МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент математического и компьютерного моделирования

РЕФЕРАТ
о практическом задание по дисциплине АИСД
«Алгоритм сжатия информации арифметическое кодирование»

направление подготовки 09.03.03 «Прикладная информатика»
профиль «Прикладная информатика в компьютерном дизайне»

Выполнил студент
гр. Б9121-09.03.03пикд
Безрукова Анастасия Леонидовна
_____
(подпись)
Руководитель практики
Доцент ИМКТ А.С Кленин
(должность, уч. звание)
_____
(подпись)
«_____» _____ 2022г.

Реферат защищен:
С оценкой _____

Рег. № _____

«____» _____ 2022 г.

г. Владивосток
2023

Глосарий

1. Вероятность (встречаемость) – кол-во повторений символов делённое на общее кол-во символов в сообщении.
2. Интервал символа – верхняя и нижняя границы символа.

Аннотация

Арифметическое кодирование — один из алгоритмов энтропийного сжатия.

При арифметическом кодировании текст представляется вещественными числами в интервале от 0 до 1. По мере кодирования текста, отображающий его интервал уменьшается, а количество битов для его представления возрастает. Очередные символы текста сокращают величину интервала исходя из значений их вероятностей. Более вероятные символы делают это в меньшей степени, чем менее вероятные, и, следовательно, добавляют меньше битов к результату.

Заданное множество символов — это, как правило, ASCII+. Для того, чтобы обеспечить остановку алгоритма распаковки вначале сжимаемого сообщения надо поставить его длину или ввести дополнительный символ-маркер конца сообщения.

Постановка задачи

Задача разделяется на следующие пункты:

1. Найти и проанализировать различные источники, в которых есть информация об арифметическом кодировании.
2. Описать алгоритм в форме научного доклада.
3. Реализовать алгоритм.
4. Сделать анализ производительности алгоритма.

Авторы и история

Базовые алгоритмы арифметического кодирования были разработаны независимо Йорма Дж. Риссаненом из IBM Research и Ричардом К. Паско, аспирантом Стэнфордского университета; оба были опубликованы в мае 1976 года.

Описание алгоритма

Кодирование:

1. На вход программы поступает сообщение и его длина - вводится либо с клавиатуры, либо из файла.
2. Из полученного сообщения создается алфавит - массив символов, исключающий повторы.
3. Для каждого символа определяется его интервал, равный вероятности его появления.
4. Вызывается функция, которая пересчитывает границы интервала каждого символа, в соответствии с вероятностями символов.

Декодирование:

1. На вход поступает закодированное сообщение.
2. Вызывается функция декодирования, которая по итоговым интервалам символов расшифровывает сообщение.

Пример работы

Составим таблицу интервалов для символов входного сообщения, указав в ней частоты символов (Таблица 1).

Символ алфавита	Вероятность символа	Диапазон для символа (границы)	
		Low	High
A	¼=0,25	0	0,25
B	¼=0,25	0,25	0,5
C	¼=0,25	0,5	0,75
E	¼=0,25	0,75	1

Таблица 1 - таблица интервалов и вероятностей символов

Выполним кодирование слова «СЕВА».

Начальные условия: Low=0, High=1.

Для символа «С»:

L1= Low + L[C] (High- Low) = 0+0,5(1-0) = 0,5

H1= Low + H[C] (High-Low) = 0+0,75(1-0) = 0,75

Для символа «Е»:

L2= Low + L[E] (H1- L1) = 0,5+0, 5(0,75-0,5) = 0,6875

H2= Low + H[E] (H1- L1) = 0,5+1(0,75-0,5) = 0,75

Для символа «В»:

L3= L2 + L[B] (H2- L2) = 0,6875+0,25(0,75-0,6875) = 0,703125

H3= L2 + H[B] (H2- L2) = 0,6875+0,5(0,75-0,6875) = 0,7185

Для символа «А»:

L4 = L3 + L[A] (H3- L3) = 0,703125+0(0,71875-0,703125) = 0,703125

H4= L3 + H[A] (H3- L3) = 0,703125+0,25(0,71875-0,703125) = 0,7070312

Таким образом, число 0,703125 однозначно кодирует сообщение «СЕВА».

Описание реализации

Реализация алгоритма состоит из 4 функций (***get_number (), get_code (), coding (), decoding ()***), а также главной функции ***main (),*** где, собственно, и вызываются функции.

Для начала через директиву ***#define*** объявим две глобальные переменные ***M*** и ***N***, где переменная ***M*** отвечает за длину сообщения (в нашем случае длина 100), и ***N*** отвечает за количество символов в используемом нами словаре (в нашем случае 4).

Далее определяем приватные переменные, присваиваем их классу ***suanshu*** и определяем публичные функции для данного класса. Публичные функции нужны для того, чтобы мы могли иметь доступ к ним из любого места кода.

~suanshu() – Это функция деструктора, для очистки памяти.

Функция ***get_number ()*** отвечает за получение символов (***number [i]***) и их вероятностей (***chance[i]***), которые мы вводим.

Функция ***get_code ()*** выполняет ввод длины кодируемого сообщения, чтобы программа знала, когда ей остановиться. Так же эта функция получает на ввод сам код сообщения, которое нам нужно закодировать.

Функция ***coding ()*** выполняет само кодирование сообщения, которое мы ввели. Сначала она обрабатывает первый введённый нами символ, оценивает его подстрочный индекс и находит его верхнюю и нижнюю границы.

Затем рассматривается второй символ. Если он равен предыдущему, то нижняя граница символа остаётся прежней и меняется только верхняя. Так же пересчитывается вероятность. Иначе если второй символ другой, то меняется и нижняя и верхняя границы, так же меняется и вероятность. В этой же функции выводится результат кодирования, в нашем случае это нижняя граница.

Функция ***decoding ()*** отвечает за декодирование нашего сообщения. В начале объявляем переменную типа char, которая отвечает за количество декодируемых символов.

Далее расшифровываем символы по одному, постоянно меняя интервал кодирования, исходя из полученного результата в предыдущей функции (нижней границы ***Low***).

Последняя главная функция ***main ()*** отвечает за вызов всех функций.

Тестирование

Список литературы

[1] Arithmetic coding. URL: https://translated.turbopages.org/proxy_u/en-ru.ru.fbaf1e08-63ce7c75-d7985506-74722d776562/https/en.wikipedia.org/wiki/Arithmetic_encoder#History_and_patents

[2] Арифметическое кодирование. URL: Арифметическое кодирование — Википедия (wikipedia.org)

[3] Алгоритмы сжатия. URL: https://mf.grsu.by/UchProc/livak/po/comprsite/theory_arithmetic.html

[4] Арифметическое кодирование. URL: Арифметическое кодирование (helpiks.org)

[5] Идея арифметического кодирования. URL: Идея арифметического кодирования (manual.ru)