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 |
|  | Мануйлов Александр Андреевич |
|  |
|  |

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.** **Использованные аппаратные средства.** 8](#_Toc37095907)

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

[**Таблицы:** 9](#_Toc37095909)

[**Графики:** 9](#_Toc37095910)

[**7.** **Заключение.** 14](#_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 (упаковка файла и распаковка архива).

1. Выполнен алгоритм Шеннона-Фано.
2. Проведен вычислительный эксперимент для исследования эффективности реализованных алгоритмов сжатия без потерь (упаковка файла и распаковка) для файлов разного типа.
3. Вычислены: энтропия, коэффициент сжатия, время упаковки, время распаковки по для каждого файла. Для измерения времени упаковки и распаковки использовались такты ЦП, измерение производилось в наносекундах.
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 реализован алгоритм LZ77.

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

В данной работе использовались следующие структуры данных:

* **unordered\_map** – неотсортированный словарь
* **map** – словарь
* **vector** – динамический массив
* **LARGE\_INTEGER** – переменная для хранения количества тактов процессора

В данной работе использовались следующие структуры данных:

* **Алгоритм сжатия Шеннона-Фано**

1. **Описание реализации алгоритмов.**

Описание сжатия алгоритмом Шеннона-Фано

Кодирование

* Создаем в динамической памяти экземпляр класса ShannonFano (shf) предоставляющий функционал для кодирования и раскодирования файлов
* Считываем побайтово исходный файл (1 char = 1 байт) и передаем байты в shf, где они учитываются в unordered\_map (в среднем обращение за константу) – считается их количество.
* Вызываем метод у shf для вычисления кодировок байтов
  1. Создаем вектор с частотами и сортируем его
  2. На основании этого вектора стоим второй вектор с кодировками байтов (как задача на контесте)
     + Делим массив частот примерно по медиане, одной половине приписываем 0, другой 1
     + Пока не дойдем до длины в 1 или 0 элемент вызываем рекурсивно для двух частей
  3. Создаем мапу <байт : кодировка>, ища совпадения по частотам в мапе с частотами, массивом частот и массивом кодировок
  4. Создаем обратную мапу <кодировка : байт>
* Идем по исходному файлу и создаем строку, заполняя ее кодировками, в соответствии с байтами
* Записываем закодированный файл, заменяя в строке кодировки каждые 8 бит на char
* Оставшиеся биты запоминаем

Декодирование

* Читаем побайтово закодированный файл, сразу генерируя строку кодировки, заменяя байт на его двоичный код
* Вызываем у shf метод раскодирования (можно было записать словарь в отдельный файл, или в файл с закодированным сообщением)
  1. Идем по строке кодировки и смотрим, увеличивая на 1 бит, когда найдем совпадение с имеющейся кодировкий байта. (так как код Фано префиксный, то декодирование однозначное)
  2. Найдя, записываем раскодированный байт в файл и смещаемся по строке вперед
  3. Добавляем в конце запомненный «хвост»

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
   5. Рассчитан коэффициент сжатия для каждого файла
   6. Рассчитана энтропия для каждого файла
   7. Посчитаны частоты каждого байта в файле
   8. Написан отчет
2. **Аппаратные средства.**

Windows 10

Процессор: Intel(R) Pentium(R) CPU N3710 @ 1.60GHz 1.60GHz

ОЗУ: 4.00 Гб

Тип системы: 64-разрядная

Процессор: 64-разрядный

1. **Результаты эксперимента.**

Исходные файлы:

1.txt

2.pdf

3.docx

4.exe

5.bin

6.pptx

7.bmp (черно-белое)

8.jpg (Черно-белое)

9.bmp

10.jpg

Размер всех файлов подобран примерно мегабайт

**Таблицы:**

Время упаковки и распаковки файлов (время в наносекундах, размер файлов в Кб)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя Файла | S1 | H | Алгоритм Шеннона-Фано | |
| tu | tp |
| 1 | 1022 | 4.32327 | 7928831200 | 1818815000 |
| 2 | 1051 | 7.28737 | 12951708800 | 2328183100 |
| 3 | 1014 | 7.99865 | 18535623100 | 2632591100 |
| 4 | 987 | 5.87762 | 9680759800 | 2005637100 |
| 5 | 1026 | 7.98625 | 13602772700 | 2461492200 |
| 6 | 1005 | 7.94575 | 13186989600 | 2409320500 |
| 7 | 1013 | 7.57499 | 12645479000 | 2306992200 |
| 8 | 1195 | 7.98723 | 15720913100 | 2820532200 |
| 9 | 1009 | 7.61529 | 12725260300 | 2286223800 |
| 10 | 1006 | 7.93664 | 13207684300 | 2372068100 |

Коэффициент сжатия файлов (размер в Кб)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя Файла | S1 | H | Алгоритм Шеннона-Фано | |
| S2 | K |
| 1 | 1022 | 4.32327 | 559 | 0.5469 |
| 2 | 1051 | 7.28737 | 968 | 0.921 |
| 3 | 1014 | 7.99865 | 1015 | 1.0009 |
| 4 | 987 | 5.87762 | 734 | 0.743 |
| 5 | 1026 | 7.98625 | 1025 | 0.999 |
| 6 | 1005 | 7.94575 | 1004 | 0.999 |
| 7 | 1013 | 7.57499 | 966 | 0.953 |
| 8 | 1195 | 7.98723 | 1197 | 1.001 |
| 9 | 1009 | 7.61529 | 968 | 0.959 |
| 10 | 1006 | 7.93664 | 1005 | 0.999 |

Вместо таблицы частот появления символов в файлах имеется all\_.csv

## **Графики:**

Все графики показывают работу алгоритма Шеннона-Фано.

1. **Заключение.**

Чем больше в файле повторяющихся символов, тем эффективнее он сжимается (меньше коэффициент сжатия) Алгоритм Шеннона-Фано Не всегда приводит к ожидаемому результату. Иногда размер заархивированного файла превышает размер исходного. Основная сложность в работе архиватора не алгоритм сжатия, а корректная обработка информации (ввод, вывод без потери символов) также немаловажно для эффективности алгоритма использовать верные типы данных.