ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

Контрольное домашнее задание по предмету

Алгоритмы и структуры данных

Алгоритмы сжатия без потерь

Исполнитель
Студент БПИ185-2
Федорова Алена Валерьевна

Москва 2020

Оглавление

1. Постановка задачи

1. Разработать на языке C++ программу, реализующую алгоритмы сжатия данных без потерь, получение архивированного файла из исходного и разархивированного файла из архивированного (упаковка файла и распаковка архива);
2. Провести вычислительный эксперимент для исследования эффективности реализованных алгоритмов сжатия без потерь (упаковка и распаковка) для файлов разного типа;
3. Подготовить отчет по итогам работы.

Для проведения эксперимента подобрать файлы примерно одного размера. В наборе должны присутствовать файлы:

- текстовый .txt
- документ Word .docx
- презентация .pptx
- документ .pdf
- исполняемый файл .exe или библиотечный .dll
- цветное изображение .jpg
- изображение черно-белое или в градациях серого .jpg
- цветное изображение .bmp
- изображение черно-белое или в градациях серого .bmp
- файлы других форматов.

В рамках эксперимента:

1. Вычислить энтропию исходных файлов;
2. Вычислить коэффициент сжатия файлов;
3. Измерить для каждого файла и каждого алгоритма время упаковки и распаковки.

В рамках работы **были реализованы** алгоритмы**:**

1. Алгоритм Шеннона-Фано;
2. Алгоритм Лемпеля-Зива LZ77.

Был проведен вычислительный эксперимент для алгоритмов:

1. Алгоритм Шеннона-Фано;
2. Алгоритм Лемпеля-Зива LZ77 с характеристиками:

a. Размер скользящего окна 5Кб, размер словаря 4Кб;
b. Размер скользящего окна 10Кб, размер словаря 8Кб;
c. Размер скользящего окна 20Кб, размер словаря 16Кб.

И для десяти файлов размером примерно 800кб каждый и форматов: .txt; .docx; .pptx; .pdf; .exe; .jpg (цветной и чб); .bmp (цветной и чб); .html;

Не был реализован алгоритм LZW.

2. Описание алгоритмов

2.1 Алгоритм Шеннона-Фано

Получение кодов символов выполняется следующим образом:

1. Вся последовательность символов упорядочивается по убыванию значения их вероятности;

2. Последовательность разбивается на две по возможности равновероятные группы(медианой);

3. Первой группе присваивается символ “0”, второй –“1”;

4. Каждая получившаяся группа разбивается на по возможности равновероятные группы первой группе присваивается символ “0”, второй –“1” и так далее, пока деление на группы возможно.

Далее в исходном тексте каждый символ заменяется на его код. Полученная последовательность будет являться закодированным текстом.

Декодирование использует кодировочную таблицу, где каждому символу сопоставлен его код. Из-за того, что ни один код не является префиксом другого, декодирование происходит однозначным образом.

2.1 Алгоритм LZ77

Результатом работы алгоритма LZ77 является набор кодовых троек значений, который получается следующим образом:

Вдоль входной последовательности скользит окно, состоящее из двух частей фиксированной длины:

1. Буфер предыстории – уже обработанный фрагмент исходных данных;
2. Буфер предпросмотра – еще не обработанный фрагмент исходных данных, следующий за буфером предыстории.

Метод LZ77 ищет в буфере предыстории (поиска) фрагмент текста максимальной длины из буфера предпросмотра.

На выход выдается код -тройка значений $[offset, length, char]$, где:

- $offset$ – позиция фрагмента в буфере предыстории (в словаре);
- $length$ – длина фрагмента;
- $char$ – первый символ буфера предпросмотра после найденного фрагмента.

При этом если фрагмент не найден, выдается «нулевая фраза» $[0, 0, char]$

Иными словами, совпадение кодируется парой [смещение, длина совпадения], к которой добавляется следующий символ.

Набор кодовых троек и будет являться закодированным текстом.

Декодирование происходит следующим образом:

Обрабатывая очередную кодовую тройку, необходимо пройтись по уже декодированной строке назад на $offset$ символов и вывести $length$ символов раскодированной строки, а затем дописать в конец строки значение $char$.

3. Описание реализаций алгоритмов и использованных структур данных

Описание реализации алгоритмов (форматы сжатых файлов для всех алгоритмов, особенности упаковки / распаковки LZ77, методы работы с битами/ байтами при сжатии / распаковке и др.)

При реализации алгоритмов использовались следующие форматы для файлов:

1. исходный файл <name>.*
2. метод упаковки, использующий алгоритм Шеннона-Фано, архивированный файл <name>.shan
3. метод распаковки, использующий алгоритм Шеннона-Фано, разархивированный файл <name>.unshan
4. метод упаковки, использующий алгоритм LZ77:
   - размер скользящего окна 5 Кб, размер словаря 4 Кб архивированный файл <name>.lz775,
   - размер скользящего окна 10 Кб, размер словаря 8 Кб архивированный файл <name>.lz7710,
   - размер скользящего окна 20 Кб, размер словаря 16 Кб архивированный файл <name>.lz7720,
5. метод распаковки, использующий алгоритм LZ77:
   - размер скользящего окна 5 Кб, размер словаря 4 Кб архивированный файл <name>.unlz775,
   - размер скользящего окна 10 Кб, размер словаря 8 Кб архивированный файл <name>.unlz7710,
   - размер скользящего окна 20 Кб, размер словаря 16 Кб архивированный файл <name>.unlz7720;

3.1 Алгоритм Шеннона-Фано

Архивированный файл <name>.shan состоит из следующих частей:

1. 256 пар значений $\{byte, f\}$, где f – сколько раз встретился $byte$ в исходном файле. $byte$ кодируется 1 байтом, f – 4 байта;
2. Количество байтов в исходном тексте, 4 байта;
3. Закодированный текст.

Архивированный файл создается следующим образом:

1. За один проход по исходному файлу подсчитываются частоты, с которыми байты с 0 по 255 встречаются в файле, байты с ненулевой

частотой сохраняются в массиве $frequencies$ как пара значений $\{frequency, byte\}$.

2. $frequencies$ сортируется по убыванию частот.
3. Вычисляется медиана $frequencies$ относительно частот, и в массиве $codes$, который хранит последовательности кодов (каждый в виде массива чисел из 0 и 1) всем байтам левее медианы дописывает 0 к их коду, правее – 1.
4. Рекурсивно пункт 3 повторяется сначала для всех байтов, которым на предыдущем шаге был присвоен 0, затем – 1. В результате в массиве $codes$ по индексу i лежит код байта $frequencies[i].second$.
5. В массивах $codes\_byte[256]$ и $frequency\_byte[256]$ сохраняются соответственно ссылки на коды каждого из 256 байтов и частота их встречаемости в тексте.
6. В архивированный файл записываются пары значений $\{byte, frequency\_byte[byte]\}$ и размер исходного файла в байтах, для дальнейшего декодирования.
7. Второй раз пройдясь по исходному файлу, каждый байт заменяется на его код, в последовательности кодов каждые 8 символов преобразуются в 1 байт, этот байт записывается в архивированный файл. Если битов в полученной последовательности не хватает для целого количества байт, последовательность дополняется нулями.

Для преобразования 8 символов кода в 1 байт используется *union storage*, где $bitset < 8 > byte$ заполняется кодами символов, в файл записывается значение ch.

```
union storage
{
    unsigned char ch;
    bitset<8> byte;

}
```

Разархивация архивированного файла происходит следующим образом:

1. Из архивированного файла считываются частоты всех символов и размер исходного файла, символы с ненулевыми частотами сохраняются в массив $frequencies$.
2. Выполняются пункты 2-4 для построения кодов всех символов.
3. Формируется словарь с реализацией хэш-таблицей $std::unordered\_map$ с парами значений $\{code, byte\}$, где $byte$ – байт (тип $unsigned\ char$), $code$ – строковое представление кода.

4. Часть архивированного файла с закодированным текстом, используя storage, преобразуется в последовательность битов, в которой, используя словарь, каждый встреченный код заменяется на соответствующий байт, этот байт записывается в файл с разархивированным текстом.

3.2 Алгоритм LZ77

Архивированный файл состоит из последовательностей кодовых троек *{offset, length, char},* где на *offset* и *length* выделено по 2 байта и на *char* – 1 байт.

При реализации алгоритма для сохранения буфера предыстории (с размером *dictionaty_size*) и буфера предпросмотра (с размером *window_size*) используется массив *text* с реализацией *std::vector<unsigned char>*, размер – сумма размеров буферов.

Он будет играть роль **кольцевого буфера**. Для этого поддерживается *start_of_window* – указатель на начало окна в *text,* index – указатель на место последнего добавления символа в буфер *text, current_size_of_dictionary* – текущий размер словаря (буфера предыстори) .

start_of_window с нахождением каждой новой кодовой тройки увеличивается на найденное значение *length* + 1, но при необходимости пройтись по буферу предпросмотра в *text start_of_window* берется по модулю размера буфера.

index увеличивается при каждом прочтении нового байта из исходного файла и добавляет новый байт в буфер на позицию *index* по модулю размера буфера.

Архивированный файл формируется следующим образом:

1. Буфер *text* заполняется первыми *window*_size байтами из исходного файла.

2. Повторяются действия до тех пор, пока весь текст не будет закодирован:

    1. До тех пор, пока весь текст не будет закодирован, осуществляется поиск наидлиннейшей подстроки в буфере предыстории, равной префиксу в буфере предпросмотра. Для этого синхронно просматриваются символы в *text*:
        a. Буфере предыстории в *text*:
        [(start_of_window – current_size_of_dictionary) ***mod buffer_size****;*
        (start_of_window – 1) ***mod buffer_size****]*;

        b. Буфер предпросмотра в *text*:
        [start_of_window ***mod buffer_size****; (start_of_window + window_size)* ***mod buffer_size****].*

2. Найденная подстрока кодируется как *{offset, length, char}* где *offset* – начало подстроки в буфере предыстории при нумерации справа налево, *length* – длина подстроки, *char* – следующий за найденной подстрокой символ. Код сохраняется в структуре *Code* и добавляется в массив кодов *codes* с реализацией *std::vector<Code>*. В силу того, что максимальный размер окна – 20КБ, а размер словаря – 16 Кб, то значения offset и length могут достигать $20 \cdot 2^{10}$ и $16 \cdot 2^{10}$, то есть для их кодирование необходим тип *short*.

```
struct Code
{
    short offset, length;
    unsigned char next_char;
};
```

3. В буфер text добавляется length + 1 новых символов, считанных из исходного файла. Они добавляются на позиции *[index* ***mod buffer_size****; index + length + 1* ***mod buffer_size****]*.

3. Все коды записываются в архивированный файл побайтово. При записи short в файл использовался метод `reinterpret_cast<const char *>`, перепредставляющий *short* как последовательность *char* *, результат которого записывался в файл с помощью метода *write* из *ofstream*.

Разархивация архивированного файла происходит следующим образом:

Промежуточный результат сохраняется в массиве *text*. Из архивированного файла последовательно считываются кодовые тройки, аналогично перепредставляя последовательность из двух байтов как *short* с помощью `reinterpret_cast<const char *>` и, воспроизводится алгоритм из раздела «описание алгоритмов», сохраняя полученные символы в *text*. Результат работы побайтово записывается в разархивированный файл.

4. План эксперимента

В рамках проведения вычислительного эксперимента были проведены следующие действия:

1. Подготовлено 10 файлов соответствующих форматов и размеров:

имя файла	Размер (Кб)
1.txt	814
2.docx	783
3.pptx	806
5.exe	797
4.pdf	823
6.jpeg (цветное)	804
7.jpeg (черно-белое)	795
8.bmp (цветное)	819
9.bmp(черно-белое)	825
10.html	818

2. Вычислена частотная встречаемость символов отдельно для каждого из файлов;
3. Вычислена энтропия каждого файла по формуле:

$$H = -\sum_{i=1}^{m} w_i log_2 w_i;$$

4. Для каждого файла и каждого из алгоритмов или их модификаций из списка:
   a. Алгоритм Шеннона-Фано;
   b. Алгоритм LZ77 с размером окна 5 Кб и словарем 4 Кб;
   c. Алгоритм LZ77 с размером окна 10 Кб и словарем 8 Кб;
   d. Алгоритм LZ77 с размером окна 20 Кб и словарем 16 Кб.

   Были выполнены действия:
   1. Вычислен коэффициент сжатия;
   2. 10 раз измерено время упаковки и найдено среднее время упаковки в миллисекундах;
   3. 10 раз измерено время распаковки и найдено среднее время распаковки в миллисекундах.

5. Описание программной реализации эксперимента и использованных дополнительных инструментов

Для проведения эксперимента был подготовлен код на языке C++, а именно:

1. Код классов ShannonFano и LZ77 с реализациями согласно пункту «Описание реализаций алгоритмов и использованных структур данных». Конструктор класса LZ77 принимает размер окна и размер словаря.
2. Файл main.cpp, в котором содержится реализация проведения всех расчетов.

Для вычисления энтропии написан метод `calc_entropy(ofstream &fout, string file)`, который подсчитывает энтропию файла *input* и записывает в файловый поток *fout* частоты всех символов файла *input*.

Для подсчета коэффициентов сжатия и времени упаковки/распаковки были подготовлены методы:

```
time_measurement_decode_ShannonFano(ShannonFano &sh, string encode, string decode)
time_measurement_encode_ShannonFano(ShannonFano &sh, string input, string encode)
time_measurement_encode_LZ77(LZ77 &lz, string input, string encode)
time_measurement_decode_LZ77(LZ77 &lz, string encode, string decode)
```

Каждый из методов запускал метод *encode* или *decode* соответствующей реализации алгоритма 13 раз, замерял время работы каждого запуска, начиная с третьего (чтобы отсечь запуски, на которых еще не произошло оптимизации компилятором), и возвращал среднее время работы алгоритма на файле. Методы *encode* к тому же возвращали коэффициент сжатия файла.

Замеры времени производились с использованием библиотеки *std::chrono*. Замеры производились в миллисекундах.

Для измерения размеров файлов был подготовлен метод int `getSize(string input)`, возвращавший размер файла в Кб.

В методе main производился запуск каждого из вышеописанных методов на каждом из подготовленных файлов, запись частот шла в файл frequencies.txt, результаты замеров времени, размеров файлов и коэффициентов сжатия – в файл result.txt.

6. Диаграмма классов

ShannonFano
public: ShannonFano(int window_size, int dictionary_size) double encode(string input, string output) void decode(string input, string output)

private:
vector<int> prefix_sum;
vector<vector<int> > codes;
vector<pair<int, int>> frequency;
 	
int get_median(int left, int right);
void build();
void build(int l, int r);
string get (int i);
void fill_prefix_sum();
 	
int count_frequencies_encode(string &input);
void count_frequencies_decode(ifstream &fin);

LZ77
public: LZ77(int window_size, int dictionary_size) double encode(string input, string output) void decode(string input, string output)
private: Code get_prefix(vector<unsigned char> &s, int start_of_window, int current_size_of_dict); pair<vector<Code>, int> get_codes(string input); private: int dictionary_size; int window_size; int buffer_size;

7. Использованные аппаратные средства для проведения эксперимента

Проведение эксперимента выполнялось с применением:

1. Среда разработки CLion;
2. Процессор Intel Core i5-8250U 1.80GHz
3. ОЗУ 8.00Гб
4. ОС Windows 10 Pro

8. Результаты экспериментов

Результаты вычислительного эксперимента оформлены в виде таблиц и графиков.

файл	1.txt	2.docx	3.pptx	4.pdf	5.exe
символ	частота появления символа				
0	0	0.0300138	0.0491959	0.00260155	0.0461856
1	0	0.00562058	0.00335135	0.00312044	0.00532232
2	0	0.00584385	0.00267962	0.00290671	0.00640125
3	0	0.00305103	0.00348472	0.00298033	0.00473013
4	0	0.00589624	0.0028324	0.00303257	0.00458791
5	0	0.00317576	0.00292333	0.00302545	0.00448124
6	0	0.00270302	0.00343501	0.00307888	0.00364384
7	0	0.00353251	0.00299245	0.00312756	0.00477917
8	0	0.00557941	0.00474451	0.00302307	0.00432921
9	3.60E-06	0.00329676	0.00279602	0.00320712	0.00390867
10	0.00563535	0.00376077	0.00315735	0.00978165	0.00398346
11	0	0.00321194	0.00309308	0.00306463	0.00366837
12	0	0.00323189	0.00268326	0.00333061	0.00434147
13	0	0.0030635	0.00358172	0.00321306	0.00371618
14	0	0.00295124	0.00336226	0.00338523	0.00359112
15	0	0.00402396	0.00319857	0.00326411	0.00433657
16	0	0.00528005	0.00445957	0.00289484	0.00960494
17	0	0.00304853	0.00565146	0.00315012	0.00386208
18	0	0.00311714	0.00283967	0.00317981	0.00356047
19	0	0.00311589	0.00301791	0.00317981	0.00339373
20	0	0.00417614	0.00347623	0.00326055	0.00363404
21	0	0.00372959	0.0031137	0.00334842	0.00387802
22	0	0.00325435	0.00356475	0.00333654	0.00338392
23	0	0.00352003	0.00359991	0.00304089	0.00340844
24	0	0.00290509	0.00287968	0.00320118	0.00363158
25	0	0.00311839	0.00315613	0.00321187	0.00354944
26	0	0.00335912	0.00343137	0.00314063	0.00425197
27	0	0.00344519	0.00381816	0.00326649	0.00339618
28	0	0.00310466	0.00338408	0.00325105	0.00348446
29	0	0.00323688	0.0035599	0.00321306	0.00330913
30	0	0.00356868	0.00331012	0.00313113	0.00334591
31	0	0.0057029	0.00379876	0.0029732	0.00326744
32	0.0875556	0.00527631	0.0027063	0.0256866	0.00490791
33	0.00072753	0.00275915	0.00524891	0.00311332	0.00328461
34	2.40E-06	0.00286268	0.0036472	0.00303732	0.00335572
35	0	0.00368967	0.0034047	0.00302426	0.00327848
36	0	0.00342648	0.00283967	0.00322255	0.0038253
37	0	0.00323189	0.00321433	0.0031525	0.00353595
38	0	0.00295498	0.00324586	0.00312044	0.0032466
39	0.0004442	0.00344145	0.0032786	0.0030955	0.00336063
40	0.00012246	0.00415369	0.00308823	0.00334367	0.00354453

41	0.00012246	0.00285395	0.00317917	0.00334367	0.00351266
42	1.08E-05	0.00320944	0.00331497	0.00334485	0.00358009
43	0	0.00395785	0.003589	0.00339472	0.0033594
44	0.0128314	0.0030186	0.00337923	0.00327242	0.00346239
45	0.00090881	0.00344769	0.00375996	0.00356927	0.00334469
46	0.00536163	0.00327555	0.0044329	0.00337335	0.00341212
47	0	0.00417864	0.00495185	0.0130695	0.00339128
48	4.32E-05	0.00262443	0.00297547	0.0265831	0.00373703
49	2.76E-05	0.00282401	0.00329921	0.00823212	0.0035384
50	1.92E-05	0.00295623	0.00354413	0.00646767	0.00333856
51	1.44E-05	0.00326058	0.00356838	0.00624088	0.004225
52	2.40E-06	0.00361733	0.00359142	0.0060224	0.00343664
53	9.60E-06	0.00373707	0.00357202	0.00622544	0.00320737
54	4.80E-06	0.00319572	0.00400731	0.00604021	0.00339986
55	9.60E-06	0.00366223	0.003977	0.00644986	0.00334346
56	1.20E-05	0.00289386	0.00346653	0.00564006	0.00345994
57	0	0.00304479	0.00345441	0.00636437	0.00357519
58	0.00027012	0.00324811	0.0036084	0.00331398	0.00375296
59	0.00034095	0.00332919	0.00385575	0.00322493	0.0035286
60	0	0.00361982	0.0036278	0.00613045	0.00341948
61	0	0.00352752	0.00403035	0.0031905	0.00355066
62	0	0.00404143	0.00399397	0.00617914	0.00335817
63	0.0006543	0.00691159	0.00392365	0.00303732	0.00345626
64	0	0.00543722	0.00274146	0.00296014	0.00617933
65	0.00011405	0.0030186	0.00306762	0.00333892	0.00557121
66	5.04E-05	0.00264314	0.00472268	0.00374856	0.00328338
67	6.36E-05	0.00309593	0.00336468	0.00426983	0.00371863
68	3.48E-05	0.00252464	0.00377693	0.00379725	0.00363526
69	3.48E-05	0.00445555	0.00327011	0.00366189	0.00498638
70	9.60E-06	0.00353126	0.00352352	0.00733327	0.00341702
71	8.40E-06	0.00426845	0.00356596	0.00329973	0.00336308
72	8.40E-06	0.00300737	0.00303367	0.00318337	0.00348323
73	0.00012726	0.00355995	0.00325071	0.00427932	0.00345503
74	5.88E-05	0.00305602	0.00338166	0.00332467	0.00351266
75	4.80E-06	0.00383187	0.00409461	0.00336267	0.0033643
76	7.20E-05	0.00260822	0.00341682	0.0036892	0.00342683
77	0.00012366	0.00356993	0.00376845	0.00349921	0.00369044
78	2.88E-05	0.00315456	0.00365932	0.00340422	0.00343909
79	1.68E-05	0.00445305	0.00377572	0.00350278	0.00329687
80	3.12E-05	0.00316329	0.0036375	0.00438738	0.00453519
81	2.16E-05	0.00405141	0.0034629	0.00319643	0.00349181
82	1.80E-05	0.00338158	0.00344107	0.00749832	0.00341948
83	3.84E-05	0.00335164	0.00346047	0.00408341	0.0040276
84	1.68E-05	0.00288887	0.00350533	0.00441112	0.00344645
85	1.08E-05	0.00359363	0.00361203	0.00358114	0.00451557
86	7.68E-05	0.00364602	0.0037927	0.00355858	0.00379342
87	8.40E-06	0.00434828	0.00383271	0.00356571	0.00358499
88	6.36E-05	0.00296496	0.00321676	0.00329498	0.00337902
89	0	0.00335663	0.003977	0.00353959	0.00396139

90	2.40E-06	0.00382314	0.0039188	0.00351465	0.00358009
91	0.00037217	0.00384809	0.00381816	0.00421283	0.00378362
92	0	0.00332794	0.00355383	0.0032558	0.00334836
93	0.00037097	0.00374331	0.00422313	0.00425795	0.00375296
94	0	0.00354373	0.00382058	0.00321662	0.00382898
95	0	0.00567047	0.00427648	0.00318218	0.00349304
96	8.40E-06	0.00250094	0.00279239	0.00305276	0.00325028
97	0.00181282	0.00300737	0.00335256	0.00683813	0.00388047
98	0.00020529	0.0026631	0.00315856	0.00598084	0.00331771
99	0.0008716	0.00339779	0.00353686	0.0047899	0.00351879
100	0.00084518	0.00289137	0.00426557	0.00565431	0.00449963
101	0.00427994	0.00369591	0.0050052	0.0124212	0.00413182
102	0.00025812	0.00312337	0.0035405	0.00349328	0.00401534
103	0.00020649	0.00384435	0.00465721	0.0050535	0.00346116
104	0.00038537	0.0033479	0.0034629	0.00403354	0.00412936
105	0.00178521	0.00411377	0.00487788	0.00505112	0.00375296
106	0.00016087	0.00341027	0.00389455	0.00577305	0.0036843
107	2.40E-05	0.00436699	0.00388121	0.00320356	0.00332997
108	0.00116933	0.00290509	0.00538835	0.00457854	0.00376155
109	0.00096044	0.00427967	0.00505491	0.00528504	0.00358009
110	0.00193888	0.00351255	0.00413098	0.00914047	0.00384737
111	0.00167956	0.00451417	0.00432983	0.00875219	0.00391848
112	0.00072153	0.00270551	0.00413705	0.00455717	0.00381549
113	0.00027853	0.00360735	0.00343743	0.00327955	0.00329319
114	0.00183923	0.00329426	0.00411886	0.00609008	0.00410484
115	0.00206013	0.00363105	0.00532287	0.00587635	0.00383634
116	0.00168076	0.00329052	0.0042777	0.0086952	0.00517519
117	0.00173358	0.00379445	0.00412371	0.00391836	0.00425565
118	0.00053784	0.00344145	0.0045008	0.00335435	0.00339618
119	1.80E-05	0.00394538	0.00398427	0.00343628	0.00352492
120	8.64E-05	0.00410754	0.00405823	0.00402522	0.00363771
121	6.72E-05	0.00341526	0.00372965	0.00437906	0.00359725
122	0.00015727	0.00352752	0.00399761	0.00335079	0.00341702
123	0	0.00366722	0.00398063	0.003181	0.0033643
124	0	0.00434453	0.00390546	0.00324393	0.0035617
125	0	0.00388426	0.00420495	0.00328905	0.00348568
126	0	0.00442686	0.00395032	0.00321424	0.00345626
127	0	0.00854936	0.0042777	0.0031145	0.0033312
128	0.0231621	0.00508546	0.00257656	0.00283903	0.00802823
129	0.0212736	0.00331297	0.00263355	0.00304207	0.00347465
130	0.0234322	0.003066	0.00271357	0.00317387	0.00313013
131	0.0118878	0.00316329	0.00307611	0.00323324	0.00469458
132	0.00099285	0.00281403	0.00441228	0.00303614	0.00365611
133	0.00355121	0.00355995	0.00314158	0.00328192	0.00424216
134	0.00177921	0.00327306	0.00340591	0.00329617	0.00327971
135	0.00545167	0.00383811	0.00348472	0.00326293	0.00337411
136	0.00399901	0.00295373	0.00356717	0.00288771	0.00342928
137	0.00120775	0.00318075	0.00306277	0.00315369	0.00496063
138	0.0002161	0.00434079	0.00307732	0.00324393	0.00341702

139	0.00800883	0.00336287	0.00376238	0.00353721	0.00860448
140	0.00810727	0.00318325	0.00317917	0.00323562	0.00328951
141	0.00099645	0.00387054	0.00379755	0.00322849	0.00465902
142	0.00282008	0.00335039	0.00381816	0.00324037	0.00325028
143	0.00959474	0.00509669	0.00381695	0.00329024	0.00326009
144	0.00089921	0.00273919	0.00310763	0.0030492	0.00442239
145	0.00098565	0.00332794	0.00338651	0.00301714	0.00329687
146	0.00086079	0.00345018	0.00317553	0.00336742	0.00338515
147	0.0002257	0.00328927	0.00361567	0.00335317	0.00332384
148	0.00411546	0.00307972	0.00325798	0.00325224	0.00339005
149	0.00014887	0.00363479	0.00369812	0.00340897	0.00336676
150	8.52E-05	0.00341401	0.00364962	0.00335435	0.00326131
151	0.00010805	0.00394663	0.00381088	0.00329142	0.00322331
152	0.00037577	0.00265936	0.00303246	0.00328311	0.00340231
153	2.40E-06	0.00315206	0.00352958	0.00342322	0.00332262
154	0.00072633	0.00376077	0.00350291	0.00324986	0.00321472
155	0.00011405	0.00336786	0.00405338	0.00339354	0.0033643
156	0.00052944	0.00306225	0.00339136	0.0034161	0.00335327
157	0.0008848	0.00342399	0.00403398	0.00327599	0.00331404
158	0.00068431	0.00368094	0.00437348	0.00331992	0.00332875
159	0.0008716	0.00525635	0.00416857	0.00324155	0.00328461
160	0.00039978	0.00321194	0.00294637	0.00303376	0.00336308
161	0.00039738	0.00297993	0.00331861	0.0031905	0.00357151
162	0.00038898	0.00390048	0.00339742	0.00328192	0.00340599
163	8.40E-05	0.00390671	0.00357323	0.00341254	0.00346361
164	6.48E-05	0.00307473	0.003201	0.00320118	0.00352982
165	5.16E-05	0.00387802	0.00396002	0.00317862	0.00342928
166	0.00057266	0.00320071	0.00381573	0.00333654	0.00325641
167	0.00019209	0.00414745	0.00385938	0.00348497	0.00331158
168	0.00020409	0.00297868	0.00330285	0.00308363	0.00342561
169	0.00039858	0.00350257	0.00384362	0.00319999	0.00329932
170	6.36E-05	0.00325559	0.0037733	0.00326768	0.00332752
171	0.00019329	0.00400151	0.00404368	0.00338523	0.00357641
172	7.20E-06	0.00319073	0.0034726	0.00328311	0.00340599
173	0.00018008	0.00424974	0.00421828	0.0035194	0.00331894
174	1.92E-05	0.00398779	0.00408612	0.00347547	0.00335204
175	0.00028213	0.00514908	0.00412371	0.00333654	0.00324905
176	0.0349382	0.00252464	0.00316583	0.00314775	0.00366101
177	0.0066246	0.00336661	0.00369206	0.00316437	0.00335695
178	0.0184307	0.00321443	0.00370903	0.0032843	0.00337534
179	0.00858509	0.00363978	0.00461598	0.00331161	0.00316813
180	0.0123236	0.00381316	0.00383392	0.00336267	0.00331281
181	0.0326332	0.0043146	0.00417948	0.00339947	0.00311787
182	0.00412267	0.00365225	0.00452383	0.00330567	0.00328829
183	0.00752381	0.00429464	0.00415402	0.00350396	0.00357396
184	0.0278658	0.00329177	0.00329557	0.00321068	0.00367695
185	0.00487901	0.0036298	0.00370297	0.00313706	0.00357641
186	0.0146562	0.00370963	0.00411522	0.00326293	0.00359971
187	0.0211476	0.00356743	0.00447776	0.00359895	0.00339741

188	0.0120258	0.00365849	0.00373571	0.00332111	0.0033876
189	0.0264444	0.00361358	0.00419767	0.00327717	0.00326009
190	0.0468968	0.0041961	0.0040837	0.00334485	0.00330545
191	0.0100582	0.00629914	0.0044717	0.00328192	0.00361319
192	0	0.00281902	0.00275237	0.00294471	0.00502439
193	0	0.00316703	0.00309915	0.00312756	0.00348323
194	0.00037457	0.00372834	0.00324343	0.00322849	0.00349917
195	0.00079716	0.00381316	0.00354898	0.00314181	0.00475833
196	0	0.0035375	0.00298396	0.00307651	0.00365978
197	0	0.00351255	0.00373086	0.00335792	0.00336308
198	0	0.00367346	0.00349806	0.00326293	0.00385105
199	0	0.00445305	0.00380361	0.00320118	0.00364017
200	0	0.00327056	0.00333316	0.0030955	0.00347097
201	0	0.00306101	0.00347017	0.00318337	0.00348201
202	0	0.00317826	0.0035211	0.00344459	0.0032981
203	0	0.00368967	0.00411037	0.00348734	0.00340599
204	0	0.00314458	0.00329678	0.00318337	0.00329932
205	0	0.00354623	0.00388	0.00337335	0.00323679
206	0	0.00321069	0.00369812	0.00321187	0.00340722
207	0	0.00432208	0.00471056	0.00316081	0.00323925
208	0.298581	0.00342024	0.00313431	0.00307294	0.00358254
209	0.122597	0.00375454	0.00372601	0.00311332	0.00345135
210	0	0.00373208	0.00382786	0.00317031	0.0034489
211	0	0.00413498	0.00409218	0.00345528	0.00341212
212	0	0.00363728	0.00366417	0.00318693	0.00332997
213	0	0.00370963	0.00424496	0.00336385	0.00335572
214	0	0.0039142	0.00392365	0.00329498	0.00338269
215	0	0.00450794	0.00432983	0.00331161	0.00334346
216	0	0.00307473	0.00378421	0.00324274	0.00404109
217	0	0.00364602	0.00421828	0.00325936	0.00344522
218	0	0.00420234	0.00429588	0.0035384	0.00338269
219	0	0.00402771	0.00476027	0.00330092	0.00354086
220	0	0.00344644	0.00366902	0.00326174	0.00342928
221	0	0.00374705	0.00438197	0.00340185	0.00336308
222	0	0.00373208	0.00424617	0.00334604	0.00318407
223	0	0.00479109	0.0044135	0.00331755	0.00339986
224	0	0.00358365	0.00286028	0.00297558	0.00358254
225	0	0.00469629	0.00328466	0.00330805	0.00344155
226	0.0042019	0.00441064	0.00366902	0.00314063	0.0033876
227	0	0.00470377	0.00389576	0.00327599	0.00345748
228	0	0.00350382	0.00359021	0.00305988	0.00365365
229	0	0.00348511	0.00388848	0.0035479	0.00341702
230	0	0.00347388	0.00372237	0.0031335	0.00334959
231	0	0.00372585	0.0038994	0.00326768	0.0033876
232	0	0.00375578	0.00338166	0.00311332	0.00636692
233	0	0.00379944	0.00385211	0.00330092	0.00386821
234	0	0.00378697	0.00408248	0.00312875	0.00348568
235	0	0.00396534	0.00429467	0.00335079	0.00402883
236	0	0.00379445	0.00424011	0.00330092	0.00383634

237	0	0.00441813	0.00465721	0.0034161	0.00329932
238	0	0.00368593	0.00408976	0.00326055	0.00337289
239	0	0.0042173	0.0043262	0.00326174	0.00338392
240	0	0.00565426	0.00328345	0.00301239	0.00470929
241	0	0.00544221	0.00404126	0.00320118	0.0033876
242	0	0.00396284	0.00390303	0.00337216	0.00341212
243	0	0.00386056	0.00410067	0.00329142	0.00334469
244	0	0.00441563	0.0037151	0.00341372	0.00349794
245	0	0.00401274	0.00444017	0.00314063	0.00332752
246	0	0.00460648	0.00432377	0.00315606	0.00340109
247	0	0.0039828	0.00402186	0.00328786	0.00340231
248	0	0.00707873	0.00356353	0.0030492	0.00398469
249	0	0.00443933	0.00427042	0.00333417	0.00345626
250	0	0.00479857	0.00418918	0.00314894	0.00336308
251	0	0.00500064	0.00435893	0.00311925	0.0035384
252	0	0.00707374	0.0039673	0.00334485	0.00427894
253	0	0.00573034	0.00411886	0.00303732	0.00342683
254	0	0.00866287	0.00425708	0.00308126	0.00365611
255	0	0.0155707	0.00446442	0.00283547	0.0125254
энтропия	4.25003	7.89967	7.86981	7.84915	7.86299

Таблица 1.1 Частоты появления символов в файле

файл	6.jpg	7.jpg (чб)	8.bmp	9.bp(чб)	10.html
символ	частота появления символа				
0	0.0382307	0.109977	0.00127321	0.000412011	0
1	0.00160753	0.0174725	0.000557254	0.000174039	0
2	0.00125637	0.0211893	0.000754142	0.000196534	0
3	0.00225516	0.00251434	0.000847217	0.000275858	0
4	0.00164276	0.000481966	0.000986829	0.000324399	0
5	0.00169501	0.0172426	0.00105604	0.000408459	0
6	0.00151275	0.000800408	0.00111451	0.000299537	0
7	0.00306196	0.00242459	0.00108229	0.000419114	0
8	0.00181287	0.000478278	0.00111451	0.000426218	0
9	0.00194775	0.00089508	0.000986829	0.000317296	0.000296249
10	0.00263547	0.0176582	0.000992795	0.000383596	0.00225651
11	0.00230376	0.00127131	0.000823352	0.000401355	0
12	0.00187363	0.000886474	0.000718345	0.000460552	0
13	0.00238274	0.00138442	0.000689706	0.00046884	0.00225651
14	0.00197205	0.000815162	0.000610951	0.000394252	0
15	0.00463303	0.00378196	0.000478499	0.000397804	0
16	0.00153827	0.00044877	0.000484465	0.000362285	0
17	0.00209355	0.000784425	0.000430768	0.000404907	0
18	0.00204981	0.000662704	0.000368718	0.000454633	0
19	0.00228553	0.000884015	0.000371105	0.000477127	0
20	0.00193195	0.0642859	0.000384231	0.000561187	0
21	0.00395503	0.00153319	0.000396163	0.000487783	0
22	0.00216889	0.000715572	0.000362752	0.000426218	0
23	0.00287241	0.001575	0.000334114	0.000440425	0
24	0.00229525	0.000693441	0.000357979	0.000522117	0
25	0.00287727	0.000623359	0.000316215	0.00042977	0

26	0.00351032	0.00192786	0.000360366	0.000511462	0
27	0.00315309	0.000967621	0.000264904	0.000490151	0
28	0.00253097	0.000593851	0.000262518	0.000493703	0
29	0.003017	0.00110041	0.000282803	0.000564739	0
30	0.00323814	0.00112008	0.00028519	0.000604993	0
31	0.00764031	0.00516884	0.000264904	0.00058605	0
32	0.00175212	0.00039713	0.000287576	0.000606177	0.0370681
33	0.0018797	0.00062213	0.000260131	0.000518565	7.65E-05
34	0.00203523	0.000780736	0.000241039	0.000596705	0.0230118
35	0.00365005	0.000817621	0.000276837	0.000628672	0.00129728
36	0.00281287	0.000781966	0.000221947	0.000612097	1.19E-06
37	0.00225516	0.000972539	0.000204048	0.000649983	0.0546245
38	0.00231591	0.000789343	0.000251779	0.00065827	0.001578
39	0.0034204	0.00159344	0.000207628	0.000600257	5.85E-05
40	0.00177399	0.0787412	0.000188536	0.000644063	0.000836186
41	0.00215674	0.000624589	0.000173023	0.000635775	0.000836186
42	0.00336451	0.000749998	0.000151544	0.000639327	1.43E-05
43	0.00456742	0.00194877	0.000243426	0.000752985	9.68E-05
44	0.00195139	0.00052254	0.000202855	0.000721019	0.00366966
45	0.00316281	0.000805326	0.000220754	0.000799159	0.0139356
46	0.00244592	0.000724179	0.000184956	0.000710363	0.0138293
47	0.00415916	0.00205942	0.000206435	0.000689053	0.0169208
48	0.00151883	0.000233606	0.000149158	0.000717467	0.0299104
49	0.00246536	0.000513933	0.000258938	0.000728123	0.0148543
50	0.0025407	0.000517622	0.000176603	0.000732858	0.00809309
51	0.0032916	0.000741392	0.00017183	0.000752985	0.00430636
52	0.00252611	0.00117172	0.000198082	0.000816918	0.00519988
53	0.00437544	0.00141393	0.000200468	0.000756537	0.00475312
54	0.00290521	0.000618441	0.000208821	0.000767193	0.00251453
55	0.00383717	0.00105123	0.000205241	0.000696156	0.00303297
56	0.00254313	0.000238524	0.000149158	0.000699708	0.0141531
57	0.00274361	0.000732785	0.000162284	0.000671294	0.0046922
58	0.00302065	0.000797949	0.000186149	0.000699708	0.00623914
59	0.00322356	0.000773359	0.00016825	0.000628672	0.00356215
60	0.00341068	0.000866802	0.000187342	0.000594337	0.0139643
61	0.00363304	0.00133155	0.000184956	0.000561187	0.0124532
62	0.00411177	0.00137213	0.00016825	0.000575394	0.0140826
63	0.0107278	0.00787744	0.000176603	0.000532773	0.000353587
64	0.00139732	0.0184168	0.00022314	0.000499622	5.02E-05
65	0.0022746	0.000879097	0.000187342	0.000490151	0.0029601
66	0.00200121	0.000580327	0.000169443	0.000481863	0.0197185
67	0.00284689	0.00114959	0.000169443	0.000518565	0.00191726
68	0.00151154	0.00044754	0.000159897	0.000515014	0.0295401
69	0.00278614	0.0488323	0.000159897	0.000486599	0.00419049
70	0.00320776	0.000591392	0.00016825	0.000490151	0.00246914
71	0.00397447	0.00230901	0.000176603	0.00046292	7.29E-05
72	0.00213851	0.000484425	0.00017183	0.000518565	0.000163654
73	0.0032831	0.000839752	0.000163477	0.000415563	0.000744206
74	0.00278006	0.00121598	0.000176603	0.000436874	4.90E-05

75	0.00294045	0.000933195	0.000167057	0.000419114	8.60E-05
76	0.00162818	0.000481966	0.000183763	0.000426218	0.000303416
77	0.00347143	0.00104508	0.000181376	0.000367021	0.000383451
78	0.00258322	0.000565573	0.000155124	0.000461736	0.000234132
79	0.00524299	0.00178278	0.000147965	0.00050791	0.000175599
80	0.00167922	0.0167975	0.000163477	0.000475944	0.000428844
81	0.002791	0.0489724	0.000153931	0.000436874	3.82E-05
82	0.00278371	0.000456147	0.000189729	0.000473576	0.000487377
83	0.00312393	0.000761064	0.000174216	0.000397804	0.000498128
84	0.00268893	0.000601228	0.000161091	0.000394252	0.000608027
85	0.00416037	0.000848359	0.000221947	0.000380044	8.24E-05
86	0.00428917	0.000595081	0.000136032	0.000412011	0.000164848
87	0.00515186	0.00252786	0.000159897	0.000397804	0.000548299
88	0.00275698	0.000382376	0.000124099	0.000440425	0.000139763
89	0.00294045	0.000526228	0.000155124	0.000376493	4.66E-05
90	0.00441554	0.00105492	0.000146771	0.000401355	2.27E-05
91	0.0037108	0.000961473	0.00016467	0.000394252	0.000444373
92	0.00358443	0.000657786	0.000149158	0.000406091	0.000150514
93	0.00366463	0.000992211	0.000157511	0.00042977	0.000444373
94	0.00406074	0.00143237	0.000162284	0.000381228	0
95	0.00648479	0.00434507	0.000156317	0.000447529	0.00505415
96	0.0014034	0.000389753	0.000170637	0.000416747	0
97	0.00211664	0.000688523	0.000124099	0.000440425	0.0287254
98	0.00210084	0.000513933	0.000133645	0.000419114	0.00593334
99	0.00409112	0.00101926	0.000134839	0.000447529	0.0118655
100	0.00239003	0.000540982	0.000200468	0.000436874	0.0112873
101	0.00275455	0.000629507	0.000156317	0.000387148	0.0309747
102	0.00296718	0.000464753	0.000169443	0.000377677	0.00792944
103	0.00409598	0.000938113	0.000173023	0.000383596	0.00792227
104	0.00250181	0.00115205	0.000156317	0.000340974	0.00892928
105	0.00339853	0.000886474	0.000183763	0.000404907	0.033549
106	0.00430011	0.00119262	0.000146771	0.000369389	0.000330891
107	0.00559172	0.00140778	0.00016467	0.000319664	0.00693318
108	0.00234872	0.000458606	0.000143192	0.000397804	0.0174142
109	0.00373146	0.00112131	0.000136032	0.000380044	0.00613402
110	0.00319197	0.000623359	0.000161091	0.00042977	0.0178538
111	0.0057035	0.00228565	0.000157511	0.000394252	0.0160285
112	0.00181773	0.00042295	0.000118133	0.000305456	0.0157167
113	0.00346536	0.000517622	0.000141998	0.000454633	0.000155292
114	0.0027424	0.000561884	0.000169443	0.000358734	0.0187676
115	0.00357593	0.000843441	0.000181376	0.00034571	0.0212666
116	0.00288699	0.00128975	0.000136032	0.000394252	0.0273182
117	0.00381165	0.00114221	0.000145578	0.000358734	0.00799872
118	0.00320776	0.000590163	0.000147965	0.000380044	0.00218484
119	0.00349816	0.00108074	0.000165864	0.000377677	0.0081337
120	0.00346536	0.00123934	0.000180183	0.000348078	0.00239269
121	0.00369865	0.000922129	0.000149158	0.000362285	0.00221948
122	0.0040741	0.00147172	0.000161091	0.000380044	0.000661782
123	0.00350181	0.000839752	0.000165864	0.000362285	0.000769291

124	0.00399999	0.00185041	0.000159897	0.000356366	9.08E-05
125	0.00420412	0.00113606	0.000177796	0.000422666	0.000769291
126	0.00415794	0.00206926	0.000210014	0.000316112	0
127	0.0164033	0.0121303	0.000213594	0.000404907	0
128	0.00157351	0.0187955	0.0002303	0.000393068	0.00783865
129	0.00179708	0.000848359	0.000249392	0.000376493	0.00869156
130	0.00206318	0.000738933	0.000251779	0.000415563	0.00907382
131	0.0022746	0.000906146	0.000342467	0.000380044	0.00290396
132	0.00222599	0.000936883	0.000535775	0.000426218	0.00287529
133	0.00325151	0.00134385	0.000749369	0.000434506	0.000911443
134	0.0027266	0.00118033	0.000984442	0.000319664	0.00124114
135	0.00378492	0.0015209	0.00140686	0.000330319	0.00161503
136	0.001904	0.000869261	0.00216458	0.000330319	0.000422871
137	0.00230133	0.000810244	0.0141795	0.000348078	0.000370311
138	0.00303766	0.0624577	0.00454037	0.000388332	5.38E-05
139	0.0029125	0.00099467	0.0138872	0.000358734	0.00183722
140	0.00326487	0.000453688	0.00183047	0.000408459	0.00183961
141	0.00416402	0.00142992	0.00132094	0.000383596	0.000415704
142	0.00355163	0.000764753	0.000951031	0.000348078	0.00070598
143	0.00607896	0.00320778	0.000730277	0.000331503	0.00349884
144	0.00212514	0.000451229	0.000565607	0.000401355	0.000365533
145	0.0034447	0.00091967	0.00051191	0.000337423	0.000698813
146	0.00313243	0.00114713	0.000490431	0.000412011	0.000474237
147	0.00353705	0.00159098	0.00039855	0.000316112	0.000244883
148	0.00196962	0.00108811	0.000357979	0.0003907	0.00118738
149	0.00391979	0.000863113	0.000324568	0.000337423	8.96E-05
150	0.00294896	0.000512704	0.000809033	0.000298353	5.85E-05
151	0.00361481	0.00112992	0.000624077	0.000348078	8.96E-05
152	0.00170838	0.000239754	0.000877049	0.000412011	0.000240105
153	0.00273268	0.000611064	0.000754142	0.000417931	3.23E-05
154	0.00349816	0.000810244	0.00121594	0.000419114	0.000439595
155	0.00354677	0.000800408	0.000498784	0.00035163	0.000170821
156	0.00264276	0.000463524	0.000970123	0.000340974	0.000339253
157	0.00312636	0.000848359	0.000447474	0.000422666	0.000265191
158	0.00368771	0.000849588	0.000509523	0.000380044	0.00057219
159	0.00726486	0.00371803	0.000387811	0.000330319	0.000494544
160	0.00201943	0.0232426	0.000620497	0.000480679	0.000407342
161	0.00282502	0.000977457	0.000294736	0.000401355	0.000642669
162	0.00219197	0.0582466	0.000221947	0.000376493	0.000166043
163	0.00380801	0.00173606	0.000196888	0.000447529	7.17E-05
164	0.00256135	0.000579097	0.000198082	0.000422666	0.000639085
165	0.0035249	0.00101434	0.000217174	0.000426218	0.00012065
166	0.00268529	0.000656556	0.000195695	0.000472392	5.97E-05
167	0.0042673	0.00120492	0.000247006	0.000412011	8.96E-05
168	0.00270716	0.00113852	0.000313828	0.000340974	0.000194712
169	0.00371323	0.000682376	0.00055964	0.000404907	2.15E-05
170	0.00368771	0.00084467	0.000303089	0.000451081	2.03E-05
171	0.0048967	0.00127377	0.000835284	0.000419114	0.000388229
172	0.00359051	0.000552048	0.00062169	0.000404907	2.39E-05

173	0.00499026	0.00111024	0.00045344	0.000376493	0.000176794
174	0.00476912	0.000961473	0.000275644	0.000362285	4.66E-05
175	0.00643133	0.00357417	0.000423608	0.000383596	3.58E-05
176	0.00195139	0.000619671	0.000947451	0.000465288	0.0107534
177	0.00368407	0.000966391	0.00045344	0.000376493	0.0015792
178	0.00282259	0.000570491	0.000371105	0.000415563	0.00497531
179	0.00375333	0.000981146	0.000264904	0.000475944	0.00171179
180	0.00344713	0.00100082	0.000248199	0.000440425	0.00312973
181	0.00543983	0.00122951	0.000256552	0.000472392	0.0103651
182	0.00327824	0.000949178	0.000328147	0.000365837	0.000700007
183	0.00500363	0.00146311	0.000659875	0.000483047	0.0024787
184	0.00293073	0.000443852	0.000495204	0.000487783	0.0149952
185	0.00376791	0.000698359	0.000631236	0.000497254	0.00176316
186	0.00356013	0.00126393	0.000441507	0.00050791	0.00610058
187	0.00375333	0.000850818	0.000293543	0.000490151	0.0066441
188	0.00374239	0.00110533	0.000761302	0.000575394	0.00382257
189	0.0038238	0.00127131	0.000467759	0.000539876	0.00839053
190	0.00412635	0.00151721	0.000601405	0.000564739	0.0141638
191	0.00976667	0.00567663	0.000532195	0.000554084	0.00248228
192	0.00188578	0.000678687	0.000577539	0.000635775	0
193	0.00213122	0.00105737	0.000510717	0.000582498	0
194	0.00365005	0.0018	0.00051907	0.000505542	0.000778848
195	0.00356864	0.00227336	0.00056322	0.000589602	9.44E-05
196	0.00270959	0.000986064	0.000689706	0.000518565	1.55E-05
197	0.00311542	0.0011041	0.00129946	0.000603809	1.19E-05
198	0.00416402	0.0012086	0.00123861	0.000610913	0
199	0.00556013	0.00182336	0.00143192	0.000610913	0
200	0.00346657	0.00112377	0.0134338	0.000660638	0
201	0.00329282	0.0017127	0.00141282	0.000610913	7.17E-06
202	0.00310691	0.000779507	0.00157033	0.000632224	1.19E-06
203	0.00353705	0.000818851	0.00124338	0.000628672	0
204	0.00287727	0.00039713	0.00102979	0.000543428	9.56E-06
205	0.00355649	0.00072049	0.00145578	0.000618016	0
206	0.00333413	0.000494261	0.000942678	0.000628672	8.00E-05
207	0.00569378	0.00252663	0.00133288	0.000618016	4.18E-05
208	0.00309719	0.0020459	0.00102979	0.000646431	0.0994703
209	0.00326852	0.00216024	0.000619304	0.000554084	0.0429704
210	0.00331591	0.00102541	0.0010811	0.000681949	7.17E-06
211	0.00373146	0.00110533	0.000661068	0.000642879	9.56E-06
212	0.00383595	0.000906146	0.000912846	0.000622752	1.19E-06
213	0.00445199	0.00106967	0.000477305	0.000689053	4.90E-05
214	0.00452489	0.00103647	0.000490431	0.000699708	1.31E-05
215	0.00617616	0.00227582	0.000643169	0.000722203	3.34E-05
216	0.00290157	0.000909834	0.000847217	0.000735226	4.78E-05
217	0.0035334	0.000831146	0.000622883	0.000717467	6.21E-05
218	0.00443133	0.00128237	0.000299509	0.000692604	4.78E-06
219	0.00449452	0.00153073	0.000459406	0.000706812	1.91E-05
220	0.00289549	0.00072172	0.000875855	0.000788503	0
221	0.00370959	0.000850818	0.000725504	0.000696156	0

222	0.0037837	0.000929506	0.00051191	0.00074233	0
223	0.00723934	0.00383483	0.000806646	0.000710363	1.55E-05
224	0.00273146	0.001425	0.000743403	0.000752985	0.00032014
225	0.0051385	0.0025377	0.00104649	0.000806263	0.000134984
226	0.00374239	0.00153319	0.000608564	0.000779032	0.00123875
227	0.00558807	0.00241106	0.00051907	0.00069734	0
228	0.00438638	0.00218975	0.000286383	0.000809814	1.19E-06
229	0.00368407	0.00119508	0.00028519	0.000866643	1.79E-05
230	0.00357471	0.000619671	0.000353206	0.000845333	5.97E-06
231	0.00517374	0.00175573	0.000420029	0.000831125	2.39E-06
232	0.0038639	0.00296803	0.000526229	0.000841781	8.36E-06
233	0.00374968	0.00134262	0.00034008	0.000969646	0
234	0.00382137	0.00105	0.000383038	0.000827573	1.19E-06
235	0.00504251	0.00158114	0.000420029	0.000994509	0
236	0.00417617	0.00132049	0.000412869	0.000969646	2.39E-06
237	0.00515186	0.00182827	0.000459406	0.000941232	2.39E-06
238	0.00352004	0.0006209	0.000814999	0.00103713	0
239	0.00514943	0.00214672	0.000529809	0.000955439	1.19E-06
240	0.00590763	0.0043463	0.00068732	0.000969646	0
241	0.00594044	0.00351639	0.000980862	0.00115789	0
242	0.00536207	0.00274917	0.00182211	0.00101937	0
243	0.005034	0.00170409	0.00246767	0.00109396	0
244	0.00470108	0.0032754	0.00186507	0.00118986	0
245	0.00508747	0.00186147	0.00211088	0.0013118	0
246	0.0061385	0.00272827	0.00260847	0.0014598	0
247	0.00513971	0.00209262	0.00320153	0.00168356	0
248	0.0084945	0.00589056	0.00503199	0.00206005	0
249	0.00679463	0.00376843	0.00564891	0.00305811	0
250	0.00580435	0.00373401	0.00930149	0.00492637	0
251	0.00768162	0.00346844	0.012534	0.00774296	0
252	0.0110486	0.0075172	0.0168572	0.0126942	0
253	0.00943131	0.0058881	0.0282457	0.0196806	0
254	0.0162296	0.0114836	0.0455612	0.0353654	0
255	0.0362234	0.0352954	0.70822	0.782442	0
энтропия	7.71074	5.91744	2.59884	2.19826	5.85361

Таблица 1.2 Частоты появления символов в файле

имя файла	S1(Кб)	H	Алгоритм Шеннона-Фано		Алгоритм LZ77, окно 5 Кб		Алгоритм LZ77, окно 10Кб		Алгоритм LZ77, окно 20 Кб	
			S2(Кб)	K	S2(Кб)	K	S2(Кб)	K	S2(Кб)	K
1.txt	814	0.531254	438	0.537335	578	0.709635	518	0.63669	467	0.573746
2.docx	783	0.987458	779	0.994922	1655	2.11385	1579	2.01556	1477	1.88534
3.pptx	806	0.983727	800	0.992162	1718	2.13206	1659	2.05938	1568	1.94588
5.exe	797	0.965814	791	0.991977	1726	2.16579	1676	2.10403	1726	2.00744
4.pdf	823	0.982873	815	0.990729	1726	2.09772	1661	2.0186	1584	1.92501
6.jpeg	804	0.963843	781	0.971475	1766	2.19725	1680	2.08968	1580	1.96497
7.jpeg (чб)	795	0.73968	593	0.746062	308	0.387632	231	0.290489	148	0.186147
8.bmp	819	0.324856	282	0.344063	255	0.31075	238	0.290804	226	0.275578
9.bmp(чб)	825	0.274782	255	0.308233	291	0.352577	260	0.3152	243	0.294594
10.html	818	0.731701	604	0.738282	397	0.485036	333	0.407193	397	0.350117

Таблица 2. Коэффициенты сжатия файлов

Где H – энтропия исходного файла,
K – коэффициент сжатия,
S1 – размер исходного файла,
S2 – размер сжатого файла

имя файла	S1(Кб)	H	Алгоритм Шеннона-Фано		Алгоритм LZ77, окно 5 Кб		Алгоритм LZ77, окно 10Кб		Алгоритм LZ77, окно 20 Кб	
			tp	tu	tp	tu	tp	tu	tp	tu
1.txt	814	0.531254	87.1	173.4	3235.3	43.2	5718.5	41.6	10160.7	41.5
2.docx	783	0.987458	82.8	251.9	4334.9	63.7	8254.3	64.7	15083.7	62.1
3.pptx	806	0.983727	95.7	255.8	4499.6	66.9	8582.1	65.3	16030.7	63.7
5.exe	797	0.965814	84.1	261.2	4510.5	66.9	8598.7	65.7	16177	65.3
4.pdf	823	0.982873	81.1	293.9	4619.7	107.4	8745.4	106.5	16505.5	105.3
6.jpeg	804	0.963843	83.8	261.2	4691.2	63.7	8813.8	62.5	16393.8	62.3
7.jpeg (чб)	795	0.73968	76	177.1	1004.5	37.1	1461.3	35.6	1716.3	34.2
8.bmp	819	0.324856	58.9	115.1	3257.8	37.1	5907.3	37.1	8401.4	36.6
9.bmp(чб)	825	0.274782	57.9	105.6	5345.8	38.7	9541.8	37.1	16027.5	37.1
10.html	818	0.731701	90.7	268.3	1424.9	130	2282.9	129.7	3821.8	127.2

Таблица 3. Время упаковки/распаковки файлов (в миллисекундах)

Где S1 – размер исходного файла,
H – энтропия исходного файла,
tp – время упаковки файла,
tu – время распаковки файлы.

Распределение частот в файлах представим в вид графиков:


Распределение относительных частот символов в файле 1.txt
Частота
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ


Распределение относительных частот в файле 2.docx
Частота
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ

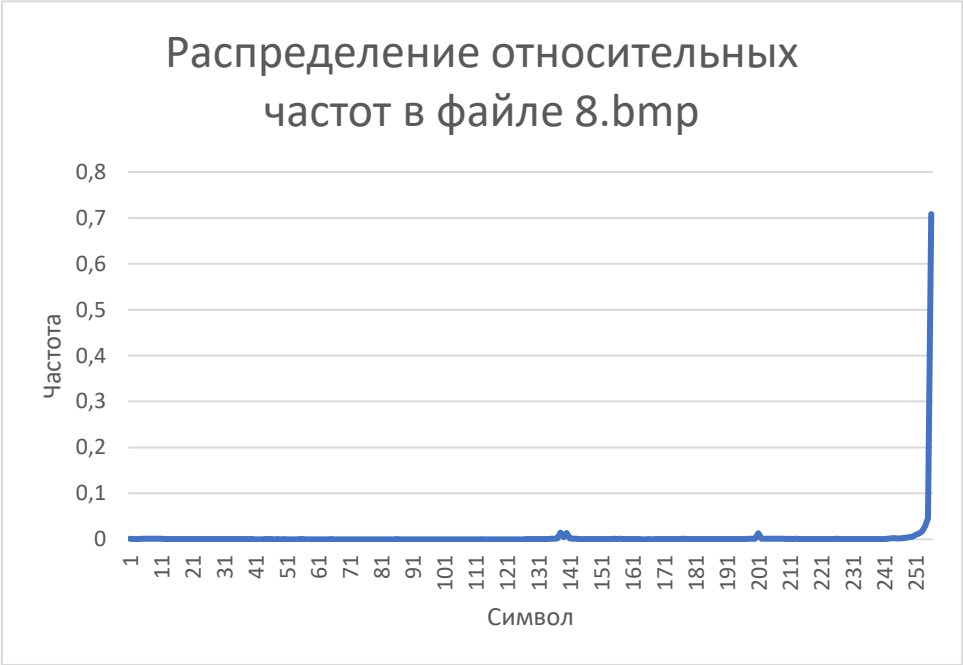

Распределение относительных частот в файле 3.pptx
Частота
0,06
0,05
0,04
0,03
0,02
0,01
0
1 17 33 49 65 81 97 113 129 145 161 177 193 209 225 241
Символ

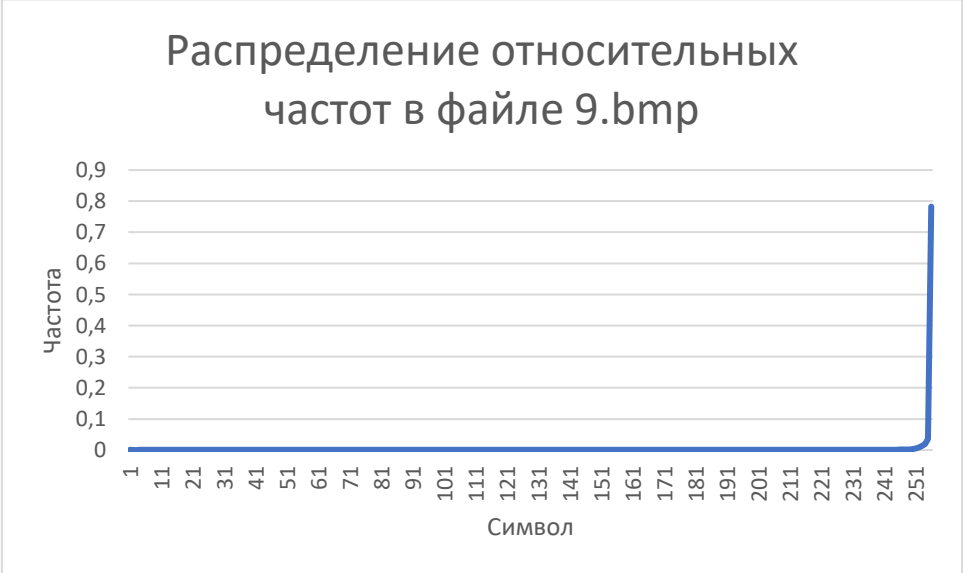

Распределение относительных частот в файле 4.pdf
Частота
0,03
0,025
0,02
0,015
0,01
0,005
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ


Распределение относительных частот в файле 5.exe
Частота
0,05
0,04
0,03
0,02
0,01
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ


Распределение относительных частот в файле 6.jpg
Частота
0,045
0,04
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ


Распределение относительных частот в файле 7.jpg
Частота
0,12
0,1
0,08
0,06
0,04
0,02
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ


Распределение относительных частот в файле 8.bmp
Частота
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ


Распределение относительных частот в файле 9.bmp
Частота
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Символ

Диаграммы 1-10. Распределение относительных частот в файлах

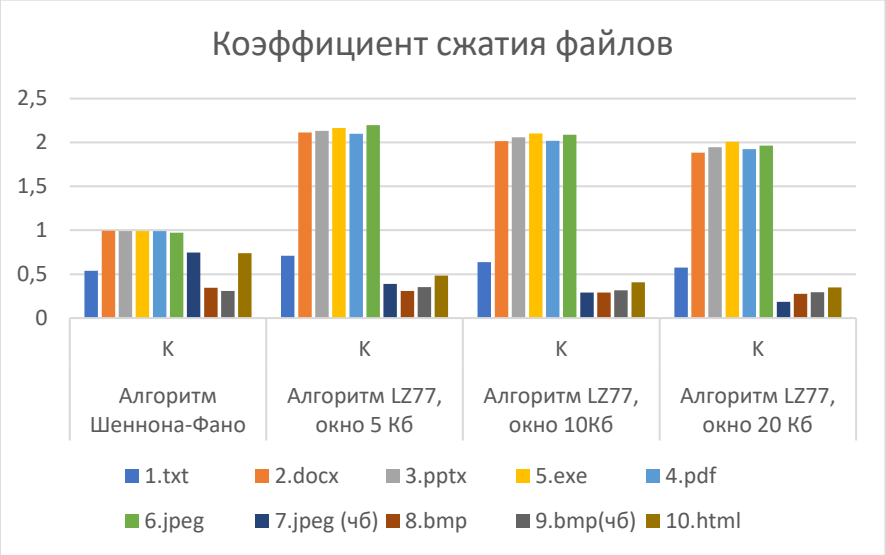


Диаграмма 11. Коэффициент сжатия файла относительно алгоритма

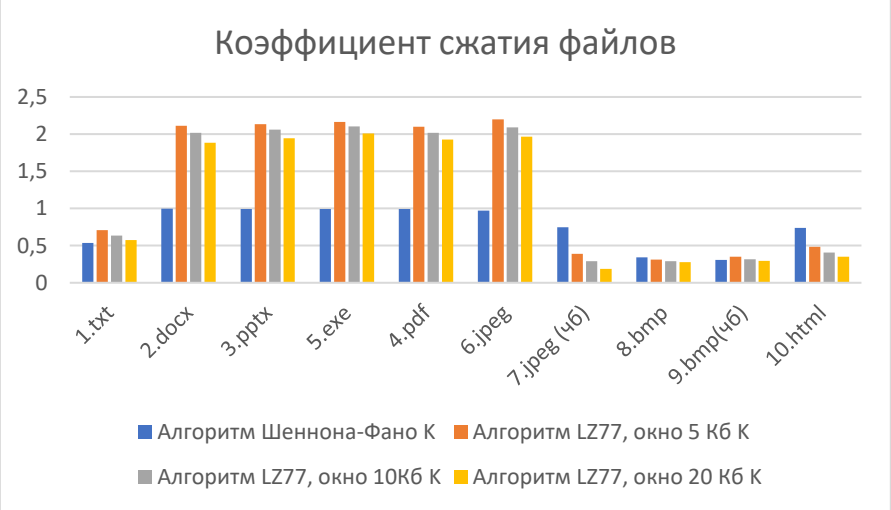


Диаграмма 12. Коэффициент сжатия алгоритмов относительно файлов

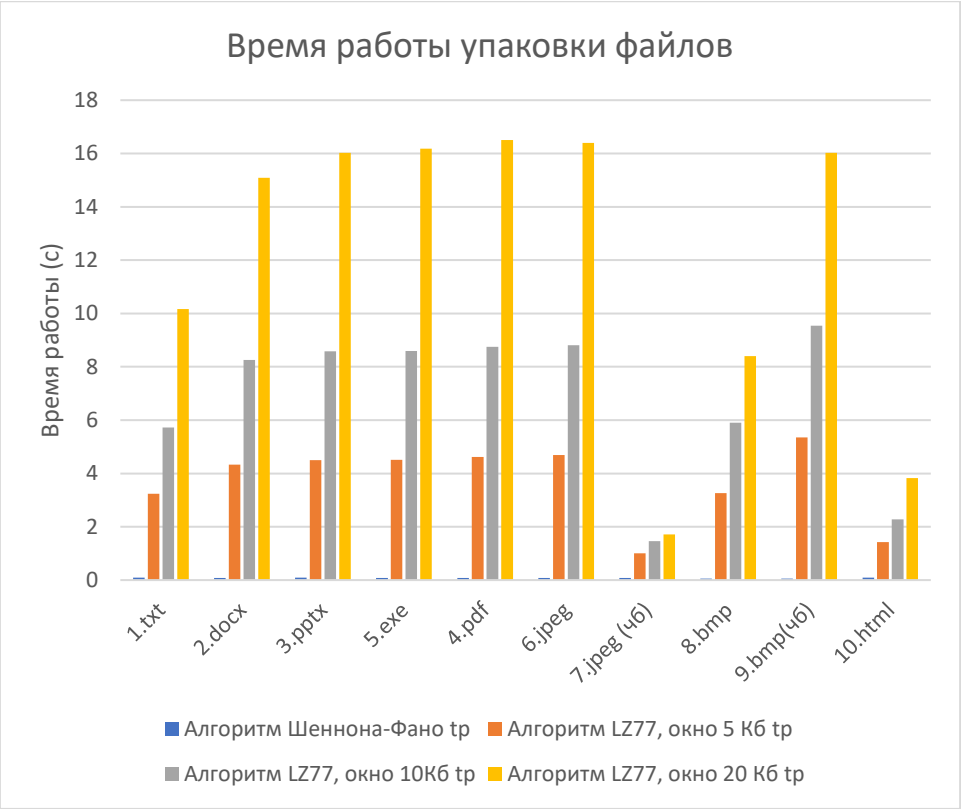


Диаграмма 13. Время работы упаковки файлов относительно алгоритмов (в секундах)

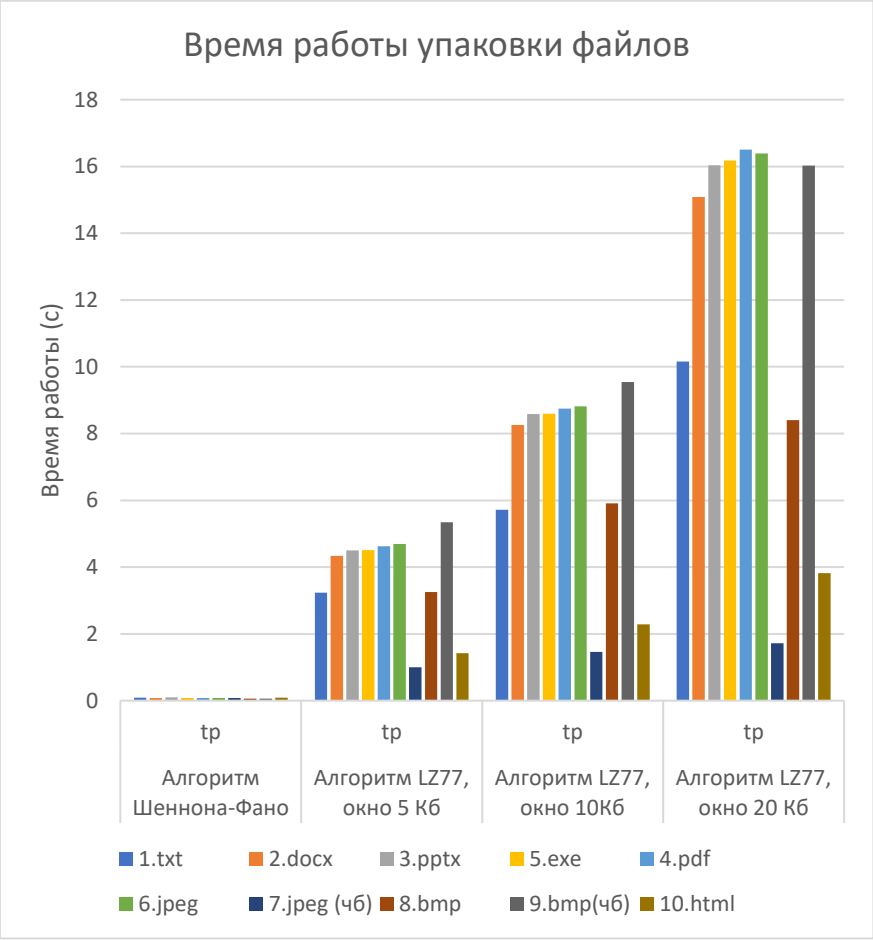


Диаграмма 14. Время работы упаковки файла относительно алгоритма (в секундах)

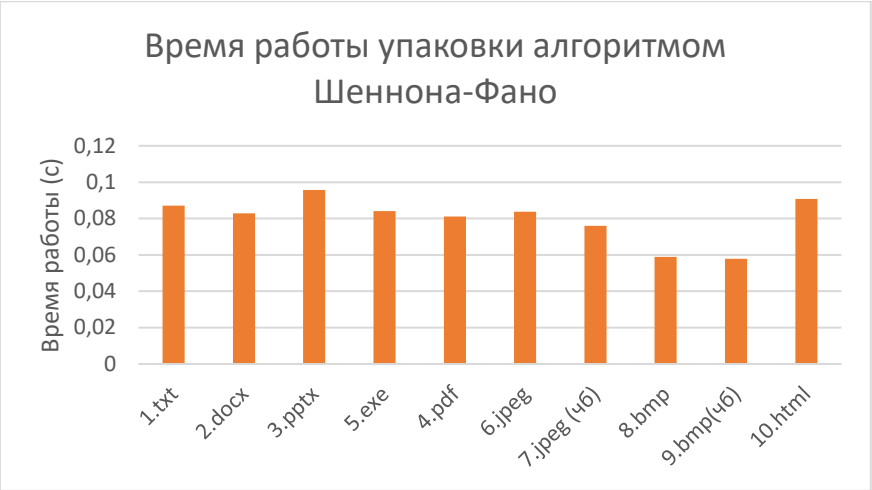


Диаграмма 15.

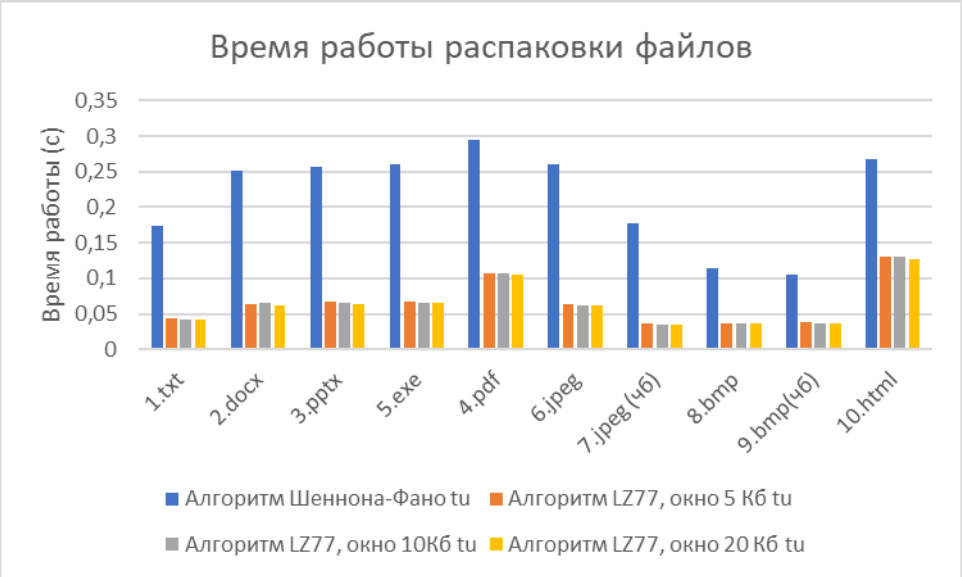


Диаграмма 16. Время работы распаковки файлов относительно алгоритмов

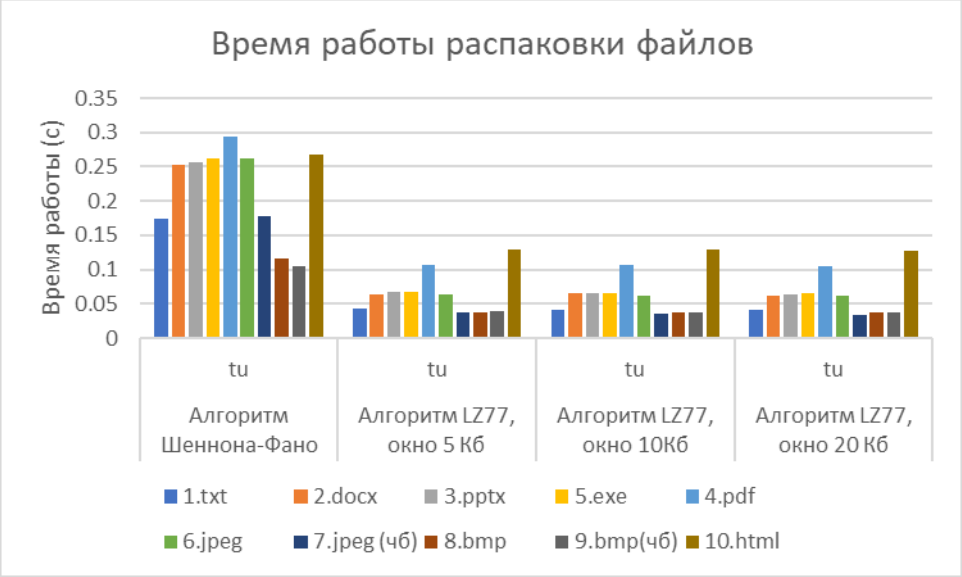


Диаграмма 17. Время работы распаковки алгоритмом относительно файла

9. Сравнительный анализ алгоритмов

Сравним средние показатели алгоритмов:

	Алгоритм Шеннона-Фано	Алгоритм LZ77, окно 5 Кб	Алгоритм LZ77, окно 10Кб	Алгоритм LZ77, окно 20 Кб
Средний коэффициент сжатия	0.761524	1.29523	1.2227626	1.140882
Среднее время упаковки	79.81	3692.42	6790.61	12031.84
Среднее время распаковки	216.35	65.47	64.58	63.53

Отсюда видно, что в среднем для данных файлов эффективнее всего работает упаковка алгоритмом Шеннона-Фано и распаковка алгоритмом LZ77 с окном 20КБ.

Теперь разберемся в причинах, посмотрев на результаты для файлов.

Для каждого файла рассмотрим лучшие для него характеристики:

имя файла	Лучший коэффициент сжатия	Лучшее время упаковки	Лучшее время распаковки
1.txt	Shannon-Fano	Shannon-Fano	LZ77_20_16
2.docx	Shannon-Fano	Shannon-Fano	LZ77_20_16
3.pptx	Shannon-Fano	Shannon-Fano	LZ77_20_16
5.exe	Shannon-Fano	Shannon-Fano	LZ77_20_16
4.pdf	Shannon-Fano	Shannon-Fano	LZ77_20_16
6.jpeg	Shannon-Fano	Shannon-Fano	LZ77_20_16
7.jpeg (чб)	LZ77_20_16	Shannon-Fano	LZ77_20_16
8.bmp	LZ77_20_16	Shannon-Fano	LZ77_20_16
9.bmp(чб)	LZ77_20_16	Shannon-Fano	LZ77_20_16
10.html	LZ77_20_16	Shannon-Fano	LZ77_20_16

Где Shannon-Fano – использование алгоритма сжатия Шеннона-Фано,
LZ_20_16 -– использование алгоритма сжатия LZ77 с размером окна 20КБ.

Результаты ожидаемые, если учитывать строение файлов:

В последних четырех файлах содержание такое, что есть много повторяющихся фрагментов (например, тэги в .html или белые фрагменты в черно-белых изображениях). Как следствие, в них сильно больше количество повторяющихся подстрок, которые эффективнее записывать с помощью кодовых троек в LZ77. При этом LZ77 с буфером 20КБ эффективнее всего, так как есть возможность кодировать равные подстроки на большем расстоянии, чем при LZ77 и меньшим размером буфера. **Поэтому алгоритм LZ77 по**

коэффициенту сжатия эффективнее для файлов с повторяющимся содержанием. Также по таблице 2 видна зависимость между энтропией файла и коэффициентом сжатия – чем энтропия ниже, тем меньше коэффициент сжатия. Это связано с тем, что у файлов с небольшой энтропией некоторые символы встречаются чаще других, так что и вероятность образования одинаковых подстрок выше.

В силу реализации упаковка алгоритмом Шеннона-Фано работает быстрее LZ77, поскольку LZ77 вынужден для нахождения очередной кодовой тройки просматривать весь буфер, размер которого может достигать 36КБ.

Однако **декодирование LZ77 происходит быстрее всего**, поскольку время распаковки зависит от длины исходного файла и количества кодовых троек (поэтому LZ77 с буфером 20Кб эффективнее всего – там наименьшее количество кодовых троек), тогда как алгоритму Шеннона-Фано изначально требуется время на построение кодов из таблицы частот символов, а затем на поиск кодов в закодированной последовательности бит.

Также интересный момент связан с коэффициентами сжатия файлов.

Сначала рассмотрим коэффициенты сжатия для алгоритма LZ77. Поскольку может быть важно не время упаковки, а время распаковки файла (а лучшее время распаковки у алгоритма LZ77_20), то важно понимать, когда LZ77 применим.

Поскольку кодовая тройка кодируется в 5 байт, то в применении LZ77 есть смысл, когда в этой кодовой тройке найденная length > 5 символов. Посмотрим, какова доля кодовых троек с length > 5 в исходных файлах и сопоставим с полученными коэффициентами сжатия:

	1.txt	2.docx	3.pptx	4.pdf	5.exe	6.jpg	7.jpg(чб)	8.bmp	9.bmp(чб)	10.html
Доля "хороших" кодовых троек	0.6589	0.0166	0.0095	0.0188	0.0123	0.0081	0.1392	0.1842	0.2836	0.4937
Коэффициент сжатия	0.5737	1.8853	1.9459	2.0074	1.9250	1.9650	0.1861	0.2756	0.2946	0.3501

Так, для большинства файлов эффективнее всего было бы оставить подстроки, закодированные кодовыми тройками длиной меньше либо равной пяти в исходном виде, либо применить к ним кодирование по Хаффману/Шеннону-Фано, как это реализовано в современных архиваторах. Однако для форматов, в которых хранится преимущественно текст (как .txt или .html и другие популярные форматы для передачи данных) или изображений с небольшой палитрой LZ77 эффективен.

Если же посмотреть на коэффициенты сжатия для алгоритма Шеннона-Фано, можно заметить, что чем меньше энтропия, тем лучше коэффициент сжатия.

Это ожидаемые результаты, поскольку длины кодов напрямую зависят от энтропии файла: чем выше энтропия, тем все более равными становятся коды по длине. При наибольшей энтропии каждый символ размером в 1 байт записывается с помощью его кода, который тоже занимает 1 байт, так как его длина достигает 8.

			Алгоритм Шеннона-Фано	
имя файла	S1(Кб)	H	S2(Кб)	K
1.txt	814	0.531254	438	0.537335
2.docx	783	0.987458	779	0.994922
3.pptx	806	0.983727	800	0.992162
5.exe	797	0.965814	791	0.991977
4.pdf	823	0.982873	815	0.990729
6.jpeg	804	0.963843	781	0.971475
7.jpeg (чб)	795	0.73968	593	0.746062
8.bmp	819	0.324856	282	0.344063
9.bmp(чб)	825	0.274782	255	0.308233
10.html	818	0.731701	604	0.738282

10. Заключение

Алгоритм Шеннона-Фано всегда сжимает с коэффициентом <= 1, тогда как упакованный файл для LZ77 может получиться большего размера, чем исходный, если в нём мала вероятность большого количества повторяющихся подстрок, как, например, для файлов .exe или .jpg.

Файлы с небольшой энтропией и алгоритм Шеннона-Фано, и LZ77 сжимают с достаточно небольшим коэффициентом. Однако время упаковки меньше для алгоритма Шеннона-Фано, а время распаковки – для LZ77/ В зависимости от того, время работы какого действия существеннее, можно выбирать соответствующий алгоритм.

Среди разных настроек алгоритма LZ77 эффективнее всего вариант с наибольшим размером буферов, поскольку во всех трёх вариантах на кодовую тройку выделяется равное количество байт, но при большем размере буфера общее количество кодовых троек наименьшее.

Файлы с большой энтропией сжимать практически бессмысленно любым из алгоритмов.

11. Использованные источники

1. cppreference.com – приведение reinterpret_cast;
2. cppreference.com – std::bitset;
3. cplusplus.com – fstream;
4. cppreference.com – функции для работы с датой и временем, библиотека chrono;
5. lms.hse.ru.