

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем

**Лабораторная работа №2**  
по дисциплине: «Теория информации»

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Твердохлеб В.В.

Белгород 2023 г.

## Исследование кодов Шеннона-Фано

**Цель работы:** изучить коды Шеннона-Фано, написать программу для кодирования сообщения методом Шеннона-Фано

**Ход работы**

**Код программы:**

```
#include <iostream>
#include <vector>
#include <iomanip>
#include "cmath"
#include <windows.h>

using namespace std;

int length;
string message;

struct SymbolCode {
    char symbol;
    int count;
    float probability;
    string code;

    SymbolCode(char symbol) {
        this->symbol = symbol;
        this->count = 1;
        this->probability = (float) 1 / (float) length;
    }
};

void pushToTable(char c, vector<SymbolCode> &symbolCodeTable) {
    for (auto &i: symbolCodeTable)
        if (i.symbol == c) {
            i.count++;
            i.probability = (float) i.count / (float) length;
            return;
        }
    symbolCodeTable.emplace_back(c);
}

void readMessage(vector<SymbolCode> &symbolCodeTable) {
    getline(cin, message);
    length = message.size();
    for (auto c: message)
        pushToTable(c, symbolCodeTable);
}
```

```

void swap(SymbolCode *a, SymbolCode *b) {
    SymbolCode tmp = *a;
    *a = *b;
    *b = tmp;
}

void sort(vector<SymbolCode> &symbolCodeTable) {
    for (int i = 0; i < symbolCodeTable.size() - 1; i++) {
        int maxIndex = i;
        for (int j = i + 1; j < symbolCodeTable.size(); j++)
            if (symbolCodeTable[j].count > symbolCodeTable[maxIndex].count)
                maxIndex = j;
        swap(&symbolCodeTable[i], &symbolCodeTable[maxIndex]);
    }
}

void CodesForSymbolCodesTable_(vector<SymbolCode> &symbolCodeTable,
                                int leftBorder, int rightBorder) {
    if (rightBorder - leftBorder == 1)
        return;

    float totalProbabilitiesSum = 0;
    for (int i = leftBorder; i < rightBorder; i++)
        totalProbabilitiesSum += symbolCodeTable[i].probability;

    int i = leftBorder;
    float partialProbabilitiesSum = 0;
    while (i < rightBorder and
           partialProbabilitiesSum < totalProbabilitiesSum / 2) {
        partialProbabilitiesSum += symbolCodeTable[i].probability;
        i++;
    }

    for (int j = leftBorder; j < i; j++)
        symbolCodeTable[j].code += '0';
    for (int j = i; j < rightBorder; j++)
        symbolCodeTable[j].code += '1';

    CodesForSymbolCodesTable_(symbolCodeTable, leftBorder, i);
    CodesForSymbolCodesTable_(symbolCodeTable, i, rightBorder);
}

void CodesForSymbolCodesTable(vector<SymbolCode> &symbolCodeTable) {
    CodesForSymbolCodesTable_(symbolCodeTable, 0, symbolCodeTable.size());
}

void outputSymbolCodeTable(vector<SymbolCode> &symbolCodeTable) {

```

```

for (auto &i: symbolCodeTable)
    cout << setprecision(4) << i.symbol << ' ' << i.probability << ' ' << i.code
        << endl;
}

```

```

void squeezeMessage(vector<SymbolCode> &symbolCodeTable) {
    for (auto i: message)
        for (auto &j: symbolCodeTable)
            if (j.symbol == i)
                cout << j.code << " ";
    cout << endl;
}

```

```

float compressionRatio(vector<SymbolCode> &symbolCodeTable) {
    int compressionMessageSize = 0;
    for (auto i: message)
        for (auto &j: symbolCodeTable)
            if (j.symbol == i)
                compressionMessageSize += j.code.size();
    return (float) (length * 8) / (float) compressionMessageSize;
}

```

```

double dispersion(vector<SymbolCode> &symbolCodeTable) {
    double res = 0;
    for (auto &i: symbolCodeTable)
        res += pow(log(i.probability), 2) * i.probability;
    return res;
}

```

```

int main() {
    setlocale(LC_ALL, "Russian");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    vector<SymbolCode> symbolCodeTable;
    readMessage(symbolCodeTable);
    sort(symbolCodeTable);
    CodesForSymbolCodesTable(symbolCodeTable);
    outputSymbolCodeTable(symbolCodeTable);
    squeezeMessage(symbolCodeTable);
    cout << compressionRatio(symbolCodeTable) << endl;
    cout << dispersion(symbolCodeTable);

    return 0;
}

```

**Кодировка:**

0.1698 000  
а 0.09434 001  
и 0.0566 01000  
л 0.0566 01001  
в 0.03774 0101  
ы 0.03774 01100  
р 0.03774 01101  
г 0.01887 0111  
ж 0.01887 100000  
ч 0.01887 100001  
щ 0.01887 100010  
б 0.01887 100011  
х 0.01887 100100  
ц 0.01887 100101  
т 0.01887 10011  
ю 0.01887 101000  
у 0.01887 101001  
с 0.01887 10101  
? 0.01887 101100  
д 0.01887 101101  
н 0.01887 10111  
о 0.01887 110000  
ф 0.01887 110001  
ь 0.01887 110010  
ш 0.01887 110011  
й 0.01887 110100  
э 0.01887 110101  
к 0.01887 11011  
з 0.01887 111000  
е 0.01887 111001  
м 0.01887 11101  
п 0.01887 111100  
я 0.01887 111101  
! 0.01887 11111

**Закодированное сообщение:**

0101'000'100001'001'100010'001'100100'000'101000'0111'001'000'100000'0  
1000'01001'000'100011'01100'000'100101'01000'10011'01101'101001'1010  
1'101100'000'101101'001'000'10111'110000'000'110001'001'01001'110010'1  
10011'01000'0101'01100'110100'000'110101'11011'111000'111001'11101'11  
1100'01001'111101'01101'11111'

**Для полученного кода рассчитать показатели коэффициента сжатия и дисперсии.**

Коэффициент сжатия: 1.669

Дисперсия: 11.24

2. Построить код для сообщения, содержащего строку «Victoria nullaest, Quam quae confessos animo quoque subjugat hostes»

**Кодировка:**

0.1324 0000  
u 0.1029 0001  
s 0.1029 001  
o 0.08824 010  
a 0.08824 011  
e 0.07353 1000  
t 0.05882 1001  
n 0.04412 10100  
i 0.04412 10101  
q 0.04412 1011  
l 0.02941 11000  
c 0.02941 11001  
m 0.02941 11010  
, 0.01471 110110  
Q 0.01471 110111  
r 0.01471 111000  
V 0.01471 111001  
f 0.01471 111010  
b 0.01471 111011  
j 0.01471 111100  
g 0.01471 111101

h 0.01471 11111

**Закодированное сообщение:**

111001'10101'11001'1001'010'111000'10101'011'0000'10100'0001'11000'11  
000'011'0000'1000'001'1001'110110'0000'110111'0001'011'11010'0000'1011'  
0001'011'1000'0000'11001'010'10100'111010'1000'001'001'010'001'0000'01  
1'10100'10101'11010'010'0000'1011'0001'010'1011'0001'1000'0000'001'000  
1'111011'111100'0001'111101'011'1001'0000'11111'010'001'1001'1000'001'

**Для полученного кода рассчитать показатели коэффициента сжатия и дисперсии:**

Коэффициент сжатия: 1.929

Дисперсия: 8.362

3. Получить кодовые представления сообщений из пунктов 1 и 2 задания по методу Хаффмана. Сравнить полученные результаты с методом Шеннона-Фано по показателям сжатия и дисперсии. Сделать соответствующие выводы.

в чащах юга жил бы цитрус? да но фальшивый экземпляр!:

Коэффициент сжатия: 0.591981132

Дисперсия: 0,729981

Victoria nulla est, Quam quae confessos animo quoque subjugat hostes:

Коэффициент сжатия: 0,511029411

Дисперсия: 0,494129

**Вывод:** в ходе лабораторной работы мы изучили коды Шеннона-Фано, написали программу для кодирования сообщения методом Шеннона-Фано.