Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Сжатие/распаковка данных методом Барроуза-Уиллера**

Студент: Дрозд А. И.

ФИТ 3 курс 2 группа

Преподаватель:

Нистюк Ольга Александровна

Минск 2024

**Цель**: приобретение практических навыков использования метода Барроуза-Уиллера для сжатия/распаковки данных.

Задачи:

1. Закрепить теоретические знания по алгебраическому описанию и использованию методов сжатия/распаковки (архивации/ разархивации) данных на основе метода Барроуза − Уилера (BurrowsWheeler transform, BWT).
2. Разработать приложение для реализации метода Барроуза-Уиллера.
3. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

# **Теоретические сведения**

Сжатие информации является одним из способов ее кодирования. Развитие методов сжатия данных имеют длинную историю, которая началась задолго до появления компьютеров и компьютерных сетей.

Основная цель сжатия – обеспечить более компактное представление данных, вырабатываемых источником, т. е. уменьшить физический объем сообщений, генерируемых источником, и сократить время его передачи по каналам связи. Фундаментальная теорема К. Шеннона о кодировании информации утверждает, что «стоимость кодирования всегда не меньше энтропии источника, хотя может быть сколь угодно близка к ней». Поэтому для любого алгоритма сжатия всегда имеется некоторый предел степени (или эффективности) сжатия, определяемый энтропией входного потока (или сжимаемого сообщения).

Все алгоритмы сжатия преобразуют входной поток данных, минимальной единицей которых является бит, а максимальной – байт или несколько байт. Основными техническими характеристиками процессов сжатия и результатов их работы являются: степень сжатия (англ. compress rating), или отношение R (англ. ratio) объемов исходного (до сжатия, Vдс) и результирующего (после сжатия, Vпс) потоков данных (сообщений); скорость сжатия − время, затрачиваемое на сжатие некоторого объема информации входного потока до получения из него эквивалентного выходного потока; качество сжатия − величина, показывающая, насколько сильно сжат выходной поток при помощи применения к нему повторного сжатия по этому же или иному алгоритму.

Степень сжатия R оценивается образом, представленном на рисунке 1.1.



Рисунок 1.1 – Нахождение степеней сжатия

Первое отношение показывает, какую часть объема сообщения (файла) до сжатия занимает сообщение (файл) после сжатия; второе отношение выражает основной физический смысл сжатия и показывает степень сжатия. Что касается третьей из приведенных технических характеристик (качества сжатия), то она показывает, по существу, совместимость данного метода с другими.

Существуют различные подходы к реализации сжатия информации. Они отличаются математической базой, уровнем сложности (простоты) практической реализации, форматом кодируемого потока данных, степенью соответствия сжимаемых и распакованных данных.

BWT-преобразование (англ. Burrows-Wheeler Transform) – техника сжатия информации (в особенности текстов), основанная на преобразовании, открытом в 1983 г. BWT не сжимает данные в классическом понимании процесса, но преобразует блок данных в формат, исключительно подходящий для сжатия. BWT оперирует сразу целым блоком данных, который выделяется из входного потока (сообщения).

Процесс BWT-преобразования включает четыре этапа:

* Выделение блока данных M длиной k символов.
* Формирование таблицы W1 размером k×k, содержащей все циклические сдвиги входной строки M.
* Лексикографическая сортировка строк таблицы W1 для получения таблицы W2 того же размера.
* Выбор последнего столбца таблицы W2 в качестве выходной строки (обозначаемой как BWT(М)), а также номера строки, соответствующей исходной строке M.

Обратное преобразование BWT осуществляется в k шагов, каждый из которых состоит из двух операций:

* Запись последовательности символов Мk в крайний справа пустой столбец матрицы.
* Лексикографическая сортировка столбцов заполненной части матрицы.

Пример преобразования и обратного преобразования с помощью BWT показан на рисунке 1.2.

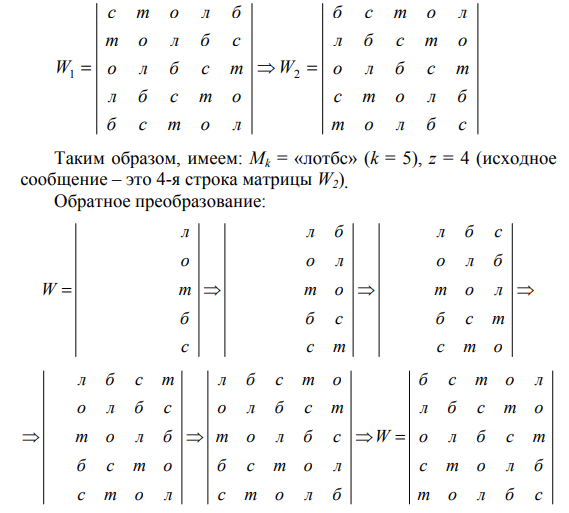


Рисунок 1.2 – Прямое преобразование и обратное преобразование слова «столб»

На каждом шаге матрица W2 восстанавливается на основе предыдущего состояния, и зная номер строки z, можно получить исходный блок М сообщения. Каждая операция восстановления матрицы W2 основана на рекурсивности преобразования и требует обращения к предыдущим значениям.

**Практическое задание**

Вариант 3



Рисунок 2.1 – Варианты заданий

Исходя из условия, необходимо разработать авторское приложение в соответствии с целью лабораторной работы. Входной блок данных может иметь произвольную длину. С помощью приложения выполнить прямое и обратное преобразования 3 отдельных блоков данных, состоящих:

а) из собственного имени (можно краткий вариант записи);

б) собственной фамилии;

в) варианта в соответствии с таблицей ниже.

Можно использовать любой из известных методов сортировки символов массива. Выполнить качественный сравнительный анализ длительности процессов прямого и обратного преобразований в зависимости от длины блока данных.

Также необходимо перевести первые 3 символа из блока данных, указанного в варианте таблицы, в бинарную последовательность в соответствии с кодами ASCII. Выполнить прямое и обратное преобразование. Оценить время прямого и обратного преобразований.

Исходное сообщение для пункта «в» берется из таблицы на рисунке 2.1.

Первоначально строим матрицу размером k×k, где k – это длина исходного блока данных. Сама матрица представляет собой все циклические сдвиги входного блока. Построение матрицы показано на рисунке 2.2.



Рисунок 2.2 – Построение матрицы циклических сдвигов

Далее необходимо отсортировать все строки полученной матрицы в алфавитном порядке. Сортировка матрицы представлена на рисунке 2.3.



Рисунок 2.3 – Сортировка матрицы циклических сдвигов

После этого необходимо выполнить обратное преобразование для получения исходного сообщения после передачи. Первоначально нужно записать полученное сжатое сообщение "ШЛАЁ" в последний столбец матрицы. Далее необходимо отсортировать данный столбец в алфавитном порядке, как это делалось при сжатии. После этого происходит получение следующего столбца, записываемого слева от предыдущего. Его также необходимо отсортировать в алфавитном порядке, но при этом происходит сортировка всей строки, а не только одного столбца. Данный алгоритм продолжается до тех пор, пока не будет получено исходное сообщение. На рисунке 2.4 изображён алгоритм обратного преобразования.



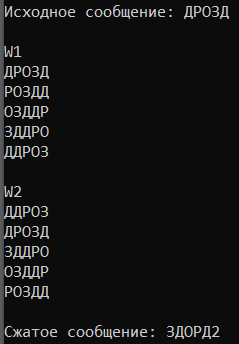
Рисунок 2.4 – Алгоритм обратного преобразования

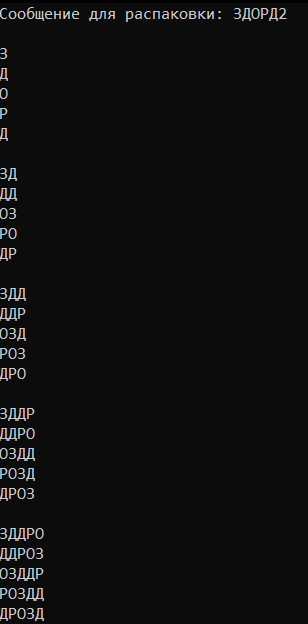
Данный алгоритм используется для выполнения всех заданий, поэтому их выполнение представлено в Приложении А.

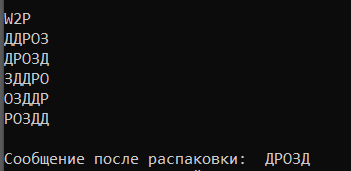
**Вывод:**

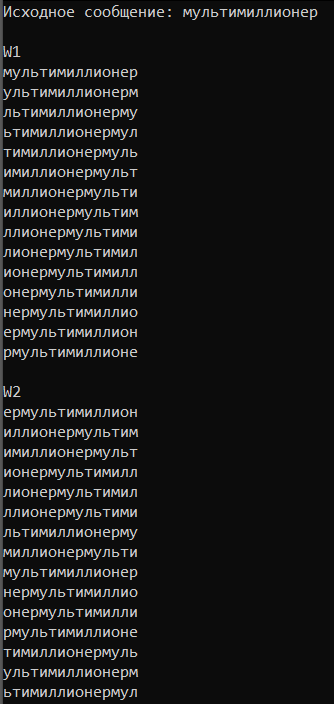
Метод сжатия данных Барроуза-Уиллера изменяет порядок символов в строке, создавая много повторяющихся последовательностей, что облегчает их сжатие. Основная идея заключается в создании матрицы всех циклических перестановок исходной строки, а затем сортировке этой матрицы по алфавиту. Последний столбец отсортированной матрицы называется преобразованием Барроуза-Уиллера. Он обладает свойством того, что подстроки схожих символов склеиваются вместе, что делает их более подходящими для последующего сжатия. Этот метод используется в таких архиваторах, как bzip2, для сжатия данных.

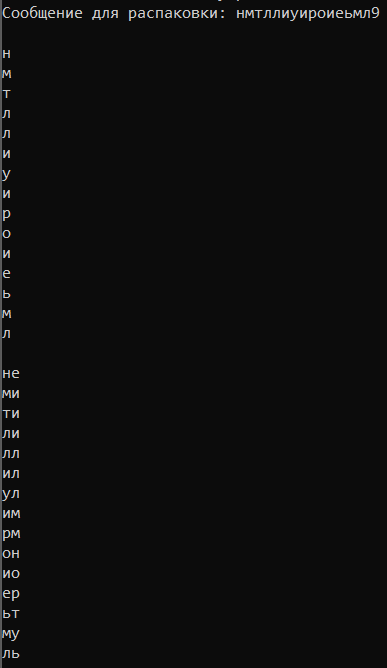
**Приложение А**

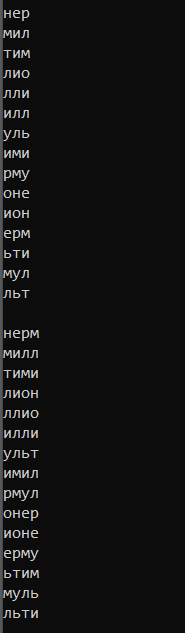
****

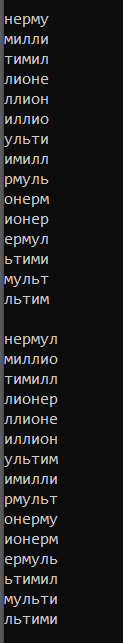
****

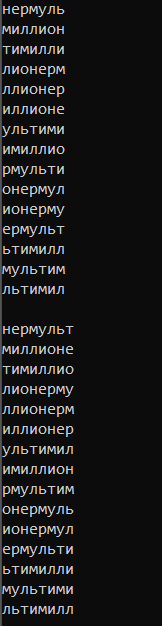
****

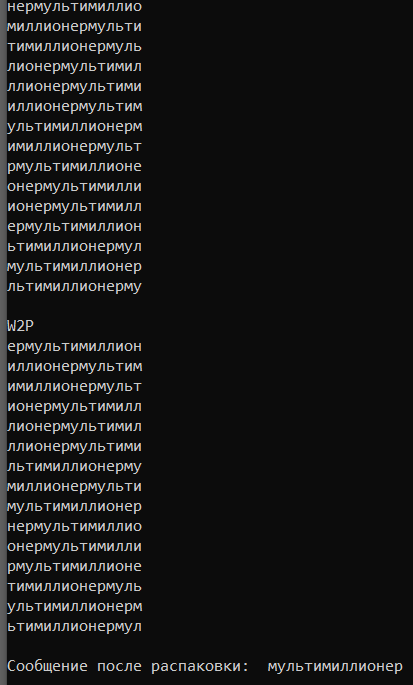
****

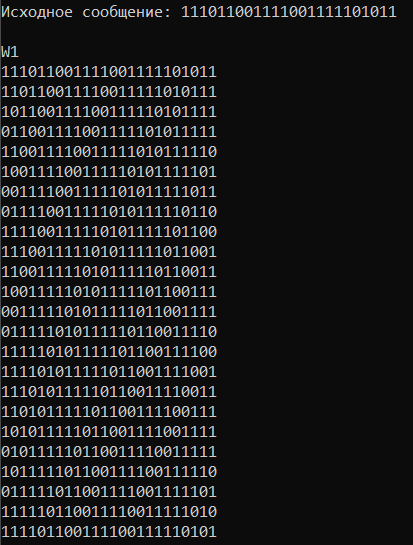
****

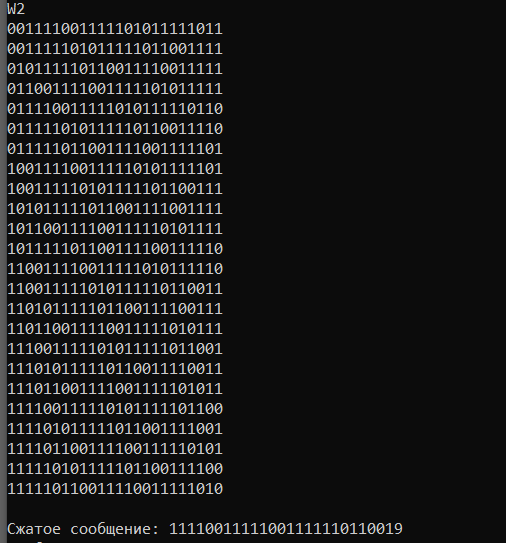
****

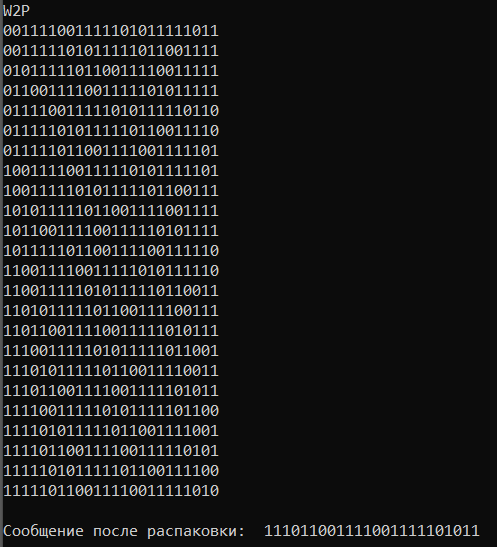
****

****

****

****

****

****