Projekt nr 2
Kody liniowe

- Michał Iwicki
- Magdalena Jeczeń
- Michał Kukla

Zadanko 1. **Metryką** na zbiorze X nazywamy każdą funkcję $D : X^2 \to \mathbb{R}$ spełniającą następujące warunki:

- $D(x,y) = 0 \Leftrightarrow x = y$,
- $D(x,y) = D(y,x)$,
- $D(x,z) \le D(x,y) + D(y,z)$

dla $x,y,z \in X$. Udowodnić, że odległość Hamminga jest metryką.

Zadanie 1

Niech $x, y, z \in X$ $\quad d(u,v) = |\{i \in [n] : u_i \neq v_i\}|$

I. $d(x,y) = 0 \iff |\{i \in [n] : x_i \neq y_i\}| = 0 \iff$

$\{i \in [n] : x_i \neq y_i\} = \emptyset \iff x = y$

II. $d(x,y) = |\{i \in [n] : x_i \neq y_i\}| = |\{i \in [n] : y_i \neq x_i\}| = d(y,x)$

III. $d(x,z) = |\{i \in [n] : x_i \neq z_i\}| \le |\{i \in [n] : x_i \neq y_i \neq z_i\}|$

Zasada [illegible] ($|A \cap B| = |A| + |B| - |A \cup B|$)

$= |\{i \in [n] : x_i \neq y_i\}| + |\{i \in [n] : y_i \neq z_i\}| - |\{i \in [n] : x_i \neq y_i \; y_i \neq z_i\}|$

$\le |\{i \in [n] : x_i \neq y_i\}| + |\{i \in [n] : y_i \neq z_i\}| = d(x,y) + d(y,z)$

Zadanko 2. Udowodnić, że dla dowolnego (n,k)–kodu liniowego $\mathcal{C}$ nad skończonym ciałem $\mathbb{K}$ i jego macierzy generującej G powstałej z bazy kodu B wynikiem kodowania dowolnego wektora $v \in \mathbb{K}^k$ jest słowo kodowe kodu $\mathcal{C}$.

Zadanie 2

Niech $v \in \mathbb{K}^k$, C - dowolny kod (n,k)-kod liniowy, $G = \begin{pmatrix} e_1^T \\ \vdots \\ e_k^T \end{pmatrix}$

$B = (e_1, e_2, \dots, e_k)$ - baza kodu

il. skalarny

$$w := (v^T \cdot G)^T = \left([v_1, v_2, \dots, v_k] \odot \begin{bmatrix} e_1^T \\ \vdots \\ e_k^T \end{bmatrix} \right)^T =$$

$$= (v_1 \cdot e_1^T + v_2 \cdot e_2^T + \dots + v_k \cdot e_k^T)^T \in \operatorname{span}(B) = C$$

bo jest kombinacją liniową

Zatem dla dowolnego $v \in \mathbb{K}^k$ jego ~~s~~kodowanie jest słowem kodowym kodu C c.k.d.

MinimizeHammingDistance($\mathcal{C}$, B, v)

IN: $\mathcal{C}$ – (n,k)–kod liniowy nad ciałem $\mathbb{K}$, B – baza kodu $\mathcal{C}$, v– dekodowany wektor

$m = \min\{d(v,w) : w \in \mathcal{C}\}$ # d to odległość Hamminga

$L = \{w \in \mathcal{C} : d(v,w) = m\}$

w = losowo wybrany wektor należący do L

r = wektor współczynników wektora w w bazie B

OUT: Wektor $r \in \mathbb{K}^k$

Zadanko 3. Udowodnić, że dla dowolnego (n,k)–kodu liniowego $\mathcal{C}$ nad skończonym ciałem $\mathbb{K}$ i jego macierzy generującej G powstałej z bazy kodu B algorytm `MinimizeHammingDistance` użyty do dekodowania słowa kodowego $w \in \mathcal{C}$ zwróci taki wektor $v \in \mathbb{K}^k$, który w wyniku zakodowania go z użyciem macierzy G da wektor w.

Zadanie 3.

C – (n,k)-kod liniowy, $B = (e_1, e_2, \dots, e_k)$, $G = B^T$

Niech $w \in C$, zatem $\min\{d(w,v) : v \in C\} = 0$, ponieważ $d(w,w) = 0$. Zatem z I własności metryki $L = \{w\}$, więc algorytm zwróci takie $v \in \mathbb{K}^k$, że $w = B \cdot v$. Zauważmy, że $(v^T \cdot G)^T = G^T \cdot (v^T)^T = B \cdot v = w$.

C.K.D.

Zadanko 4. Udowodnić, że dla dowolnej przestrzeni liniowej V nad ciałem $\mathbb{K}$ odległość Hamminga jest niezmiennicza ze względu na przesunięcia (czyli dla dowolnych wektorów $u, v, x \in \mathbb{K}^n$ odległość Hamminga słów u i v jest taka sama jak odległość słów $u + x$ i $v + x$).

Zadanie 4.

- Niech $x, y, z \in V$ oraz weźmy dane $i \in [n]$.

Wiemy, że $x_i + z_i \neq y_i + z_i$ wtedy i tylko wtedy, gdy $x_i \neq y_i$. Zatem z dowolności i mamy:

$\{i \in [n] : x_i \neq y_i\} = \{i \in [n] : x_i + z_i \neq y_i + z_i\}$, więc

$d(x, y) = d(x+z, y+z)$

Dla ścisłości, ta oczywista własność wynika z istnienia elementu przeciwnego do skalara z ciała K o który przesuwamy inny skalar. Gdyby to nie była prawda, moglibyśmy przesunąć 2 różne skalary najpierw o pewny skalar **z**, a potem o **–z** i otrzymalibyśmy 2 takie same skalary. Sprzeczność.

Zadanko 5 ($\star$). Obliczyć odległość Hamminga dla wektorów $(1,2,0,1)^T$ i $(0,0,0,1)^T$. Które z wektorów ze zbioru

$$\left\{\begin{pmatrix}1\\2\\1\\2\\0\end{pmatrix},\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix},\begin{pmatrix}0\\0\\2\\1\\1\end{pmatrix},\begin{pmatrix}2\\2\\2\\1\\0\end{pmatrix}\right\}$$

znajdują się najbliżej siebie w sensie Hamminga? W Mathematice przydać się może polecenie `HammingDistance[]`.

```
fprintf('Odległość wektorów')
HammingDist([1, 2, 0, 1]', [0, 0, 0, 1]')

A =[1 1 0 2;
    2 1 0 2;
    1 1 2 2;
    2 1 1 1;
    0 1 1 0;];
min_d = 5;
for i = 1 : 4
    for j = i+1:4
        if (HammingDist(A(:,i),A(:,j)) < min_d)
           min_d = HammingDist(A(:,i),A(:,j));
        end
    end
end
for i = 1 : 4
    for j = i+1:4
        if (HammingDist(A(:,i),A(:,j)) == min_d)
            fprintf("Najblizsze sobie 2 wektory")
            [A(:,i),A(:,j)]
        end
    end
end
```

```
function [result] = HammingDist(x, y)
result = 0;
n = length(x);
for i = 1 : n
    if ~(x(i) == y(i))
        result = result + 1;
    end
end
```

```
Command Window

Odległość wektorów
ans =

     2

Najblizsze sobie 2 wektory
ans =

     1     1
     2     1
     1     1
     2     1
     0     1

Najblizsze sobie 2 wektory
ans =

     1     2
     2     2
     1     2
     2     1
     0     0

Najblizsze sobie 2 wektory
ans =

     1     0
     1     0
     1     2
     1     1
     1     1

Najblizsze sobie 2 wektory
ans =

     0     2
     0     2
     2     2
     1     1
     1     0

fx >>
```

Zadanko 6 ($\star$). Wygenerować w wybranym języku wszystkie słowa kodowe dla $(5,3)$-kodu liniowego $\mathcal{C}$ nad ciałem $\mathbb{Z}_7$ takiego, że bazą kodu liniowego $\mathcal{C}$ jest

$$B = \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 4 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 5 \\ 6 \end{pmatrix} \right).$$

```
B =[1 0  0;
    0 1 0;
    0 0 1;
    2 1 5;
    4 0 6];
r = 1;
C = [];
for i = 0:6
    for j = 0:6
        for k = 0:6
            v = i*B(:,1)+j*B(:,2)+k*B(:,3);
            v = mod(v, 7);
            C(:,r) = v;
            r= r+1;

        end
    end
end
% każdy wektor jest wyznaczony jednoznacznie względem bazy
% ponieważ Z7 to przestrzeń liniowa,
% więc nie musimy sprawdzać ich unikalności
C
```

Wyniki zadania 6

Columns 1 through 14

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	1	2	3	4	5	6	0	1	2	3	4	5	6
0	5	3	1	6	4	2	1	6	4	2	0	5	3
0	6	5	4	3	2	1	0	6	5	4	3	2	1

Columns 15 through 28

0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	3	3	3	3	3	3	3
0	1	2	3	4	5	6	0	1	2	3	4	5	6
2	0	5	3	1	6	4	3	1	6	4	2	0	5
0	6	5	4	3	2	1	0	6	5	4	3	2	1

Columns 29 through 42

0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	5	5	5	5	5	5	5
0	1	2	3	4	5	6	0	1	2	3	4	5	6
4	2	0	5	3	1	6	5	3	1	6	4	2	0
0	6	5	4	3	2	1	0	6	5	4	3	2	1

Columns 43 through 56

0	0	0	0	0	0	0	1	1	1	1	1	1	1
6	6	6	6	6	6	6	0	0	0	0	0	0	0
0	1	2	3	4	5	6	0	1	2	3	4	5	6
6	4	2	0	5	3	1	2	0	5	3	1	6	4
0	6	5	4	3	2	1	4	3	2	1	0	6	5

Columns 57 through 70

1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	2	2	2	2	2	2
0	1	2	3	4	5	6	0	1	2	3	4	5	6
3	1	6	4	2	0	5	4	2	0	5	3	1	6
4	3	2	1	0	6	5	4	3	2	1	0	6	5

Columns 71 through 84

1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	4	4	4	4	4	4	4
0	1	2	3	4	5	6	0	1	2	3	4	5	6
5	3	1	6	4	2	0	6	4	2	0	5	3	1
4	3	2	1	0	6	5	4	3	2	1	0	6	5

Columns 85 through 98

1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	5	5	5	5	5	5	6	6	6	6	6	6	6
0	1	2	3	4	5	6	0	1	2	3	4	5	6
0	5	3	1	6	4	2	1	6	4	2	0	5	3
4	3	2	1	0	6	5	4	3	2	1	0	6	5

Columns 99 through 112

2	2	2	2	2	2	2	2	2	2	2	2	2	2
0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	1	2	3	4	5	6	0	1	2	3	4	5	6
4	2	0	5	3	1	6	5	3	1	6	4	2	0
1	0	6	5	4	3	2	1	0	6	5	4	3	2

Columns 113 through 126

2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	3	3	3	3	3	3	3
0	1	2	3	4	5	6	0	1	2	3	4	5	6
6	4	2	0	5	3	1	0	5	3	1	6	4	2
1	0	6	5	4	3	2	1	0	6	5	4	3	2

Columns 127 through 140

2	2	2	2	2	2	2	2	2	2	2	2	2	2
4	4	4	4	4	4	4	5	5	5	5	5	5	5
0	1	2	3	4	5	6	0	1	2	3	4	5	6
1	6	4	2	0	5	3	2	0	5	3	1	6	4
1	0	6	5	4	3	2	1	0	6	5	4	3	2

Columns 141 through 154

2	2	2	2	2	2	2	3	3	3	3	3	3	3
6	6	6	6	6	6	6	0	0	0	0	0	0	0
0	1	2	3	4	5	6	0	1	2	3	4	5	6
3	1	6	4	2	0	5	6	4	2	0	5	3	1
1	0	6	5	4	3	2	5	4	3	2	1	0	6

Columns 155 through 168

3	3	3	3	3	3	3	3	3	3	3	3	3	3
1	1	1	1	1	1	1	2	2	2	2	2	2	2
0	1	2	3	4	5	6	0	1	2	3	4	5	6
0	5	3	1	6	4	2	1	6	4	2	0	5	3
5	4	3	2	1	0	6	5	4	3	2	1	0	6

Columns 169 through 182

3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	4	4	4	4	4	4	4
0	1	2	3	4	5	6	0	1	2	3	4	5	6
2	0	5	3	1	6	4	3	1	6	4	2	0	5
5	4	3	2	1	0	6	5	4	3	2	1	0	6

Columns 183 through 196

3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	5	5	5	5	5	5	6	6	6	6	6	6	6
0	1	2	3	4	5	6	0	1	2	3	4	5	6
4	2	0	5	3	1	6	5	3	1	6	4	2	0
5	4	3	2	1	0	6	5	4	3	2	1	0	6

Columns 197 through 210

4	4	4	4	4	4	4	4	4	4	4	4	4	4
0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	1	2	3	4	5	6	0	1	2	3	4	5	6
1	6	4	2	0	5	3	2	0	5	3	1	6	4
2	1	0	6	5	4	3	2	1	0	6	5	4	3

Columns 211 through 224

4	4	4	4	4	4	4	4	4	4	4	4	4	4
2	2	2	2	2	2	2	3	3	3	3	3	3	3
0	1	2	3	4	5	6	0	1	2	3	4	5	6
3	1	6	4	2	0	5	4	2	0	5	3	1	6
2	1	0	6	5	4	3	2	1	0	6	5	4	3

Columns 225 through 238

4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	5	5	5	5	5	5	5
0	1	2	3	4	5	6	0	1	2	3	4	5	6
5	3	1	6	4	2	0	6	4	2	0	5	3	1
2	1	0	6	5	4	3	2	1	0	6	5	4	3

Columns 239 through 252

4	4	4	4	4	4	4	5	5	5	5	5	5	5
6	6	6	6	6	6	6	0	0	0	0	0	0	0
0	1	2	3	4	5	6	0	1	2	3	4	5	6
0	5	3	1	6	4	2	3	1	6	4	2	0	5
2	1	0	6	5	4	3	6	5	4	3	2	1	0

Columns 253 through 266

5	5	5	5	5	5	5	5	5	5	5	5	5	5
1	1	1	1	1	1	1	2	2	2	2	2	2	2
0	1	2	3	4	5	6	0	1	2	3	4	5	6
4	2	0	5	3	1	6	5	3	1	6	4	2	0
6	5	4	3	2	1	0	6	5	4	3	2	1	0

Columns 267 through 280

5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	4	4	4	4	4	4	4
0	1	2	3	4	5	6	0	1	2	3	4	5	6
6	4	2	0	5	3	1	0	5	3	1	6	4	2
6	5	4	3	2	1	0	6	5	4	3	2	1	0

Columns 281 through 294

5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	6	6	6	6	6	6	6
0	1	2	3	4	5	6	0	1	2	3	4	5	6
1	6	4	2	0	5	3	2	0	5	3	1	6	4
6	5	4	3	2	1	0	6	5	4	3	2	1	0

Columns 295 through 308

6	6	6	6	6	6	6	6	6	6	6	6	6	6
0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	1	2	3	4	5	6	0	1	2	3	4	5	6
5	3	1	6	4	2	0	6	4	2	0	5	3	1
3	2	1	0	6	5	4	3	2	1	0	6	5	4

Columns 309 through 322

6	6	6	6	6	6	6	6	6	6	6	6	6	6
2	2	2	2	2	2	2	3	3	3	3	3	3	3
0	1	2	3	4	5	6	0	1	2	3	4	5	6
0	5	3	1	6	4	2	1	6	4	2	0	5	3
3	2	1	0	6	5	4	3	2	1	0	6	5	4

Columns 323 through 336

6	6	6	6	6	6	6	6	6	6	6	6	6	6
4	4	4	4	4	4	4	5	5	5	5	5	5	5
0	1	2	3	4	5	6	0	1	2	3	4	5	6
2	0	5	3	1	6	4	3	1	6	4	2	0	5
3	2	1	0	6	5	4	3	2	1	0	6	5	4

Columns 337 through 343

6	6	6	6	6	6	6
6	6	6	6	6	6	6
0	1	2	3	4	5	6
4	2	0	5	3	1	6
3	2	1	0	6	5	4

Zadanko 7 (⋆). W wybranym języku utwórz macierz generującą G dla kodu liniowego $\mathcal{C}$ i bazy kodu B z zadanka 6. Następnie utwórz dowolnie wybrany przez siebie wektor $v \in \mathbb{Z}_7^5$ i wykonaj dekodowanie tego wektora używając algorytmu `MinimizeHammingDistance` dla kodu $\mathcal{C}$ i bazy B.

```
fprintf("Macierz G to zwyczajnie transpozycja B")
G = B'
fprintf("Wektor v")
v = [2 3 4 1 2]'
fprintf('Współrzędne losowego najbliższego wektora v w bazie
r = MinimizeHammingDistance(C,B,v)
fprintf("Ten wektor zapisany w bazie B")
mod(B*r,7)


function [w] = MinimizeHammingDistance(C, B, v)
min_d = length(B);
L = [];
r = 1;
for i = 1: length(C)
    d = HammingDist(v, C(:,i));
    if(d == min_d)
            L(:, r) = C(:,i);
            r = r+1;
    elseif(d < min_d)
            L = C(:,i);
            r = 2;
            min_d = d;
    end
end
w = L(:,randi(size(L,2))); %losowa kolumna
% W tym zadaniu jest już zeschodkowana macierz
% więc uprościmy kod
w = w(1:size(B,2));
```

```
Macierz G to zwyczajnie transpozycja B
G =

     1     0     0     2     4
     0     1     0     1     0
     0     0     1     5     6

Wektor v
v =

     2
     3
     4
     1
     2

Współrzędne losowego najbliższego wektora v w bazie B
r =

     5
     6
     4

Ten wektor zapisany w bazie B
ans =

     5
     6
     4
     1
     2

fx >>
```

Zadanko 8 ($\star$). Celem tego zadania jest symulacja przesłania zakodowanej wiadomości. Do wykonania zadania użyj wybranego przez siebie języka.

a) Wygeneruj losową macierz o 10 kolumnach i 4 wierszach o wyrazach z ciała $\mathbb{Z}_5$.

b) Dokonaj unormowania macierzy z podpunktu a) do przedziału $[0, 1]$ dzieląc wszystkie wyrazy macierzy przez 4 (w tym podpunkcie potraktuj elementy macierzy jako liczby całkowite, a nie elementy z ciała $\mathbb{Z}_5$, czyli dzielenie przez 4 to standardowa operacja dzielenia na dwóch liczbach całkowitych). Na podstawie unormowanej macierzy utwórz obraz (przydatne w Mathematice może być polecenie `Image[]`)

```
% a)
fprintf('a)')
fprintf("Losowa macierz")
A = randi(5, 4, 10)-1

% b)
fprintf('b)')
fprintf("Unormowana macierz")
A_norm = A/4
fprintf("Jej obraz")
Im = im_A(A_norm)
```

```
function [Im] = im_A(A)
B=rref(A);
s = size(A);
i=1;
j=1;
while(i <= s(1) && j <= s(2))
    while(B(i,j) == 0)
        j= j+1;
    end
    Im(:,i) = A(:,j);
    i = i + 1;
    j = j + 1;
end
end
```

```
a)Losowa macierz
A =

     3     1     2     3     1     4     4     2     2     0
     4     0     3     2     3     1     3     3     4     0
     1     3     3     4     2     1     2     2     1     2
     3     2     1     4     2     0     3     3     0     2

b)Unormowana macierz
A_norm =

  Columns 1 through 8

    0.7500    0.2500    0.5000    0.7500    0.2500    1.0000    1.0000    0.5000
    1.0000         0    0.7500    0.5000    0.7500    0.2500    0.7500    0.7500
    0.2500    0.7500    0.7500    1.0000    0.5000    0.2500    0.5000    0.5000
    0.7500    0.5000    0.2500    1.0000    0.5000         0    0.7500    0.7500

  Columns 9 through 10

    0.5000         0
    1.0000         0
    0.2500    0.5000
         0    0.5000

Jej obraz
Im =

    0.7500    0.2500    0.5000    0.7500
    1.0000         0    0.7500    0.5000
    0.2500    0.7500    0.7500    1.0000
    0.7500    0.5000    0.2500    1.0000
```

c) Dana jest macierz

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 4 & 4 & 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 3 & 0 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 4 & 3 & 0 \end{pmatrix}$$

o wyrazach z ciała $\mathbb{Z}_5$. Udowodnij, że istnieje $(11, 4)$–kod liniowy $\mathcal{C}$ nad ciałem $\mathbb{Z}_5$ taki, że G jest macierzą generującą kodu $\mathcal{C}$.

```
% c)
fprintf('c)')
G = [1 0 0 0 0 4 4 2 0 1 1;
    0 1 0 0 0 3 0 2 2 1 0;
    0 0 1 0 0 2 0 1 1 1 1;
    0 0 0 1 1 0 0 0 4 3 0];
fprintf("Wystarczy pokazać, że wymiar obrazu B jest równy 4")
size(im_A(G'), 2)
```

```
c)Wystarczy pokazać, że wymiar obrazu B jest równy 4
ans =

     4
```

d) Dla dowolnej kolumny v macierzy z podpunktu a) zakoduj wektor v używając macierzy generującej G.

```
% d)
fprintf('d)')
fprintf("Zakodowana pierwsza kolumna")
mod((A(:,1)'*G)', 5)
```

```
d)Zakodowana pierwsza kolumna
ans =

     3
     4
     1
     3
     3
     1
     2
     0
     1
     2
     4
```

e) Dla każdego zakodowanego wektora z podpunktu d) zasymuluj wysłanie go do pewnego użytkownika poprzez kanał, który dla przesyłanego wektora v dla każdej pozycji dodaje modulo 5 losową liczbę ze zbioru $\{0, 3\}$, przy czym prawdopodobieństwo dodania liczby 0 wynosi $0{,}95$, zaś prawdopodobieństwo dodania 3 jest równe $0{,}05$.

```
% e)
fprintf('e)Zakodowana cała macierz A')
E = mod(G'*A, 5);

for i = 1: size(E, 1)
    for j = 1 : size(E, 2)
        p = rand(1)* 100;
        if (p >= 95)
            E(i,j) = mod(E(i,j) + 3, 5);
        end
    end
end
E
```

```
e)Zakodowana cała macierz A
E =

     3     1     2     3     1     2     4     2     2     0
     4     0     3     2     3     1     3     3     4     0
     1     3     3     4     2     1     2     2     1     2
     3     2     1     4     2     0     3     3     0     2
     3     2     1     4     2     0     3     3     0     2
     1     0     3     1     2     1     4     4     2     4
     2     4     3     2     4     1     1     3     3     0
     0     0     3     4     0     1     1     2     3     2
     1     1     3     4     1     1     0     0     4     3
     2     0     1     1     2     1     3     1     2     3
     4     4     0     2     3     0     1     2     3     2
```

f) Dla każdego zakodowanego wektora po przesłaniu go przez kanał odkoduj ten wektor używając algorytmu `MinimizeHammingDistance`.

g) Z odkodowanych wektorów utwórz macierz odpowiadającą macierzy kodowanej z podpunktu a).

h) Porównaj macierze z podpunktu a) i g). Ile kolumn macierzy z podpunktu a) zostało poprawnie odkodowanych?

```
% f, g)
%Trzeba wygenerować C
fprintf('f,g,')
B = G';
r = 1;
C = [];
for i = 0:4
    for j = 0:4
        for k = 0:4
            for l = 0:4
                v = i*B(:,1)+j*B(:,2)+k*B(:,3)+ l*B(:,4);
                v = mod(v, 5);
                C(:,r) = v;
                r= r+1;
            end
        end
    end
end
E_kod = [];
for i = 1:size(E,2)
    E_kod(:,i) = MinimizeHammingDistance(C,B,E(:,i));
end

% h)
fprintf('h)')
fprintf("Wszystkie kolumny dobrze się odkodowały")
A
E_kod
```

```
f,g,h)Wszystkie kolumny dobrze się odkodowały
A =

     3     1     2     3     1     4     4     2     2     0
     4     0     3     2     3     1     3     3     4     0
     1     3     3     4     2     1     2     2     1     2
     3     2     1     4     2     0     3     3     0     2


E_kod =

     3     1     2     3     1     4     4     2     2     0
     4     0     3     2     3     1     3     3     4     0
     1     3     3     4     2     1     2     2     1     2
     3     2     1     4     2     0     3     3     0     2
```

i) Wyrazy odkodowanej macierzy z podpunktu g) unormuj do przedziału $[0,1]$ analogiczną metodą jak w podpunkcie b). Następnie dla unormowanej macierzy utwórz obraz analogicznie jak w podpunkcie b).

```
% i)
fprintf('i)Macierz unormowana oraz jej obraz')
E_kod = E_kod/4
im_A(E_kod)
```

```
i)Macierz unormowana oraz jej obraz
E_kod =

  Columns 1 through 8

    0.7500    0.2500    0.5000    0.7500    0.2500    1.0000    1.0000    0.5000
    1.0000         0    0.7500    0.5000    0.7500    0.2500    0.7500    0.7500
    0.2500    0.7500    0.7500    1.0000    0.5000    0.2500    0.5000    0.5000
    0.7500    0.5000    0.2500    1.0000    0.5000         0    0.7500    0.7500

  Columns 9 through 10

    0.5000         0
    1.0000         0
    0.2500    0.5000
         0    0.5000


ans =

    0.7500    0.2500    0.5000    0.7500
    1.0000         0    0.7500    0.5000
    0.2500    0.7500    0.7500    1.0000
    0.7500    0.5000    0.2500    1.0000

>>
```