



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

---

Отчет по выполнению практического задания 8-1

**Тема: «Алгоритмы кодирования и сжатия данных»**

Дисциплина: Структуры и алгоритмы обработки данных

Выполнил студент Фамилия И. О.  
Группа АААА-00-00

**Москва 2024**

## СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ.....	3
1.1 Постановка задачи.....	3
1.2 Индивидуальный вариант.....	4
ЗАДАНИЕ 1.....	5
2.1 Метод Шеннона–Фано.....	5
2.2 Метод Лемпеля–Зива LZ77.....	6
2.3 Метод Лемпеля–Зива LZ78.....	7
ЗАДАНИЕ 2.....	8
3.1 Программа Шеннона–Фано.....	8
3.2 Программа Хаффмана.....	9
4 ВЫВОД.....	10
5 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ.....	11

# 1 ВВЕДЕНИЕ

## 1.1 Постановка задачи

Задание 1 Исследование алгоритмов сжатия на примерах

1) Выполнить каждую задачу варианта, представив алгоритм решения в виде таблицы и указав результат сжатия. Примеры оформления решения представлены в Приложении1 этого документа.

2) Описать процесс восстановления сжатого текста.

3) Сформировать отчет, включив задание, вариант задания, результаты выполнения задания варианта.

Задание 2 Разработать программы сжатия и восстановления текста методами Хаффмана и Шеннона – Фано.

1) Реализовать и отладить программы.

2) Сформировать отчет по разработке каждой программы в соответствии с требованиями.

- По методу Шеннона-Фано привести: постановку задачи, описать алгоритм формирования префиксного дерева и алгоритм кодирования, декодирования, код и результаты тестирования. Рассчитать коэффициент сжатия. Сравнить с результат сжатия вашим алгоритмом с результатом любого архиватора.

- по методу Хаффмана выполнить и отобразить результаты выполнения всех требований, предъявленных в задании и оформить разработку программы: постановка, подход к решению, код, результаты тестирования.

## 1.2 Индивидуальный вариант

Таблица 1 — Индивидуальный вариант

Вариант	Закодировать фразу методами Шеннона–Фано	Сжатие данных по методу Лемпеля–Зива LZ77 Используя двухсимвольный алфавит (0, 1) закодировать следующую фразу:	Закодировать следующую фразу, используя код LZ78
19	Перводан, другодан, На колоде барабан; Свистель, коростель, Пятерка, шестерка, утюг.	0001000010101001101	comconcomconacom

## ЗАДАНИЕ 1

### 2.1 Метод Шеннона–Фано

Оформим таблицу символов для заданной фразы: «Перводан, другодан, На колоде барабан; Свистель, коростель, Пятерка, шестерка, утюг.». Результат представлен в таблице 1.

Таблица 1 - Кодирование символов

Сим вол	К/о	1-я цифра	2-я цифра	3-я цифра	4-я цифра	5-я цифра	6-я цифра	7-я цифра	Код	К/о бит
Пробел	9	0	0	0					000	27
а	8	0	0	1	0				0010	32
е	7	0	0	1	1				0011	28
,	6	0	1	0					010	18
о	6	0	1	1	0				0110	24
р	6	0	1	1	1				0111	24
т	5	1	0	0	0				1000	20
с	4	1	0	0	1				1001	16
к	4	1	0	1	0				1010	16
н	4	1	0	1	1	0			10110	20
д	4	1	0	1	1	1			10111	20
л	3	1	1	0	0	0			11000	15
ь	2	1	1	0	0	1			11001	10
б	2	1	1	0	1	0			11010	10
г	2	1	1	0	1	1	0		110110	12
у	2	1	1	0	1	1	1		110111	12
в	2	1	1	1	0	0			11100	10
п	2	1	1	1	0	1	0		111010	12
.	1	1	1	1	0	1	1		111011	6
ю	1	1	1	1	1	0			11110	5
ш	1	1	1	1	1	0			11110	5
я	1	1	1	1	1	1	0		111110	6
и	1	1	1	1	1	1	1	0	1111110	7
;	1	1	1	1	1	1	1	1	1111111	7

Незакодированная фраза –  $84 \cdot 8$  бит = 672 бит.

Закодированная фраза – 362 бит.

## 2.2 Метод Лемпеля–Зива LZ77

Фраза: «110101011001100001001». Алфавит: 0, 1.

Таблица 2 - Последовательное кодирование подпоследовательностей

Содержимое окна (сжимаемый) текст	Содержимое управляющего буфера	Код назначенный последовательности
	110101011001100001001	
1	10101011001100001001	1
10	101011001100001001	10
101	011001100001001	11
01	1001100001001	100
100	1100001001	101
11	00001001	110
00	001001	111
001	001	1000
001		001

Таблица 3 - Результат кодирования

Исходный текст	110101011001100001001 1.10.101.01.100.11.00.001.001		
LZ-код	1.10.101.01.100.11.00.1111.1111		
R	2	3	4
Вводимые коды	- 10 11 100 101 110 111 1000 1001		

## 2.3 Метод Лемпеля–Зива LZ78

Фраза: «comcomcomcomacom».

Таблица 4 - Кодирование подпоследовательностей

Словарь	Считываемое содержимое	Код
	c	<0, c>
c = 1	o	<0, o>
c = 1 o = 2	m	<0, m>
c = 1 o = 2 m = 3	co	<1, o>
c = 1 o = 2 m = 3 co = 4	n	<0, n>
c = 1 o = 2 m = 3 co = 4 n = 5	com	<4, m>
c = 1 o = 2 m = 3 co = 4 n = 5 com = 6	con	<4, n>
c = 1 o = 2 m = 3 co = 4 n = 5 com = 6 con = 7	a	<0, a>
c = 1 o = 2 m = 3 co = 4 n = 5 com = 6 con = 7 a = 8	com	<6, EOF>

Результат кодирования:

<0, c><0, o><0, m><1, o><0, n><4, m><4, n><0, a><6, EOF>

## ЗАДАНИЕ 2

### 3.1 Программа Шеннона–Фано

Необходимо реализовать программу сжатия и восстановления текста методом Шеннона-Фано.

Первым шагом определяются вероятности символов в строке. Формируется список структур символов с самим символом, его кодом и вероятностью, и сортируется по вероятности.

Далее используется рекурсивная функция формирования кодов символов. Каждый этап рекурсии функция определяет точку разрыва в текущем списке по сумме вероятностей и добавляет код символу в зависимости от нахождения в нужной половине списка.

Далее, из полученных кодов формируется словарь символов с и их кодами, и полученный код записывается в файл. Для дешифровки происходит чтение файла и дешифруется в соответствии с таблицей кодов.

Результаты сравнения представлены в таблице 5. Результаты тестирования на рис. 2.

Таблица 5 - Сравнительная таблица

Метод	Исходная строка	Шеннона-фано	zip	tar.gz
Размер	38 Байт	17 байт	38 байт	153 байт
Коэф. Сжатия	1	2.23	1	0.25

```
[rubicus@rubicus output]$ ./"shannon"
Исходный размер: 38 bytes
Исходная строка: Лаба Илья Михайлович
Зашифрованная строка: 001110000100000101111001000100100111010010100111110011111010010011011100101100
111011000100000110010010100011000001101000101100111111
Сжатый размер: 17 bytes
Коэффициент сжатия: 2.23529
Дешифрованная строка: Лаба Илья Михайлович
```

Рисунок 1 - Результаты тестирования



### 3.2 Программа Хаффмана

Необходимо реализовать программу сжатия и восстановления текста методом Хаффмана. Реализуемый алгоритм похож на предыдущий построением таблицы частот. Однако построение дерева кодов начинается с листьев, в отличие от корня. Для построения дерева используется приоритетная очередь, с приоритетом по частоте. И также используется рекурсивный алгоритм для построения дерева Хаффмана и формирования кодов.

Сравнительная таблица представлена в таблице 6. Результаты тестирования на рис. 2.

Таблица 6 - Сравнительная таблица

Метод	Исходная строка	Хаффмана
Размер	38 Байт	17 байт

```
[rubicus@rubicus output]$ ./"huffman"
Исходная строка: Лаба Илья Михайлович
Исходный размер: 38 bytes
Зашифрованная строка: 110110011101111000011101100111110101111000010001010100111100111100010111001010
0010011101111011011110001100011110111001010100
Сжатый размер: 17 bytes
Дешифрованная строка: Лаба Илья Михайлович
[rubicus@rubicus output]$
```

Рисунок 2 - Результаты тестирования

## **4 ВЫВОД**

Были изучены алгоритмы сжатия данных LZ77, LZ78, Шеннона-Фано и Хаффмана. Алгоритмы были использованы для сжатия тестовых данных.

Для алгоритмов Шеннона-Фано и Хаффмана были написаны программы на языке программирования C++.

## **5 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ**

1. Структуры и алгоритмы обработки данных (часть 2): Лекционные материалы / Рысин М. Л. МИРЭА — Российский технологический университет, 2022/23. – 82 с.