Учреждение образования «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа №8

Метод Хаффмана

Выполнила: Михнюк А.В.

Проверил: Шиман Д.В.

Минск, 2017

Оглавление

[Цели работы 3](#_Toc484040822)

[Теоретическое введение 4](#_Toc484040823)

[Реализация алгоритма 5](#_Toc484040824)

[Вывод 8](#_Toc484040825)

# Цели работы

В ходе лабораторной работы необходимо реализовать вероятностный (статический) метод сжатия Хаффмана.

# Теоретическое введение

Один из первых алгоритмов эффективного кодирования информации был предложен Д. А. Хаффманом в 1952 году. Идея алгоритма состоит в следующем: зная вероятности символов в сообщении, можно описать процедуру построения кодов переменной длины, состоящих из целого количества битов. Символам с большей вероятностью ставятся в соответствие более короткие коды. Коды Хаффмана обладают свойством префиксности (то есть ни одно кодовое слово не является префиксом другого), что позволяет однозначно их декодировать.

Классический алгоритм Хаффмана на входе получает таблицу частот встречаемости символов в сообщении. Далее на основании этой таблицы строится дерево кодирования Хаффмана (Н-дерево).

1. Символы входного алфавита образуют список свободных узлов. Каждый лист имеет вес, который может быть равен либо вероятности, либо количеству вхождений символа в сжимаемое сообщение.
2. Выбираются два свободных узла дерева с наименьшими весами.
3. Создается их родитель с весом, равным их суммарному весу.
4. Родитель добавляется в список свободных узлов, а два его потомка удаляются из этого списка.
5. Одной дуге, выходящей из родителя, ставится в соответствие бит 1, другой — бит 0. Битовые значения ветвей, исходящих от корня, не зависят от весов потомков.
6. Шаги, начиная со второго, повторяются до тех пор, пока в списке свободных узлов не останется только один свободный узел. Он и будет считаться корнем дерева.

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

Для декодирования можно воспользоваться построенным деревом. Начинаем с корня дерева и в качестве текущего бита берем начало текста.

В цикле делаем следующее

1) Смотрим, какое значение у текущего бита, и идем в низ по дереву по дуге, на которой указано такое же значение.

2) Переходим к следующему биту.

3) Если сейчас находимся в листе дерева: читаем символ находящийся в этом листе и записываем его в результат декодирования, переходим снова в корень дерева.

# Реализация алгоритма

В реализации программного средства в качестве языка программирования использовался C#.

Прямое преобразование - рисунок 1. Необходимо ввести сообщение, нажать кнопку «Сжать». После чего заполняться поля «Вероятности» и «Код», а также сжатое сообщение в поле «Итоговый результат».

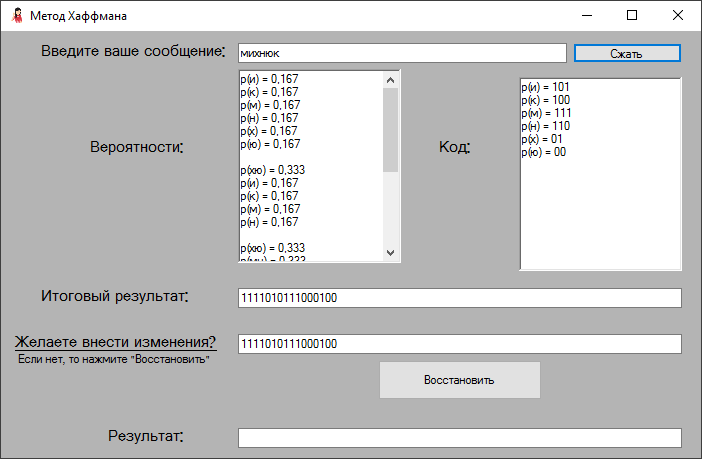


Рисунок 1 – Прямое преобразование

Обратное преобразование продемонстрировано на рисунке 2. Результат безошибочной работы алгоритма.

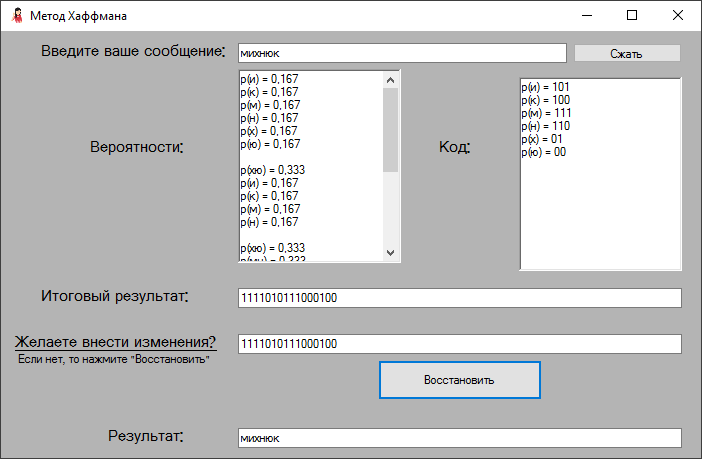


Рисунок 2– Обратное преобразование

На рисунке 3 показан результат выполнения алгоритма с внесением изменений в итоговый результат.

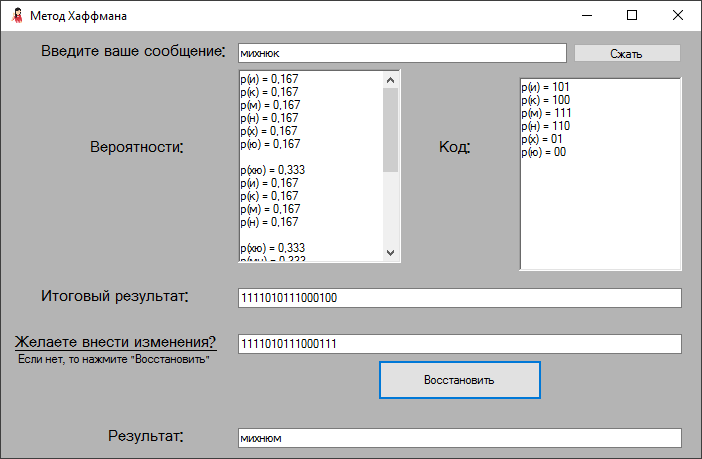


Рисунок 3 – Изменение последовательности

На рисунке 4 показан результат выполнения программы с дополнением кодовой последовательности.

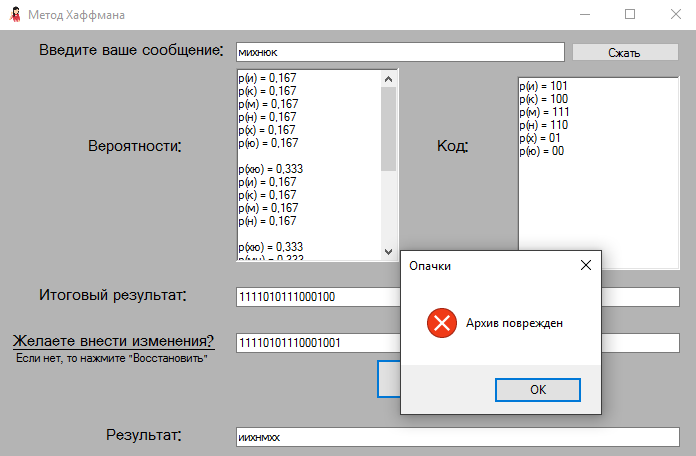


Рисунок 4 – Добавление несуществующего кода

# **Вывод**

В ходе выполнения лабораторной работы было разработано программное средство, которое реализовывает вероятностный метод сжатия Хаффмана.