## НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет Компьютерных Наук

Департамент Программной Инженерии

Контрольное Домашнее Задание

по дисциплине «Алгоритмы и Структуры Данных»

ОТЧЕТ

Выполнил: Студент 2 курса группы БПИ151 Куприрянов Кирилл Игорвеич

Москва 2016

Сол	ержани	ſе
$\sim$ $\sim$	C P211CH111.	

Постановка задачи	3
-------------------	---

## 1 Постановка задачи

Необходимо было реализовать с использованием языка C++ программы для архивирования и разархивирования текстовых файлов. При этом использовать два известных алгоритма кодирования информации:

- 1. Хаффмана (не адаптивный, простой)
- 2. Шеннона-Фано

Обе реализации поместить в одном файле main.cpp, содержащем соответствующие методы:

- 1. метод архивирования, использующий алгоритм Хаффмана, вход: текстовый файл <name>.txt (кодировка UTF-8) выход: архивированный файл <name>.haff
- 2. метод разархивирования, использующий алгоритм Хаффмана, вход: архивированный файл <name>-unz-h.txt (кодировка UTF-8)
- 3. метод архивирования, использующий алгоритм Шеннона-Фано, вход: текстовый файл <name>.txt (кодировка UTF-8) выход: архивированный файл <name>.shan
- 4. метод разархивирования, использующий алгоритм Шеннона-Фано. вход: архивированный файл <name>-unz-s.txt (кодировка UTF-8)

Выбор алгоритма осуществляется с помощью флага командной строки.

Оба алгоритма работают в два прохода. Сначала строится таблица частот встречаемости символов в конкретном архивируемом файле (кодируем только те символы из набора допустимых, которые реально встречаются в файле). Затем строится кодовое дерево (не обязательное). По нему (или по таблице кодов) и архивируется файл. Для разархивирования алгоритмам потребуется знать таблицу, которая использовалась при архивировании. Соответствующая таблица должна сохраняться в архивном файле в самом его начале и использоваться при разархивировании. В начале пишется количество различных символов п, имеющихся в кодируемом файле, а затем п пар (код символа UTF-8, битовый код в архиве). Порядок — по убыванию частоты встречаемости символа в кодируемом файле.

Провести вычислительный эксперимент с целью оценки реализованных алгоритмов архивации / разархивации. Оценить количество элементарных операций каждого алгоритма. Для этого

- 1. Подготовить тестовый набор из нескольких текстовых файлов разного объема (20, 40, 60, 80, 100 Кб; 1, 2, 3 Мб всего 8 файлов) на разных языках (ru, en кодировка UTF-8) с разным набором символов в каждом файле, а именно:
  - (а) первый набор: символы латинского алфавита и пробел
  - (b) второй набор: символы из первого набора + символы русского алфавита
  - (c) третий набор: символы из второго набора + следующие знаки и спецсимволы: знаки арифметики "+ \* / =", знаки препинания ". , ; : ? !", "% @ # \$ '', скобки разных типов "( ) [ ] < >" , кавычки """"),
- 2. Измерить (экспериментально) количество операций (в рамках модели RAM (взять из лекционного материала)), выполняемых за время работы (архивирования, разархивирования) каждого алгоритма на нескольких различных (не менее трех) файлах для каждого размера входного файла и набора символов (итого получается  $8 \times 3 \times 3 = 72$  эксперимента по архивированию и 72 по разархивированию для каждого алгоритма,

т.е. Всего минимум  $144 \times 2 = 288$ ). Для повышения достоверности результатов каждый эксперимент можно повторить несколько (5-10) раз на различных файлах (с одним возможным набором символов) одного размера с последующим усреднением результата.

Подготовить отчет по итогам работы, содержащий постановку задачи, описание алгоритмов и задействованных структур данных, описание реализации, обобщенные результаты измерения эффективности алгоритмов, описание использованных инструментов (например, если использовались скрипты автоматизации), выводы о соответствии результатов экспериментальной проверки с теоретическими оценками эффективности исследуемых алгоритмов. Отчет также должен содержать измерения качества архивации (степень сжатия = отношение размеров выходного и входного файлов), оценку связи между степенью сжатия для различных входных файлов (как влияют объем, язык, набор символов, их разнообразие?) и временем работы (количеством операций) для каждого алгоритма.