Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе № 5

по курсу «Защита информации»

на тему: «Программная реализация алгоритма Хаффмана для сжатия данных»

Вариант № 2

Студент ИУ7-73Б (Группа) _____ (Подпись, дата) Марченко В. (И. О. Фамилия)

Преподаватель _____ (Подпись, дата) Чиж И. С. (И. О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 Алгоритм Хаффмана 4

2 Требования к входным данным 6

3 Тестирование программного обеспечения 7

ЗАКЛЮЧЕНИЕ 8

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 9

ВВЕДЕНИЕ

Алгоритм Хаффмана — жадный алгоритм оптимального префиксного кодирования алфавита с минимальной избыточностью. Был разработан в 1952 году аспирантом Массачусетского технологического института Дэвидом Хаффманом при написании им курсовой работы. В настоящее время используется во многих программах сжатия данных [1].

Целью данной лабораторной работы является программная реализация алгоритма Хаффмана для сжатия данных.

Задачи лабораторной работы:

1) изучить принцип работы алгоритма Хаффмана;

2) разработать программное обеспечение для сжатия и восстановления файлов с помощью алгоритма Хаффмана;

3) протестировать разработанное программное обеспечение.

1 Алгоритм Хаффмана

Идея алгоритма состоит в следующем: зная вероятности появления 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 частотностей символов в сообщении. Далее на основании этой таблицы строится дерево кодирования Хаффмана (Н-дерево) по следующему алгоритму.

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.

Допустим, у есть следующая таблица абсолютных частотностей: А — 15 раз, Б — 7 раз, В — 6 раз, Г — 6 раз и Д — 5 раз. Этот процесс можно представить как построение дерева, корень которого — символ с суммой вероятностей объединенных символов, получившийся при объединении символов из последнего шага, его n_0 потомков — символы из предыдущего шага и т. д.

Чтобы определить код для каждого из символов, входящих в сообщение, нужно пройти путь от листа дерева, соответствующего текущему символу,

до его корня, накапливая биты при перемещении по ветвям дерева (первая ветвь в пути соответствует младшему биту). Полученная таким образом последовательность битов является кодом данного символа, записанным в обратном порядке.

Для данной таблицы символов коды Хаффмана будут выглядеть следующим образом: А — 0, Б — 100, В — 101, Г — 110 и Д — 111.

Поскольку ни один из полученных кодов не является префиксом другого, они могут быть однозначно декодированы при чтении их из потока. Кроме того, наиболее частый символ сообщения А закодирован наименьшим количеством бит, а наиболее редкий символ Д — наибольшим.

При этом общая длина сообщения, состоящего из приведенных в таблице символов, составит 87 бит (в среднем 2.2308 бита на символ). При использовании равномерного кодирования общая длина сообщения составила бы 117 бит (ровно 3 бита на символ). Энтропия источника, независимым образом порождающего символы с указанными частотностями, составляет 2.1858 бита на символ, то есть избыточность построенного для такого источника кода Хаффмана, понимаемая как отличие среднего числа бит на символ от энтропии, составляет менее 0.05 бита на символ.

2 Требования к входным данным

Программа принимает три аргумента командной строки.

1. Первый аргумент — опция (-c для сжатия и -d для декомпрессии).
2. Второй аргумент — путь к файлу, который необходимо сжать/восстановить.
3. Третий аргумент — путь к сжатому/восстановленному файлу.

При наличии ошибок в аргументах командной строки или при передаче на вход программе пустого файла программа выдаст сообщение об ошибке и завершится.

Программное обеспечение для сжатия и восстановления файлов с помощью алгоритма Хаффмана было написано на языке программирования C.

Программа может сжимать/восстанавливать файлы любых типов.

Ограничение: алгоритм сжатия не будет работать в случае, если сжимаемый файл состоит из одного и того же символа. Для корректной работы алгоритма в файле должно быть хотя бы два разных символа.

3 Тестирование программного обеспечения

В таблице 3.1 приведены тесты для проверки корректности работы реализованного программного обеспечения.

Таблица 3.1 – Тесты

Описание	**Размер исходного файла, Б**	**Размер сжатого файла, Б**
Пустой входной файл	0	Error: input file is empty.
Кол-во аргументов командной строки не равно трем	0	Error: program requires 3 parameters.
Первый аргумент неправильный	0	Error: incorrect option.
Рисунок формата JPG (небо)	5127	5054
Рисунок формата PNG (красный прямоугольник)	6958	2320
Архив формата RAR	216	156

Все тесты пройдены успешно.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы был реализован алгоритм Хаффмана для сжатия данных.

Были выполнены следующие задачи:

1) изучен принцип работы алгоритма Хаффмана;
2) разработано программное обеспечение для сжатия и восстановления файлов с помощью алгоритма Хаффмана;
3) протестировано разработанное программное обеспечение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Википедия*. Код Хаффмана. — 2023. — (Дата обращения: 13.12.2023). https://en.wikipedia.org/wiki/Huffman_coding.