**АЛГОРИТМ LZ77**

**Руководство программиста**

**Выполнили: Талеев Василий,**

**Никитина Надежда**

**15 листов**

**АННОТАЦИЯ**

В данном документе приведено руководство программиста по настройке и использованию алгоритма сжатия в виде программного кода LZ77.py, которая предназначена для сжатия данных. Алгоритм LZ77 является одним из базовых методов сжатия без потерь и широко применяется в различных областях, таких как архивирование файлов, передача данных в сети, и другие приложения, где эффективное использование ресурсов и пропускной способности играют важную роль.

В разделе “Назначение и условия применения программы” указаны назначение программы, список всех функций и пояснение по ним, а также требования как к оборудованию, так и к самому программисту.

В разделе “Характеристика программы” указаны особенности кода, тестирования и работа с функциями.

В разделе “Обращение к программе” указано способ запуска работы с функциями и тестирования.

В разделе “Входные и выходные данные” указано, как работать с каждой из функций (что должно быть на входе и что на выходе).

В разделе “Сообщения” указаны возможные ошибки и способы их решения.

**Содержание**

Аннотация 2

Содержание 3

1. Назначение и условия применения 5

1.1 Область применения 5

1.2 Функции кода 5

1.3 Требования к выполнению 6

1.3.1 Программные и аппаратные требования 6

1.3.1.1 Операционные системы 6

1.3.1.2 Системная конфигурация 6

1.3.1.3 Программное обеспечение 7

1.3.2 Требования к программисту 7

2. Характеристика программы 8

2.1 Дистрибутив кода 8

2.2 Работа с программой 8

2.2.1 test.py 8

2.2.2 main.py 8

2.3 Тестирование 9

3. Обращение к программе 12

3.1 Запуск программы 12

3.2 Выполнения программы 12

3.2.1 Выполнение тестирования 12

3.2.2 Выполнение работы с функциями 12

4. Входные и выходные данные 13

4.1. Входные данные 13

4.2. Входные и выходные данные 13

4.2.1. Сжатые данные 13

4.2.2. Распакованные данные 13

5. Сообщения 15

1. **Назначение и условия применения программы**

**1.1 Область применения**

Программный код main.py предназначен для реализации алгоритма сжатия данных методом LZ77. Этот код обеспечивает компрессию данных с использованием словаря и фраз, что полезно для эффективного уменьшения объема передаваемых или хранимых данных, таких как архивирование файлов, передача информации по сети и другие приложения.

**1.2 Функции кода**

LZ77\_c (LZ77 Compression) - Данная функция предназначена для сжатия входных данных с использованием алгоритма сжатия LZ77. Она принимает два параметра: data (входные данные для сжатия) и window\_size (размер окна поиска, по умолчанию установлен в 4096).

В ходе выполнения алгоритма, функция проходит по входным данным, ищет наилучший совпадающий подстроку в заданном окне и сохраняет информацию о совпадении в виде кортежа (offset, length, next\_char). Полученные результаты компрессии сохраняются в списке compr, который затем возвращается;

LZ77\_dec (LZ77 Decompression) - Данная функция предназначена для распаковки данных, ранее сжатых алгоритмом LZ77. Принимает на вход сжатые данные в виде списка кортежей compressed и возвращает распакованный текст.

В процессе распаковки функция проходит по списку кортежей и восстанавливает оригинальный текст. Если длина сжатой фразы равна 0, то она добавляется к распакованным данным как отдельный символ. В противном случае функция использует смещение offset и длину length, чтобы восстановить подстроку из предыдущих символов;

**1.3 Требования к выполнению**

Весь дистрибутив (будет рассмотрен дальше) должен находиться в одной папке для корректного запуска.

**1.3.1** **Программные и аппаратные требования**

В данном разделе описаны требующиеся пункты, без которых установка и запуск программного кода не является возможным.

**1.3.1.1 Операционная система**

ОС Windows XP x86 (и выше).

**1.3.1.2 Системная конфигурация**

Особых требований нет, система должна корректно работать, иметь достаточное количество свободного ОЗУ и дискового пространства для установки полного дистрибутива.

**1.3.1.3 Программное обеспечение**

Python 3.4 (или новее) и интерпретатор IDLE.

**1.3.2 Требования к программисту**

Программист должен обладать пониманием алгоритма сжатия LZ77 для корректной интерпретации ответов, а также должен владеть языком программирования Python.

1. **Характеристика программы**
   1. **Дистрибутив кода**

LZ77.py – программный код.

test.py – файл с тестами.

main.py – главный код программы для работы с функциями через интерфейс.

* 1. **Работа с программой**
     1. **test.py**

В данном файле находятся тесты для программного кода, реализующей алгоритм сжатия LZ77. Каждая функция теста проверяет корректность работы алгоритма на различных входных данных. При запуске файла выводится сообщение "Проведенные тесты соответствуют ожидаемым результатам", если представленные тесты были пройдены успешно.

* + 1. **Main.py**

В данном файле реализовано взаимодействие с пользователем через графический интерфейс с использованием библиотеки Tkinter. Пользователю предоставляется окно, в котором можно ввести текст для сжатия, выполнить сжатие, а затем просмотреть результат. Основные компоненты интерфейса включают поле ввода текста, кнопки "Сжать" и "Очистить", а также текстовое поле для вывода результатов.

Функциональность предоставлена методами класса LZ77App, включающими создание виджетов, сжатие данных, очистку полей ввода и вывода, а также отображение результатов. Приложение запускается при выполнении файла и ожидает действий пользователя через графический интерфейс. В случае ошибки смотрите пункт сообщения.

* 1. **Тестирование**

Для тестирования программы были применены различные методы тестирования, включая тест-раннер, кормление собак, а также модульное и системное тестирование.

**Тест-раннер:**

Описание: Тест-раннер представляет собой автоматизированный инструмент для запуска набора тестов. В данном случае, тест-раннер был использован для автоматического выполнения всех тестов из файла test.py.

Цель: Обеспечить быстрый и эффективный способ запуска и анализа результатов всех тестов, что позволяет обнаруживать ошибки и несоответствия в ранних стадиях разработки.

**«Кормление собак»:**

Описание: Термин "кормление собак" (dogfooding) относится к использованию собственного продукта разработчиками для проверки его функциональности и выявления возможных проблем. В данном контексте, программу, реализующую алгоритм сжатия LZ77, тестировали на реальных данных и использовали результаты для проверки соответствия ожиданиям.

Цель: Обеспечить более реалистичное и практическое тестирование программы, идентифицируя потенциальные слабые места в алгоритме на основе реальных данных.

**Модульное тестирование:**

Описание: Модульное тестирование направлено на проверку отдельных компонентов (модулей) программы, в то время как системное тестирование оценивает работу программы в целом. В данном случае, каждая функция теста в файле test.py представляет собой модульный тест, проверяющий определенный аспект алгоритма сжатия LZ77.

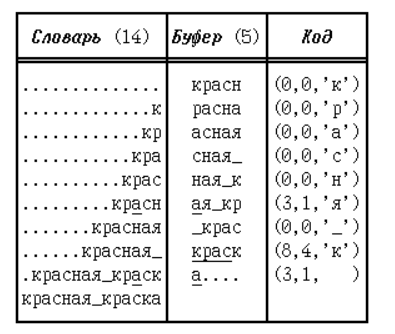
Цель: Выявить и устранить ошибки и несоответствия в отдельных частях программы, а также гарантировать корректное взаимодействие между компонентами.

**Ручное тестирование:**

Описание: Ручное тестирование представляет собой процесс проверки программы или ее отдельных компонентов вручную, без привлечения автоматизированных средств тестирования. В данном случае, ручное тестирование будет использоваться для проверки алгоритма сжатия LZ77 на примере сжатия слова "красная краска".

Цель: Выявление и детальное изучение процесса сжатия, а также проверка корректности работы алгоритма на конкретном примере.

Пример ручного тестирования для слова "красная краска":



Использование разнообразных методов тестирования обеспечивает более полное и надежное покрытие функциональности программы, позволяя выявлять и устранять проблемы на всех этапах разработки.

**3. Обращение к программе**

**3.1 Запуск программы**

Запуск производится нажатие клавиши Enter по файлу тестирования или работы с функциями (LZ77.py и main.py соответственно)

**3.2 Выполнение программы**

**3.2.1 Выполнение тестирования**

После выведения результата, указанного в пункте 2.2.1, программа будет ждать отклика для закрытия программы или ее повторного запуска.

**3.2.2 Выполнение работы с функциями**

После выполнения основной функции программы, выводится сообщение с результатом сжатия. Для повторного использования, нажмите на кнопку “Очистить”.

**4. Входные и выходные данные**

В данной главе предоставляется информация о форматах входных данных, параметрах, которые программа ожидает, и форматах выходных данных, которые программа генерирует. Эта информация является ключевой для понимания того, как правильно взаимодействовать с программой и как оценить ее результаты.

**4.1. Входные Данные**

Описание: Основным входным параметром программы является текст, который предполагается сжать. Текст может содержать символы из расширенного набора ASCII, включая буквы, цифры, знаки препинания и специальные символы.

Формат: Текст представлен в виде строки символов. Пример: "Это текст для сжатия."

**4.2. Выходные Данные**

**4.2.1 Сжатые Данные:**

Описание: Результат сжатия текста алгоритмом LZ77. Состоит из последовательности кортежей, каждый из которых содержит информацию о смещении, длине фразы и следующем символе.

Формат: Список кортежей вида (offset, length, next\_char). Пример: [(0, 0, 'к'), (5, 3, 'р'), ...]

**4.2.2 Распакованные Данные:**

Описание: Результат распаковки сжатых данных. Представляет собой строку, идентичную исходному тексту.

Формат: Строка символов на языке Python. Пример: "Это текст для сжатия."

**5. Сообщения**

Ошибки могут возникнуть при неправильном вводе текста, при этом программа завершит выполнение. В дальнейшем ожидается исправление данной ситуации.

1. **Ошибка**: если введен пустой текст, алгоритм сжатия не сможет корректно выполнить свою задачу.

**Способ решения**: проверьте, что вводимый текст не является пустым. Введите текст для сжатия и повторите запрос.

2. **Ошибка**: при вводе некорректных данных, не соответствующих ожидаемому формату.

**Способ решения**: проверьте введенные данные на соответствие формату, убедитесь, что используются только допустимые символы, и повторите запрос с правильными данными.

3. **Ошибка**: в случае ограниченных ресурсов памяти алгоритм может столкнуться с ограничениями выполнения.

**Способ решения**: попробуйте уменьшить размер входных данных или выбрать другие параметры алгоритма для снижения нагрузки на память.

4. **Ошибка**: в случае использования других компонентов Python, несовместимых с текущей реализацией алгоритма, могут возникнуть ошибки.

**Способ решения**: убедитесь, что используется совместимая версия Python и необходимые библиотеки. При необходимости обновите или установите необходимые компоненты.