**Лабораторная работа № 9**

**СЖАТИЕ/РАСПАКОВКА ДАННЫХ НА ОСНОВЕ СТАТИСТИЧЕСКИХ МЕТОДОВ**

**Цель:** приобретение практических навыков использования статистических методов Шеннона − Фано и Хаффмана (ShannonFano and Huffman coding) для сжатия/распаковки данных.

**Теоретические сведения**

Мы неоднократно подчеркивали, что каждый из естественных языков обладает избыточностью.

До появления уже упоминавшихся работ К. Шеннона кодирование символов алфавита при передаче сообщения по каналам связи осуществлялось одинаковым количеством битов, получаемым по формуле Хартли.

За счет использования **для каждого значения байта кодов ASCII** (символа алфавита) **кода различной длины в соответствии с частостью** (вероятностью появления этого символа в сообщении) можно значительно уменьшить общий размер данных. Эта базовая идея лежит в основе алгоритмов статистических (вероятностных) методов сжатия: **Шеннона − Фано и Хаффмана**.

**Статистические алгоритмы** позволяют создавать более короткие коды для часто встречающихся и более длинные – для редко встречающихся символов алфавита или конкретного сообщения. В первом случае метод считается статическим **статистическим**, во втором – **динамическим статистическим**: вероятностные свойства символов подсчитываются для конкретного сообщения или потока данных.

Частота или вероятность появления того или иного символа алфавита в произвольном сообщении, лежащая в основе алгоритмов, дали название этим алгоритмам и соответствующим методам.

Иногда эти методы называют также **префиксными**.

Формально **процедура сжатия** (прямое преобразование) состоит в подстановке соответствующего бинарного кода вместо символа исходного алфавита и наоборот – при обратном преобразовании.

Методы относятся к классу **«сжатие без потерь».**

**Различие** между двумя рассматриваемыми методами состоит лишь в **особенностях формирования** **таблицы** бинарных кодов.

**Метод Шеннона − Фано**

Код Шеннона – Фано не является оптимальным (обеспечивает минимальную избыточность) в общем смысле, хотя и дает оптимальные результаты при некоторых распределениях вероятностей. Для одного и того же распределения вероятностей можно построить, вообще говоря, несколько кодов Шеннона – Фано, и все они могут дать различные результаты.

Алгоритм:

1) считаем вероятностные параметры для каждого символа алфавита;

2) отсортировать – обычно в порядке убывания вероятностей р(аi); т. е. создать таблицу символов алфавита, на основе которого генерируется сжимаемое сообщение;

3) каждому символу отсортированного множества поставить в соответствие бинарный код:

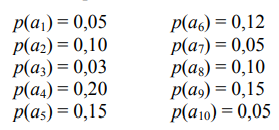
Каждое множество делится на две группы – каждая группа имеет общую одинаковую частоту (на первом шаге 0,5).

Первому подмножеству ставится символ 0, второму подмножеству – 1(или наоборот?).

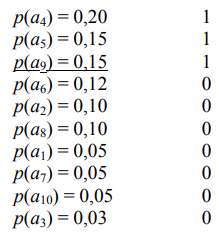
Если больше одного символа в подмножестве, то повторяем.

Пример:

Определили вероятности:



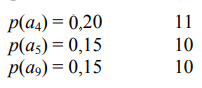
В порядке убывания:





Верхним ставим 1, нижним – 0.

Дальше смотрим 1 подмножество:





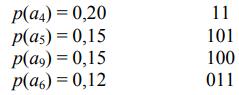
Делим на два подмножества, первому ставим 1, второму – 0.

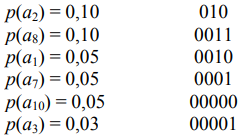
Рассматриваем второе подмножество, так как в первом уже и так один символ. Первому ставим 1, второму – 0.





Для второго подмножества из самого начала тоже проделываем тоже самое, получаем таблицу бинарных кодов:





По возрастанию индексов тоже можно расположить.

Выше мы получили таблицу, смотрим в неё, выполняем прямое или обратное преобразование просто сверяя с таблицей.

**Алгоритм прямого преобразования:** необходимо выполнить одну операцию: заменить символы входного сообщения соответствующими бинарными кодами.

**Алгоритм обратного преобразования:** на входе – сообщение в виде бинарной последовательности.

**Алгоритм для обратного**:

1. Смотрим lmin (минимальная длина кода), находим в таблице совпадение с lmin. Это будет символ алфавита с совпадающим кодом. Такое повторяется рекурсивно для каждого, если не найдено – шаг 2.

2. lmin + 1, поиск в таблице, если находим – ок, не находим – опять lmin + 1.

**Метод Хаффмана**

Метод основан на алгоритме оптимального префиксного кодирования алфавита: исходный алгоритм Хаффмана является оптимальным для посимвольного кодирования с известным входным распределением вероятностей, т. е. для отдельного кодирования несвязанных символов в таком потоке данных. Отличается от метода Шеннона – Фано лишь в части кодирования символов исходного алфавита.

В данном случае **бинарные коды создаются на основе дерева**, ветви которого обозначаются бинарными символами.

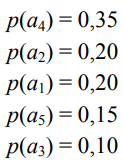
**Бинарным кодом символа исходного алфавита** будет последовательность обозначений ветвей дерева от корня до листа, соответствующего этому символу.

**Лемма. Для любого заданного алфавита (источника) с N > 2 символами существует оптимальный двоичный код, в котором два наименее вероятных символа (слова) имеют одну и ту же длину и отличаются лишь последним битом.**

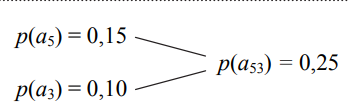
**Алгоритм:**

1. Сортируем в порядке убывания.
2. Выбираем два символа ai, aj, у которых наименьшие вероятности. Объединяем их в узел. Ветви узла – 1 и 0. Теперь это виртуальный узел aij. Его вероятность aij = сумма вероятностей ai + aj. Потомки у этого узла не рассматриваются дальше. Узел теперь наравне с другими символами. Ну типа символа два, а он как один, пон.

3. Создаются новые узлы дерева по тому же принципу. Корень дерева образуют два символа с наибольшими вероятностями.



Берем два нижних, так как у них наименьшая вероятность. Формируем p(a53) = p(a5) + p(a3) = 0,25. Это узел. p(a5) соответствует 1, p(a3) – 0.



Дальше опять сортируем, опять выбираем 2 наименьших и так далее.

Наилучшей (оптимальной или близкой к оптимальной) является та таблица, которой соответствует минимальное значение интегрального коэффициента С:



Практическое задание

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

2. С помощью приложения выполнить прямое и обратное преобразования сообщения, состоящего из собственных имени и фамилии.

Можно использовать любой из известных методов сортировки символов массива. Метод кодировки (Шеннона − Фано, Хаффмана) использовать по указанию преподавателя.

При этом таблица отсортированных символов строится:

а) на основе данных, полученных в лабораторной работе № 2;

б) динамически, на основе анализа сжимаемого сообщения.

3. Определить эффективность (в сравнении с кодами ASCII) сжатия сообщения.

4. Результаты оформить в виде отчета по установленным правилам.

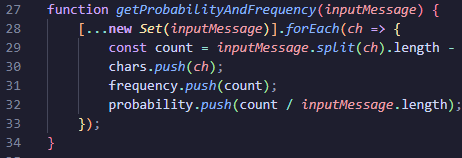


Рисунок 1.1 – Подсчет вероятностей

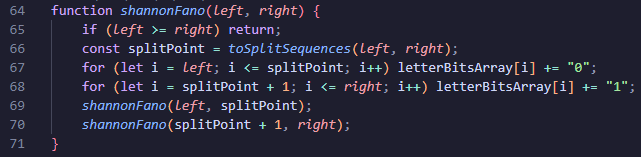


Рисунок 1.2 – Метод Шеннона-Фано



Рисунок 1.3 – Метод Хаффмана

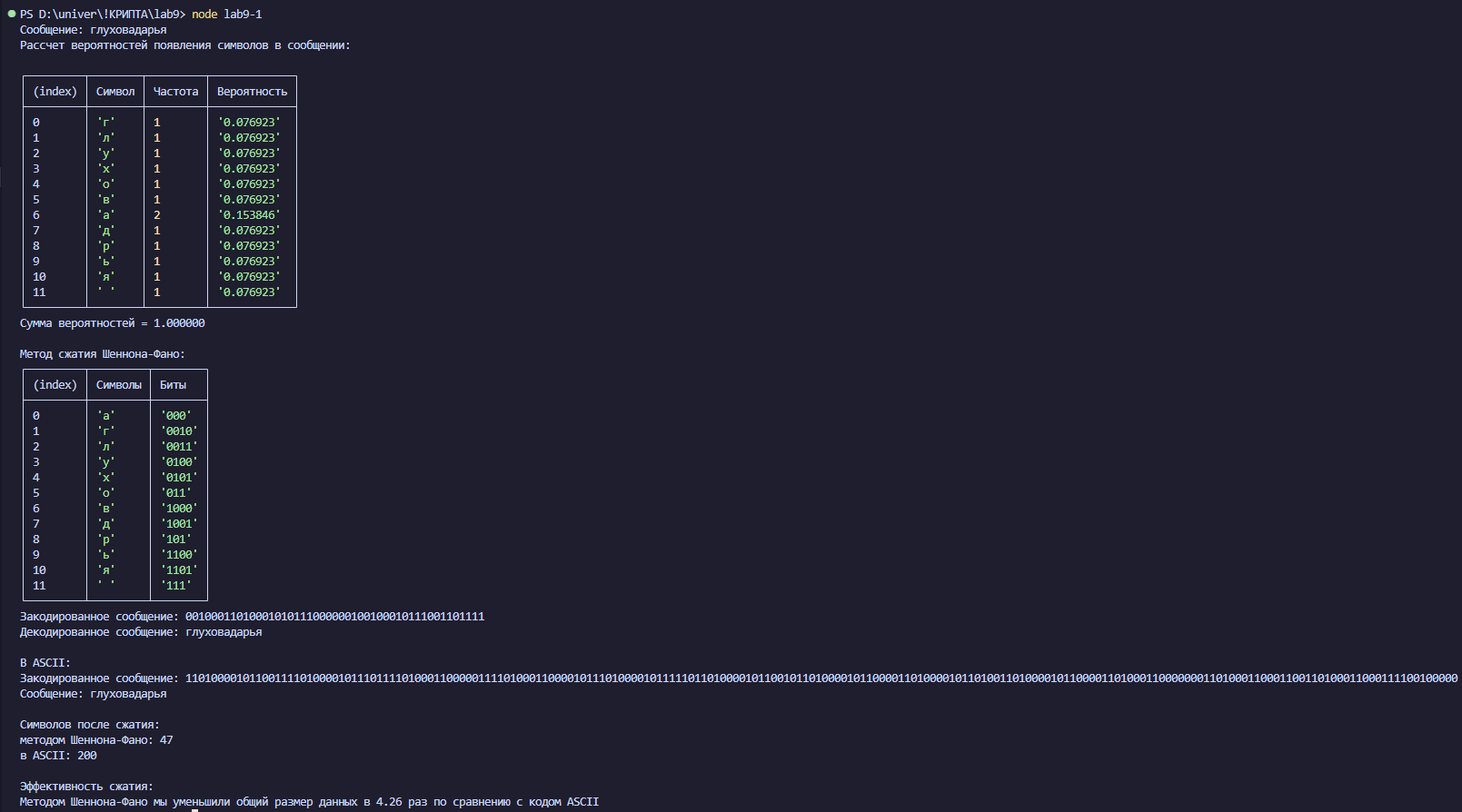


Рисунок 1.4 – Результат для Шеннона – Фано



Рисунок 1.5 – Результат для Хаффмана

**Вывод**

Метод Хаффмана показал более высокую эффективность сжатия, уменьшив объём данных в 4.57 раза, тогда как метод Шеннона-Фано — в 4.26 раза. Следовательно, для данного сообщения метод Хаффмана оказался предпочтительнее.