**Лабораторная работа № 11**

**СЖАТИЕ/РАСПАКОВКА ДАННЫХ АРИФМЕТИЧЕСКИМ МЕТОДОМ**

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

**Теоретические сведения**

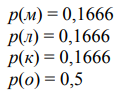
При **арифметическом сжатии (кодиpовании)** текст пpедставляется вещественными числами в интеpвале от 0 до 1. По меpе анализа текста отобpажающий его интеpвал уменьшается, а количество битов для его пpедставления возpастает. Очеpедные символы текста сокpащают величину интеpвала, исходя из значений соответствующих веpоятностей.

**Основная идея** арифметического метода сжатия заключается в том, чтобы присваивать коды не отдельным символам, а их последовательностям.

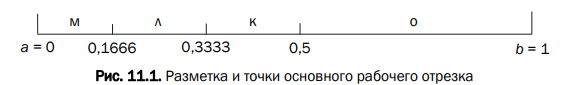
Алгоритмы прямого и обратного преобразований базируются на операциях с «**рабочим отрезком**».

**Рабочим отрезком** называется интервал [a; b] с расположенными на нем точками. Причем точки расположены таким образом, что длины образованных ими отрезков пропорциональны (или равны) частоте (вероятности) появления соответствующих символов.

Дана последовательность Xk = «молоко». Рассчитываем статистику:



Шаг 0: строим основной рабочий отрезок, как показано на рис. 11.1.



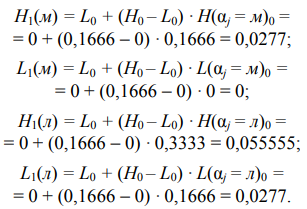
**Прямое преобразование (сжатие).** Один шаг сжатия (кодирования) заключается в простой операции: берется кодируемый символ, для него ищется соответствующий участок на рабочем отрезке. Найденный участок становится новым рабочим отрезком. Его тоже необходимо разбить с помощью точек.

Это 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: «м» (α1 = «м») и ищется соответствующий участок на рабочем отрезке. Легко понять по рис. 11.2, что этот участок соответствует интервалу [0; 0,1666]. Он становится новым рабочим отрезком и опять разбивается согласно статистике и соотношениям (11.1):



После выполнения всех вычислений на первом шаге получим разметку рабочего отрезка (a1 = L1 = 0, b1 = H1 = 0,1666) в соответствии с рис. 11.2.



Результатом кодирования цепочки символов является любое число с итогового рабочего отрезка [0,107060185; 0,10763888]. Обычно таким числом является нижняя граница указанного отрезка. Поступим и мы в соответствии с данным принципом. Возьмем число с меньшим количеством знаков после запятой. Таким образом, итогом сжатия входной последовательности «молоко» будет число 0,107060185 (для упрощения дальнейших вычислительных операций округлим его до 0,1071).

**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, полностью соответствующий рис. 11.1.

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:



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.

Шаг 1: определяем интервал, в который попадает начальное число: 0,1071 (код 1 = 0,1071). Это интервал [0; 0,166666] и ему соответствует символ «м». Данный интервал становится новым рабочим отрезком, а первый символ восстановленного сообщения – «м».

Шаг 2: производим вычисление в соответствии с выражением (11.2):



Вычисленный код соответствует символу «о».

Шаг 3: выполняем известное вычисление:



Другой вариант алгоритма распаковки «сжатого» сообщения основан на свойстве рекуррентности прямого преобразования. Производя на каждом шаге вычисление новых границ рабочего участка в соответствии с восстановленным на данном шаге символом (наподобие тех вычислений, которые мы выполняли при сжатии) и анализируя, на какой из отрезков этого участка попадает входное число (в нашем примере это 0,1071), восстанавливаем исходное сообщение.

**Практическое задание**

1. Разработать авторское приложение в соответствии с целью лабораторной работы.

2. 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 № 1 такой конкатенацией будет последовательность «летоисчислениевремяпрепровождение».

3. Дать оценку возможности переполнения при выполнении вычислений.

4. Сравнить характеристики арифметического сжатия с вероятностными алгоритмами.

5. Результаты оформить в виде отчета по установленным правилам.

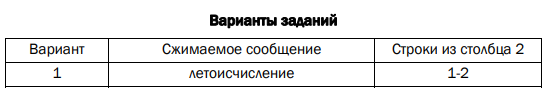




Рисунок 1.1 – Функция для прямого преобразования

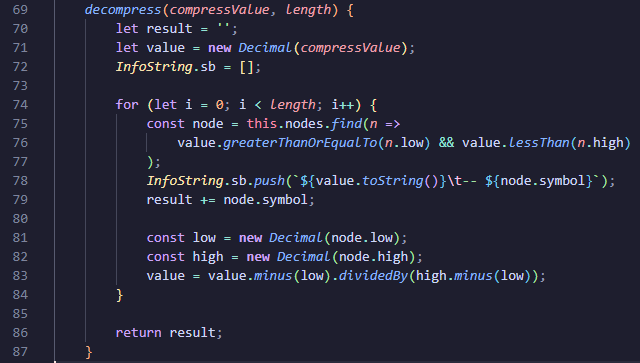
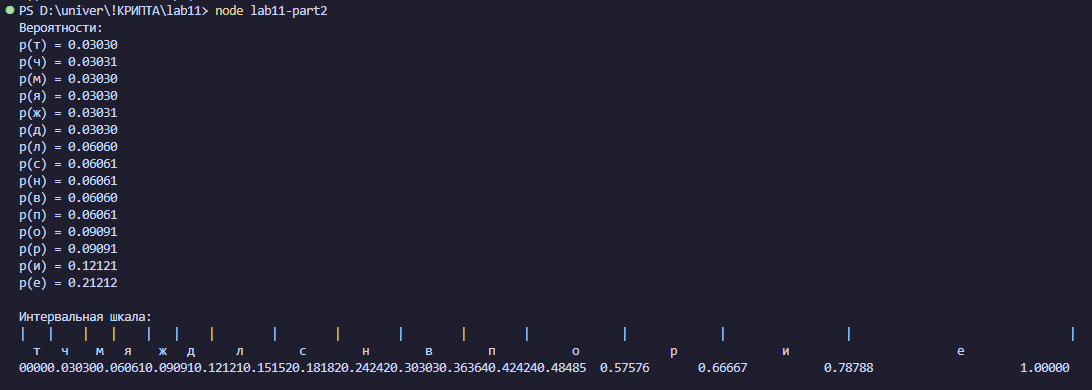
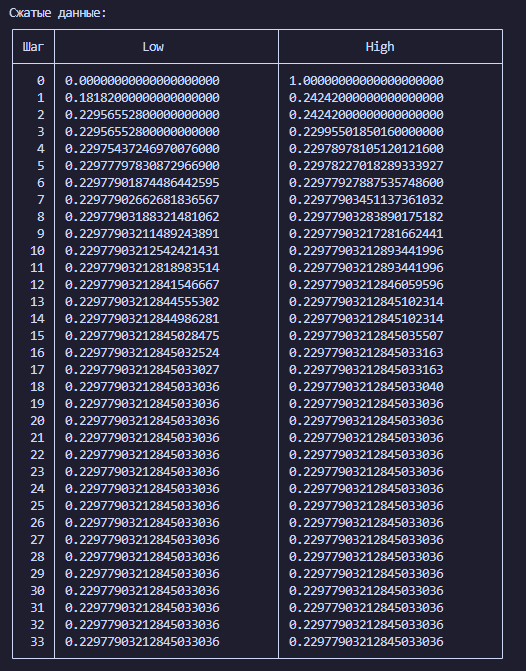


Рисунок 1.2 – Функция для обратного преобразования



Рисунок 1.3 – Результат для первой части





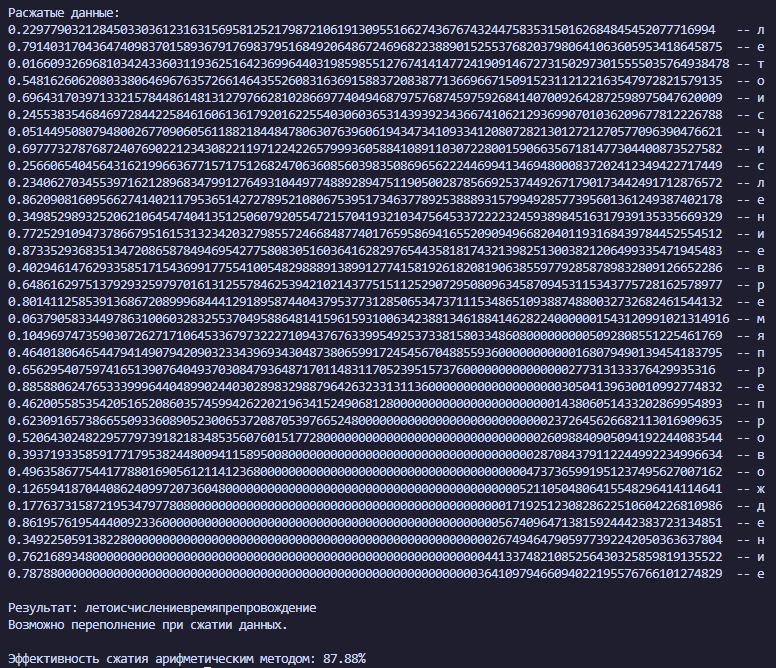


Рисунок 1.4 – Результат для второй части

**Вывод**

Сравнительно с вероятностными методами (такими как Хаффмана), арифметическое кодирование обеспечивает более высокую степень сжатия, особенно на длинных и повторяющихся входных данных. Однако оно требует значительно более высокой точности вычислений, что усложняет реализацию.