**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Сибирский государственный университет науки и технологий   
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедры информационно-управляющих систем

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Теория информации

|  |
| --- |
| Определение избыточности сообщений. Оптимальное неравномерное кодирование. |

Руководитель А.Н. Бочаров

подпись, дата инициалы, фамилия

Обучающийся БПИ22-02, 221219037 М.А. Соколов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2024 г.

**ЦЕЛЬ РАБОТЫ**

Закрепление знаний по методам кодирования информации.

**постановка задачи**

1. Используя текстовый файл из лабораторных работ 1–2, определить:   
 – избыточность заданного текста, вызванную неравновероятностью появления символов в сообщении Dp;   
 – избыточность, вызванную статистической связью между соседними символами Ds;   
 – полную избыточность D.   
 2. Используя этот же текстовый файл, построить оптимальный неравномерный код, применяя:   
 – метод Шеннона–Фано;   
 – метод Хаффмана.   
 3. Определить для каждого метода среднюю длину символа исходного алфавита lср, коэффициент статистического сжатия Kcc, коэффициент относительной эффективности Kоэ.   
 4. Написать программу кодирования и декодирования исходного текста методом Шеннона–Фано и методом Хаффмана. В отчете привести отрывок текста объемом не менее 4-х абзацев, закодированного и декодированного обоими методами.

**ХОД РАБОТЫ**

**Задание 1.**

Избыточность текста, вызванная неравновероятностью появления символов в сообщении находиться по формулe: . Пользуясь результатами прошлых вычислений: H(X) = 4,323, , получаем .

Теперь найдем избыточность, вызванную статистической связью между соседними символами . Пользуясь результатами прошлых вычислений: H(Y/X) = 3,5005, получаем

Полная избыточность

**Задание 2.**

Для выполнения лабораторной работы была написана программа на языке Python. С помощью разработанной программы был закодирован алфавит методом Шеннона-Фано и Хаффмана. Код программы:

from collections import defaultdict

import heapq

# чтение вероятностей символов из файла

def read\_probs(file\_name):

probs = {}

with open(file\_name, 'r', encoding='utf-8') as file:

for line in file:

sym, prob = line.strip().split()

probs[sym] = float(prob)

return probs

# кодировка текста

def encode\_text(text, codes):

encoded\_text = ''

for char in text:

encoded\_text += codes.get(char, '')

return encoded\_text

# построение кодов Хаффмана

def huffman\_method(probs):

heap = [[weight, [sym, ""]] for sym, weight in probs.items()]

heapq.heapify(heap)

while len(heap) > 1:

lo = heapq.heappop(heap)

hi = heapq.heappop(heap)

for p in lo[1:]:

p[1] = '0' + p[1]

for p in hi[1:]:

p[1] = '1' + p[1]

heapq.heappush(heap, [lo[0] + hi[0]] + lo[1:] + hi[1:])

huffman\_codes = dict(heapq.heappop(heap)[1:])

return huffman\_codes

# построение кодов Шеннона-Фано

def sf\_method(probs):

sorted\_probs = sorted(probs.items(), key=lambda x: x[1], reverse=True)

codes = defaultdict(str)

def build\_code(syms, start, end):

if start == end:

return

if start + 1 == end:

codes[syms[start]] += '0'

codes[syms[end]] += '1'

return

total\_prob = sum(probs[sym] for sym in syms[start:end+1])

cumulative\_prob = 0

index = start

for i in range(start, end+1):

cumulative\_prob += probs[syms[i]]

if cumulative\_prob >= total\_prob / 2:

index = i

break

for i in range(start, index+1):

codes[syms[i]] += '0'

for i in range(index+1, end+1):

codes[syms[i]] += '1'

build\_code(syms, start, index)

build\_code(syms, index+1, end)

syms = [sym for sym, \_ in sorted\_probs]

build\_code(syms, 0, len(syms)-1)

return dict(codes)

# декодирование текста с использованием кодов Шеннона-Фано

def decode\_text\_shannon\_fano(encoded\_text, codes):

reversed\_codes = {code: sym for sym, code in codes.items()}

decoded\_text = ''

code\_buffer = ''

for bit in encoded\_text:

code\_buffer += bit

if code\_buffer in reversed\_codes:

decoded\_text += reversed\_codes[code\_buffer]

code\_buffer = ''

return decoded\_text

# декодирование текста с использованием кодов Хаффмана

def decode\_text\_huffman(encoded\_text, codes):

reversed\_codes = {code: sym for sym, code in codes.items()}

decoded\_text = ''

code\_buffer = ''

for bit in encoded\_text:

code\_buffer += bit

if code\_buffer in reversed\_codes:

decoded\_text += reversed\_codes[code\_buffer]

code\_buffer = ''

return decoded\_text

# чтение вероятностей символов из файла

probs = read\_probs('probs.txt')

# построение кодов Шеннона-Фано

codes = sf\_method(probs)

# кодировка текста с использованием построенных кодов

input\_text = open("text.txt", encoding='utf-8').read()

encoded\_text = encode\_text(input\_text, codes)

# сохранение закодированного текста и кодов в файл

with open('encoded\_text.txt', 'w', encoding='utf-8') as file:

file.write(encoded\_text + '\n')

with open('codes\_fano.txt', 'w', encoding='utf-8') as file:

for sym, code in codes.items():

file.write(f"{sym}: {code}\n")

# построение кодов Хаффмана

codes\_huffman = huffman\_method(probs)

# кодировка текста с использованием построенных кодов

input\_text = open("text.txt", encoding='utf-8').read()

encoded\_text\_huffman = encode\_text(input\_text, codes\_huffman)

with open('encoded\_text\_huffman.txt', 'w', encoding='utf-8') as file:

file.write(encoded\_text\_huffman + '\n')

with open('codes\_huffman.txt', 'w', encoding='utf-8') as file:

for sym, code in codes.items():

file.write(f"{sym}: {code}\n")

# декодирование закодированного текста Шеннона-Фано

encoded\_text\_sf = open("encoded\_text.txt", encoding='utf-8').readline().strip()

decoded\_text\_sf = decode\_text\_shannon\_fano(encoded\_text\_sf, codes)

with open('decoded\_text\_sf.txt', 'w', encoding='utf-8') as file:

file.write(decoded\_text\_sf + '\n')

# декодирование закодированного текста Хаффмана

encoded\_text\_huffman = open("encoded\_text\_huffman.txt", encoding='utf-8').readline().strip()

decoded\_text\_huffman = decode\_text\_huffman(encoded\_text\_huffman, codes\_huffman)

with open('decoded\_text\_huffman.txt', 'w', encoding='utf-8') as file:

file.write(decoded\_text\_huffman + '\n')

Для построения кодов использовались вероятности появления букв, показанные в таблице 1.

Таблица 1 — Частота появления символов в тексте

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Буква | Вероятность | Буква | Вероятность | Буква | Вероятность | Буква | Вероятность |
| Пробел | 0,171 | р | 0,042 | я | 0,024 | х | 0,008 |
| о | 0,084 | в | 0,036 | ы | 0,018 | ж | 0,003 |
| е | 0,069 | л | 0,041 | з | 0,016 | ю | 0,004 |
| а | 0,070 | к | 0,034 | ъ | 0,001 | ш | 0,008 |
| и | 0,055 | м | 0,028 | б | 0,016 | ц | 0,001 |
| н | 0,056 | д | 0,025 | г | 0,013 | щ | 0,002 |
| т | 0,050 | п | 0,026 | ч | 0,013 | э | 0,002 |
| с | 0,045 | у | 0,025 | й | 0,012 | ф | 0,001 |
| ь | 0,020 |  |  |  |  |  |  |

Результаты кодирования методом Шеннона-Фано отображены в таблице 2.

Таблица 2 – Результат кодирования методом Шеннона-Фано

|  |  |
| --- | --- |
| Буквы | Код |
| 1 | 2 |
| пробел | 000 |
| о | 0010 |
| а | 0011 |
| е | 0100 |
| н | 0101 |
| и | 0110 |
| т | 0111 |
| с | 10000 |
| р | 10001 |
| л | 1001 |
| в | 10100 |
| к | 10101 |
| м | 10110 |
| п | 10111 |
| д | 110000 |
| у | 110001 |
| я | 11001 |
| ь | 11010 |
| ы | 11011 |
| з | 111000 |
| б | 111001 |
| г | 111010 |
| ч | 111011 |
| й | 1111000 |
| х | 1111001 |
| ш | 111101 |
| ю | 11111000 |
| ж | 11111001 |
| щ | 1111101 |
| э | 11111100 |
| ъ | 11111101 |
| ц | 11111110 |
| ф | 11111111 |

Результаты кодирования методом Хаффмана предоставлены в таблице 3.

Таблица 3 – Результат кодирования методом Хаффмана

|  |  |
| --- | --- |
| Буквы | Код |
| 1 | 2 |
| ь | 00000 |
| я | 00001 |
| с | 0001 |
| д | 00100 |

Продолжение таблицы 3

|  |  |
| --- | --- |
| 1 | 2 |
| й | 001010 |
| г | 001011 |
| т | 0011 |
| у | 01000 |
| п | 01001 |
| и | 0101 |
| ч | 011000 |
| ж | 01100100 |
| ъ | 011001010 |
| ф | 0110010110 |
| ц | 0110010111 |
| х | 0110011 |
| м | 01101 |
| н | 0111 |
| б | 100000 |
| з | 100001 |
| к | 10001 |
| е | 1001 |
| а | 1010 |
| ш | 1011000 |
| щ | 101100100 |
| э | 101100101 |
| ю | 10110011 |
| ы | 101101 |
| в | 10111 |
| л | 11000 |
| р | 11001 |
| о | 1101 |
| пробел | 111 |

**Задание 3.**

**По Шеннону-Фано:**

Средняя длина кода:

Lср = 4,524

Коэффициент статистического сжатия:

Ксс = = = 1,107

Коэффициент относительной эффективности:

Коэ = = = 0,944

**По Хаффману:**

Средняя длина кода:

Lср = 4,495

Коэффициент статистического сжатия:

Ксс = = = 1,14

Коэффициент относительной эффективности:

Коэ = = = 0,972

**Задание 4.**

Для кодирования был выбран данный фрагмент текста. Предварительно были удалены все лишние символы:

У Маньяка я просидел до позднего вечера. "Гиннес" сменился "Балтикой" номер шесть, а на десерт Шурка откопал банку рождественского "Кроненбурга". Ни ирландское, ни питерское, ни французское пиво не подкачали.

В глубине души я был рад, что хоть кому-то открылся. Мои друзья-хакеры делятся на две группы - одна хранит тайны до первой бутылки пива, вторая, после этой самой бутылки, ее как бы забывает. Шурка - из второй.

По крайней мере теперь он будет знать, для чего мне весь разнообразный вирусный софт, который я правдами и неправдами выманиваю у него.

Насколько проще было бы, не затягивай глубина так сильно, думал я в такси по дороге к дому. Насколько правильнее и легче.

Не было бы деления на счастливчиков и неудачников, которое ничем не сломать. Не было бы этого безумия - великолепных программистов, неспособных перейти грань между иллюзией и явью, и неумех вроде меня, не замечающих этого барьера.

Не было бы зависти друг к другу - и вечной охоты.

Но разве я виноват? Я и сам не знаю, почему так происходит, какая ошибка сознания, а ведь это именно ошибка - мы в меньшинстве, - делает из человека дайвера. Не пользоваться своей способностью глупо. Предлагать ее для всеобщего изучения - страшно.

Текст, закодированный методом Шеннона-Фано:

0011010111010110011010100111100110111100010010100000110110000010010011100000010101110010111000110000010101001110100010101000100111011010010001001101100101010101001000010000101100100010101101001100001100100111001011101101010100101111000010100101011001001000111110101001000001111101000110101001111000001001000001001000101111100011000110101001100100111101010010101110011100111100100110101101011100011000100101111100111000001001000001111010001000101100001010100101110100010100010010010101000101111001110001100011110100011011001101000110010011010111000010000101010010010001010110101110110011101001000110000101010010010001010110111111111000100110101111111101100011110001000010101001001001011101101010000100101010010111001011000010101001111101100111001011011101010011100011110010110010101001100001100011111010110110011110011101110011000100111100001110110111001011110010010011111010101010010101101100010111001000100111101011000111011100110000110010010011011000010001110001111000110101100111110010011101010100100011101111000001001001110010111100001100101010011110000101000100111010100011100011011110111110110010110000010100111111001100010011010101100111011100111111000010111011110000001010111010010001101000010111100011100111000101111101110011010101101011101101010000111010001110010100010011110011011100101000010010100111111000111001011110001000000111011000101111000111001110001011111011100110101011001000100101010011101011110011101111100000111110011101110100001101000111110001100011010100110110111000101000111001010001001011110000010101011000100111111000010101001111000101100100100010100011101001011101001000111010001001011110011100011100000100011111100001010011011111010110000100111001111011010011101000101011001010100101000100100001101010001001111100001010010001011100110001001111100001011101111110001010001101000111000110000010111011111100010000001011111111011110101001001110010100011101111110001100110111100010011101001100000011101100110011001010100101111000100111010011000000111011001101010011011101100011010101101010000111111100011000101010100111010001000111000010101001010011101010101001010111100010010111110101001110011101110010010111001110110101010011100000110111110011110100110101000011111100011101010011100011110010110010100110111001110101100000110100111010010100101100001100011011000111001110011010001110011101011000001101011100101100000010100010010111010010010101110000001010110110001001110000101010010100111010101010010101111000100111010001101001110100101010001000110100101001110101110110100010011100111011100100101110011101111000001001001010001010110110010101001110000111011001110000011110010110101001110110110101010010101000110010101001100011100000011111011010101101010100101010010101001001110010100010010010001010110111011010010110010101001000010010010101100011011111010010011100111011100100101110011101111111100011100101110100010111001010011100011000110110011011001101000100100101101010100101001010010111010111011111100110111100010010111010100010011101101011001101000001110010101000101010010000101110010100000010111001010111011111100110111010010001010011110000111011011101010001001101011101010110010011111001110000110001011010011001111110001110000110010011110000110110011010011010111110000110010101001100011011001001111001101001000100101100000100101100100010111001010101001110000011101100100111011001111111000111110101101111001111111000111001011101000101110010011100011101001001000100110100111001110111001001011100111011111000001110100011010000011101101100001000111000111101010101110000100011100011110101100010110101000100111011010100101111000001011110010010011111011001010001001111100010100010011001101000110010100101010000110111011010000001110110010101001110000101001111111000101110010111011010010110110001011100111010110111100010010011010000111100100101100000110011110101001110101001111001001011110101101110011010100111000000101110000101001101010110110010011101000100110000110101111110001110010011010110010001010101001000101111010110111001101010011101101101110100101100100010111010111101011001011000001111010001001100000100100100110100011101101110001110110100100100101010001001010100111100000011111100010100010010001001101001011100101001110101110000010101000011011111010100001100110000101000010010011110001000010111001010000001011100101010010100000111110101111100011101010011100011011100101000101001100001001001111101000110111110100100010011000010011100110100100000100001011100111111010100111010001001101110001100011110110100010101101100110000011110001001111110101010010

Текст, закодированный методом Хаффмана:

1010011100000000011000110100000101001110011101000101010010010011100000100110101001110110000100100011110010010111101101111001011000100111001101001010111011110010001000101101100101110101110000001000011010110000011010110001110100101001111101011011001110011011000100100010011000001010011110100010010010001100111001001101000110011000110101101001110001110101001101011000100000101001111000101000110011101011001000010010010001001110111100101110001100011101001011110111001110101111001011110000001000110010010111010010101011100111000101001110010000011000111011001011101010100101010011100111001000110001110110010111010101100101101100110100111011001011101000100001000110001110110010100101011011111010111100101001110100100100011010011000101011000010100101111000010001000000101011110010010001000101100001010000110000010110111000110011010001000110000011110101100111101001100000100011101011010100000111101110100111000111001101101110000001000011101010100100110010100010000100000000010110011101010001100111001101101001001001110000000100110001000010111101000100101111001001011110010100001001010011011011101001000111101001100111100110100111010100110011101000101001111011010010011010100110011100110111110100101010000001000001110110111000100010101010010101101111010101110011110111001101000001010011101000111000100110110010100111101001010000110100110111010010101000000100000111011011100010001010110011001100011010100011000001011011000011010100000101101101111010100100110100011001100011010010110000110111001111011100111010010101101100011100110100010100111100100101001101100111001100100111001010011001110010000011010111100000010000010010010011100001011110100011000000010011000000010110001001001011110101101011110011011110010001000001100110101000010111110111011000001100110101000010111101101001010101110101110010100000010111101101001010000111010110010110001110001110100111101110011011010010100000101001110011010101110010010100110101010101011110010100111001101010111001001010011010101101111011010110110100111010110111101010110011010000111100100101111011010000110001110111000000001000111010100111001110110110010010011000001011011100011011000001011010111100110000110100011000010010110101101111010001010001011110000100010000001010111101000111010100010001010111000000000111110100100010000110110101100000001101110011101010001000101010100111010010011011100111010010111001100010010011010110101000101000011000111011100000000100011101010011100110101011101011100000000011110011001010111000100100101101100010011001100000101101110001101100000101101001001001110001001011101010000101111010000101100010100001001111000010110111011000010110001110110111010101111001010000010010100110000111010110001110110111100011101001111011100111011001011101010110001001011010111100100011100011010110110100011000001001100000101101110001101100000101101101100101001111010010111101100000100110000101000011010101000011011110011100001011000111011100010010100101111011010110011010011100111010010111100110100110101101010100010011110110111011110010001010011101000111011000000111101101011001101001100111001100100101000110101001011110011010011100000011011001011001000010001000010111000110001011001110000101011001001010010100001101110000010110011010101111001010000110110010110011101111100111010010010010110110010111000010111100110000110100110110010110001010101100111011001000101011001110110010100111101001011110110000010101100100000100111001101010011000001011011100011011000001011011000011010101110101000100110101001001100101000001011100010010011001010000010110100001011011110010110000111110100101011010110011110100111011011101110011010100001101111001000011011101010111110110111101000110101000110100110101111001100001011110101011001101001110101100010010110101000001110101000101001110011101010100010110011110100100010100111000110101000110100000111011011000010110000010001101000011101100001011110100111010100001101010111100100100000001011001010011110101010110110010111011111011101101100001011000001000110100110110110110111011011001011100000101100001010111000100111011110010010010011100010101001001101011000010110001001110001101101111001100011010001001010001010101111001110011010100101001110111000000001000011101101111010001100000000100001000110111110110010010100001010011101000111011000000111110100010011000001011001100101111000010000100111011100110010010011000101000101110100011000001001100100100110000000110111000110011101100000101100100100100101111010101100001010000110001001011101010000100010011110011010101100001111101

Результат декодирования методом Шеннона-Фано:

у-маньяка-я-просидел-до-позднего-вечера-гиннес-сменился-балтикой-номер-шесть-а-на-десерт-шурка-откопал-банку-рождественского-кроненбурга-ни-ирландское-ни-питерское-ни-французское-пиво-не-подкачалив-глубине-души-я-был-рад-что-хоть-комуто-открылся-мои-друзьяхакеры-делятся-на-две-группы--одна-хранит-тайны-до-первой-бутылки-пива-вторая-после-этой-самой-бутылки-ее-как-бы-забывает-шурка--из-второйпо-крайней-мере-теперь-он-будет-знать-для-чего-мне-весь-разнообразный-вирусный-софт-который-я-правдами-и-неправдами-выманиваю-у-негонасколько-проще-было-бы-не-затягивай-глубина-так-сильно-думал-я-в-такси-по-дороге-к-дому-насколько-правильнее-и-легчене-было-бы-деления-на-счастливчиков-и-неудачников-которое-ничем-не-сломать-не-было-бы-этого-безумия--великолепных-программистов-неспособных-перейти-грань-между-иллюзией-и-явью-и-неумех-вроде-меня-не-замечающих-этого-барьеране-было-бы-зависти-друг-к-другу--и-вечной-охотыно-разве-я-виноват-я-и-сам-не-знаю-почему-так-происходит-какая-ошибка-сознания-а-ведь-это-именно-ошибка--мы-в-меньшинстве--делает-из-человека-дайвера-не-пользоваться-своей-способностью-глупо-предлагать-ее-для-всеобщего-изучения—страшно

Результат декодирования методом Хаффмана:

у-маньяка-я-просидел-до-позднего-вечера-гиннес-сменился-балтикой-номер-шесть-а-на-десерт-шурка-откопал-банку-рождественского-кроненбурга-ни-ирландское-ни-питерское-ни-французское-пиво-не-подкачалив-глубине-души-я-был-рад-что-хоть-комуто-открылся-мои-друзьяхакеры-делятся-на-две-группы--одна-хранит-тайны-до-первой-бутылки-пива-вторая-после-этой-самой-бутылки-ее-как-бы-забывает-шурка--из-второйпо-крайней-мере-теперь-он-будет-знать-для-чего-мне-весь-разнообразный-вирусный-софт-который-я-правдами-и-неправдами-выманиваю-у-негонасколько-проще-было-бы-не-затягивай-глубина-так-сильно-думал-я-в-такси-по-дороге-к-дому-насколько-правильнее-и-легчене-было-бы-деления-на-счастливчиков-и-неудачников-которое-ничем-не-сломать-не-было-бы-этого-безумия--великолепных-программистов-неспособных-перейти-грань-между-иллюзией-и-явью-и-неумех-вроде-меня-не-замечающих-этого-барьеране-было-бы-зависти-друг-к-другу--и-вечной-охотыно-разве-я-виноват-я-и-сам-не-знаю-почему-так-происходит-какая-ошибка-сознания-а-ведь-это-именно-ошибка--мы-в-меньшинстве--делает-из-человека-дайвера-не-пользоваться-своей-способностью-глупо-предлагать-ее-для-всеобщего-изучения--страшно

**ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. **Что лежит в основе определения избыточности?**

Наличие лишней информации в сообщении, что позволяет обнаруживать и исправлять ошибки при передаче данных.

1. **Какие частные избыточности вы знаете?**

Избыточность по Шеннону, которая обеспечивает надежность передачи, и избыточность алфавита, формируемую разными вероятностями символов.

1. **В чем заключается оптимальное неравномерное кодирование?**

В присвоении более длинных кодов символам с низкой вероятностью и более коротких кодов символам с высокой вероятностью для эффективной передачи данных.

1. **Как проводится кодирование информации методом Шеннона-Фано?**

Происходит разделение алфавита символов на две части с близкими по вероятности символами на каждом шаге кодирования.

1. **Как кодируется информация методом Хаффмана?**

Метод Хаффмана – это метод оптимального префиксного кодирования, при котором более часто встречающиеся символы получают более короткие коды.

1. **Как определяется средняя длина кода?**

Сумма произведений вероятности каждого символа на длину его кода в процессе кодирования.

1. **Как вычисляются коэффициенты статистического сжатия и коэффициент относительной эффективности?**

Коэффициент статистического сжатия – отношение исходного объема данных к объему данных после сжатия. Коэффициент относительной эффективности – отношение энтропии исходных данных к средней длине кода.

1. **Как формируется информационная модель канала связи?**

Информационная модель канала связи описывает его основные параметры, такие как скорость передачи данных, вероятность ошибки и пропускную способность.

1. **Как определяется пропускная способность канала связи?**

Пропускная способность канала связи определяется как максимальная скорость передачи данных без искажения информации.

1. **В чем заключаются недостатки оптимального кодирования?**

Высокие вычислительные затраты при построении кода для больших объемов данных и сложность процесса кодирования.

1. **В чем суть основной теоремы Шеннона о кодировании для дискретного канала без помех.**

Основная теорема Шеннона утверждает, что для дискретного канала без помех с заданной пропускной способностью существует кодирование, при котором скорость передачи данных может быть сколь угодно близкой к пропускной способности канала и вероятность ошибки может быть сделана сколь угодно малой.

1. **В чем суть теоремы Шеннона для побуквенного кодирования.**

Теорема Шеннона для побуквенного кодирования утверждает, что существуют коды, которые позволяют передавать информацию с минимальной вероятностью ошибки при условии, что скорость кодирования не превышает пропускной способности канала.

**ВЫВОДЫ**

Были освоены практические методики кодирования информации.

Также, можно утверждать, что процесс кодирования был выполнен оптимально и правильно, так как коэффициент статического сжатия по Шеннону-Фано и по Хаффману превышает 1, а коэффициент относительной эффективности по Шеннону-Фано и по Хаффману, наоборот, не превышает 1.