Федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

###### Факультет компьютерных наук

**КОНТРОЛЬНАЯ ДОМАШНЯЯ РАБОТА**

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

по направлению подготовки Алгоритмы и структуры данных

образовательная программа «Программная инженерия»

|  |  |
| --- | --- |
|  | Выполнил: |
|  | Студент группы БПИ183 |
|  | Имамов Радмир Маратович |
|  |
|  |

Москва 2018

Оглавление

[**1.** **Краткая постановка задачи.** 3](#_Toc37095901)

[**2.** **Описание алгоритмов и использованных структур данных.** 4](#_Toc37095902)

[**3.** **Описание реализации алгоритмов.** 5](#_Toc37095903)

[**4.** **План эксперимента.** 6](#_Toc37095904)

[**5.** **Описание программной реализации эксперимента.** 7](#_Toc37095905)

[**6.** **Диаграмма классов.** 8](#_Toc37095906)

[**7.** **Аппаратные средства.** 8](#_Toc37095907)

[**8.** **Результаты эксперимента.** 9](#_Toc37095908)

[**Таблицы:** 9](#_Toc37095909)

[**Графики:** 10](#_Toc37095910)

[**9.** **Сравнение алгоритмов.** 14](#_Toc37095911)

[**10.** **Заключение.** 15](#_Toc37095912)

1. **Краткая постановка задачи.**

Разработать на языке С++ программу, реализующую алгоритмы сжатия без потерь, получение архивного файла из исходного и разархивированного файла из архивированного.

1. Выполнен алгоритм Шеннона-Фано.
2. Проведен вычислительный эксперимент для исследования эффективности реализованных алгоритмов сжатия без потерь для файлов разного типа.
3. Вычислены: энтропия, коэффициент сжатия, время упаковки, время распаковки по 10 раз для каждого файла. Для измерения времени упаковки и распаковки использовались такты ЦП.
4. Подготовлен отчет по итогам работы, включающий таблицы и графики, описание алгоритмов и структур данных, план эксперимента, описание реализации, диаграмму классов, анализ результатов и заключение.
5. Не выполнен алгоритм LZ77.

1. **Описание алгоритмов и использованных структур данных.**

В данной работе использовались следующие структуры данных:

* **multimap** – отсортированный словарь
* **map** – стандартный словарь
* **vector** – реализация динамического массива

В работе использовался алгоритм сжатия данных Шеннона-Фано.

В программе использовались файлы приведенные в архиве формата:

cmake-build-debug/DATA:

1.txt, 2.docx, 3.ppt, 4.pdf, 5.exe, 6.jpg(color), 7.jpg(bw), 8.bmp(color), 9.bmp(bw)

cmake-build-debug/SHAN-заархивированные файлы с раcширением .shan

cmake-build-debug/UOUT – разархивированные файлы с расширением .unshan

1. **Описание реализации алгоритмов.**

Описание сжатия алгоритмом Шеннона-Фано:

* В сжатом файле в начале записан словарь, после записаны n битов, при n<8, тк не был найдет другой способ как их запомнить. Последняя часть - это зашифрованные с помощью словаря, байтовый код файла.
* Упаковка происходит в 3 этапа
  1. Зашифровывается словарь Шеннона-Фано сначала байт закодированного символа, после побитово код символа, предварительно записан его размер
  2. После запоминается позиция каретки и проходится файл до конца, чтобы найти те n (n<8) битов и записываются.
  3. Каретка перемещается на заранее запомненную позицию и шифруется весь файл
* Распаковка проходит в 2 этапа
  1. Читается размер словаря, после читается словарь.
  2. После читается условный хвост, те самые n (n<8) битов
  3. Читаем байты и расшифровываем их в последовательность из 8 битов и записываем их в переменную, пытаясь расшифровать их.
  4. После дописываем хвост и расшифровываем эту строку обратно в символы.
* После проверяются изначальный и расшифрованный файл на идентичность.

1. **План эксперимента.**
   1. Был написан алгоритм Шеннона-Фано
   2. Написан алгоритм шифрования и записи шифра в файл
   3. После алгоритм дешифрования и записи дешифрованного кода в новый
   4. Был произведен замер времени шифрования и дешифрования, чтобы измерения были репрезентативными они были произведены по 10 раз для каждого файла, также высчитан коэффициент сжатия файла
   5. Был написан алгоритм прочтения символов и рассчитана вероятность встречи каждого символа и энтропия
   6. Написан отчет)
2. **Описание программной реализации эксперимента.**

Работа не автоматизирована. Используется метод проверки совпадения двух файлов.

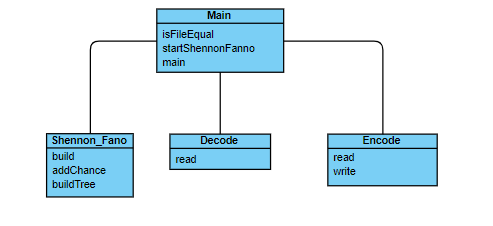
Программа начинает работу с метода startShennonFanno. После читается изначальный файл записывается массив с частотой появления символов.

После с использованием сортированного словаря данные подаются в алгоритм Шеннона-Фано, после чего записываются в файл с помощью приведенный выше алгоритмов.

После архивированный файл читается и возвращается с первоначальный вид.

Попутно происходят замеры времени и другие сопутствующие дейсвия.

1. **Диаграмма классов.**



1. **Аппаратные средства.**

Windows 10

Процессор: Intel(R) Core(TM) u5-7300HW CPU @ 2.50GHz 2.50GHz

ОЗУ: 6.00 Гб

Тип системы: 64-разрядная

1. **Результаты эксперимента.**

**Таблицы:**

Время упаковки и распаковки файлов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя Файла | S1 | H | Алгоритм Шеннона-Фано | |
| tu | tp |
| 1 | 1521736 | 4.40616 | 1,14E+10 | 4,45E+10 |
| 2 | 12.401337 | 12.4013 | 1,17E+10 | 7,62E+10 |
| 3 | 1496320 | 18.4069 | 1,23E+10 | 6,83E+10 |
| 4 | 1931029 | 26.3387 | 1,87E+10 | 1,15E+11 |
| 5 | 1347360 | 34.3334 | 1,3E+10 | 8,26E+10 |
| 6 | 1836108 | 42.2807 | 1,77E+10 | 1,12E+11 |
| 7 | 466626 | 44.2832 | 3,17E+09 | 8,86E+09 |
| 8 | 2211853 | 49.9909 | 1,8E+10 | 8,99E+10 |
| 9 | 983053 | 55.8874 | 8,11E+09 | 3,99E+10 |

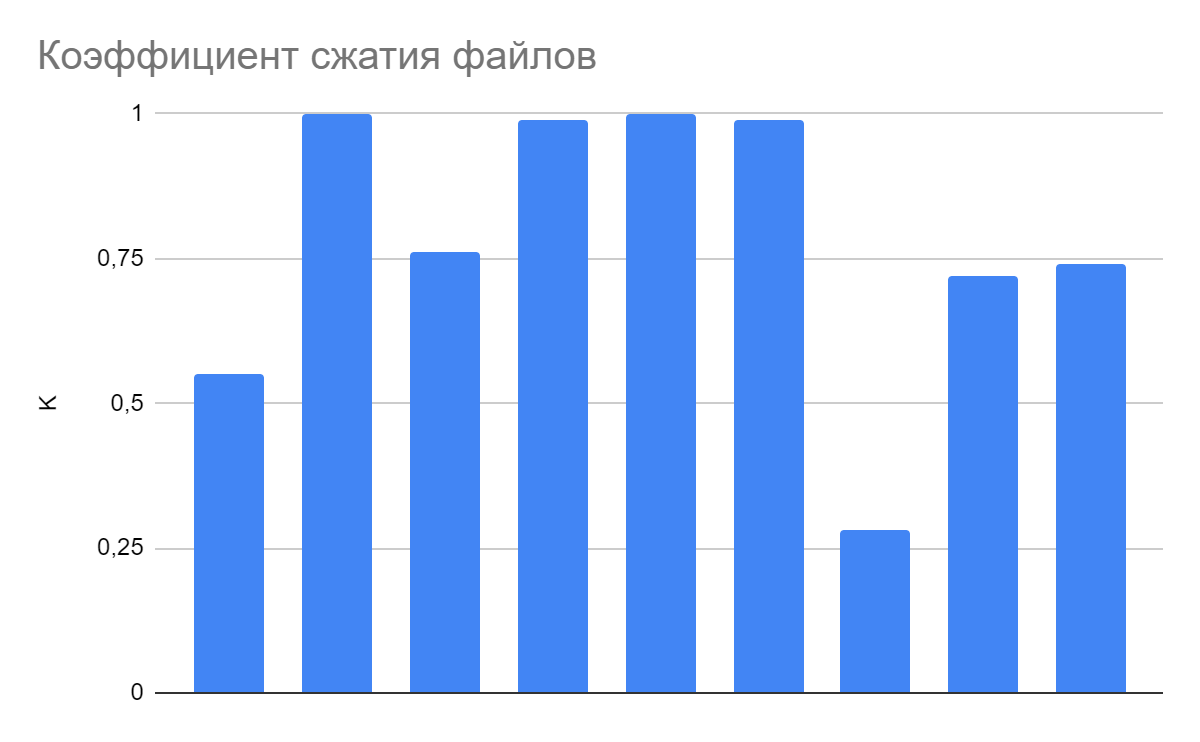
Коэффициент сжатия файлов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя Файла | S1 | H | Алгоритм Шеннона-Фано | |
| S2 | K |
| 1 | 15217 | 4.40616 | 8444 | 0.554906 |
| 2 | 12031 | 12.4013 | 12065 | 1.002826 |
| 3 | 14963 | 18.4069 | 11368 | 0.759741 |
| 4 | 19310 | 26.3387 | 19274 | 0.998136 |
| 5 | 13473 | 34.3334 | 13496 | 1.001707 |
| 6 | 18361 | 42.2807 | 18358 | 0.999837 |
| 7 | 4666 | 44.2832 | 1315 | 0.281826 |
| 8 | 22118 | 49.9909 | 15890 | 0.718419 |
| 9 | 9830 | 55.8874 | 7302 | 0.742828 |

Таблица частота появления символов в файле представлена в виде csv таблицы в документах эксперимента.

## **Графики:**

Все графики показывают работу алгоритма Шеннона-Фано.

****

1. **Сравнение алгоритмов.**

Формально был выполнен толь Шеннон-Фано, но во время эксперимента была попытка реализации LZ77, поэтому некоторые заключения можно утвердить. Отслеживается зависимость чем дольше мы кодируем файл, тем меньше коэффициент сжатия, те Шеннон-Фано архивирует файл не так долго, но сжимает файлы не сильно. LZ77 работает существенно дольше при большом буфере), но сжимает файлы часто в разы сильнее. Также чем больше буфер LZ77 тем дольше он работает, и тем сильнее он сжимает файлы.

1. **Заключение.**

Главным выводом из данного эксперимента является зависимость между временем на сжатие и коэффициентом сжатия. Также все представленные алгоритмы сжимают файл лучше, если там много повторяющийся символов. Поэтому лучше всего сжимают черно-белые картинки jpg, а еще лучше картинки bmp. Также хорошо сжимаются txt файлы в связи частой встречаемостью в них символов. Хуже всего сжимаются exe и pdf файлы в связи с одинаковой частотой встречи символов.