

4. АЛГОРИТМЫ КОДИРОВАНИЯ ДАННЫХ

4.1. Задания по алгоритму кода Шеннона

1. Для набора из 10 первых последовательных неповторяющихся символов ФИО студента построить вручную код Шеннона. Вероятности символов оценить по всей ФИО с точностью до двух знаков после запятой. Для построенного кода Шеннона проверить выполнение неравенства Крафта, вычислить среднюю длину кодового слова и сравнить ее с энтропией. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Реализовать алгоритм построения побуквенного статического кода Шеннона для текста на русском языке. Для оценки вероятностей символов использовать файл не менее 10 Кбайт. Вывести на экран полученную кодовую таблицу в виде:

Код Шеннона

Символ	Вероятность символа	Кодовое слово	Длина кодового слова

3. Для построенного статического кода Шеннона проверить выполнение неравенства Крафта, сравнить среднюю длину кодового слова с энтропией исходного файла, определить избыточность построенного кода Шеннона, построить таблицу вида:

Код Шеннона

Неравенство Крафта	Энтропия исходного текста	Средняя длина кодового слова	Избыточность кода Шеннона

4. При помощи построенного кода Шеннона закодировать русский текст из 100 символов и вывести на экран закодированную символьную последовательность. Вычислить длину закодированной последовательности и оценить коэффициент сжатия данных.

5*. Реализовать кодовое дерево для построенного кода Шеннона и вывести его на экран.

4.2. Задания по алгоритму кода Фано

1. Для набора из 10 первых последовательных неповторяющихся символов ФИО студента построить вручную код Фано. Вероятности символов оценить по всей ФИО с точностью до двух знаков после запятой. Для построенного кода Фано проверить выполнение неравенства Крафта, вычислить среднюю

длину кодового слова и сравнить ее с энтропией и со средней длиной построенного ранее кода Шеннона. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Реализовать алгоритм построения побуквенного статического кода Фано для текста на русском языке. Для оценки вероятностей символов использовать файл не менее 10 Кбайт. Вывести на экран полученную кодовую таблицу в виде:

Код Фано

Символ	Вероятность символа	Кодовое слово	Длина кодового слова

3. Для построенного статического кода Фано проверить выполнение неравенства Крафта, сравнить среднюю длину кодового слова с энтропией исходного файла, определить избыточность построенного кода Фано построить таблицу вида:

Код Фано

Неравенство Крафта	Энтропия исходного текста	Средняя длина кодового слова	Избыточность кода Фано

4. Реализовать алгоритм построения побуквенного статического кода Фано, используя для вычисления медианы функцию из алгоритма А2 построения почти оптимального дерева поиска. Для построенного варианта кода Фано вывести на экран таблицы п.2 и п.3.

5*. Реализовать кодовое дерево для построенного кода Фано и вывести его на экран.

4.3. Задания по алгоритму кода Хаффмана

1. Для набора из 10 первых последовательных неповторяющихся символов ФИО студента построить вручную оптимальный код Хаффмана. Вероятности символов оценить по всей ФИО с точностью до двух знаков после запятой. Для построенного кода Хаффмана проверить выполнение неравенства Крафта, вычислить среднюю длину кодового слова и сравнить ее с энтропией и со средней длиной построенных ранее кодов Шеннона и Фано. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Реализовать алгоритм построения оптимального побуквенного кода Хаффмана для текста на русском языке. Для оценки вероятностей символов использовать файл не менее 10 Кбайт. Вывести на экран полученную кодовую таблицу в виде:

Код Хаффмана

Символ	Вероятность символа	Кодовое слово	Длина кодового слова

3. Для построенного оптимального кода Хаффмана проверить выполнение неравенства Крафта, сравнить среднюю длину кодового слова с энтропией исходного файла, определить избыточность построенного кода Хаффмана построить таблицу вида:

Код Хаффмана

Неравенство Крафта	Энтропия исходного текста	Средняя длина кодового слова	Избыточность кода Хаффмана

Сравнить со средней длиной построенных ранее кодов Шеннона и Фано.

4. При помощи построенного кода Хаффмана закодировать русский текст из 100 символов и вывести на экран закодированную символьную последовательность. Вычислить длину закодированной последовательности и оценить коэффициент сжатия данных.

5*. Реализовать кодовое дерево для построенного кода Хаффмана и с его помощью декодировать закодированную символьную последовательность, сравнить ее с исходной.

4.4. Задания по алгоритму кода Гилберта-Мура

1. Для набора из 10 первых последовательных неповторяющихся символов ФИО студента построить вручную алфавитный код Гилберта-Мура. Вероятности символов оценить по всей ФИО с точностью до двух знаков после запятой. Для построенного кода Гилберта-Мура проверить выполнение неравенства Крафта, вычислить среднюю длину кодового слова и сравнить ее с энтропией и со средней длиной построенных ранее кодов Шеннона, Фано и Хаффмана. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Реализовать алгоритм построения алфавитного кода Гилберта-Мура для текста на русском языке. Для оценки вероятностей символов использовать файл не менее 10 Кбайт. Вывести на экран полученную кодовую таблицу в виде:

Код Гилберта-Мура

Символ	Вероятность символа	Кодовое слово	Длина кодового слова

3. Для построенного алфавитного кода Гилберта-Мура проверить выполнение неравенства Крафта, сравнить среднюю длину кодового слова с энтропией исходного файла, определить избыточность построенного кода Гилберта-Мура построить таблицу вида:

Код Гилберта-Мура

Неравенство Крафта	Энтропия исходного текста	Средняя длина кодового слова	Избыточность кода Гилберта-Мура

Сравнить со средней длиной построенных ранее кодов Шеннона, Фано и Хаффмена.

4. При помощи построенного кода Гилберта-Мура закодировать русский текст из 100 символов и вывести на экран закодированную символьную последовательность. Вычислить длину закодированной последовательности и оценить коэффициент сжатия данных.

5*. Реализовать кодовое дерево для построенного кода Гилберта-Мура и с его помощью декодировать закодированную символьную последовательность, сравнить ее с исходной.