря, затем каждый символ словаря, длина его код и непосредственно код. После считывания словаря происходит считывание длины кода, кодирующего исходную строку, и самого кода.	Распаковка происходит аналогично упаковке, особенностью является то, что размер сдвига всегда увеличивается на один, если длина не нулевая – иначе сдвиг остается равен нулю. Все содержимое считанного файла подается функции распаковщика на вход и тройки считываются друг за другом.	Распаковка аналогично упаковке. Номер распакованного однобуквенного слова в словаре восстанавливается в порядке считывания. Сначала считывается словарь, а затем массив с кодами.

Таблица 1 – Структура классов, реализующих упаковку/распаковку.

3.3. Файловый ввод и вывод

Задача работы с файлами вынесена в класс Converter. В таблице 2 приведено краткое описание его методов.

Название метода	Описание
<code>void writeBinaryStringToFile(CharSequence& bits, const std::string& outputFileName)</code>	Принимает на вход последовательность символов '0' и '1' и записывает ее в бинарный файл, упаковывая данные в байты. Для упаковки 8-ми идущих подряд символов использует вспомогательную функцию <i>packByte</i> , принимающую на вход всю последовательность и индекс, откуда

	следует начинать кодировать. При помощи битовых операций собирается новое число, записывается в файл и алгоритм переходит к следующей восьмерке символов. Если предварительно переданная последовательность символов не кратна 8, дописывает в конец недостающее количество нулей.
<pre>void writeCharSequenceToABinaryFile(const std::string& fileName, const CharSequence& source)</pre>	В бинарный файл с переданным именем записывает переданную последовательность символов. Каждый символ представляет собой один байт.
<pre>CharSequence readBinaryFile(const std::string& filename)</pre>	Открывает бинарный файл с переданным именем на чтение и считывает его побайтово, записывая ответ в последовательность символов.

Таблица 2 – Методы класса, реализующего файловый ввод/вывод.

4. ПЛАН ЭКСПЕРИМЕНТА

Разделим проведение эксперимента на несколько частей:

1. Реализация алгоритмов кодирования, методов сжатия и распаковки данных.
2. Подготовка файлов, содержимое которых будет кодироваться / декодироваться.
3. Проведения анализа подготовленных файлов: подсчет для каждого символа частоты встречаемости в каждом файле, вычисление энтропии файлов.
4. Применение алгоритмов с анализом времени работы каждого из них, а также с измерением памяти, требуемой для хранения полученного закодированного файла, вычисление коэффициента сжатия.
5. Анализ полученных данных: составление таблиц, графиков и диаграмм.
6. Подведение итогов об эффективности тех или иных алгоритмов при работе с различными форматами данных.

5. ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ ЭКСПЕРИМЕНТА

Для проведения эксперимента создано несколько вспомогательных классов.

- Класс *Watch* используется для измерения времени в наносекундах, время получает с помощью функции языка C **int clock_gettime(clockid_t clock_id, struct timespec *t_time);** определенной в заголовочном файле **time.h**, и позволяющей определить текущее время с высокой точностью.
- Класс *CSVWriter* используется для записи данных построчно в файл с расширением csv. Функция **void addRow(T first, T last);** принимает два итератора, указывающих на начало и конец записываемого фрагмента коллекции.
- Класс *Experimenter* содержит различные методы, используемые для проведения экспериментов. Описание методов для наглядности представлено в таблице 3.

Метод	Назначение
void overallTesting(const std::string csvFileName)	<p>Используется для запуска всех по-очереди алгоритмов кодирования для каждого тестового файла. Результаты построчно записываются в csv-файл с переданным именем csvFileName. В столбцах слева направо расположены следующие характеристики файлов:</p> <ul style="list-style-type: none"> • размер исходного файла в Кб • энтропия файла • размер файла, сжатого алгоритмом Шеннона-Фано • коэффициент сжатия алгоритмом Шеннона-Фано • время распаковки алгоритмом Шеннона-Фано • время упаковки алгоритмом Шеннона-Фано • размер файла, сжатого алгоритмом LZ77 с размером скользящего окна 5 Кб • коэффициент сжатия алгоритмом LZ77 с размером скользящего окна 5 Кб • время распаковки алгоритмом LZ77 с размером скользящего окна 5 Кб • время упаковки алгоритмом LZ77 с размером скользящего окна 5 Кб • размер файла, сжатого алгоритмом LZ77 с размером скользящего окна 10 Кб

	<ul style="list-style-type: none"> • коэффициент сжатие алгоритмом LZ77 с размером скользящего окна 10 Кб • время распаковки алгоритмом LZ77 с размером скользящего окна 10 Кб • время упаковки алгоритмом LZ77 с размером скользящего окна 10 Кб • размер файла, сжатого алгоритмом LZ77 с размером скользящего окна 20 Кб • коэффициент сжатие алгоритмом LZ77 с размером скользящего окна 20 Кб • время распаковки алгоритмом LZ77 с размером скользящего окна 20 Кб • время упаковки алгоритмом LZ77 с размером скользящего окна 20 Кб • размер файла, сжатого алгоритмом LZW • коэффициент сжатие алгоритмом LZW • время распаковки алгоритмом LZW • время упаковки алгоритмом LZW
<code>double calculateEntropy(const std::string& filename)</code>	<p>Вычисляет энтропию файла с переданным именем в соответствии с формулой:</p> $-\sum_{i=1}^m w_i \log_2 w_i$ <p>где w_i — частота, с которой в файле встречается i-ый символ, m — общее количество символов.</p>
<code>long long getFileSize(const std::string& filename) const</code>	<p>Вычисляет размер файла с переданным именем в байтах.</p>
<code>double calculateEntropy(const std::string& filename) const</code>	<p>Вычисляет размер файла с переданным именем в Кб.</p>
<code>void measureFilesPrintResult(const std::string& filename) const</code>	<p>Записываем в csv-файл с переданным именем информацию о частоте появления каждого символа из $[0, 255]$ в файле. В i-ой строке j-ом столбце записана информация о символе i в файле с номером j. Для подсчета используется структура данных <code>std::map</code>.</p>

Таблица 3 – Описание методов класса Experimenter.

6. ДИАГРАММА КЛАССОВ

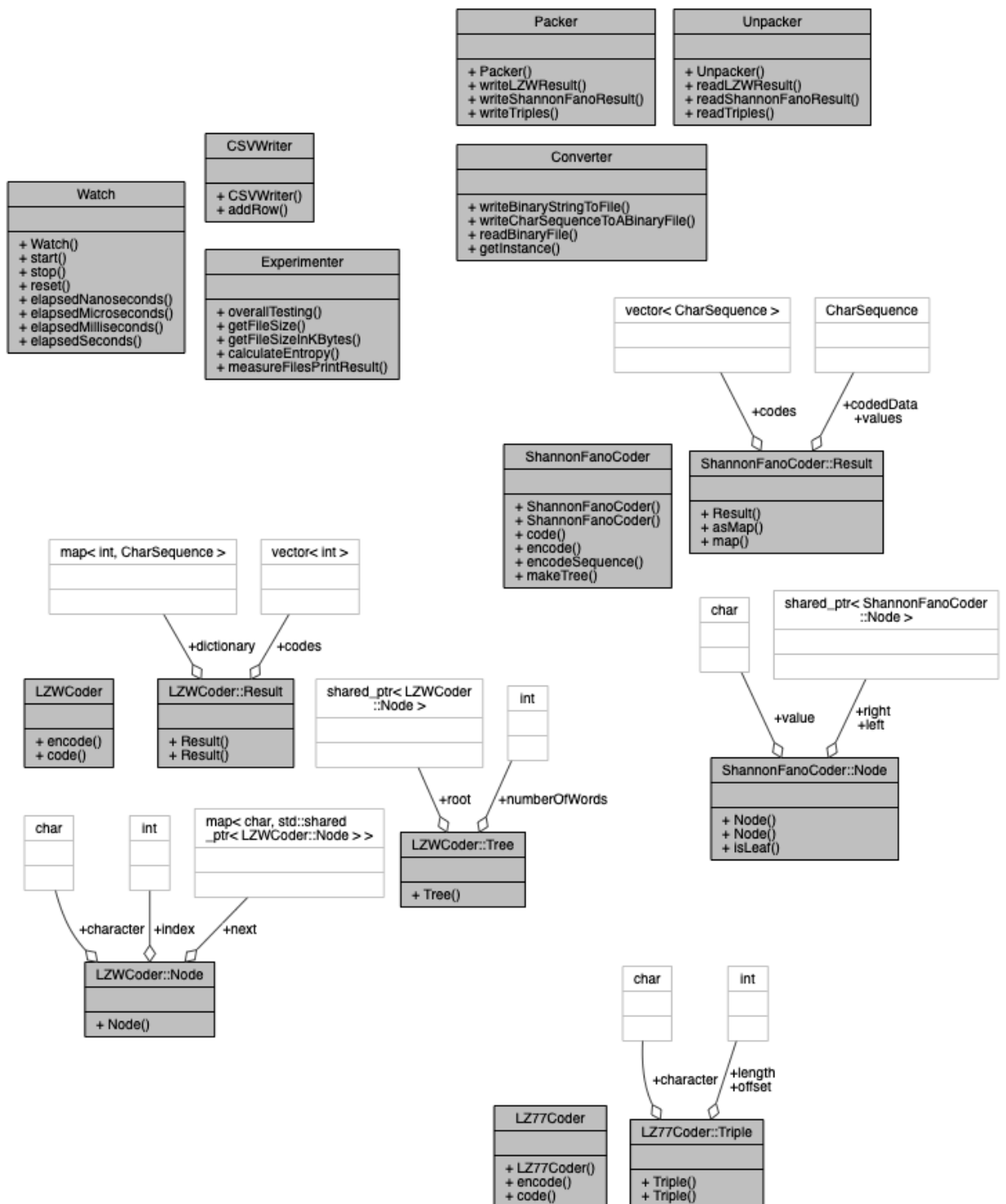


Рисунок 1 – Диаграмма классов

7. ИСПОЛЬЗОВАННЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТА

Разработка велась на операционной системе macOS Catalina с использованием среды разработки CLion 2019.2.5. В процессе проведения эксперимента было сформировано два csv-файла: `frequency.csv`, содержащий частоты появления символов в каждом файле, и `overall.csv`, содержащий характеристики файлов (размер и энтропию) и результаты работы каждого алгоритма и упаковщика.

На основе данных, записанных в этих файлах, были построены таблицы и диаграммы. Для этого использовалась командная оболочка для интерактивных вычислений **Jupyter Notebook**, язык программирования **Python 3.7** и библиотеки **pandas**, **numpy**, **matplotlib**. Исходный файл Jupyter Notebook *analyze.ipynb* расположен в папке `src/experiments/output`.

Для построения диаграммы классов использовалась кроссплатформенная система документирования исходных текстов **Doxygen** и фреймворк **Graphviz**.

8. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Перечень исходных файлов, выбранных для эксперимента:

- 1.txt – Текстовый файл, содержащий несколько книг.
- 2.pptx – Презентация.
- 3.pdf – Книга по английскому языку.
- 4.docx – Файл с текстом и картинками.
- 5.jpg – Цветное изображение.
- 6.jpg – Черно-белое изображение.
- 7.bmp – В цвете.
- 8.bmp – Черно-белая картинка.
- 9.dll – Динамически подключаемая библиотека.
- 10.exe – Исполняемый файл.
- 11.mp4 – Видео.
- 12.txt – Текстовый файл, полученный повторением одних и тех же строк.

В таблицах 4-5 представлены частоты встречаемости символов в файлах из тестового набора (полные таблицы находятся в приложении А). Диаграммы частот для каждого файла из тестового набора изображены на рисунке 2.

файл	1	2	3	4	5	6
символ	Частота появления символа					
0	0.00e+00	3.24e-02	9.05e-03	9.22e-03	7.91e-03	1.51e-02
1	0.00e+00	4.31e-03	3.26e-03	4.22e-03	3.21e-03	3.50e-03
2	0.00e+00	3.75e-03	3.43e-03	3.63e-03	2.18e-03	2.16e-03
...
Энтропия H	4.627848	7.933153	7.978228	7.992566	7.958278	7.898119

Таблица 4 – Частоты появления символов в файлах 1-6, энтропия файлов 1-6

файл	7	8	9	10	11	12	
символ	Частота появления символа						
0	8.75e-02	1.19e-03	1.36e-01	1.45e-02	5.08e-02	0.00e+00	1.51e-02
1	2.44e-05	5.59e-03	9.46e-03	4.52e-03	9.37e-03	0.00e+00	3.50e-03
2	0.00e+00	1.26e-02	4.70e-03	4.53e-03	5.48e-03	0.00e+00	2.16e-03
...	
Энтропия H	6.968415	7.568010	6.626704	7.982869	7.866040	4.116830	

Таблица 5 – Частоты появления символов в файлах 7-12, энтропия файлов 7-12

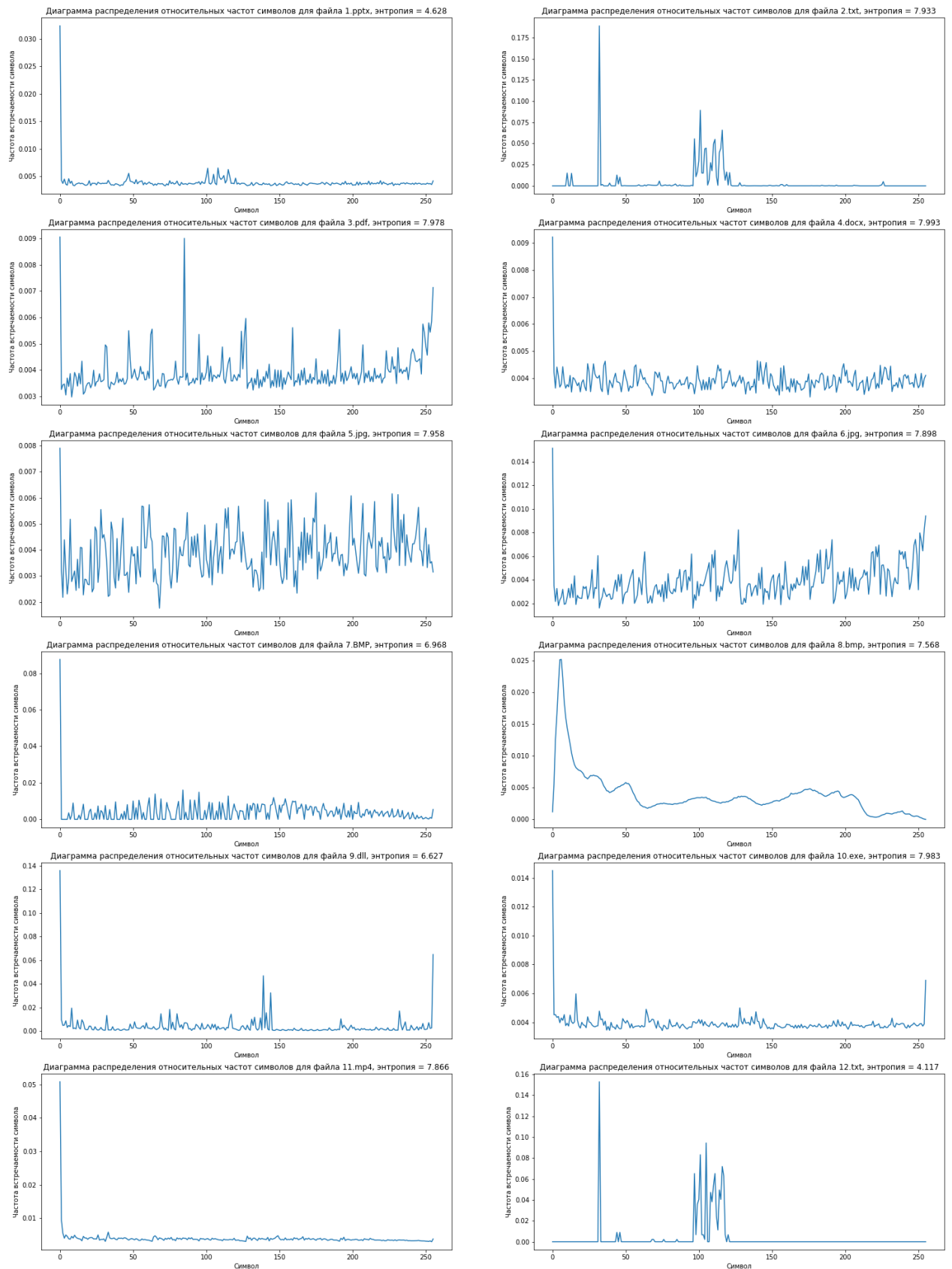


Рисунок 2 – Диаграммы распределения относительных частот символов для тестовых файлов

Таблица 6 отображает коэффициенты сжатия данных (K), размер каждого файла (S1, измеряется в Кб), энтропию каждого файла (H), размер сжатого файла исследуемыми алгоритмами (S2, измеряется в Кб).

Имя файла	S1, Кб	H	Алгоритм Шеннона-Фано		Алгоритм LZ77, окно 5 Кб		Алгоритм LZ77, окно 10 Кб		Алгоритм LZ77, окно 20 Кб		Алгоритм LZW	
			S2, Кб	K	S2, Кб	K	S2, Кб	K	S2, Кб	K	S2, Кб	K
1.pptx	4568.51	7.933	4548.54	0.996	7671.620	1.679	7870.040	1.723	7784.460	1.704	8076.540	1.768
2.txt	4600.58	4.628	2692.57	0.585	3394.900	0.738	3272.960	0.711	3106.420	0.675	2875.510	0.625
3.pdf	4308.05	7.978	4317.73	1.002	7649.990	1.776	7848.470	1.822	7929.250	1.841	8071.890	1.874
4.docx	4519.20	7.993	4529.84	1.002	8344.040	1.846	8590.880	1.901	8645.800	1.913	8778.820	1.943
5.jpg	4782.90	7.958	4789.46	1.001	8840.110	1.848	9071.310	1.897	9086.790	1.900	8945.350	1.870
6.jpg	4392.70	7.898	4365.48	0.994	7966.580	1.814	8134.480	1.852	8120.940	1.849	8077.570	1.839
7.bmp	4164.37	6.968	3661.23	0.879	5152.740	1.237	4510.110	1.083	4278.350	1.027	4671.730	1.122
8.bmp	4602.20	7.568	4384.53	0.953	2690.500	0.585	2606.760	0.566	2523.960	0.548	2421.320	0.526
9.dll	4760.00	6.627	3965.51	0.833	3996.580	0.840	3938.120	0.827	3898.870	0.819	4764.880	1.001
10.exe	3812.18	7.983	3814.88	1.001	6966.550	1.827	7208.590	1.891	7284.610	1.911	7418.540	1.946
11.mp4	4295.13	7.866	4250.32	0.990	7598.490	1.769	7861.280	1.830	7946.000	1.850	8030.210	1.870
12.txt	4106.69	4.117	2139.94	0.521	16.136	0.004	8.919	0.002	5.076	0.001	236.317	0.058

Таблица 6 – Таблица, отображающая коэффициенты сжатия данных

Таблица 7 отображает размер каждого файла (S1, измеряется в Кб), энтропию каждого файла (H), время упаковки данных исследуемыми алгоритмами (TP, в секундах) и время распаковки данных (TU, в секундах).

Имя файла	S1, Кб	H	Алгоритм Шеннона-Фано		Алгоритм LZ77, окно 5 Кб		Алгоритм LZ77, окно 10 Кб		Алгоритм LZ77, окно 20 Кб		Алгоритм LZW	
			TU, с	TP, с	TU, с	TP, с	TU, с	TP, с	TU, с	TP, с	TU, с	TP, с
1.pptx	4568.51	7.933	8.842	9.069	7.769	85.754	7.649	144.766	7.559	251.138	18.394	17.929
2.txt	4600.58	4.628	5.407	6.501	3.614	42.182	3.518	69.383	3.365	115.918	7.202	7.687
3.pdf	4308.05	7.978	9.027	8.676	7.507	83.870	7.646	144.639	7.665	255.697	18.344	17.865
4.docx	4519.20	7.993	8.783	9.079	8.950	91.507	9.112	158.769	9.093	278.757	20.701	20.101
5.jpg	4782.90	7.958	9.241	9.637	9.374	96.719	9.530	167.075	9.499	294.076	21.217	20.590
6.jpg	4392.70	7.898	8.507	8.751	7.813	87.496	7.911	148.942	8.210	262.409	18.587	18.385
7.bmp	4164.37	6.968	6.843	7.257	5.372	60.012	4.837	87.812	4.659	145.271	11.375	10.699
8.bmp	4602.20	7.568	8.369	7.687	3.015	30.650	2.938	49.686	2.959	85.254	6.656	7.154
9.dll	4760.00	6.627	7.696	7.915	4.247	48.376	4.032	78.427	3.983	133.286	11.640	11.486
10.exe	3812.18	7.983	7.231	7.357	7.073	78.715	7.271	136.458	7.085	237.017	16.881	16.349
11.mp4	4295.13	7.866	8.293	8.330	7.693	84.877	7.907	149.430	7.918	265.031	19.688	19.552
12.txt	4106.69	4.117	5.287	5.868	0.397	0.905	0.404	1.150	0.437	2.231	1.399	3.008

Таблица 7 – Таблица, отображающая время упаковки и распаковки данных

На рисунке 3 представлены столбчатые диаграммы, отражающие коэффициенты сжатия каждого файла из тестового набора для каждого алгоритма.

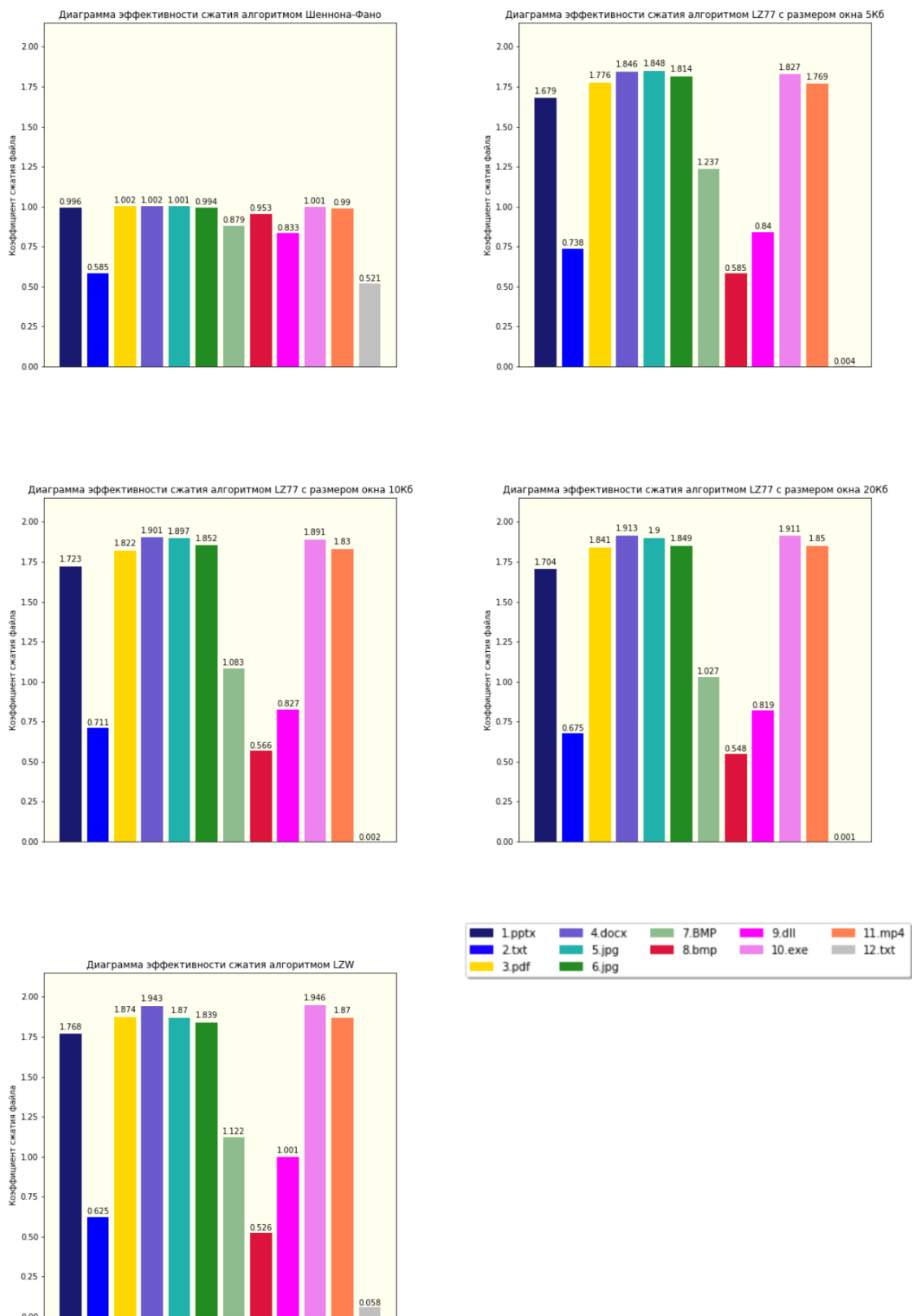


Рисунок 3 – Диаграммы, отражающие коэффициенты сжатия файлов исследуемыми алгоритмами

На рисунке 4 представлены столбчатые диаграммы, отражающие время упаковки каждого файла из тестового набора для каждого алгоритма.

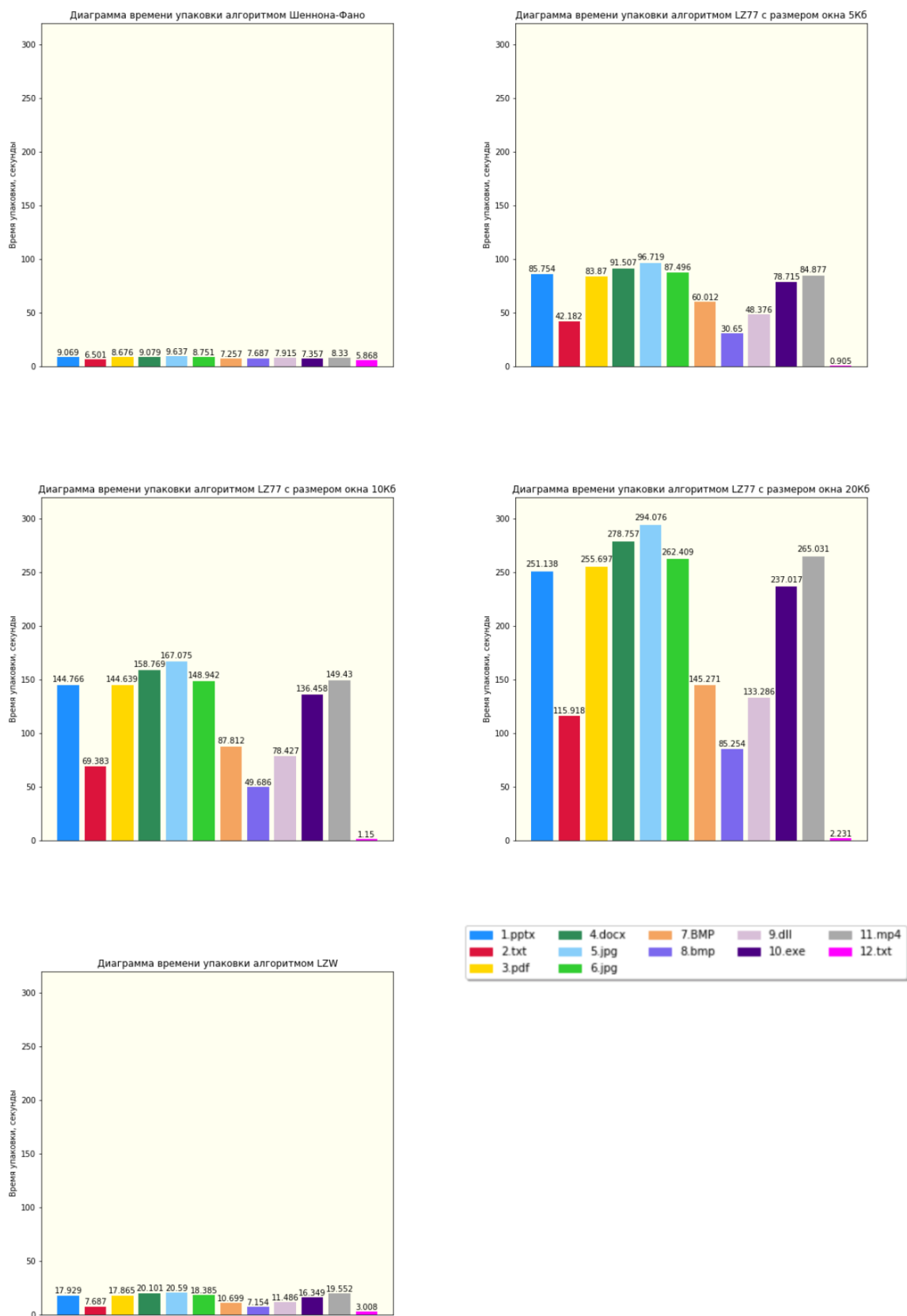


Рисунок 4 – Диаграммы, отражающие время упаковки файлов исследуемыми алгоритмами

На рисунке 5 представлены столбчатые диаграммы, отражающие время распаковки каждого файла из тестового набора для каждого алгоритма.

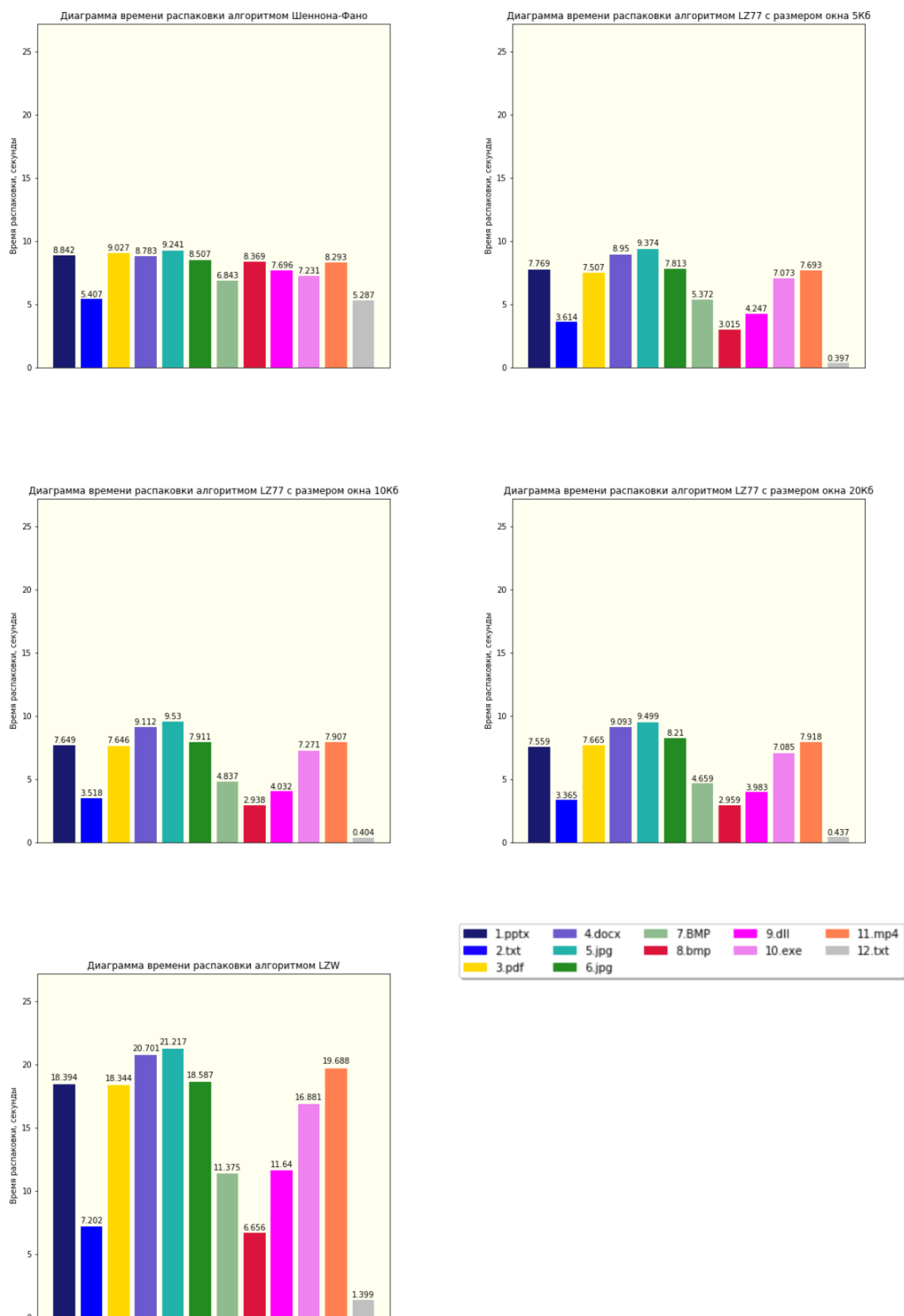


Рисунок 5 – Диаграммы, отражающие время распаковки файлов исследуемыми алгоритмами

На рисунке 6 представлены столбчатые диаграммы, отражающие суммарное время распаковки и упаковки каждого файла из тестового набора для каждого алгоритма. Они построены для того, чтобы оценивать время работы алгоритмов в целом, так как в реальной ситуации нельзя упаковав одним из них, использовать другой для распаковки.

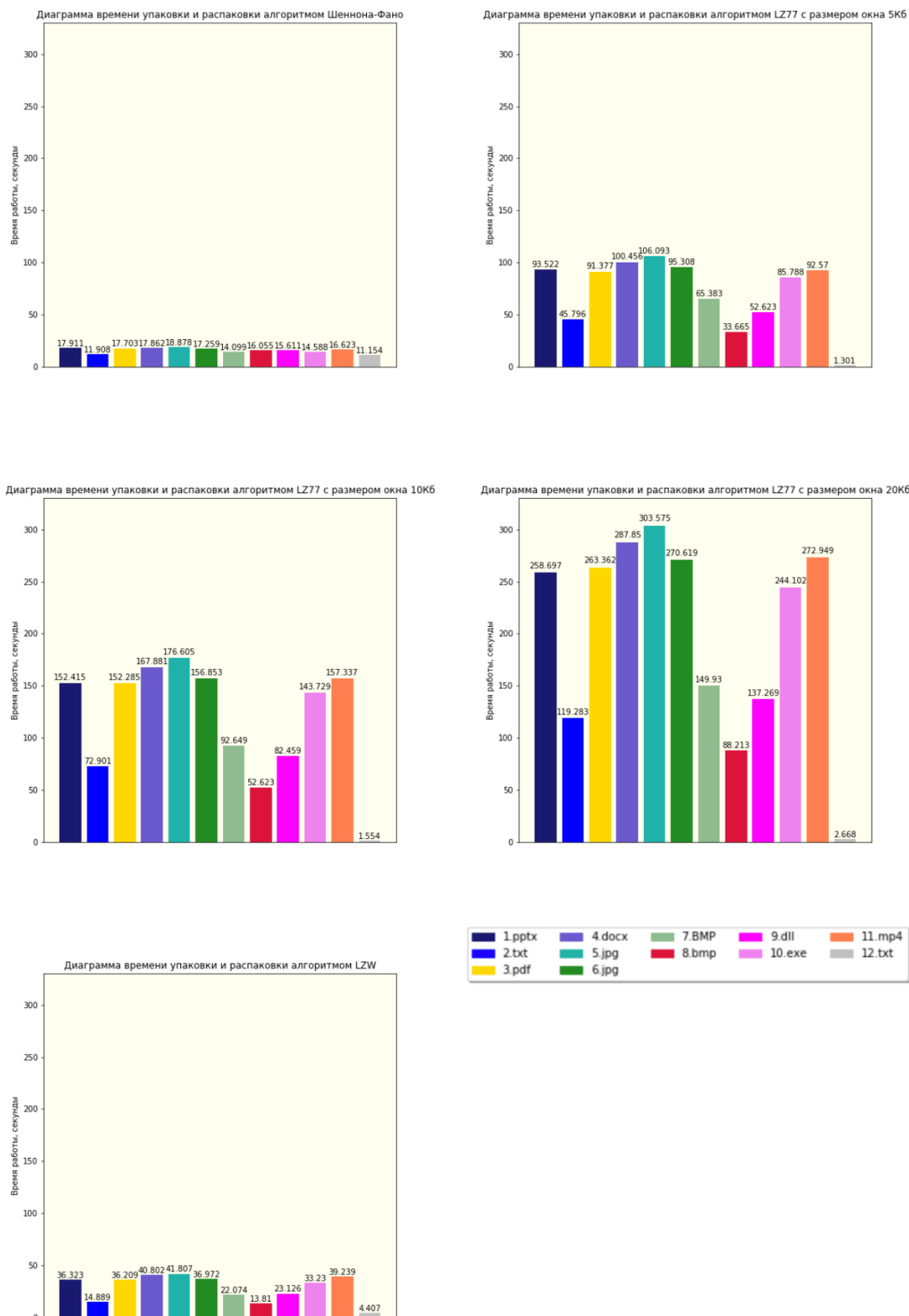


Рисунок 6 – Диаграммы, отражающие суммарное время упаковки и распаковки файлов исследуемыми алгоритмами

9. СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ

1. Во-первых, проанализируем тестовые файлы. Они были выбраны так, чтобы их размер порядка 4-5 Мб. Заметим, что наименьшая энтропия у текстовых файлов 2.txt и 12.txt. Наибольшая энтропия у файлов форматов .docx и .pptx. Взглянув на диаграммы распределения относительных частот символов, можно выделить среди файлов те, в которых очень велика частота появления одних и тех же символов – к таким относятся 1.pptx, 2.txt, 7.bmp (цветное изображение), 8.bmp (черно-белое изображение), 9.dll, 10.exe, 11.mp4, 12.txt.
2. Во-вторых, рассмотрим показатели коэффициентов сжатия для каждого из исследуемых алгоритмов. Коэффициенты сжатия алгоритмом Шеннона-Фано для каждого файла (за исключением текстовых) были довольно близки к единице. Работа алгоритма на текстовых данных показала сжатие в два раза.
3. Наилучшие коэффициенты сжатия алгоритм LZ77 показал на текстовых файлах, причем на файле, созданном путем повторения одной и той же строки, этот коэффициент минимален из всех представленных. Хорошие коэффициенты сжатия LZ77 также были получены для файлов 8.bmp и 9.dll – в них частота появления одинаковых символов высокая, поэтому LZ77, использующий сжатие за счет поиска повторяющихся фрагментов, показал снижение размера закодированного файла по сравнению с исходным. Заметим, что при увеличении размера окна коэффициенты сжатия этих файлов уменьшались. А вот коэффициенты сжатия изображений в формате .jpg и файла в формате .docx увеличивались с увеличением размера окна.
4. Показатели работы алгоритма LZW низкие на текстовых файлах, а также на черно-белом изображении в формате bmp. Сжатие данных происходит в примерно 2 раза. На остальных файлах размер упакованного файла превышает размер исходного.
5. Далее перейдем к рассмотрению результатов измерения времени работы алгоритмов. Упаковка методом Шеннона-Фано по времени практически не превышала распаковку. Наилучшие показатели при работе с текстовыми данными – с файлами, чья энтропия наименьшая.
6. Время упаковки методом LZ77 в десятки раз превосходит время распаковки. Наиболее быстро алгоритм упаковал текстовые файлы 2.txt и 12.txt, а также 8.bmp и 9.dll, худшие результаты он показал при работе с изображениями формата jpg.
7. Про алгоритм LZW однозначно сказать, была ли быстрее распаковка чем упаковка или наоборот нельзя, так как показатели варьировались от файла к файлу. Наиболее быстро алгоритм упаковал и распаковал текстовые файлы, оба изображения в формате .bmp, а также файл 9.dll. Худший результат по времени работы LZW показал на файлах 4.docx и 5.jpg.

10. ВЫВОДЫ И ЗАКЛЮЧЕНИЕ

На основе проведенного анализа и поставленного эксперимента для каждого формата можно сказать, какой алгоритм следует выбрать для достижения либо хорошего сжатия, либо небольших затрат по времени. Полученные выводы представлены в таблице 8.

Название файла	Лучший алгоритм для достижения хорошего коэффициента сжатия	Лучший алгоритм для достижения быстрого времени работы
1.pptx	Шеннон-Фано	Шеннон-Фано
2.txt	Шеннон-Фано	Шеннон-Фано
3.pdf	Шеннон-Фано	Шеннон-Фано
4.docx	Шеннон-Фано	Шеннон-Фано
5.jpg	Шеннон-Фано	Шеннон-Фано
6.jpg	Шеннон-Фано	Шеннон-Фано
7.bmp	Шеннон-Фано	Шеннон-Фано
8.bmp	LZW	LZW
9.dll	LZ77 (размер окна 20 Кб)	Шеннон-Фано
10.exe	Шеннон-Фано	Шеннон-Фано
11.mp4	Шеннон-Фано	Шеннон-Фано
12.txt	LZ77 (размер окна 20 Кб)	LZ77 (размер окна 5 Кб)

Таблица 8 – Выбор наилучшего алгоритма для кодирования файла с учетом коэффициента сжатия и времени работы.

Отдельно по времени распаковки LZ77 является самым быстрым алгоритмом среди исследуемых на всех файлах, кроме 4.docx и 5.jpg, однако во времени упаковки очень сильно проигрывает остальным. Будем оценивать время работы алгоритма в целом, включая и упаковку, и распаковку, так как результат упаковки одного алгоритма не может являться входными данными для другого.

Получается, что алгоритмы LZ77 и LZW хорошо работают с файлами, распределение встречаемости символов в которых далеко от равномерного, в которых много повторяющихся символов и энтропия которых мала. LZW работает максимально быстро и получает минимальный по весу упакованный файл при работе с черно-белыми изображениями формата .bmp. LZ77 показывает хорошие результаты по эффективности упаковки файла .dll и текстовых файлов, в которых содержится много повторяющихся фрагментов. По времени работы LZ77 быстро распаковывает те же файлы с повторениями. В остальных случаях наиболее выгодным является использование алгоритма Шеннона-Фано – в среднем на каждом файле показатели работы алгоритма не сильно различаются, частота появления тех или иных символов не оказывает на алгоритм Шеннона-Фано такое сильное влияние, как на LZ77 и LZW, для которых малые повторения символов становятся катастрофическими.

11. ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Документация к кроссплатформенной среде документирования исходных кодов Doxygen. [Электронный ресурс] // Doxygen: [сайт]. URL: <http://www.doxygen.nl> (дата обращения: 02.04.2020).
2. Справка по языку C++. [Электронный ресурс] // Справка по языку C++: [сайт]. URL: <https://ru.cppreference.com/w/> (дата обращения: 02.04.2020).
3. Документация к библиотеке Pandas. [Электронный ресурс] // Pandas: [сайт]. URL: <https://pandas.pydata.org/pandas-docs/stable/index.html> (дата обращения: 02.04.2020).
4. Википедия. Свободная энциклопедия. [Электронный ресурс] // Википедия. Свободная энциклопедия: [сайт]. URL: <https://ru.wikipedia.org/wiki/> (дата обращения: 01.04.2020).
5. Ватолин Д., Ратушняк А., Смирнов М., Юкин В., Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: ДИАЛОГ-МИФИ, 2003.

ПРИЛОЖЕНИЕ А

ТАБЛИЦА С ЧАСТОТОЙ ПОЯВЛЕНИЯ СИМВОЛОВ В ФАЙЛАХ.

файл	1	2	3	4	5	6
символ	Частота появления символа					
0	0.00e+00	3.24e-02	9.05e-03	9.22e-03	7.91e-03	1.51e-02
1	0.00e+00	4.31e-03	3.26e-03	4.22e-03	3.21e-03	3.50e-03
2	0.00e+00	3.75e-03	3.43e-03	3.63e-03	2.18e-03	2.16e-03
3	0.00e+00	4.54e-03	3.46e-03	4.41e-03	4.39e-03	3.27e-03
4	0.00e+00	3.62e-03	3.05e-03	4.08e-03	3.22e-03	1.82e-03
5	0.00e+00	3.42e-03	3.69e-03	3.67e-03	2.31e-03	2.30e-03
6	0.00e+00	4.61e-03	3.37e-03	3.77e-03	2.89e-03	2.53e-03
7	0.00e+00	3.77e-03	3.86e-03	4.42e-03	5.18e-03	3.20e-03
8	0.00e+00	4.14e-03	2.97e-03	3.88e-03	2.78e-03	1.93e-03
9	0.00e+00	3.41e-03	3.37e-03	3.63e-03	2.98e-03	1.98e-03
10	1.51e-02	3.33e-03	3.90e-03	3.76e-03	3.19e-03	2.53e-03
11	0.00e+00	3.63e-03	3.74e-03	3.68e-03	2.44e-03	3.28e-03
12	0.00e+00	3.72e-03	3.39e-03	4.10e-03	3.64e-03	2.48e-03
13	1.48e-02	3.84e-03	3.87e-03	3.49e-03	2.54e-03	3.64e-03
14	0.00e+00	3.68e-03	3.47e-03	4.02e-03	4.10e-03	2.50e-03
15	0.00e+00	3.77e-03	4.33e-03	3.97e-03	4.08e-03	4.34e-03
16	0.00e+00	3.66e-03	3.08e-03	3.85e-03	2.27e-03	1.91e-03
17	0.00e+00	3.45e-03	3.16e-03	3.72e-03	2.88e-03	2.68e-03
18	0.00e+00	3.41e-03	3.39e-03	3.83e-03	2.85e-03	2.48e-03
19	0.00e+00	3.55e-03	3.50e-03	3.49e-03	2.68e-03	2.45e-03
20	0.00e+00	4.23e-03	3.52e-03	3.84e-03	2.66e-03	2.42e-03
21	0.00e+00	3.36e-03	3.32e-03	3.94e-03	4.40e-03	3.44e-03
22	0.00e+00	3.78e-03	3.47e-03	3.68e-03	2.39e-03	3.26e-03
23	0.00e+00	3.75e-03	3.99e-03	3.54e-03	2.52e-03	3.43e-03
24	0.00e+00	3.77e-03	3.38e-03	4.54e-03	4.88e-03	2.37e-03
25	0.00e+00	3.49e-03	3.55e-03	4.23e-03	4.73e-03	2.76e-03
26	0.00e+00	3.95e-03	3.58e-03	3.50e-03	2.75e-03	3.41e-03
27	0.00e+00	3.77e-03	3.85e-03	4.00e-03	3.37e-03	4.86e-03
28	0.00e+00	3.67e-03	3.55e-03	4.53e-03	5.55e-03	2.43e-03
29	0.00e+00	3.78e-03	3.58e-03	4.15e-03	4.45e-03	3.35e-03
30	0.00e+00	3.72e-03	3.63e-03	4.03e-03	4.59e-03	3.26e-03
31	0.00e+00	3.78e-03	4.95e-03	4.01e-03	4.26e-03	6.06e-03
32	1.89e-01	3.75e-03	4.88e-03	4.10e-03	3.48e-03	1.61e-03
33	6.11e-04	4.30e-03	3.38e-03	3.62e-03	2.21e-03	2.24e-03
34	1.68e-03	3.79e-03	3.27e-03	3.50e-03	2.27e-03	2.56e-03

35	1.27e-06	3.46e-03	3.54e-03	4.46e-03	5.07e-03	3.31e-03
36	1.91e-06	3.49e-03	3.48e-03	4.63e-03	4.70e-03	2.87e-03
37	8.49e-07	3.41e-03	3.44e-03	3.78e-03	2.94e-03	2.57e-03
38	3.82e-06	3.69e-03	3.56e-03	3.39e-03	2.54e-03	2.75e-03
39	3.26e-03	3.63e-03	3.91e-03	3.92e-03	4.44e-03	2.80e-03
40	9.19e-05	3.52e-03	3.51e-03	3.80e-03	2.82e-03	2.35e-03
41	9.15e-05	3.31e-03	3.65e-03	3.65e-03	3.26e-03	2.43e-03
42	3.93e-05	3.49e-03	3.55e-03	3.99e-03	4.33e-03	3.26e-03
43	2.55e-06	3.44e-03	3.69e-03	4.18e-03	5.22e-03	3.99e-03
44	1.28e-02	4.00e-03	3.45e-03	3.86e-03	3.02e-03	3.03e-03
45	2.12e-03	4.15e-03	3.49e-03	3.85e-03	3.03e-03	4.44e-03
46	1.02e-02	4.74e-03	3.67e-03	3.81e-03	3.15e-03	3.45e-03
47	2.44e-05	5.58e-03	5.49e-03	3.49e-03	2.37e-03	4.47e-03
48	9.04e-05	4.16e-03	4.41e-03	3.85e-03	3.40e-03	1.98e-03
49	2.44e-04	4.08e-03	3.68e-03	4.30e-03	4.12e-03	2.71e-03
50	1.35e-04	3.98e-03	3.82e-03	4.04e-03	3.76e-03	2.95e-03
51	8.72e-05	3.70e-03	4.03e-03	3.86e-03	3.85e-03	2.97e-03
52	6.11e-05	4.43e-03	3.74e-03	3.52e-03	2.69e-03	3.52e-03
53	5.41e-05	3.67e-03	3.61e-03	3.72e-03	4.13e-03	4.10e-03
54	5.07e-05	4.02e-03	3.76e-03	3.65e-03	3.28e-03	4.46e-03
55	4.22e-05	4.11e-03	4.13e-03	3.69e-03	2.96e-03	4.87e-03
56	5.16e-05	4.22e-03	3.85e-03	4.42e-03	5.68e-03	2.01e-03
57	4.59e-05	3.49e-03	3.92e-03	4.49e-03	5.66e-03	2.31e-03
58	2.84e-04	3.89e-03	3.62e-03	3.71e-03	4.08e-03	2.70e-03
59	1.24e-03	3.58e-03	3.64e-03	3.95e-03	4.08e-03	4.20e-03
60	2.55e-06	3.79e-03	3.96e-03	4.34e-03	4.81e-03	3.71e-03
61	7.64e-06	3.99e-03	3.68e-03	4.13e-03	5.74e-03	2.74e-03
62	1.27e-06	3.72e-03	5.34e-03	3.95e-03	4.50e-03	5.11e-03
63	1.00e-03	3.72e-03	5.55e-03	4.02e-03	4.32e-03	6.38e-03
64	2.76e-06	3.39e-03	3.23e-03	3.82e-03	2.77e-03	3.50e-03
65	1.14e-03	3.66e-03	3.33e-03	3.78e-03	3.27e-03	2.03e-03
66	1.08e-03	3.43e-03	3.47e-03	3.66e-03	2.73e-03	2.16e-03
67	8.96e-04	3.74e-03	3.64e-03	3.61e-03	2.64e-03	2.65e-03
68	6.79e-04	3.68e-03	3.39e-03	3.36e-03	1.76e-03	2.03e-03
69	5.95e-04	3.66e-03	3.39e-03	3.59e-03	2.87e-03	2.95e-03
70	4.57e-04	3.71e-03	3.87e-03	4.23e-03	4.54e-03	3.43e-03
71	5.67e-04	3.44e-03	3.84e-03	4.18e-03	4.52e-03	3.59e-03
72	1.20e-03	3.31e-03	3.32e-03	4.01e-03	3.71e-03	2.77e-03
73	5.66e-03	3.67e-03	3.51e-03	4.22e-03	4.65e-03	3.14e-03
74	3.87e-04	3.45e-03	3.61e-03	3.90e-03	4.48e-03	2.57e-03
75	1.24e-04	4.24e-03	3.61e-03	3.89e-03	3.08e-03	3.59e-03
76	5.72e-04	3.71e-03	3.64e-03	3.47e-03	2.53e-03	2.15e-03
77	1.11e-03	3.92e-03	3.62e-03	3.45e-03	3.24e-03	3.86e-03
78	5.03e-04	3.64e-03	3.71e-03	3.88e-03	4.84e-03	2.44e-03

79	4.41e-04	3.73e-03	4.34e-03	4.08e-03	4.78e-03	4.52e-03
80	9.64e-04	4.17e-03	3.66e-03	3.81e-03	2.76e-03	3.12e-03
81	3.63e-05	3.56e-03	3.46e-03	3.85e-03	3.49e-03	2.90e-03
82	3.14e-04	3.37e-03	3.75e-03	3.82e-03	4.12e-03	2.82e-03
83	1.13e-03	3.92e-03	3.72e-03	3.59e-03	3.80e-03	3.24e-03
84	2.16e-03	3.61e-03	3.73e-03	3.67e-03	3.78e-03	2.98e-03
85	1.21e-04	3.64e-03	9.00e-03	3.80e-03	4.34e-03	4.15e-03
86	9.62e-05	3.69e-03	3.61e-03	3.82e-03	4.42e-03	4.16e-03
87	1.11e-03	3.53e-03	3.88e-03	3.94e-03	5.43e-03	4.88e-03
88	5.71e-05	3.77e-03	3.42e-03	3.74e-03	3.43e-03	2.96e-03
89	4.37e-04	3.74e-03	3.52e-03	3.79e-03	3.35e-03	3.58e-03
90	3.40e-06	3.68e-03	3.52e-03	3.77e-03	4.50e-03	4.72e-03
91	1.40e-05	3.63e-03	3.68e-03	3.95e-03	3.79e-03	4.83e-03
92	0.00e+00	3.67e-03	3.49e-03	4.05e-03	4.54e-03	3.14e-03
93	1.40e-05	3.88e-03	3.69e-03	3.61e-03	3.72e-03	4.26e-03
94	0.00e+00	3.82e-03	3.62e-03	3.84e-03	4.62e-03	3.90e-03
95	6.38e-04	4.06e-03	5.35e-03	3.80e-03	3.93e-03	6.19e-03
96	1.27e-06	3.56e-03	3.45e-03	3.76e-03	3.33e-03	1.60e-03
97	5.56e-02	4.08e-03	3.89e-03	3.42e-03	2.95e-03	2.74e-03
98	1.08e-02	3.79e-03	3.62e-03	3.78e-03	3.02e-03	2.27e-03
99	1.63e-02	3.78e-03	3.68e-03	4.46e-03	4.96e-03	3.86e-03
100	2.96e-02	5.05e-03	3.92e-03	3.98e-03	3.64e-03	2.65e-03
101	8.93e-02	6.51e-03	4.54e-03	3.84e-03	3.40e-03	3.63e-03
102	1.52e-02	3.77e-03	3.57e-03	3.55e-03	2.93e-03	3.51e-03
103	1.50e-02	3.62e-03	4.14e-03	3.94e-03	4.36e-03	3.50e-03
104	4.36e-02	3.96e-03	3.56e-03	3.58e-03	2.66e-03	3.88e-03
105	4.46e-02	5.41e-03	3.81e-03	3.87e-03	3.70e-03	4.42e-03
106	8.36e-04	3.79e-03	3.76e-03	3.57e-03	4.18e-03	4.87e-03
107	6.82e-03	3.51e-03	3.68e-03	4.03e-03	5.01e-03	5.45e-03
108	2.74e-02	6.56e-03	3.82e-03	3.57e-03	3.13e-03	3.00e-03
109	1.78e-02	4.89e-03	3.73e-03	3.97e-03	3.79e-03	6.14e-03
110	4.95e-02	4.46e-03	3.98e-03	3.99e-03	4.13e-03	5.03e-03
111	5.49e-02	4.70e-03	4.87e-03	3.80e-03	3.09e-03	6.50e-03
112	1.15e-02	5.18e-03	3.66e-03	4.10e-03	4.33e-03	2.21e-03
113	5.68e-04	3.82e-03	3.50e-03	4.49e-03	5.58e-03	3.50e-03
114	3.96e-02	4.33e-03	3.93e-03	4.27e-03	4.84e-03	2.57e-03
115	4.42e-02	6.28e-03	4.27e-03	4.35e-03	5.62e-03	3.45e-03
116	6.59e-02	5.05e-03	4.47e-03	3.61e-03	3.65e-03	3.10e-03
117	2.07e-02	3.75e-03	3.59e-03	3.68e-03	4.01e-03	4.14e-03
118	6.57e-03	3.78e-03	3.58e-03	3.87e-03	4.03e-03	5.09e-03
119	1.63e-02	3.68e-03	3.82e-03	3.84e-03	3.84e-03	5.61e-03
120	8.70e-04	4.71e-03	3.70e-03	4.08e-03	4.29e-03	4.80e-03
121	1.57e-02	3.62e-03	3.60e-03	4.28e-03	4.32e-03	3.90e-03
122	5.11e-04	3.81e-03	3.83e-03	3.96e-03	5.68e-03	3.69e-03

123	6.37e-06	3.58e-03	3.72e-03	3.71e-03	4.49e-03	3.98e-03
124	3.82e-06	3.77e-03	5.47e-03	3.86e-03	3.53e-03	6.11e-03
125	2.12e-07	3.77e-03	4.04e-03	3.68e-03	4.69e-03	4.75e-03
126	0.00e+00	3.75e-03	5.27e-03	3.86e-03	4.10e-03	5.63e-03
127	0.00e+00	3.53e-03	5.96e-03	3.92e-03	3.60e-03	8.23e-03
128	3.73e-03	3.33e-03	3.30e-03	4.10e-03	3.25e-03	3.50e-03
129	1.94e-04	3.41e-03	3.51e-03	3.81e-03	3.31e-03	1.97e-03
130	1.62e-04	3.49e-03	3.52e-03	4.04e-03	3.40e-03	1.95e-03
131	5.23e-04	3.92e-03	3.70e-03	3.85e-03	3.67e-03	2.45e-03
132	1.32e-04	3.37e-03	3.24e-03	3.41e-03	2.57e-03	2.07e-03
133	5.60e-05	3.57e-03	3.68e-03	3.81e-03	3.23e-03	3.40e-03
134	4.33e-05	3.86e-03	3.44e-03	3.87e-03	3.22e-03	3.67e-03
135	8.79e-05	3.80e-03	4.01e-03	3.55e-03	3.02e-03	3.64e-03
136	2.17e-04	3.56e-03	3.31e-03	3.70e-03	2.42e-03	2.61e-03
137	1.38e-04	3.55e-03	3.63e-03	3.68e-03	2.54e-03	3.00e-03
138	1.41e-04	3.50e-03	3.36e-03	3.97e-03	3.92e-03	3.36e-03
139	1.32e-04	3.83e-03	3.74e-03	3.45e-03	2.49e-03	3.54e-03
140	7.51e-05	3.59e-03	3.54e-03	4.65e-03	5.93e-03	2.69e-03
141	1.11e-04	3.65e-03	3.95e-03	3.91e-03	3.71e-03	4.56e-03
142	5.73e-05	3.56e-03	3.68e-03	4.61e-03	5.84e-03	3.25e-03
143	3.18e-05	3.69e-03	4.22e-03	4.15e-03	4.79e-03	5.05e-03
144	2.25e-04	3.34e-03	3.32e-03	3.78e-03	3.43e-03	1.93e-03
145	1.64e-04	3.41e-03	3.60e-03	4.27e-03	4.39e-03	3.26e-03
146	3.44e-05	3.57e-03	3.38e-03	4.58e-03	4.71e-03	2.94e-03
147	3.57e-05	3.79e-03	4.01e-03	3.91e-03	4.26e-03	3.03e-03
148	4.85e-04	3.35e-03	3.34e-03	3.62e-03	3.40e-03	2.33e-03
149	4.27e-04	3.49e-03	3.99e-03	4.17e-03	5.14e-03	3.73e-03
150	7.51e-05	3.72e-03	3.36e-03	4.06e-03	3.69e-03	4.19e-03
151	5.86e-05	3.58e-03	3.99e-03	3.75e-03	3.00e-03	4.42e-03
152	1.37e-04	3.45e-03	3.26e-03	3.62e-03	2.70e-03	2.36e-03
153	3.68e-04	3.50e-03	3.71e-03	3.52e-03	2.94e-03	3.07e-03
154	7.51e-05	3.87e-03	3.45e-03	3.73e-03	4.06e-03	4.19e-03
155	7.90e-05	4.05e-03	3.69e-03	3.38e-03	2.86e-03	4.54e-03
156	1.53e-03	3.72e-03	3.92e-03	4.38e-03	5.81e-03	1.89e-03
157	1.57e-03	3.71e-03	3.78e-03	3.52e-03	3.72e-03	3.52e-03
158	6.24e-05	3.85e-03	3.63e-03	4.46e-03	5.93e-03	3.18e-03
159	7.64e-06	3.69e-03	5.61e-03	3.97e-03	4.57e-03	5.56e-03
160	1.51e-03	3.54e-03	3.38e-03	3.58e-03	2.58e-03	3.88e-03
161	7.90e-05	3.66e-03	3.59e-03	3.68e-03	3.20e-03	2.57e-03
162	7.51e-05	3.56e-03	3.53e-03	3.49e-03	2.34e-03	2.32e-03
163	4.59e-05	3.68e-03	3.85e-03	3.98e-03	4.11e-03	3.64e-03
164	2.80e-05	3.39e-03	3.43e-03	3.68e-03	3.61e-03	3.09e-03
165	5.60e-05	3.41e-03	3.99e-03	3.97e-03	4.69e-03	3.67e-03
166	3.57e-05	3.89e-03	3.60e-03	3.48e-03	3.23e-03	3.27e-03

167	2.29e-05	3.66e-03	4.07e-03	3.91e-03	5.23e-03	4.17e-03
168	3.95e-05	3.57e-03	3.45e-03	3.78e-03	3.50e-03	3.61e-03
169	1.85e-05	3.42e-03	3.63e-03	3.79e-03	4.65e-03	3.56e-03
170	4.37e-05	3.68e-03	3.52e-03	3.55e-03	3.82e-03	3.90e-03
171	4.97e-05	3.80e-03	3.85e-03	3.59e-03	4.57e-03	5.15e-03
172	1.27e-05	3.79e-03	3.49e-03	3.64e-03	3.67e-03	3.53e-03
173	6.24e-05	3.72e-03	3.72e-03	3.89e-03	5.22e-03	5.11e-03
174	1.27e-05	3.71e-03	3.63e-03	3.86e-03	5.07e-03	3.99e-03
175	6.37e-06	3.66e-03	4.42e-03	4.16e-03	6.19e-03	5.70e-03
176	9.17e-05	3.61e-03	3.46e-03	3.30e-03	2.88e-03	2.34e-03
177	1.63e-04	3.67e-03	3.63e-03	3.94e-03	3.92e-03	3.42e-03
178	9.17e-05	3.69e-03	3.47e-03	3.77e-03	3.19e-03	3.38e-03
179	1.13e-04	3.94e-03	3.96e-03	3.64e-03	3.37e-03	4.08e-03
180	5.09e-05	3.81e-03	3.50e-03	3.93e-03	3.70e-03	4.74e-03
181	1.26e-04	3.55e-03	3.84e-03	3.89e-03	4.97e-03	6.20e-03
182	4.33e-05	3.91e-03	3.47e-03	3.83e-03	3.69e-03	4.66e-03
183	4.08e-05	3.77e-03	4.01e-03	4.12e-03	4.25e-03	6.53e-03
184	5.03e-04	3.58e-03	3.38e-03	4.19e-03	4.30e-03	3.71e-03
185	2.34e-04	3.41e-03	3.59e-03	4.10e-03	4.67e-03	3.73e-03
186	5.86e-05	3.87e-03	3.49e-03	3.56e-03	3.56e-03	4.52e-03
187	6.37e-05	3.69e-03	3.79e-03	3.66e-03	3.87e-03	6.59e-03
188	1.10e-04	3.70e-03	3.46e-03	4.08e-03	3.85e-03	4.90e-03
189	3.13e-04	3.81e-03	3.79e-03	3.62e-03	4.50e-03	5.03e-03
190	1.13e-04	3.72e-03	4.56e-03	3.68e-03	3.70e-03	5.83e-03
191	1.63e-04	3.61e-03	5.54e-03	3.73e-03	3.39e-03	7.40e-03
192	0.00e+00	3.50e-03	3.57e-03	4.12e-03	3.80e-03	1.99e-03
193	0.00e+00	3.87e-03	3.88e-03	4.05e-03	3.83e-03	2.36e-03
194	7.48e-04	3.62e-03	3.47e-03	3.48e-03	3.00e-03	3.31e-03
195	1.85e-05	4.12e-03	3.96e-03	3.75e-03	3.48e-03	4.65e-03
196	3.82e-06	3.59e-03	3.59e-03	4.04e-03	3.22e-03	3.63e-03
197	5.09e-06	3.76e-03	3.72e-03	3.81e-03	3.55e-03	3.98e-03
198	0.00e+00	3.62e-03	3.89e-03	4.35e-03	4.97e-03	3.84e-03
199	0.00e+00	3.82e-03	4.12e-03	4.53e-03	6.08e-03	4.70e-03
200	0.00e+00	3.42e-03	3.68e-03	4.08e-03	4.18e-03	2.40e-03
201	1.27e-05	3.49e-03	3.87e-03	4.30e-03	4.45e-03	2.80e-03
202	7.64e-06	3.45e-03	3.66e-03	3.84e-03	4.00e-03	3.65e-03
203	1.15e-05	3.99e-03	3.87e-03	3.89e-03	3.50e-03	5.02e-03
204	0.00e+00	3.44e-03	3.44e-03	3.82e-03	3.10e-03	3.31e-03
205	0.00e+00	3.97e-03	3.64e-03	3.69e-03	3.65e-03	4.95e-03
206	5.77e-04	3.53e-03	3.92e-03	4.09e-03	4.73e-03	3.05e-03
207	3.17e-04	3.74e-03	4.96e-03	4.29e-03	5.78e-03	4.22e-03
208	2.80e-04	3.67e-03	3.52e-03	3.54e-03	3.07e-03	3.34e-03
209	1.40e-04	3.78e-03	3.88e-03	3.59e-03	3.00e-03	3.13e-03
210	0.00e+00	3.60e-03	3.69e-03	3.82e-03	4.22e-03	3.98e-03

211	1.27e-06	4.16e-03	3.97e-03	3.85e-03	4.66e-03	4.15e-03
212	1.27e-06	3.51e-03	3.56e-03	3.92e-03	4.33e-03	4.28e-03
213	0.00e+00	3.91e-03	3.75e-03	3.40e-03	4.06e-03	4.84e-03
214	0.00e+00	3.66e-03	3.55e-03	3.68e-03	4.51e-03	4.39e-03
215	1.27e-06	3.74e-03	4.07e-03	4.05e-03	5.86e-03	4.69e-03
216	0.00e+00	3.67e-03	3.67e-03	3.53e-03	3.36e-03	2.78e-03
217	0.00e+00	3.90e-03	3.80e-03	3.69e-03	3.15e-03	3.88e-03
218	0.00e+00	3.69e-03	3.69e-03	3.94e-03	4.33e-03	6.28e-03
219	0.00e+00	4.24e-03	3.86e-03	3.99e-03	4.13e-03	6.39e-03
220	0.00e+00	3.57e-03	3.50e-03	4.21e-03	4.48e-03	4.27e-03
221	0.00e+00	4.00e-03	3.68e-03	3.61e-03	3.67e-03	6.81e-03
222	0.00e+00	3.75e-03	3.71e-03	3.81e-03	3.75e-03	5.95e-03
223	0.00e+00	3.70e-03	4.73e-03	3.64e-03	3.12e-03	6.94e-03
224	5.26e-04	3.50e-03	4.05e-03	4.48e-03	4.27e-03	2.77e-03
225	1.19e-03	3.68e-03	3.93e-03	3.77e-03	3.62e-03	5.23e-03
226	4.99e-03	3.66e-03	3.92e-03	4.14e-03	3.71e-03	5.27e-03
227	6.37e-06	3.90e-03	4.50e-03	4.41e-03	6.16e-03	4.61e-03
228	0.00e+00	3.35e-03	4.03e-03	4.35e-03	4.87e-03	2.62e-03
229	0.00e+00	3.66e-03	4.13e-03	3.98e-03	4.17e-03	4.94e-03
230	0.00e+00	3.56e-03	3.48e-03	3.93e-03	3.91e-03	4.76e-03
231	0.00e+00	3.64e-03	4.85e-03	4.45e-03	6.13e-03	3.73e-03
232	0.00e+00	3.49e-03	3.90e-03	3.50e-03	3.40e-03	2.57e-03
233	0.00e+00	3.78e-03	4.04e-03	3.74e-03	5.15e-03	3.65e-03
234	0.00e+00	3.75e-03	3.86e-03	3.71e-03	4.28e-03	4.18e-03
235	0.00e+00	3.84e-03	3.99e-03	3.84e-03	5.37e-03	3.75e-03
236	0.00e+00	3.64e-03	3.94e-03	3.65e-03	3.40e-03	3.66e-03
237	0.00e+00	3.86e-03	4.10e-03	4.02e-03	4.58e-03	6.50e-03
238	0.00e+00	3.72e-03	3.63e-03	3.83e-03	4.01e-03	6.14e-03
239	2.55e-06	3.69e-03	4.07e-03	3.62e-03	3.73e-03	6.38e-03
240	0.00e+00	3.62e-03	4.60e-03	4.09e-03	3.53e-03	5.75e-03
241	0.00e+00	3.77e-03	4.80e-03	4.13e-03	4.23e-03	6.39e-03
242	0.00e+00	3.56e-03	4.66e-03	3.99e-03	4.23e-03	4.99e-03
243	0.00e+00	3.84e-03	4.35e-03	4.10e-03	4.46e-03	5.06e-03
244	0.00e+00	3.64e-03	4.31e-03	3.78e-03	4.96e-03	3.18e-03
245	0.00e+00	3.76e-03	4.38e-03	3.80e-03	5.64e-03	3.80e-03
246	0.00e+00	3.74e-03	4.43e-03	3.84e-03	4.02e-03	5.00e-03
247	0.00e+00	3.58e-03	3.84e-03	3.66e-03	3.94e-03	6.91e-03
248	0.00e+00	3.66e-03	5.74e-03	4.16e-03	3.39e-03	7.42e-03
249	0.00e+00	3.65e-03	5.44e-03	3.84e-03	4.35e-03	6.14e-03
250	0.00e+00	3.60e-03	4.91e-03	3.64e-03	4.84e-03	3.15e-03
251	0.00e+00	3.79e-03	4.56e-03	3.75e-03	3.32e-03	7.97e-03
252	0.00e+00	3.66e-03	5.79e-03	4.18e-03	4.21e-03	7.33e-03
253	0.00e+00	3.71e-03	5.43e-03	3.67e-03	3.49e-03	6.45e-03
254	0.00e+00	3.54e-03	5.86e-03	3.98e-03	3.55e-03	8.23e-03

255	0.00e+00	4.21e-03	7.13e-03	4.10e-03	3.15e-03	9.41e-03
Энтропия Н	4.627848	7.933153	7.978228	7.992566	7.958278	7.898119

Таблица А.1 – Частоты появления символов в файлах 1-6, энтропия файлов 1-6.

файл	7	8	9	10	11	12
символ	Частота появления символа					
0	8.75e-02	1.19e-03	1.36e-01	1.45e-02	5.08e-02	0.00e+00
1	2.44e-05	5.59e-03	9.46e-03	4.52e-03	9.37e-03	0.00e+00
2	0.00e+00	1.26e-02	4.70e-03	4.53e-03	5.48e-03	0.00e+00
3	1.22e-05	1.65e-02	5.00e-03	4.34e-03	4.00e-03	0.00e+00
4	1.31e-05	2.10e-02	8.65e-03	4.40e-03	4.97e-03	0.00e+00
5	1.29e-05	2.51e-02	3.01e-03	3.97e-03	4.55e-03	0.00e+00
6	3.60e-03	2.52e-02	4.68e-03	4.29e-03	3.82e-03	0.00e+00
7	1.55e-05	2.23e-02	3.70e-03	4.13e-03	3.71e-03	0.00e+00
8	2.24e-03	1.85e-02	1.94e-02	4.53e-03	4.55e-03	0.00e+00
9	8.98e-03	1.61e-02	2.06e-03	3.76e-03	4.04e-03	0.00e+00
10	2.02e-05	1.44e-02	2.52e-03	3.93e-03	4.90e-03	0.00e+00
11	3.12e-05	1.31e-02	1.81e-03	3.74e-03	4.24e-03	0.00e+00
12	0.00e+00	1.19e-02	9.29e-03	4.49e-03	3.96e-03	0.00e+00
13	2.32e-03	1.05e-02	2.82e-03	4.03e-03	3.83e-03	0.00e+00
14	1.69e-05	9.43e-03	1.70e-03	3.92e-03	3.72e-03	0.00e+00
15	3.62e-03	8.62e-03	9.47e-03	4.03e-03	3.22e-03	0.00e+00
16	8.32e-03	8.15e-03	7.18e-03	5.97e-03	4.55e-03	0.00e+00
17	1.57e-05	7.93e-03	1.81e-03	4.15e-03	4.07e-03	0.00e+00
18	2.81e-05	7.73e-03	1.05e-03	3.75e-03	4.07e-03	0.00e+00
19	1.97e-05	7.67e-03	1.11e-03	3.60e-03	3.68e-03	0.00e+00
20	4.24e-03	7.47e-03	4.14e-03	3.99e-03	4.08e-03	0.00e+00
21	5.60e-03	7.19e-03	4.07e-03	3.82e-03	4.19e-03	0.00e+00
22	1.55e-05	6.70e-03	1.22e-03	3.78e-03	4.17e-03	0.00e+00
23	2.60e-05	6.56e-03	8.99e-04	3.62e-03	3.79e-03	0.00e+00
24	3.53e-03	6.38e-03	3.59e-03	4.38e-03	3.80e-03	0.00e+00
25	2.25e-05	6.63e-03	1.39e-03	4.02e-03	3.76e-03	0.00e+00
26	7.34e-03	6.91e-03	7.08e-04	3.98e-03	4.98e-03	0.00e+00
27	1.57e-05	6.87e-03	1.24e-03	3.81e-03	3.43e-03	0.00e+00
28	4.83e-03	6.96e-03	2.98e-03	3.72e-03	3.64e-03	0.00e+00
29	3.89e-03	6.86e-03	1.21e-03	3.69e-03	3.61e-03	0.00e+00
30	2.04e-05	6.83e-03	6.55e-04	3.73e-03	3.79e-03	0.00e+00

31	7.60e-03	6.73e-03	6.85e-04	3.76e-03	3.06e-03	0.00e+00
32	4.69e-07	6.48e-03	1.32e-02	4.77e-03	4.30e-03	1.53e-01
33	2.51e-05	6.32e-03	8.67e-04	4.18e-03	5.83e-03	0.00e+00
34	5.41e-03	5.85e-03	8.93e-04	4.23e-03	4.09e-03	0.00e+00
35	1.57e-05	5.28e-03	9.23e-04	3.79e-03	3.87e-03	0.00e+00
36	1.27e-05	4.94e-03	3.64e-03	4.07e-03	3.91e-03	0.00e+00
37	0.00e+00	4.51e-03	8.85e-04	3.46e-03	4.07e-03	0.00e+00
38	9.61e-03	4.46e-03	6.09e-04	3.71e-03	3.68e-03	0.00e+00
39	2.60e-05	4.22e-03	1.52e-03	3.44e-03	3.55e-03	0.00e+00
40	1.43e-05	4.32e-03	1.65e-03	3.99e-03	4.05e-03	0.00e+00
41	1.41e-05	4.46e-03	5.63e-04	3.68e-03	3.94e-03	0.00e+00
42	3.02e-03	4.55e-03	5.20e-04	3.66e-03	4.01e-03	0.00e+00
43	0.00e+00	4.86e-03	1.23e-03	3.52e-03	3.86e-03	0.00e+00
44	4.42e-03	4.96e-03	1.66e-03	3.83e-03	4.15e-03	8.99e-03
45	2.18e-05	5.14e-03	8.36e-04	3.62e-03	4.05e-03	0.00e+00
46	8.09e-03	5.16e-03	8.37e-04	3.55e-03	3.73e-03	8.99e-03
47	2.51e-05	5.36e-03	1.00e-03	3.56e-03	3.45e-03	0.00e+00
48	0.00e+00	5.43e-03	5.88e-03	4.25e-03	3.73e-03	0.00e+00
49	2.60e-05	5.58e-03	2.34e-03	4.07e-03	3.90e-03	0.00e+00
50	1.01e-02	5.80e-03	2.44e-03	3.94e-03	3.73e-03	0.00e+00
51	2.37e-05	5.62e-03	7.76e-03	4.12e-03	3.53e-03	0.00e+00
52	6.37e-03	5.63e-03	3.11e-03	3.93e-03	3.85e-03	0.00e+00
53	0.00e+00	5.28e-03	2.40e-03	3.61e-03	3.68e-03	0.00e+00
54	1.05e-02	4.80e-03	2.28e-03	3.89e-03	3.33e-03	0.00e+00
55	6.37e-03	4.23e-03	2.16e-03	3.61e-03	3.37e-03	0.00e+00
56	2.51e-05	3.83e-03	3.72e-03	3.69e-03	3.78e-03	0.00e+00
57	3.78e-03	3.27e-03	4.52e-03	3.73e-03	3.53e-03	0.00e+00
58	2.74e-05	2.88e-03	1.94e-03	3.73e-03	3.62e-03	0.00e+00
59	0.00e+00	2.70e-03	6.96e-03	3.75e-03	3.49e-03	0.00e+00
60	7.08e-03	2.28e-03	2.58e-03	3.66e-03	3.42e-03	0.00e+00
61	1.18e-02	2.14e-03	2.39e-03	3.78e-03	3.37e-03	0.00e+00
62	3.40e-05	1.99e-03	1.92e-03	3.68e-03	3.20e-03	0.00e+00
63	1.10e-05	1.96e-03	2.00e-03	3.71e-03	3.08e-03	0.00e+00
64	0.00e+00	1.83e-03	3.74e-03	4.90e-03	4.42e-03	0.00e+00
65	1.39e-02	1.75e-03	1.97e-03	4.52e-03	4.72e-03	0.00e+00
66	3.69e-03	1.89e-03	1.21e-03	4.00e-03	4.26e-03	0.00e+00
67	2.72e-05	1.90e-03	2.00e-03	4.06e-03	3.54e-03	0.00e+00
68	1.76e-05	2.03e-03	2.90e-03	4.21e-03	4.09e-03	2.25e-03
69	1.12e-02	2.13e-03	1.46e-02	4.18e-03	3.94e-03	2.25e-03
70	0.00e+00	2.19e-03	4.35e-03	3.87e-03	3.73e-03	0.00e+00
71	2.65e-05	2.35e-03	1.25e-03	3.59e-03	3.37e-03	0.00e+00
72	4.08e-05	2.42e-03	2.52e-03	4.10e-03	3.96e-03	0.00e+00
73	9.26e-03	2.43e-03	1.28e-03	3.70e-03	3.69e-03	0.00e+00
74	5.86e-03	2.52e-03	5.34e-04	3.63e-03	4.03e-03	0.00e+00

75	3.85e-03	2.47e-03	1.83e-02	3.42e-03	3.75e-03	0.00e+00
76	0.00e+00	2.56e-03	1.76e-03	3.80e-03	4.25e-03	2.25e-03
77	1.62e-05	2.50e-03	7.43e-03	3.72e-03	3.50e-03	0.00e+00
78	1.90e-05	2.46e-03	2.04e-03	3.55e-03	3.64e-03	0.00e+00
79	2.93e-05	2.41e-03	8.48e-04	3.56e-03	3.26e-03	0.00e+00
80	5.77e-03	2.42e-03	1.46e-02	4.20e-03	4.09e-03	0.00e+00
81	9.85e-03	2.38e-03	6.79e-03	3.84e-03	3.80e-03	0.00e+00
82	3.54e-05	2.34e-03	3.04e-03	3.69e-03	3.97e-03	0.00e+00
83	0.00e+00	2.45e-03	5.29e-03	3.77e-03	3.62e-03	0.00e+00
84	1.61e-02	2.46e-03	1.81e-03	3.75e-03	4.06e-03	0.00e+00
85	3.87e-05	2.43e-03	6.81e-03	3.99e-03	3.96e-03	2.25e-03
86	3.94e-03	2.51e-03	6.99e-03	3.75e-03	3.95e-03	0.00e+00
87	2.35e-05	2.57e-03	6.37e-03	3.59e-03	3.56e-03	0.00e+00
88	1.76e-05	2.58e-03	1.71e-03	3.78e-03	4.12e-03	0.00e+00
89	1.07e-02	2.60e-03	2.08e-03	3.86e-03	3.82e-03	0.00e+00
90	0.00e+00	2.79e-03	4.86e-04	3.72e-03	4.14e-03	0.00e+00
91	3.56e-05	2.77e-03	2.03e-03	3.61e-03	3.62e-03	0.00e+00
92	1.06e-02	2.93e-03	1.58e-03	3.53e-03	3.69e-03	0.00e+00
93	5.14e-03	3.12e-03	5.61e-03	3.64e-03	3.68e-03	0.00e+00
94	4.88e-05	3.14e-03	4.50e-03	3.68e-03	3.61e-03	0.00e+00
95	1.49e-02	3.23e-03	2.86e-03	3.64e-03	3.19e-03	0.00e+00
96	3.07e-05	3.27e-03	1.18e-03	4.03e-03	3.93e-03	0.00e+00
97	0.00e+00	3.31e-03	6.57e-03	4.03e-03	3.80e-03	6.52e-02
98	2.56e-05	3.37e-03	1.93e-03	3.93e-03	3.83e-03	6.74e-03
99	4.95e-03	3.43e-03	1.90e-03	3.99e-03	3.59e-03	3.60e-02
100	2.53e-05	3.42e-03	2.80e-03	4.19e-03	3.76e-03	4.04e-02
101	4.58e-03	3.41e-03	5.52e-03	3.91e-03	3.89e-03	8.31e-02
102	9.45e-03	3.48e-03	3.70e-03	4.17e-03	3.57e-03	6.74e-03
103	3.17e-05	3.42e-03	1.46e-03	3.76e-03	3.44e-03	6.74e-03
104	9.05e-03	3.44e-03	5.82e-03	4.04e-03	3.99e-03	2.25e-03
105	0.00e+00	3.48e-03	2.71e-03	3.84e-03	3.73e-03	9.44e-02
106	4.55e-05	3.24e-03	5.58e-03	3.77e-03	3.87e-03	0.00e+00
107	4.72e-03	3.25e-03	8.52e-04	3.70e-03	3.71e-03	0.00e+00
108	2.81e-05	3.04e-03	3.97e-03	3.92e-03	3.57e-03	4.72e-02
109	9.59e-03	2.94e-03	1.77e-03	4.03e-03	3.49e-03	3.82e-02
110	7.46e-05	2.88e-03	2.02e-03	3.80e-03	3.51e-03	5.39e-02
111	8.99e-03	2.84e-03	2.72e-03	3.87e-03	3.25e-03	6.52e-02
112	4.64e-03	2.73e-03	2.81e-03	3.73e-03	3.71e-03	2.47e-02
113	3.45e-05	2.69e-03	7.79e-04	3.65e-03	3.53e-03	1.12e-02
114	0.00e+00	2.62e-03	3.07e-03	3.72e-03	3.54e-03	4.94e-02
115	1.28e-02	2.55e-03	2.54e-03	3.86e-03	3.50e-03	4.04e-02
116	1.62e-05	2.70e-03	1.06e-02	3.96e-03	3.61e-03	7.19e-02
117	1.45e-05	2.72e-03	1.42e-02	3.90e-03	3.59e-03	6.29e-02
118	4.72e-03	2.70e-03	2.36e-03	3.73e-03	3.51e-03	6.74e-03

119	8.47e-03	2.80e-03	1.99e-03	3.96e-03	3.97e-03	0.00e+00
120	5.07e-03	2.78e-03	1.68e-03	3.65e-03	3.36e-03	6.74e-03
121	4.22e-03	2.93e-03	1.06e-03	3.75e-03	3.40e-03	0.00e+00
122	9.85e-06	3.01e-03	5.44e-04	3.66e-03	3.46e-03	0.00e+00
123	4.84e-03	3.13e-03	7.56e-04	3.86e-03	3.23e-03	0.00e+00
124	4.60e-03	3.39e-03	3.66e-03	3.66e-03	3.25e-03	0.00e+00
125	4.47e-03	3.37e-03	4.51e-03	3.86e-03	3.17e-03	0.00e+00
126	3.05e-05	3.39e-03	2.36e-03	3.64e-03	3.12e-03	0.00e+00
127	1.57e-05	3.62e-03	1.10e-03	3.76e-03	3.02e-03	0.00e+00
128	8.45e-03	3.52e-03	4.91e-03	5.00e-03	4.62e-03	0.00e+00
129	2.65e-05	3.66e-03	1.82e-03	4.06e-03	3.82e-03	0.00e+00
130	7.64e-03	3.60e-03	5.89e-04	3.88e-03	4.09e-03	0.00e+00
131	4.92e-03	3.64e-03	1.00e-02	4.27e-03	3.40e-03	0.00e+00
132	8.81e-03	3.59e-03	4.69e-03	3.94e-03	3.96e-03	0.00e+00
133	8.17e-03	3.50e-03	1.17e-02	3.96e-03	4.02e-03	0.00e+00
134	2.42e-05	3.27e-03	2.34e-03	3.99e-03	3.64e-03	0.00e+00
135	8.48e-03	3.24e-03	6.94e-04	3.80e-03	3.32e-03	0.00e+00
136	4.10e-03	3.05e-03	2.03e-03	4.39e-03	4.02e-03	0.00e+00
137	4.31e-03	2.90e-03	1.11e-02	4.04e-03	3.70e-03	0.00e+00
138	8.39e-03	2.76e-03	8.83e-04	3.89e-03	3.94e-03	0.00e+00
139	8.24e-03	2.54e-03	4.68e-02	4.73e-03	3.76e-03	0.00e+00
140	7.93e-03	2.48e-03	2.59e-03	4.08e-03	3.69e-03	0.00e+00
141	1.43e-05	2.38e-03	1.55e-02	4.03e-03	3.60e-03	0.00e+00
142	3.58e-03	2.32e-03	1.75e-03	3.57e-03	4.23e-03	0.00e+00
143	1.85e-05	2.23e-03	7.57e-04	3.67e-03	3.21e-03	0.00e+00
144	7.84e-03	2.39e-03	3.24e-02	4.25e-03	3.94e-03	0.00e+00
145	8.11e-03	2.35e-03	9.08e-04	3.92e-03	3.68e-03	0.00e+00
146	1.19e-02	2.45e-03	4.84e-04	3.78e-03	3.84e-03	0.00e+00
147	8.31e-03	2.48e-03	5.50e-04	3.56e-03	4.04e-03	0.00e+00
148	3.91e-03	2.53e-03	1.40e-03	3.66e-03	4.45e-03	0.00e+00
149	1.71e-05	2.63e-03	1.02e-03	3.54e-03	4.75e-03	0.00e+00
150	7.86e-03	2.69e-03	5.53e-04	3.61e-03	3.90e-03	0.00e+00
151	3.77e-03	2.86e-03	4.35e-04	3.54e-03	3.51e-03	0.00e+00
152	7.84e-03	2.91e-03	1.27e-03	3.90e-03	3.65e-03	0.00e+00
153	7.92e-03	2.97e-03	9.56e-04	3.87e-03	3.48e-03	0.00e+00
154	1.13e-02	2.94e-03	4.38e-04	3.73e-03	4.07e-03	0.00e+00
155	7.52e-03	2.89e-03	4.59e-04	3.73e-03	3.53e-03	0.00e+00
156	3.36e-03	3.07e-03	1.26e-03	3.65e-03	3.66e-03	0.00e+00
157	6.80e-06	3.18e-03	8.05e-04	3.61e-03	3.56e-03	0.00e+00
158	6.92e-03	3.22e-03	1.07e-03	3.64e-03	3.81e-03	0.00e+00
159	1.00e-02	3.38e-03	4.47e-04	3.67e-03	3.44e-03	0.00e+00
160	9.58e-03	3.47e-03	2.35e-03	3.86e-03	4.14e-03	0.00e+00
161	9.94e-03	3.55e-03	1.06e-03	3.83e-03	3.87e-03	0.00e+00
162	3.11e-03	3.74e-03	6.44e-04	3.83e-03	3.99e-03	0.00e+00

163	6.41e-03	4.13e-03	4.02e-04	3.70e-03	3.55e-03	0.00e+00
164	6.04e-03	3.99e-03	1.04e-03	3.73e-03	3.80e-03	0.00e+00
165	8.42e-03	4.02e-03	2.23e-03	3.60e-03	3.92e-03	0.00e+00
166	5.66e-03	4.09e-03	4.16e-04	3.80e-03	4.45e-03	0.00e+00
167	6.57e-06	4.05e-03	5.71e-04	3.60e-03	3.45e-03	0.00e+00
168	8.08e-03	4.18e-03	1.09e-03	3.79e-03	3.92e-03	0.00e+00
169	5.15e-03	4.22e-03	3.54e-04	3.62e-03	3.71e-03	0.00e+00
170	5.26e-03	4.32e-03	4.40e-04	3.86e-03	4.02e-03	0.00e+00
171	7.22e-03	4.46e-03	1.01e-03	3.62e-03	3.66e-03	0.00e+00
172	6.84e-03	4.66e-03	1.23e-03	3.73e-03	3.70e-03	0.00e+00
173	2.12e-03	4.71e-03	3.79e-04	3.70e-03	3.88e-03	0.00e+00
174	6.69e-03	4.69e-03	3.67e-04	3.57e-03	3.65e-03	0.00e+00
175	6.84e-03	4.76e-03	5.91e-04	3.67e-03	3.39e-03	0.00e+00
176	4.19e-03	4.82e-03	1.29e-03	3.83e-03	3.80e-03	0.00e+00
177	4.31e-03	4.68e-03	4.29e-04	3.78e-03	3.77e-03	0.00e+00
178	4.92e-06	4.51e-03	4.39e-04	3.71e-03	3.97e-03	0.00e+00
179	5.75e-03	4.61e-03	7.17e-04	3.79e-03	3.47e-03	0.00e+00
180	8.75e-03	4.43e-03	1.09e-03	3.69e-03	3.77e-03	0.00e+00
181	5.41e-03	4.37e-03	1.12e-03	3.69e-03	3.82e-03	0.00e+00
182	5.07e-03	4.11e-03	9.51e-04	3.89e-03	3.63e-03	0.00e+00
183	1.59e-03	4.10e-03	6.75e-04	3.79e-03	3.34e-03	0.00e+00
184	3.31e-03	3.97e-03	2.67e-03	3.64e-03	3.47e-03	0.00e+00
185	4.60e-03	3.81e-03	1.20e-03	3.67e-03	3.69e-03	0.00e+00
186	4.46e-03	3.66e-03	4.36e-04	3.70e-03	3.69e-03	0.00e+00
187	1.49e-03	3.65e-03	7.25e-04	3.81e-03	3.45e-03	0.00e+00
188	2.92e-03	3.77e-03	1.29e-03	3.58e-03	3.43e-03	0.00e+00
189	6.83e-03	3.88e-03	7.72e-04	3.80e-03	3.44e-03	0.00e+00
190	2.71e-03	4.06e-03	1.55e-03	3.72e-03	3.30e-03	0.00e+00
191	5.65e-03	4.21e-03	1.02e-03	3.77e-03	3.09e-03	0.00e+00
192	7.50e-06	4.15e-03	1.03e-02	4.09e-03	3.97e-03	0.00e+00
193	4.03e-03	4.35e-03	2.55e-03	3.85e-03	3.60e-03	0.00e+00
194	8.85e-03	4.39e-03	4.96e-03	3.70e-03	4.30e-03	0.00e+00
195	2.45e-03	4.48e-03	3.32e-03	4.17e-03	3.32e-03	0.00e+00
196	1.22e-03	4.30e-03	1.66e-03	3.82e-03	3.61e-03	0.00e+00
197	4.78e-03	3.76e-03	5.10e-04	3.73e-03	3.68e-03	0.00e+00
198	1.22e-03	3.55e-03	2.84e-03	4.02e-03	3.55e-03	0.00e+00
199	7.74e-03	3.44e-03	4.92e-03	3.78e-03	3.42e-03	0.00e+00
200	1.10e-05	3.52e-03	2.23e-03	3.86e-03	3.56e-03	0.00e+00
201	4.52e-03	3.62e-03	3.16e-03	3.68e-03	3.55e-03	0.00e+00
202	3.28e-03	3.71e-03	8.87e-04	3.52e-03	3.67e-03	0.00e+00
203	3.11e-03	3.83e-03	1.50e-03	3.74e-03	3.63e-03	0.00e+00
204	9.33e-03	3.89e-03	1.57e-03	4.02e-03	3.38e-03	0.00e+00
205	1.97e-03	3.86e-03	5.55e-04	3.77e-03	3.41e-03	0.00e+00
206	1.03e-03	3.70e-03	2.09e-03	3.82e-03	3.56e-03	0.00e+00

207	3.13e-03	3.51e-03	1.34e-03	3.80e-03	3.29e-03	0.00e+00
208	1.92e-03	3.27e-03	2.01e-03	3.78e-03	3.84e-03	0.00e+00
209	5.21e-03	3.08e-03	9.99e-04	3.80e-03	3.67e-03	0.00e+00
210	5.50e-03	2.59e-03	1.03e-03	3.70e-03	3.64e-03	0.00e+00
211	2.77e-03	2.11e-03	8.70e-04	3.78e-03	3.48e-03	0.00e+00
212	4.65e-03	1.69e-03	1.53e-03	3.66e-03	3.96e-03	0.00e+00
213	8.40e-04	1.29e-03	5.39e-04	3.63e-03	3.69e-03	0.00e+00
214	3.51e-03	9.86e-04	9.83e-04	3.76e-03	3.65e-03	0.00e+00
215	5.25e-03	6.89e-04	1.01e-03	3.77e-03	3.46e-03	0.00e+00
216	3.59e-03	5.76e-04	3.23e-03	3.80e-03	3.42e-03	0.00e+00
217	3.92e-03	4.74e-04	1.75e-03	3.82e-03	3.46e-03	0.00e+00
218	1.85e-03	4.38e-04	7.05e-04	3.81e-03	3.45e-03	0.00e+00
219	3.46e-03	3.99e-04	2.06e-03	4.08e-03	3.42e-03	0.00e+00
220	4.64e-03	3.68e-04	2.08e-03	3.68e-03	3.28e-03	0.00e+00
221	1.55e-03	3.48e-04	1.40e-03	3.79e-03	3.42e-03	0.00e+00
222	3.13e-03	3.94e-04	9.29e-04	3.81e-03	3.24e-03	0.00e+00
223	3.72e-03	4.28e-04	8.78e-04	3.89e-03	3.27e-03	0.00e+00
224	4.42e-03	5.52e-04	2.71e-03	3.61e-03	3.54e-03	0.00e+00
225	1.31e-03	6.77e-04	7.39e-04	3.65e-03	3.41e-03	0.00e+00
226	2.60e-03	7.42e-04	5.62e-04	3.61e-03	3.38e-03	0.00e+00
227	5.87e-03	8.09e-04	6.51e-04	3.74e-03	3.42e-03	0.00e+00
228	1.22e-03	9.52e-04	2.12e-03	3.63e-03	3.30e-03	0.00e+00
229	1.79e-03	9.57e-04	7.16e-04	3.64e-03	3.51e-03	0.00e+00
230	2.37e-03	9.19e-04	7.02e-04	3.80e-03	3.27e-03	0.00e+00
231	5.49e-03	8.86e-04	6.98e-04	3.83e-03	3.44e-03	0.00e+00
232	1.44e-03	8.40e-04	1.70e-02	4.28e-03	3.49e-03	0.00e+00
233	1.48e-03	9.62e-04	5.52e-03	3.77e-03	3.42e-03	0.00e+00
234	4.88e-03	1.03e-03	8.70e-04	3.66e-03	3.43e-03	0.00e+00
235	2.75e-04	1.05e-03	4.12e-03	3.93e-03	3.44e-03	0.00e+00
236	1.38e-03	1.18e-03	7.74e-03	3.86e-03	3.32e-03	0.00e+00
237	2.32e-03	1.17e-03	6.72e-04	3.83e-03	3.40e-03	0.00e+00
238	3.84e-03	1.23e-03	8.61e-04	3.90e-03	3.24e-03	0.00e+00
239	1.92e-04	1.34e-03	6.99e-04	3.93e-03	3.27e-03	0.00e+00
240	9.10e-04	1.04e-03	4.85e-03	3.89e-03	3.22e-03	0.00e+00
241	3.82e-03	8.54e-04	2.49e-03	3.67e-03	3.21e-03	0.00e+00
242	1.09e-03	8.45e-04	7.77e-04	3.59e-03	3.23e-03	0.00e+00
243	1.67e-04	8.57e-04	1.82e-03	3.81e-03	3.26e-03	0.00e+00
244	2.26e-03	8.49e-04	3.68e-03	3.64e-03	3.27e-03	0.00e+00
245	6.15e-04	6.58e-04	7.95e-04	3.78e-03	3.35e-03	0.00e+00
246	1.03e-03	5.33e-04	3.21e-03	3.82e-03	3.23e-03	0.00e+00
247	1.69e-03	4.72e-04	1.89e-03	3.95e-03	3.18e-03	0.00e+00
248	2.41e-04	5.14e-04	6.38e-03	3.83e-03	3.17e-03	0.00e+00
249	5.26e-04	5.38e-04	1.36e-03	3.74e-03	3.17e-03	0.00e+00
250	1.07e-03	4.68e-04	1.31e-03	3.78e-03	3.11e-03	0.00e+00

251	5.00e-04	3.23e-04	1.76e-03	3.89e-03	3.10e-03	0.00e+00
252	2.01e-04	2.54e-04	7.07e-03	3.91e-03	3.03e-03	0.00e+00
253	1.05e-03	1.28e-04	2.22e-03	3.73e-03	3.19e-03	0.00e+00
254	5.83e-04	5.16e-05	2.61e-03	3.88e-03	2.94e-03	0.00e+00
255	5.44e-03	4.67e-06	6.48e-02	6.90e-03	3.78e-03	0.00e+00
Энтро- пия H	6.968415	7.568010	6.626704	7.982869	7.866040	4.116830

Таблица А.2 – Частоты появления символов в файлах 7-12, энтропия файлов 7-12.