МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО

ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Математический факультет

Кафедра математического моделирования

Методы сжатия информации и создание резервных копий

Лабораторная работа №1

Направление 10.05.04 Информационно-аналитические системы безопасности

Обучающийся: Ручкин Р. А.

Преподаватель: Завгородний М. Г.

Воронеж 2024

**Задания.**

1. Изучить работу программ архивирования и создания архивных копий.

Алгоритм решения:

а) установить программы-архиваторы WinRAR и WinZip

б) запустить их и рассмотреть все функции, которые они способны выполнять, посредством изучения действия всех предоставленных иконок действий и пунктов выпадающих меню

в) прочесть дополнительную информацию на официальных сайтах данных программ

г) прочесть информацию о встроенной в Windows утилите создания резервной копии

2. Изучить один из методов сжатия информации.

Алгоритм решения:

а) задать в поисковом окне браузера запрос по типу «методы сжатия информации» и изучить материалы какого-либо сайта

б) выбрать один из алгоритмов и более подробно изучить его, посредством поиска дополнительной информации сети Интернет

3. Создать алгоритм и написать программу, реализующую изученный Вами метод сжатия информации.

4. Создать алгоритм и написать программу, выполняющую декомпрессию.

Эти 2 программы объединены в одну.

Ниже приведен код программы на Python

def compress\_running(text):

    compressed\_text = ""

    i = 0

    while i < len(text):

        char = text[i]

        count = 1

        while i + count < len(text) and text[i + count] == char:

            count += 1

        if count > 2:

            compressed\_text += str(count) + char

        else:

            compressed\_text += char \* count

        i += count

    return compressed\_text

def decompress\_running(compressed\_text):

    decompressed\_text = ""

    i = 0

    while i < len(compressed\_text):

        if compressed\_text[i].isdigit():

            count\_str = ""

            while i < len(compressed\_text) and compressed\_text[i].isdigit():

                count\_str += compressed\_text[i]

                i += 1

            count = int(count\_str)

            char = compressed\_text[i]

            decompressed\_text += char \* count

            i += 1

        else:

            decompressed\_text += compressed\_text[i]

            i += 1

    return decompressed\_text

def compress\_file(input\_filename, output\_filename):

    with open(input\_filename, 'r') as file:

        text = file.read()

    compressed\_text = compress\_running(text)

    with open(output\_filename, 'w') as file:

        file.write(compressed\_text)

def decompress\_file(input\_filename, output\_filename):

    with open(input\_filename, 'r') as file:

        compressed\_text = file.read()

    decompressed\_text = decompress\_running(compressed\_text)

    with open(output\_filename, 'w') as file:

        file.write(decompressed\_text)

def main():

    input\_filename = "c:/Users/karny/OneDrive/Рабочий стол/study/ИБ/lab\_1/Pushkin.txt"

    output\_filename = "compressed.txt"

    decompressed\_output\_filename = "decompressed.txt"

    compress\_file(input\_filename, output\_filename)

    decompress\_file(output\_filename, decompressed\_output\_filename)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

5. Проверить работоспособность подготовленных Вами программ. Для этого выберите файл для сжатия. Преобразуйте его с помощью Вашей программы. Посмотрите на сколько изменился объем файла. Разархивируйте преобразованный файл и сравните его с исходным файлом.

Файлы, использованные для проверки работоспособности программы, прилагаются в электронном виде.

Опишем в текстовом виде проведенные проверки.

Проверка:

Файл с расширением txt.

Текст:

ddddddddddddhjgtktgggggggggggkkkkykullli

chhdrazaaaaaaaaaaxgggggggggggggggggbbbbbbbb

mmmmmmmmmmmhhhhhhhhhhffffffffffeeeeeeeeeee

(2,КБ КБ)

Результат компрессии:

12dhjgtkt11g4kyku3li

chhdraz10ax17g8b

11m10h10f11e

(52 байт)

Результат декомпрессии:

ddddddddddddhjgtktgggggggggggkkkkykullli

chhdrazaaaaaaaaaaxgggggggggggggggggbbbbbbbb

mmmmmmmmmmmhhhhhhhhhhffffffffffeeeeeeeeeee

(2,38 КБ) разницы с исходным файлом нет

**Выводы.**

В заключении отметим, что в ходе работы мы познакомились с такими программами, как WinZip и WinRAR, а также встроенной в Windows утилиты создания резервной копии системы, изучили алгоритм сжатия данных RLE и реализовали компрессию и декомпрессию файла с его помощью в программе, проверили ее работоспособность, оценили преимущества и недостатки как программы так и самого метода сжатия, назовем их:

Плюсы:

* сильное сжатие, экономящее большие объемы памяти при преобразовании файлов с большими последовательностями поочередно идущих одинаковых символов
* сжатие без потерь при кодировании непрерывных строк

Минусы:

* игнорирование цифр как знаков
* игнорирование пробелов
* 0 эффективность при 2 символах подряд
* отрицательная при отсутствии или малом количестве последовательностей одинаковых символов подряд