Лабораторная работа №3

Дисциплина: Основы кодирования

Тема: Основы кодирования информации. Равномерные и префиксные коды

1. Цель работы

Научиться строить кодовые деревья и двоичные коды для алфавита источника; реализовать равномерное (фиксированной длины) кодирование и префиксное кодирование по правилу Шеннона; вычислять среднюю длину кода, эффективность и выигрыш относительно равномерного кода. Работа выполняется на статистике источника из ЛР №1 (те же исходные данные).

2. Исходные данные

В данной лабораторной работе используются те же данные, что и в предыдущих работах. Студент не формирует новые данные.

Используются:
- источник информации, сформированный из ФИО студента и даты рождения;
- алфавит источника информации;
- вероятности появления символов;
- энтропия источника информации.

3. Теоретические сведения

3.1. Код, кодовое слово и длина кода

В теории информации кодирование рассматривается как отображение элементов исходного алфавита A в кодовые слова над другим алфавитом. Цель кодирования — представить информацию в форме, удобной для хранения, передачи или обработки.

В данной лабораторной работе используется двоичное кодирование, где кодовые слова состоят из символов $\{0,1\}$. Каждому символу $x_i \in A$ ставится в соответствие двоичное слово длины l_i.

3.2. Источник информации и статистическая модель

Источник информации моделируется как последовательность символов конечного алфавита. В лабораторной работе источником служит строка, составленная из ФИО и дат рождения, записанных без разделителей. Каждый символ рассматривается независимо от смысловой нагрузки.

Для статистического описания источника подсчитываются частоты появления символов n_i и вычисляются вероятности:

$p_i = n_i / N$

где N — общее количество символов в сообщении. Алфавит источника включает только те символы, для которых $n_i > 0$.

3.3. Энтропия и информационная эффективность

Энтропия источника информации характеризует среднее количество информации, приходящееся на один символ:

$H = -\sum p_i \log_2 p_i$
Энтропия задаёт теоретический предел сжатия информации. Ни один код не может иметь среднюю длину меньшую, чем энтропия источника.

3.4. Равномерное кодирование

Равномерное кодирование не учитывает статистику источника. Каждому символу алфавита назначается кодовое слово одинаковой длины.
Минимальная длина равномерного двоичного кода определяется из условия существования достаточного количества кодовых слов:
$L = \lceil \log_2 |A| \rceil$
Общий объём закодированного сообщения при равномерном кодировании равен:
$B\_uniform = N \cdot L$

3.5. Двоичное дерево кодов

Двоичное дерево кодов является наглядной моделью процесса кодирования. Каждая вершина дерева соответствует частичному коду, а каждый путь от корня к листу — полному кодовому слову.
Левое ребро соответствует добавлению бита 0, правое — бита 1. Если кодовое слово соответствует листу дерева, то код является префиксным.

3.6. Префиксные коды и условие Крафта (уровень B)

Префиксным называется код, в котором ни одно кодовое слово не является началом другого кодового слова. Префиксность гарантирует однозначное декодирование.
Для существования префиксного двоичного кода длины кодовых слов должны удовлетворять условию Крафта:
$\Sigma\ 2^{(-l_i)} \leq 1$

3.7. Код Шеннона и принцип оптимального кодирования (уровень C)

Принцип оптимального кодирования заключается в назначении более коротких кодовых слов символам с большей вероятностью появления.
В коде Шеннона длины кодовых слов определяются по формуле:
$l_i = \lceil -\log_2 p_i \rceil$
На основе рассчитанных длин строится префиксный код, который приближает среднюю длину кода к энтропии источника.

3.8. Средняя длина кода и оценка эффективности

Средняя длина кода характеризует среднее число бит, необходимых для кодирования одного символа:
$\bar{L} = \sum p_i \cdot l_i$
Для кода Шеннона выполняется фундаментальное неравенство:
$H \leq \bar{L} < H + 1$

3.9. Сравнение кодов и подготовка к оптимальному кодированию

Сравнивая равномерный и префиксный коды, можно оценить выигрыш от учёта статистики источника. Чем ближе средняя длина кода к энтропии, тем код эффективнее.
Оптимальный префиксный код с минимальной средней длиной строится алгоритмом Хаффмана, который рассматривается в последующих лабораторных работах.

4. Задания

1. Получить алфавит и вероятности p_i (по данным ЛР №1).

2. Построить равномерный двоичный код: $L = \lceil \log_2 |A| \rceil$ и таблицу «символ → код».
3. Определить общий объём сообщения при равномерном кодировании: B_uniform = N·L.
4. Описать кодовое дерево равномерного кода и показать декодирование 3 символов.
5. Построить префиксный код Шеннона: $l_i = \lceil -\log_2 p_i \rceil$, таблица «символ → l_i → код».
6. Проверить условие Крафта и объяснить префиксность через дерево.
7. Вычислить среднюю длину $\bar{L}$ и объём B_shannon для кода Шеннона.
8. Сравнить коды ($\bar{L}$, η, ΔB и % выигрыша).
9. Сделать выводы для перехода к оптимальному кодированию (Хаффман).

5. Пример

5.1. Алфавит и вероятности

N = 86, |A| = 29. Энтропия (контроль): H = 4.534286 бит/символ; Hmax = 4.857981; избыточность = 6.7%.

Символ	n_i	p_i	$-\log_2(p_i)$
А	12	0.139535	2.841302
0	7	0.081395	3.618910
Р	6	0.069767	3.841302
1	4	0.046512	4.426265
2	4	0.046512	4.426265
И	4	0.046512	4.426265
Л	4	0.046512	4.426265
Н	4	0.046512	4.426265
7	3	0.034884	4.841302
В	3	0.034884	4.841302
Е	3	0.034884	4.841302
С	3	0.034884	4.841302
Т	3	0.034884	4.841302
У	3	0.034884	4.841302
5	2	0.023256	5.426265
9	2	0.023256	5.426265
Г	2	0.023256	5.426265
М	2	0.023256	5.426265
О	2	0.023256	5.426265
П	2	0.023256	5.426265
Ч	2	0.023256	5.426265
Ы	2	0.023256	5.426265
3	1	0.011628	6.426265
4	1	0.011628	6.426265
Д	1	0.011628	6.426265
Й	1	0.011628	6.426265
К	1	0.011628	6.426265
Х	1	0.011628	6.426265
Я	1	0.011628	6.426265

5.2. Равномерный код

$L = \lceil \log_2 |A| \rceil = \lceil \log_2(29) \rceil = 5$ бит/символ.

Символ	Код (L=5)
0	00000
1	00001
2	00010
3	00011
4	00100
5	00101
7	00110
9	00111
А	01000
В	01001
Г	01010
Д	01011
Е	01100
И	01101
Й	01110
К	01111
Л	10000
М	10001
Н	10010
О	10011
П	10100
Р	10101
С	10110
Т	10111
У	11000
Х	11001
Ч	11010
Ы	11011
Я	11100

5.3. Объём при равномерном кодировании

Формула: $B_{uniform} = N \cdot L$

$B_{uniform} = 86 \cdot 5 = 430$ бит.

5.4. Дерево кода и декодирование (равномерный код)

Равномерный код задаётся полным двоичным деревом глубины L. Декодирование выполняется чтением каждых L бит как одного кодового слова.

Пример: 00000→0, 00001→1, 00010→2.

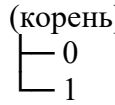
5.5. Код Шеннона (префиксный)

Формула длин: $l_i = \lceil -\log_2 p_i \rceil$. Далее строятся кодовые слова по кумулятивным вероятностям.

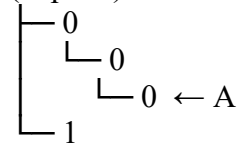
Символ	p_i	l_i	Код Шеннона
А	0.139535	3	000
0	0.081395	4	0010
Р	0.069767	4	0011
1	0.046512	5	01001
2	0.046512	5	01010
И	0.046512	5	01100
Л	0.046512	5	01101
Н	0.046512	5	01111
7	0.034884	5	10000
В	0.034884	5	10001
Е	0.034884	5	10010
С	0.034884	5	10100
Т	0.034884	5	10101
У	0.034884	5	10110
5	0.023256	6	101110
9	0.023256	6	110000
Г	0.023256	6	110001
М	0.023256	6	110011
О	0.023256	6	110100
П	0.023256	6	110110
Ч	0.023256	6	110111
Ы	0.023256	6	111001
3	0.011628	7	1110101
4	0.011628	7	1110111

Д	0.011628	7	1111000
Й	0.011628	7	1111010
К	0.011628	7	1111011
Х	0.011628	7	1111101
Я	0.011628	7	1111110

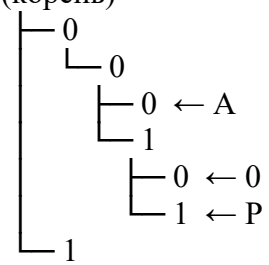
Пояснение к построению префиксного кода (на основе дерева кодов)
Префиксный код удобно представлять в виде двоичного дерева. Каждому символу соответствует путь от корня дерева до листа. Переход влево обозначается битом 0, переход вправо — битом 1.
Главное правило: каждому символу соответствует лист дерева. Если код заканчивается в листе, он не может быть префиксом другого кода.
В начале имеется только корень дерева. От него отходят две ветви: 0 и 1.



Самому вероятному символу назначается самый короткий путь от корня. В нашем примере символ «А» получает код 000.

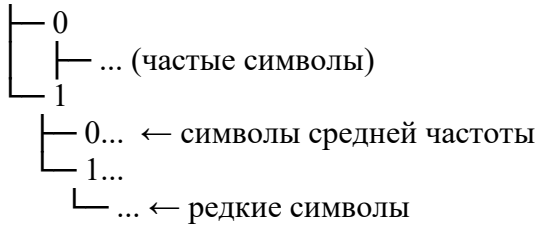


После этого путь 000 закрывается и не может быть использован для других символов.
Следующие символы получают коды большей длины, занимая свободные пути дерева, которые не пересекаются с уже занятыми.



Когда короткие пути заняты, новые символы размещаются глубже в дереве. Чем реже встречается символ, тем глубже расположен соответствующий лист.

(корень)



Редкие символы получают самые длинные коды и располагаются на наибольшей глубине дерева, например:
1110101 ← 3
1110111 ← 4
1111000 ← Д
1111010 ← Й
1111011 ← К
1111101 ← Х
1111110 ← Я

Поскольку все кодовые слова соответствуют листьям дерева, ни одно кодовое слово не может быть началом другого. Это гарантирует префиксность и однозначность декодирования.
Построение кода сводится к заполнению свободных листьев двоичного дерева. Самые короткие пути отдаются наиболее вероятным символам, а самые длинные — наименее вероятным.
Таким образом, код автоматически учитывает статистику источника и реализует принцип оптимального кодирования.

5.6. Проверка Крафта и префиксности
Условие Крафта: $\Sigma\ 2^{(-l_i)} \le 1$

$\Sigma\ 2^{(-l_i)} = 0.773438 \le 1$ (условие выполняется).

Префиксность: кодовые слова соответствуют листьям двоичного дерева; ни одно слово не является префиксом другого.

5.7. Средняя длина и объём для кода Шеннона
$\bar{L} = \Sigma\ p_i \cdot l_i$

$\bar{L} = 4.918605$ бит/символ

$B\_shannon = \Sigma\ n_i \cdot l_i$

B_shannon = 423 бит

Контроль неравенства: $H \le \bar{L} < H+1 \rightarrow 4.534286 \le 4.918605 < 5.534286$

5.8. Сравнение кодов (эффективность и выигрыш)
Равномерный код: $\bar{L} = 5.000000$, $\eta = H/\bar{L} = 0.906857$, объём = 430 бит.

Код Шеннона: $\bar{L} = 4.918605$, $\eta = H/\bar{L} = 0.921864$, объём = 423 бит.

Выигрыш: ΔB = 7 бит, что составляет 1.63%.

5.9. Выводы для перехода к Хаффману

Наиболее вероятные символы должны иметь самые короткие коды. Топ-5 по вероятности: A(p=0.1395), 0(p=0.0814), P(p=0.0698), 1(p=0.0465), 2(p=0.0465).

Для ЛР №6 (Хаффман) используется таблица частот n_i из Задания 1 как входные данные.

6. Требования к программе на Python

Студенту разрешается и рекомендуется дополнять и расширять программу, разработанную в лабораторных работах №1 и №2. Создание новой программы с нуля не является обязательным.

Программа должна:
- принимать алфавит и вероятности символов;
- строить равномерный двоичный код;
- строить префиксный код на основе дерева;
- вычислять длины кодовых слов и среднюю длину кода;
- выводить таблицу «символ – вероятность – код – длина».

Запрещено:
- Использование готовых библиотек кодирования (Huffman, compression и т.п.).
- Реализовывать алгоритм Хаффмана в ЛР №3. Игнорировать вероятности символов.

7. Требования к отчёту

Отчёт должен содержать цель работы, исходные данные, описание построенных кодов, расчёты средней длины кодов и выводы.
Исходный код программы приводится в приложении.

8. Контрольные вопросы

10. Что такое код и кодовое слово?
11. Что называют равномерным кодом?
12. Что такое префиксный код?
13. В чём преимущество префиксных кодов?
14. Что такое дерево кодов?
15. Почему вероятности символов важны при кодировании?
16. Почему равномерный код не является оптимальным?
17. Почему средняя длина кода не может быть меньше энтропии?

8. Выводы

В выводах необходимо проанализировать эффективность различных способов кодирования и обосновать необходимость оптимального кодирования.