

МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ (ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ МАШИН СИСТЕМ И СЕТЕЙ

Лабораторная работа №11  
по курсу «Методы и средства передачи информации»

Тема: «Циклические кольца»

Выполнил:  
Балашов С.А., А-08-19  
Проверил:  
доц. Оцоков Ш.А.

Москва 2021

Задача 1 2

Задача 2 3

Задача 3 5

## Задача 1

Построить таблицу синдромов для кода Хемминга (7,4).

```
import itertools

import numpy as np

H = np.array([[1, 0, 1, 0, 1, 0, 1], [0, 1, 1, 0, 0, 1, 1], [0, 0, 0, 1, 1, 1, 1]])

G = np.array([[1, 0, 0, 0, 0, 1, 1], [0, 1, 0, 0, 1, 0, 1], [0, 0, 1, 0, 1, 1, 0], [0, 0, 0, 1, 1, 1, 1]])

U = np.asarray(list(itertools.product(*[[0,1]]*4)))

codes = np.asarray(U.dot(G))

codes %= 2

Tcodes = codes.transpose()

print('Транспонированные коды:\n', Tcodes)

B = np.eye(7)

print('Ошибки в лидере:\n', B)

s = []

for i in range(7):
    s.append(np.dot(B[i], H.transpose()))

print('Синдромы:\n', s)
```

Транспонированные коды:

```
[[0 0 0 0 0 0 0 1 1 1 1 1 1 1 1]
 [0 0 0 0 1 1 1 1 0 0 0 0 1 1 1]
 [0 0 1 1 0 0 1 1 0 0 1 1 0 0 1]
 [0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]
 [0 1 1 0 1 0 0 1 0 1 1 0 1 0 0]
 [0 1 1 0 0 1 1 0 1 0 0 1 1 0 0]
 [0 1 0 1 1 0 1 0 1 0 1 0 0 1 0]]
```

Ошибки в лидере:

```
[[1. 0. 0. 0. 0. 0. 0.]  
[0. 1. 0. 0. 0. 0. 0.]  
[0. 0. 1. 0. 0. 0. 0.]  
[0. 0. 0. 1. 0. 0. 0.]  
[0. 0. 0. 0. 1. 0. 0.]  
[0. 0. 0. 0. 0. 1. 0.]  
[0. 0. 0. 0. 0. 0. 1.]]
```

Синдромы:

```
[array([1., 0., 0.]), array([0., 1., 0.]), array([1., 1.,  
0.]), array([0., 0., 1.]), array([1., 0., 1.]), array([0.,  
1., 1.]), array([1., 1., 1.])]
```

## Задача 2

Составить программу, которая выводит смежный класс и лидеров смежного класса. Таблица кодовых слов и не кодовое слово вводится в программу.

```
import numpy as np
```

```
def SideClass(codes, n, m, not_cword):  
    for i in range (n):  
        for j in range(m):  
            codes[i, j] += not_cword[i]  
    print('Смежный класс:\n', codes%2)  
    return codes
```

```
def Leader(codes, n, m):  
    minima = n  
    tmp = 0  
    for i in range (n):  
        tmp = 0
```

```

        for j in range (m):
            tmp += codes[i, j]
        if tmp > 0:
            minima = min(minima, tmp)
    for j in range (n):
        tmp = 0
        for i in range (m):
            tmp += codes[i, j]
        if tmp == minima:
            print('Лидер:')
            print(codes[:, j])

def func():
    print('Введите количество строк:')
    n = int(input())
    print('Введите количество столбцов:')
    m = int(input())
    codes = np.zeros((n, m))
    for i in range(n):
        for j in range(m):
            print('Введите [', i, ', ', j, ']-й элемент матрицы:')
            codes[i, j] = int(input())
    print('Кодовые слова:', codes)
    print('Введите не кодовое слово:')
    not_cword = [0] * n
    for i in range(n):
        not_cword[i] = int(input())
    side_codes = SideClass(codes, n, m, not_cword)
    Leader(side_codes%2, n, m)

```

Кодовые слова: [[1. 0. 0. 1. 0.]

[1. 0. 0. 0. 0.]

[1. 0. 1. 1. 0.]

[0. 1. 1. 1. 0.]

[1. 0. 0. 1. 1.]]

Введите не кодовое слово:

1

0

1

1

0

Смежный класс:

[[0. 1. 1. 0. 1.]

[1. 0. 0. 0. 0.]

[0. 1. 0. 0. 1.]

[1. 0. 0. 0. 1.]

[1. 0. 0. 1. 1.]]

Лидер:

[1. 0. 0. 0. 0.]

Лидер:

[0. 0. 0. 0. 1.]

## Задача 3

**Доказать, что алгоритм синдромного декодирования позволяет исправить любое количество ошибок, не превосходящее  $\left\lfloor \frac{d-1}{2} \right\rfloor$ , где  $d$  — кодовое расстояние.**

**У к а з а н и е.** Достаточно проверить, что все векторы веса  $\left\lfloor \frac{d-1}{2} \right\rfloor$  и меньше попадают в различные смежные классы и, следовательно, являются лидерами в своих смежных классах.

Допустим, есть 2 лидера с одним синдромом

$$v_1 * H_t = v_2 * H_t$$

$$H_t * (v_1 - v_2) = 0$$

$$w(v_1) = w(v_2) = (d - 1) / 2 \Rightarrow \text{синдром} = 0$$

$v_1 - v_2 = c$  - кодовый вектор

$w(c)$  должно быть больше  $d$

Пусть  $w(v_1) < (d - 1) / 2$ , чтобы синдром не был равен 0

Тогда  $w(c) < d$  - противоречие  $\Rightarrow$  исходное предположение неверно.

Можно исправить не более  $(d - 1) / 2$  ошибок

чтд