Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

«СЖАТИЕ/РАСПАКОВКА ДАННЫХ АРИФМЕТИЧЕСКИМ МЕТОДОМ»

Студент: Скалкович С.Л.

ФИТ 3 курс 2 группа

Преподаватель: Нистюк О.А.

Минск 2024

**Вариант 12**

**Цель:** приобретение практических навыков использования арифметических методов сжатия/распаковки данных.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию и использованию арифметических методов сжатия/распаковки (архивации/разархивации) данных.

2. Разработать приложение для реализации арифметических методов.

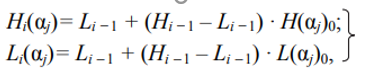
3. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Теоретические сведения**

При арифметическом сжатии (кодировании) текст представляется вещественными числами в интервале от 0 до 1. По мере анализа текста отображающий его интервал уменьшается, а количество битов для его представления возрастает. Очередные символы сокращают величину интеpвала, исходя из значений, соответствующих вероятностей.

Прямое преобразование (сжатие). Один шаг сжатия (кодирования) заключается в простой операции: берется кодируемый символ, для него ищется соответствующий участок на рабочем отрезке. Найденный участок становится новым рабочим отрезком. Его тоже необходимо разбить с помощью точек.

Э то и последующие разбиения отрезка (на шаге i) подразумевают определение новых значений верхней (Hi) и нижней (Li) границ для всего участка и осуществляются по следующим правилам:



где αj – j-й символ сжимаемой последовательности, Li − 1 и Hi − 1 – соответственно нижняя и верхняя границы рабочего отрезка на (i − 1)-м шаге, L(αj)0 и H(αj)0 – соответственно исходные нижняя и верхняя границы символа αj.

Шаг 1: в нашем примере на первом шаге берется первый символ последовательности и ищется соответствующий участок на рабочем отрезке. Он становится новым рабочим отрезком и опять разбивается согласно статистике и соотношениям. Повторяем первый шаг до тех пор, пока не останется 1 символ. В качестве результата кодирования берём минимальная граница отрезка последнего символа.

Обратное преобразование (декомпрессия). Для восстановления исходного сообщения необходима информация:

• о значении числа, являющегося итогом сжатия сообщения (в нашем случае 0,1071);

• количестве символов в сжатом сообщении;

• вероятностных параметрах всех символов исходного сообщения (таблица вероятностей).

Как и при сжатии, вначале необходимо начальный рабочий отрезок [0; 1) разбить на интервалы, длины которых равны вероятностям появления соответствующих символов, т. е. создать рабочий отрезок.

Анализ будем проводить с использованием конкретных данных, взятых из последнего примера.

Итак, в качестве исходного участка для обратного преобразования принимается исходный участок для прямого преобразования с одинаковыми точками его разбиения.

На каждом шаге обратного преобразования выбираем отрезок, в который попадает текущее число (код). Символ, который соответствует данному отрезку, является очередным символом восстановленного (распакованного) сообщения.

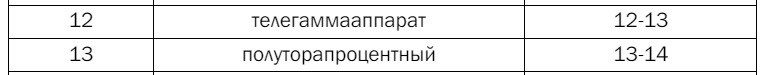
В общем случае код символа, восстанавливаемого на шаге i, вычисляется соотношением: код i = [код (i − 1) − L(αi − 1)0] / [H(αi − 1)0 − L(αi − 1)0], где код (i − 1) – число, анализ которого производился на предыдущем шаге – (i − 1)-м; H(αi − 1)0 и L(αi − 1)0 – соответственно верхняя и нижняя исходные границы символа сообщения, восстановленного на предыдущем шаге.

**Задание 1**

С помощью приложения выполнить прямое и обратное преобразования сообщений в соответствии с таблицей.

Каждый студент выполняет задание, состоящее из двух частей. Первая часть предусматривает кодирование/декодирование сообщения, указанного в 2-м столбце, вторая часть – составного сообщения, полученного конкатенацией последовательностей из 2-го столбца, указанных в 3-м столбце. Например, для варианта № 12 такой конкатенацией будет последовательность «телегаммааппаратполуторапроцентный».

Согласно 12 варианту необходимо входным словом взять «телегаммааппарат». Чтоб проверить возможность переполнения берем «телегаммааппаратполуторапроцентный».



Для выполнения этого задания при помощи арифметического метода нужно посчитать вероятности появления символов в исходном сообщении и отсортировать в порядке возрастания. На рисунке 1 будут представлены вероятности появления символов.

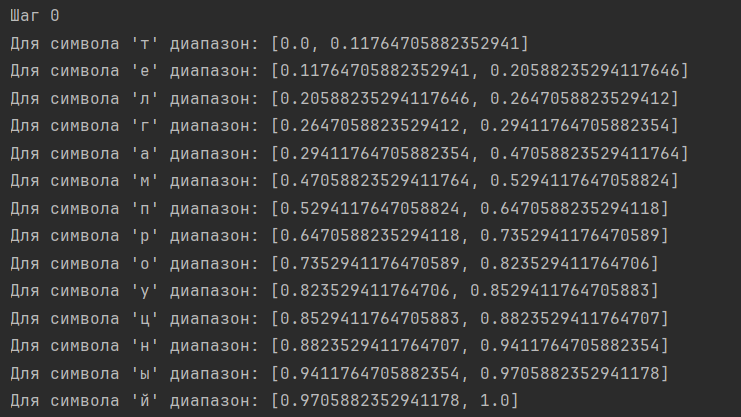


Рисунок 1 – Вероятности появления символов

После этого шага нужно распределить интервалы для каждой буквы от 0 до 1 в соответствии с их вероятностями, чтобы интервалы занимали пропорциональные вероятности. На рисунке 2 будут представлены эти интервалы.

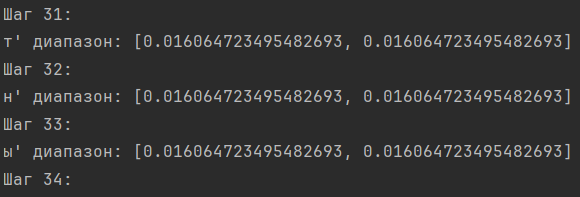


Рисунок 2 – Интервалы от 0 до 1

После этого процесса необходимо взять каждую букву по отдельности. Начнем с символа "т" и рассчитаем для него интервал, затем пересчитаем вероятности, оставляя их в процентном диапазоне. Затем берем следующий символ и продолжаем этот процесс до тех пор, пока не обработаем все символы.

Вам выбрать значение с результатом, у которого наименьшее кол-во символов. Для декодирования данных необходимо иметь статистические данные о вероятностях появления символов. На основе этих данных мы строим аналогичные интервалы. Затем, имея закодированное число, мы определяем, к какому интервалу оно относится. Таким образом, мы можем определить символ, который соответствует этому интервалу.



Рисунок 3 –Результат работы распаковки

**Вывод**:

Эксперименты в ходе лабораторной работы позволили исследовать арифметические методы сжатия и распаковки данных. Разработанная программа наглядно демонстрирует эти методы на практике.

По сравнению с другими вероятностными методами, арифметические методы обладают значительной гибкостью и эффективностью, особенно при работе с большими объемами данных. Однако они также требуют больше вычислительных ресурсов.