Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Сибирский государственный университет телекоммуникаций и
информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

Практическая работа № 4

По дисциплине «Теория информации»

Оптимальное побуквенное кодирование

Выполнили:
студенты группы ИП-711
Логинов В.С.
Щерба А.С.

Работу проверил:
Доцент кафедры ПМиК
Мачикина Е.П.

Новосибирск 2021 г.

Оглавление

Задание

1. Запрограммировать процедуру двоичного кодирования текстового файла. В качестве метода кодирования использовать или метод Шеннона, или метод Фано, или метод Хаффмана. Текстовые файлы использовать те же, что и в практических работах 1, 2, 3.

2. Проверить, что построенный код для каждого файла является префиксным. Вычислить среднюю длину кодового слова и оценить избыточность каждого построенного кода.

3. После кодирования текстового файла вычислить оценки энтропии выходной последовательности, используя частоты отдельных символов, пар символов и троек символов и заполнить таблицу.

Программная реализация

```python
class Symbol(object):
    code = ''
    def __init__(self, arg, parg):
        self.a = arg
        self.p = parg

    def __repr__(self):
        return f'{repr(self.a)} - {str(self.p)} - {self.code}'

    def __lt__(self, other):
        return self.p < other.p

    def __add__(self, other):
        return Symbol('', self.p + other.p)

    def __sub__(self, other):
        return Symbol('', abs(self.p - other.p))

    def __radd__(self, other):
        if other == 0:
            return self
        else:
            return self.__add__(other)


def fano(l):
    if len(l) == 1:
        return

    n = min(enumerate([sum(l[:i]) - sum(l[i:]) for i in range(1, len(l))]), key=operator.itemgetter(1))[0] + 1

    for i, e in enumerate(l):
        e.code += ('0' if i < n else '1')

    fano(l[:n])
    fano(l[n:])
```

Рис 1. Код процедуры

Была реализована процедура двоичного кодирования методом Фано. При этом список букв алфавита источника разбивается на две части таким образом, чтобы разность сумм вероятностей была минимальна.

Результаты и анализ

```
File ../lab1/f1.txt
'a' - 0.1939453125 - 000
'c' - 0.19912109375 - 001
'e' - 0.2 - 01
'd' - 0.20263671875 - 10
'b' - 0.204296875 - 11
Средняя длина кодового слова - 2.39306640625
Step 1 - 0.9824638263647659
Step 2 - 0.9820020682196073
Step 3 - 0.9810667199188918
Избыточность кода - 0.07136576210299062
Encoded to file 0.bin
```

Рис 2. Вывод программы для первого файла

Программа выводит кодовые слова, видно что код префиксный.

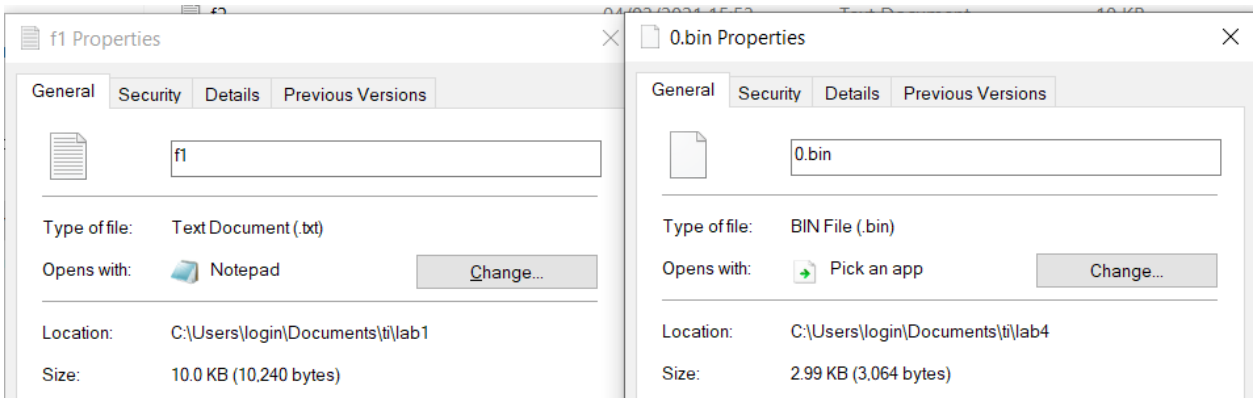

Рис 3. Размеры файлов

Программа создала файлы, которые содержат закодированную информацию из соответствующих входных файлов.

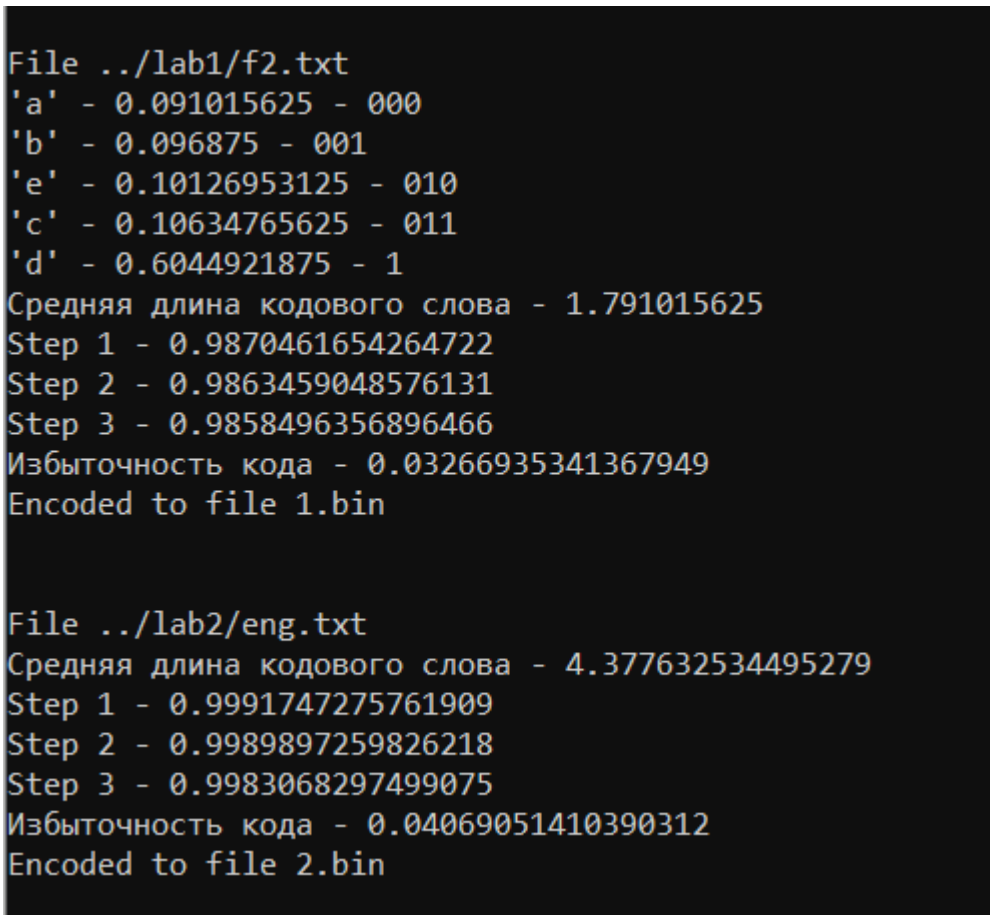

Рис 4. Вывод программы для остальных файлов

Таблица с результатам

Метод кодирования	Название текста	Оценка избыточности кодирования	Оценка энтропии выходной посл-ти (частоты символов)	Оценка энтропии выходной посл-ти (пары символов)	Оценка энтропии выходной посл-ти (тройки символов)
Фано	f1.txt	0.0713	0.9824	0.982	0.981
Фано	f2.txt	0.0326	0.987	0.9863	0.9858
Фано	eng.txt	0.0406	0.9991	0.9989	0.9983

Оценки энтропии выходных последовательностей очень близки к единице, т. к. после кодирования информация сжата.