

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего

образования

**«МИРЭА - Российский технологический университет»**

РТУ МИРЭА

Институт информационных технологий

Кафедра математического обеспечения и стандартизации информационных технологий

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 8**

***«*Кодирование и сжатие данных методами без потерь»**

по дисциплине

**«**Структуры и алгоритмы обработки данных**»**

Выполнил студент группы *ИКБО-03-22 Хохлинов Д.И.*

Принял *Сорокин А.В.*

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | « » 2023 г. |  |
|  |  |  |
| «Зачтено» | « » 2023 г. |  |

Москва 2023

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc152791400)

[2 РАЗРАБОТКА ПРИЛОЖЕНИЯ 4](#_Toc152791401)

[2.1 Метод «грубой силы» и оценка количества переборов 4](#_Toc152791402)

[2.2 Метод динамического программирования 6](#_Toc152791403)

[2.3 Класс field 8](#_Toc152791404)

[2.4 Код программы 9](#_Toc152791405)

[2.5 Тестирование 13](#_Toc152791406)

[3 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc152791407)

# **1 ЗАДАНИЕ 1**

## 1.1 Постановка задачи

Сжать текст, используя метод RLE (run length ecoding/кодирование длин серий/групповое кодирование).

1) Описать процесс сжатия алгоритмом RLE.

2) Придумать текст, в котором есть длинные (в разумных пределах) серии из повторяющихся символов. Выполнить сжатие текста. Рассчитать коэффициент сжатия.

3) Придумать текст, в котором много неповторяющихся символов и между ними могут быть серии. Выполнить групповое сжатие, показать коэффициент сжатия. Применить алгоритм разделения текста при групповом кодировании, позволяющий повысить эффективность сжатия этого текста. Рассчитать коэффициент сжатия после применения алгоритма.

## 1.2 Описание алгоритма RLE

При сжатии алгоритмом RLE каждая группа из одинаковых символов (или «серия») заменяется на один символ и количество его повторений. Каждая серия обычно кодируется в два байта. Первый байт представляет количество символов в пробеге и называется счетчиком прогона. На практике кодированный прогон может включать от 1 до 128 или от 1 до 256 символов. Счетчик обычно содержит число символов минус один (значение в диапазоне значений от 0 до 127 или от 0 до 255). Второй байт — это значение символа в прогоне, которое содержится в диапазоне значений от 0 до 255 и именуется значением запуска.

## 1.3 Сжатие «удобного» текста алгоритмом RLE

Допустим, что имеется следующий текст: AAAAAAAAABBBBBAAAAACCCCCCCCCCCCCCDDD. Его длина в исходном состоянии равна 36 символам или 36 байтам, при условии, что на 1 символ выделяется 1 байт. Выполним сжатие этой строки алгоритмом RLE:

1) серия AAAAAAAAA будет записана как 9A;

2) серия BBBBB – как 5B;

3) серия CCCCCCCCCCCCCC – как 14C;

4) серия DDD – как 3D.

Итоговая запись будет выглядеть как 9A5B14C3D. Каждое число будет занимать 1 байт, следовательно, размер этой записи – 8 байт. Коэффициент сжатия равен .

## 1.5 Сжатие «неудобного» текста алгоритмом RLE. Модификация алгоритма RLE для сжатия неповторяющихся символов

Возьмем следующий текст: ABCDEFGGGGHABCDDDDEEFGHABC (26 байт). При попытке сжатия с помощью RLE получим следующую запись: 1A1B1C1D1E1F4G1H1A1B1C4D2E1F1G1H1A1B1C (38 байт). Коэффициент сжатия равен , что означает, что сжатая запись занимает больше места, чем исходная. Следовательно, метод RLE неэффективен для сжатия текстов с цепочками неповторяющихся символов.

Для решения этой проблемы можно провести модификацию метода RLE: теперь байт, хранящий число повторов, будет также хранить информацию о том, является ли следующий символ повторяющимся: 0 в старшем разряде байта - счетчика прогона будет означать повторение символа, записанного следующим байтом, указанное количество раз, а 1 в старшем разряде байта - счетчика прогона будет означать серию из стольких неповторяющихся символов, сколько указано в остальных разрядах байта - счетчика прогона.

Проведем сжатие того же текста модифицированным методом, в результате чего получим следующую запись (значения байтов – счетчиков прогона приведены в десятичной системе счисления): 134ABCDEF 4G 132HABC 4D 2E 134FGHABC (25 байт). Коэффициент сжатия в этом случае будет равен .

# **2 ЗАДАНИЕ 2**

## 2.1 Постановка задачи

1) Выполнить каждую задачу варианта, представив алгоритм решения в виде таблицы и указав результат сжатия.

**Задачи:**

Сжатие данных по методу Лемпеля – Зива LZ77: Используя двухсимвольный алфавит (0, 1) закодировать следующую фразу: 101000100101010001011

Закодировать следующую фразу, используя код LZ78: какатанекатанекатата

2) Описать процесс восстановления сжатого текста.

## 2.2 Описание метода LZ77 и его применение к заданному тексту

Описание алгоритма кодирования методом LZ77:

Первоначально каждому символу алфавита присваивается определенный код (коды - порядковые номера, начиная с 0).

1. Выбирается первый (один) символ сообщения и заменяется на его код.

2. Выбираются следующие два символа и заменяются своими кодами. При этом комбинации двух символов присваивается свой код. Обычно это номер, равный числу уже использованных кодов. Так, если алфавит включает 8 символов, имеющих коды от 000 до 111, то первая двух символьная комбинация получит код 1000, следующая - код 1001 и т.д.

3. Выбираются из исходного текста очередные 2, 3,...N символов до тех пор, пока не образуется еще не встречавшаяся комбинация. Тогда этой комбинации присваивается очередной код, и поскольку совокупность А из первых N-1 символов уже встречалась, то она имеет свой код, который и записывается вместо этих N-1 символов. Т.е. можно представить формирование кода в этом случае так: хххххххх код из N(=8) символов выбран из кодируемой последовательности, тогда если для первых N-1 символов уже был сформирован код, то заменяем эти N-1 символы на их код. Каждый акт введения нового кода назовем шагом кодирования.

4. Процесс продолжается до исчерпания исходного текста.

Для примера рассмотрим строку 101000100101010001011. Проведем сжатие методом LZ77 (таблица 1):

Таблица 1 – Процесс сжатия методом LZ77

|  |  |  |  |
| --- | --- | --- | --- |
| Содержимое окна | Содержимое буфера | Код новой последовательности | Текущий код |
|  | 101000100101010001011 |  |  |
| 1 | 01000100101010001011 | 1 | 1 |
| 0 | 1000100101010001011 | 0 | 1.0 |
| 10 | 00100101010001011 | 10 | 1.0.10 |
| 00 | 100101010001011 | 11 | 1.0.10.11 |
| 10 | 0101010001011 |  | 1.0.10.11.10 |
| 010 | 1010001011 | 100 | 1.0.10.11.10. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **3 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Задание на самостоятельную работу: <https://online-edu.mirea.ru/pluginfile.php?file=%2F1144127%2Fmod_assign%2Fintroattachment%2F0%2FСиАОД%20Самостоятельная%20работа%207%20%28алгоритмические%20стратегии%29.pdf>, дата обращения: 06.12.23
2. Структуры и алгоритмы обработки данных – лекция 2.8: <https://online-edu.mirea.ru/pluginfile.php?file=%2F1126225%2Fmod_folder%2Fcontent%2F0%2F%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D1%8B%20%D0%B8%20%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D1%8B%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85_%D0%9B%D0%B5%D0%BA%D1%86%D0%B8%D1%8F_2.8.pptx>, дата обращения: 06.12.23