

МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ (ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ МАШИН СИСТЕМ И СЕТЕЙ

Лабораторная работа №4
по курсу «Методы и средства передачи информации»

Тема: «Кодирование информации методом Шеннона и Хаффмана»

Задание 1	2
Задание 2	2
Задание 3	3
Приложение 1. Код программы к заданию 3	4

Задание 1

Найти префиксные коды с указанными ниже длинами слов:

$l_1 = l_2 = 2, l_3 = l_4 = 3, l_5 = l_6 = l_7 = 4$

$$q_N = y = \sum_{i=1}^{N-1} 2^i$$

$$q_1 = 0$$

$$v_1 = 0\ 0$$

$$q_2 = 2^{-2} = 1/4$$

$$v_2 = 0\ 1$$

$$q_3 = 2^{-2} + 2^{-2} = 1/2$$

$$v_3 = 1\ 0\ 0$$

$$q_4 = 2^{-2} + 2^{-2} + 2^{-3} = 5/8$$

$$v_4 = 1\ 0\ 1$$

$$q_5 = 2^{-2} + 2^{-2} + 2^{-3} + 2^{-3} = 3/4$$

$$v_5 = 1\ 1\ 0\ 0$$

$$q_6 = 2^{-2} + 2^{-2} + 2^{-3} + 2^{-3} + 2^{-4} = 13/16$$

$$v_6 = 1\ 1\ 0\ 1$$

$$q_7 = 2^{-2} + 2^{-2} + 2^{-3} + 2^{-3} + 2^{-4} + 2^{-4} = 7/8$$

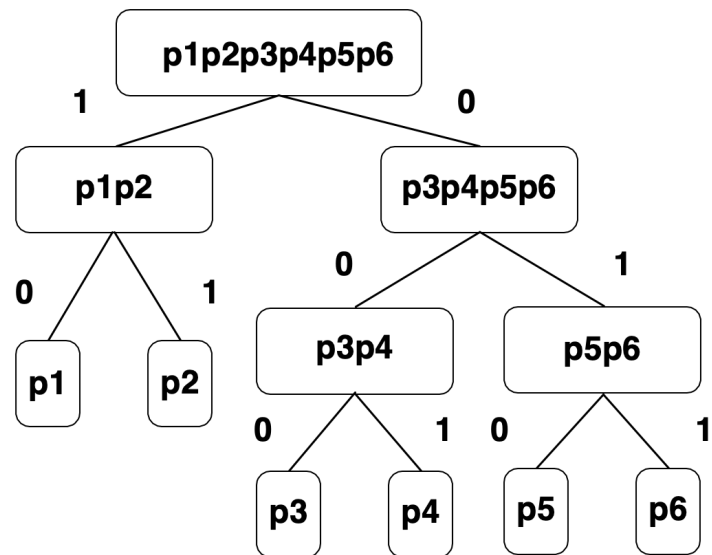
$$v_7 = 1\ 1\ 1\ 0$$

Задание 2

Закодировать двоичным кодом Хаффмена множество сообщений, имеющих вероятности:

Табл. 1 Кодировка методом Хаффмана

Сообщение	Вероятность	Код
p1	0.25	10
p2	0.2	11
p3	0.15	000
p4	0.15	001
p5	0.15	010
p6	0.1	011



Задание 3

Составить программу для кодирования и декодирования слов из букв алфавита кодовой таблицы задания 2.

Строка для кодирования	Закодированная строка	Строка для декодирования	Декодированная строка
123456	1011000001010011	11010	25
456123	0010100111011000	1110	21
246	11001011	010011	56
125	1011010	0010100111011000	456123

Приложение 1. Код программы к заданию 3

```
int main() {
    map<char, string>
        dictionary = {
            {'1', "10"},
            {'2', "11"},
            {'3', "000"},
            {'4', "001"},
            {'5', "010"},
            {'6', "011"}
        };
    TryAgain:
    cout << "Enter line of messages (1 2 3 4 5 6) to encode:" << endl;
    string input;
    cin >> input;
    bool check = true;
    for (const auto &symbol : input)
    {
        if (symbol != '1' &&
            symbol != '2' &&
            symbol != '3' &&
            symbol != '4' &&
            symbol != '5' &&
            symbol != '6')
        {
            check = false;
            break;
        }
    }
    if (!check)
    {
        cout << "Incorrect input!" << endl;
        goto TryAgain;
    }
    else
    {
        string output;
        for(const auto & symb : input)
        {
            output += dictionary[symb];
        }
        cout << "Encoded line is " << output << endl;
    }

    cout << "Enter line to decode" << endl;
    string code;
    cin >> code;
    string decoded;
    int i = 0;
    while (i < code.size()) {
        if (code[i] == '1') {
            if (code[i + 1] == '0') {
                decoded += '1';
            } else if (code[i + 1] == '1') {
                decoded += '2';
            }
            i += 2;
        } else if (code[i] == '0') {
            if (code[i + 1] == '0' && code[i + 2] == '0') {
                decoded += '3';
            } else if (code[i + 1] == '0' && code[i + 2] == '1') {
                decoded += '4';
            } else if (code[i + 1] == '1' && code[i + 2] == '0') {
                decoded += '5';
            } else if (code[i + 1] == '1' && code[i + 2] == '1') {
                decoded += '6';
            }
            i += 3;
        }
    }
    cout << "Decoded line is " << decoded << endl;
    return 0;
}
```