ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Отчёт к контрольному домашнему заданию**

**По дисциплине**

**«Алгоритмы и структуры данных»**

Работу выполнила

Студентка группы БПИ185

Фролова Е.Д.

Оглавление

[Постановка задачи 3](#_Toc37138453)

[Описание алгоритмов и использованных структур данных 4](#_Toc37138454)

[Описание реализации алгоритмов 4](#_Toc37138455)

[План эксперимента 5](#_Toc37138456)

[Описание программной реализации эксперимента 5](#_Toc37138457)

[Диаграмма классов 5](#_Toc37138458)

[Использованные аппаратные средства для проведения эксперимента 5](#_Toc37138459)

[Результаты экспериментов - таблицы и графики 5](#_Toc37138460)

[Сравнительный анализ алгоритмов по эффективности сжатия файлов разных типов, по скорости работы 5](#_Toc37138461)

[Заключение 6](#_Toc37138462)

# Постановка задачи

1. Разработать на языке C++ программу, реализующую алгоритмы сжатия данных без потерь, получение архивного файла из исходного и разархивированного файла из архивного (упаковка файла и распаковка архива)
   * + 1. алгоритм Хаффмана (простой) - студенты, у которых логин заканчивается на четное число. ***(done)***
       2. алгоритм Лемпеля-Зива LZ77 (со скользящим окном), ***(done)***
       3. 3. алгоритм Лемпеля-Зива-Велча LZW\* ***(not done)***  
          \*(Бонус-задача. Включение данного алгоритма в исследование не обязательно. Его реализация и включение в состав экспериментального исследования позволяет получить дополнительные баллы)
2. Провести вычислительный эксперимент для исследования эффективности реализованных алгоритмов сжатия без потерь (упаковка и распаковка) для файлов разного типа.  
    Для проведения эксперимента с алгоритмами сжатия без потерь необходимо использовать набор из N файлов различных типов с именами 1.\*...N.\*.   
   Мы должны самостоятельно подобрать исходные файлы примерно одного размера, желательно не менее 12000 Кб. В наборе должны присутствовать файлы:
   * текстовый .txt ***(done)***
   * документ Word .docx ***(done)***
   * презентация .pptx ***(done)***
   * документ .pdf ***(done)***
   * исполняемый файл .exe или библиотечный .dll ***(done)***
   * цветное изображение .jpg ***(done)***
   * изображение черно-белое или в градациях серого .jpg ***(done)***
   * цветное изображение .bmp ***(done)***
   * изображение черно-белое или в градациях серого .bmp ***(done)***
   * файлы других форматов. ***(done, 10.mp3)***

Вычислить:

* энтропию исходных файлов; ***(done)***
* коэффициент сжатия как отношение размера сжатого файла к размеру исходного файла. ***(done)***
* Измерить для каждого файла и для каждого алгоритма:
* время упаковки; ***(done)***
* время распаковки. ***(done)***

1. Подготовить отчет по итогам работы. В отчете необходимо **явно** указать, какие части задания были сделаны, а какие нет. ***(done)***
2. Результаты работы надо загрузить в ЛМС. ***(almost done)***

# Описание алгоритмов и использованных структур данных

В программе использовался алгоритм сжатия данных Шеннона-Фано и алгоритм сжатия данных LZ77 (с различными вариациями размеров скользящего окна и окна предыстории).

Все использованные СД являются частью STL.

# Описание реализации алгоритмов

**Форматы имен файлов:**

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

Алгоритм LZ77:

Данные для алгоритма считывались в строку байтов, после чего строились кодирующие «тройки» (более подробное описание действий приведено в комментариях к коду), которые записывались в вектор и выводились архивированным файлом.  
Дополнительное сжатие «троек» не использовалось.  
Распаковка происходила сначала считыванием всех «троек», после чего применялся алгоритм декодирования, получающий закодированные байты, которые в последствие выводились через поток байтов.

Алгоритм Шеннона-Фано:

На основе считанных данных строится таблица частот файлов, после чего она сортируется по невозрастанию. По таблице ищется индекс медианы. Все элементы, индекс которых в таблице невозрастания меньше индекса медианы, получают код 0, остальные код 1. Операция поиска медианы-получения кода повторяется для каждой из частей «до» медианы и «после» до тех пор, пока не будут построены все коды.  
В файл записывается количество дозаписанных «1» (для удобства кодировки), с новой строки число N – количество закодированных символов.  
Далее N «пар» вида *symb***code,** где *symb* - символ, **code** – код этого символа.При распаковке использовались полученные пары, хранящиеся в словаре, после чего подбирался код для декодирования.Алгоритм включал в себя использование std::bitset<8> для упрощённого кодирования/декодирования байтов.

# План эксперимента

Реализация алгоритмов Шеннона-Фано и LZ77;

Многократная обработка файлов различных форматов, примерно одинакового размера (1500KB)

Анализ работы алгоритма

# Описание программной реализации эксперимента

Эксперимент был реализован с использованием языка программирования C++ 17 и среды разработки CLion 2019.3.5  
Для подсчёта частот встречаемости байтов был использован скрипт на языке программирования Python3.7 с использованием среды разработки JetBrains Pycharm.

# Диаграмма классов

Реализованы 2 класса:

* ShannonFano.cpp
* LZ77.cpp

# Использованные аппаратные средства для проведения эксперимента

Ноутбук Xiaomi Mi Notebook Pro 15.6" GTX (Intel Core i5 8250U 1600 MHz/1920x1080/8Gb/256Gb SSD/GeForce MX150 /Win10 Home RUS) серый

# Результаты экспериментов - таблицы и графики

Приведены в файле «Таблицы и графики (Фролова, 185).xlsx»

# Сравнительный анализ алгоритмов по эффективности сжатия файлов разных типов, по скорости работы

Алгоритм Шеннона-Фано лучше проявляет себя как по времени, так и по коэффициенту сжатия. Алгоритм LZ77 эффективен только в распаковке. В остальном же сжимаемые файлы имеют больший объём, времени затрачивается в несколько раз больше, алгоритм работает несколько эффективнее (себя же) на менее энтропичных файлах и с большим размером окна.

# Заключение

Алгоритм LZ77 проявляет себя плохо из-за большого количества «троек». Была попытка сжатия файла .lzz775 методом Шеннона-Фано, что уменьшило файл до приемлемого размера, но полученный размер был не только больше файла .shan, но и время его упаковки было большим. Данный факт говорит о том, что нужно либо искать способ оптимального хранения «троек», либо использовать алгоритм Шеннона-Фано, который тоже не так хорош при сжатии презентации (pptx), файла типа word (docx), mp3 файла или цветного изображения (jpg).

# Источники

<https://ru.wikipedia.org/wiki/LZ77>

<https://ru.qwe.wiki/wiki/LZ77_and_LZ78>

<https://studfile.net/preview/6390380/page:9/>

<http://www.cplusplus.com/reference/bitset/bitset/>

<https://www.geeksforgeeks.org/c-bitset-interesting-facts/?ref=rp>

<https://stackoverflow.com/questions/7897050/adding-to-a-vector-of-pair>

<http://compression.ru/download/articles/huff/tiger_shannon-fano.html>

<https://stackoverflow.com/questions/7349689/how-to-print-using-cout-a-number-in-binary-form/7349767>

<https://stackoverflow.com/questions/21433194/c-read-part-of-a-binary-file-into-a-bitset>