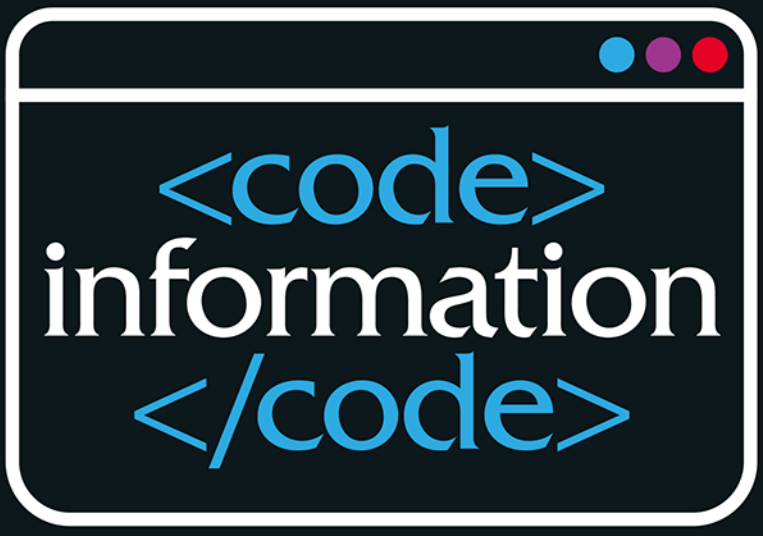
Wojciech Kordecki

INFORMACJA I KODOWANIE

KRÓTKIE WPROWADZENIE Z PRZYKŁADAMI ZASTOSOWAŃ



Wydawnictwo Naukowe Helion

Informacja i kodowanie. Krótkie wprowadzenie z przykładami zastosowań PDF

Redaktor prowadzący: Małgorzata Kulik

Projekt okładki: Studio Gravite/Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

Skład komputerowy w systemie LaTeX wykonał autor.

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/infkod
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-289-0088-2

Printed in Poland.

Spis treści

Opowiadanie 8

Macierze a kodowanie

8.1. Macierze

O pożytkach z tablic

Macierz to tablica, w naszym przypadku tablica liczb, o wymiarze $m \times n$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

mająca m wierszy i n kolumn.

Będziemy korzystać z następujących działań na macierzach:

- dodawania macierzy,
- mnożenia macierzy przez liczbę,
- mnożenia macierzy przez macierz.

Dodawanie:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \dots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \dots & a_{2n}+b_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \dots & a_{mn}+b_{mn} \end{bmatrix}.$$

Mnożenie macierzy przez liczbę:

$$\alpha \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \dots & \alpha a_{1n} \\ \alpha a_{21} & \alpha a_{22} & \dots & \alpha a_{2n} \\ \dots & \dots & \dots & \dots \\ \alpha a_{m1} & \alpha a_{m2} & \dots & \alpha a_{mn} \end{bmatrix}.$$

Mnożenie macierzy:

$$\begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{i,k-1} & a_{ik} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \cdots & b_{1j} & \cdots \\ \cdots & b_{2j} & \cdots \\ \dots & \dots & \dots \\ \cdots & b_{k-1,j} & \cdots \\ \cdots & b_{kj} & \cdots \end{bmatrix}$$
$$= \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \cdots & a_{i1}b_{1j} + \cdots + a_{ik}b_{kj} & \cdots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix}.$$

Uwaga: Dodawać można tylko macierze o tych samych wymiarach. Mnożyć można tylko macierze takie, że lewy mnożnik ma tyle samo kolumn, co prawy mnożnik wierszy.

Macierzą transponowaną A^T macierzy A jest macierz, w której wiersze zamieniono na kolumny, czyli

$$A^T = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \dots & \dots & \dots & \dots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{bmatrix}.$$

Od tego miejsca, jeśli wyraźnie nie zaznaczymy, że jest inaczej, dodawanie i mnożenie zawsze będzie oznaczać działanie $\bmod 2$.

Przykład 8.1.

Mnożenie i dodawanie:

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Niech $w_1, w_2, \ldots, w_k$ będą wybranymi wierszami macierzy A. Suma tych wierszy pomnożonych przez dowolne liczby $\alpha_1, \alpha_2, \ldots, \alpha_k$

$$\alpha_1 w_1 + \alpha_2 w_2 + \cdots + \alpha_k w_k$$

jest się kombinacją liniową tych wierszy. Jeżeli macierz jest binarna, to dodawanie jest $\bmod 2$, więc kombinacja liniowa jest sumą $\bmod 2$ wybranych wierszy.

Jeżeli żaden z wierszy nie jest kombinacją liniową pozostałych wierszy, to te wiersze są niezależne. Wiersze niezależne, takie że wszystkie inne są ich kombinacjami liniowymi, tworzą bazę.

Przykład 8.2.

W macierzy binarnej

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

wiersz czwarty jest sumą wiersza drugiego i trzeciego. Niezależne są trzy pierwsze wiersze – są bazą. Wiersze pierwszy, drugi i czwarty też tworzą bazę.

Wszystkie definicje niezależności odnoszą się również do kolumn macierzy. Możemy więc też mówić o niezależnych kolumnach i o bazie kolumn, tak jak o bazie wierszy.

Przykład 8.3.

W macierzy binarnej (tej samej co poprzednio)

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

kolumna czwarta jest sumą trzech pierwszych kolumn, a kolumna pierwsza jest sumą trzech ostatnich.

8.2. Generowanie kodów przez macierze

Piszemy kody na tablicy

W tym punkcie ponownie odwołamy się do artykułu Aydin [3]. Zawarto w nim sporo bardzo użytecznych informacji o macierzach generujących kod.

Niech G będzie macierzą, której wiersze są słowami kodowymi pewnego kodu liniowego. Zbiór wszystkich niezależnych wierszy macierzy G nazywa się bazą kodu liniowego $[n,k]$, gdzie n jest liczbą kolumn, a k jest liczbą wierszy tej macierzy.

Przykład 8.4.

Kontrola parzystości w słowie kodowym czterobitowym daje nam kod blokowy $[4,3]$. W słowie kodowym trzy pierwsze bity to bity informacyjne, a czwarty bit to bit parzystości.

i	słowo kodowe				w_i
0	0	0	0	0	w_0
1	0	0	1	1	$w_1 = b_1$
2	0	1	0	1	$w_2 = b_2$
3	0	1	1	0	$w_3 = b_1 + b_2$
4	1	0	0	1	$w_4 = b_3$
5	1	0	1	0	$w_5 = b_1 + b_3$
6	1	1	0	0	$w_6 = b_2 + b_3$
7	1	1	1	1	$w_7 = b_1 + b_2 + b_3$

Wiersze, którymi są słowa kodowe b_1, b_2, b_3, tworzą bazę. Słowa kodowe b_i są wierszami macierzy

$$G_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Za bazę można też przyjąć wiersze w odwrotnej kolejności: b_3,b_2,b_1, co daje macierz

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Przykład 8.5.

Wróćmy znów do kodu Hamminga $(7,4)$ w postaci podanej w tabeli 4.1 na str. 69. Bazą są słowa kodowe zawierające ciągi złożone z bitów informacyjnych:

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Odpowiadającymi słowami kodowymi są wiersze macierzy

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Zapiszmy ciąg bitów informacyjnych w postaci jednowierszowej macierzy $[i_1\, i_2\, \ldots\, i_k]$. Macierz o k wierszach, której wiersze są słowami kodowymi i tworzą bazę kodu, nazywamy macierzą generującą kod. Jeśli G jest macierzą generującą kod $[n,k]$, czyli G ma k wierszy i n kolumn, to

$$[x_1\, x_2\, \ldots\, x_k]\, G$$

jest słowem kodowym dla ciągu $[i_1\, i_2\, \ldots\, i_k]$ w postaci jednowierszowej macierzy o n elementach.

Przykład 8.6.

Wróćmy do kontroli parzystości. Słowo trzybitowe kodujemy z kontrolą parzystości, korzystając z macierzy generujących G_1 i G_2.

$$[i_1\, i_2\, i_3] \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} = [i_3\, i_2\, i_1\, i_1 + i_2 + i_3]\,.$$

$$[i_1\, i_2\, i_3] \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = [i_1\, i_2\, i_3\, i_1 + i_2 + i_3]\,.$$

Dla tych dwóch przypadków dostajemy słowa kodowe o parzystej liczbie jedynek.

W następnym przykładzie znów wracamy do kodu Hamminga. Jest to kod wystarczająco prosty, aby dobrze zilustrować metodę, ale jednak nie tak banalny, jakim jest kontrola parzystości.

Przykład 8.7.

Kod Hamminga (7,4) w postaci podanej na str. 69 otrzymany za pomocą macierzy generującej z przykładu 8.5.

$$[i_1\, i_2\, i_3\, i_4] \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = [i_1 + i_3 + i_4\; i_1 + i_2 + i_4\; i_1\; i_1 + i_2 + i_3\; i_2\; i_3\; i_4]\,.$$

Przykład 8.8.

Kod Hamminga (7,4) w postaci (6.1) otrzymamy po odpowiednim przestawieniu kolumn macierzy z przykładu 8.7. Macierz G generująca kod będzie miała postać:

$$G = [I_k | A_{k\times m}]\,, \tag{8.1}$$

gdzie I_k jest macierzą jednostkową (wymiaru $k\times k$, jedynki na głównej przekątnej, poza tym zera), natomiast macierz $A_{k\times m}$ wskazuje, które bity informacyjne wchodzą do bitów kontrolnych jako sumy. W tym przykładzie $k = 4$ i $m = 3$:

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}. \tag{8.2}$$

W wyniku otrzymujemy słowa kodowe, w których na pierwszych czterech pozycjach są bity informacyjne, a na ostatnich trzech znajdują się bity kontrolne.

$$[i_3\, i_2\, i_1\, i_0] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [i_3\; i_2\; i_1\; i_0\; i_3 + i_2 + i_1\; i_2 + i_1 + i_0\; i_3 + i_2 + i_0]\,.$$

Na przykład ciąg czterech bitów informacyjnych 1011 da słowo kodowe 1011000.

Macierz generująca kod postaci (8.1) ma walor ogólniejszy niż tylko taki, że ma bardziej uporządkowany wygląd. Wprowadzimy bowiem macierz H kontroli parzystości, mającą n wierszy i m kolumn. Ma ona własność taką, że jeśli $c_1c_2\ldots c_n$ jest słowem kodowym o $k = n - m$ bitach informacyjnych i m bitach kontrolnych, to

$$H\,[c_1c_2\ldots c_n]^T = [0\ 0\ \ldots\ 0]^T\,. \tag{8.3}$$

Inaczej mówiąc, każdy wiersz macierzy H z każdym słowem kodowym tworzą parę ciągów (wektorów) ortogonalnych.

Twierdzenie 11.
Jeżeli G jest macierzą generującą kod postaci (8.1)*, to macierz kontroli parzystości H wyraża się wzorem*

$$H = \ A^T|I_m\ \ , \tag{8.4}$$

gdzie A^T jest macierzą transponowaną macierzy $A = A_{m\times k}$, ma więc m wierszy i k kolumn. □

Przykład 8.9.
Dla macierzy A określonej wzorem (8.2) macierz kontroli parzystości wyrazi się wzorem

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Sprawdźmy to dla słowa kodowego 1011001:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Jeśli zostanie przekłamany jeden bit i otrzymamy na przykład 1011001, to

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Daje to syndrom 001.

Ogólnie dla kodu Hamminga $(7,4)$, jeśli wystąpił jeden błąd, a odebranym słowem (w postaci jednowierszowej macierzy) jest $B = [b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7]$, to HB^T jest macierzą jednokolumnową. Cyfry w tej macierzy dają pozycję, w której wystąpił błąd w sposób określony w tabeli 8.1.

Tabela 8.1. Określenie pozycji błędu za pomocą syndromu

Syndrom	Pozycja
000	0000000
001	0000001
010	0000010
100	0000100
011	0001001
110	0010000
111	0100000
101	1000000

Przykład 8.10.

Z tabeli 8.1 otrzymujemy, że syndrom wskazuje na błąd na pozycji bitu siódmego, co istotnie ma miejsce.

Powróćmy jeszcze na chwilę do kodu dualnego do kodu Hamminga $(7,4)$, czyli do kodu $(7,3)$ – str. 96. Kod $(7,3)$ ma trzy bity informacyjne i cztery bity kontrolne.

Przykład 8.11.
Baza kodu, co widać w tabeli 6.4 (wiersze drugi, trzeci i piąty), to

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Słowa kodowe mają więc następującą postać:

$$\begin{aligned} &[i_1\ i_2\ i_3] \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \\ &\quad = [i_1\ i_2\ i_3\ i_2 + i_1\ i_3 + i_2\ i_3 + i_2 + i_1\ i_3 + i_1]. \end{aligned}$$

Otrzymany wynik sprawdźmy dla słów 011 i 111 i porównajmy z tabelą 6.4:

$$[0\,1\,1] \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} = [0\,1\,1\,1\,0\,0\,1].$$

$$[1\,1\,1] \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} = [1\,1\,1\,0\,0\,1\,0].$$

Macierz kontroli parzystości wyraża się wzorem (8.4), więc

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Przykład 8.12.
Słowo kodowe z przykładu 8.11: 0111001 zostało odebrane jako 0001000 – błąd polega na przekłamaniu bitów na trzech pozycjach. Sprawdzenie:

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Znaleziono błąd! Został wykryty, ale trzech błędów poprawić nie można. Jeśli jednak wiemy (a raczej zakładamy, bo tego na pewno nie wiemy), że nastąpił jeden błąd, to można go poprawić.

Jeśliby został przekłamany bit tylko na jednej pozycji, na przykład odebrano by 0111000, to

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Łatwo ten błąd poprawić po odnalezieniu w tabeli 6.4 słowa kodowego (jest ich tylko osiem) różniącego się od otrzymanego na jednej pozycji.

Z przykładów 8.7, 8.9 i 8.11 wynika, że macierz generująca kod w kodzie Hamminga $(7,4)$ jest macierzą kontroli parzystości w kodzie Hamminga $(7,3)$ dualnym do $(7,4)$ i na odwrót. Nie jest to przypadek. Zachodzi bowiem twierdzenie:

Twierdzenie 12.

Jeżeli C jest binarnym kodem liniowym o macierzy generującej kod G i macierzy kontroli parzystości H, to kod dualny C^d jest kodem liniowym o macierzy generującej kod H i macierzy kontroli parzystości G. □

Wróćmy do ternarnego kodu Hamminga $(4,2)$ przedstawionego w opowiadaniu 4 na str. 73.

Słowo kodowe ma postać (d_1,d_2,c_1,c_2). Macierzą generującą kod może być macierz G:

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix}.$$

Wtedy

$$[d_1\ d_2]\, G = [d_1\ d_2\ d_1 + d_2\ d_1 - d_2]\,,$$

co daje wynik zgodny z równaniami (4.8) i (4.9). Zauważmy też, że kod ten jest kodem liniowym, ale nie jest kodem cyklicznym.

Skorowidz

Naukę programowania zacznij od solidnych podstaw

Jak sądzisz, co stanowi bazę informatyki? Od czego powinien zacząć przyszły programista? Może od opanowania jednego z najpopularniejszych języków programowania? Oczywiście mógłby od tego rozpocząć, tyle że to trochę tak, jakby uczyć się korzystać z narzędzia bez świadomości, do czego ono właściwie służy. Języki programowania to praktyka. Tymczasem niezbędną wiedzą, którą także powinien opanować każdy przyszły informatyk, są podstawy dotyczące teorii informacji i kodowania.

Wraz z tą książką przyswoisz je bez konieczności odwoływania się do zaawansowanej matematyki i samej informatyki. Ten podręcznik obejmuje najważniejsze kwestie: od znaków, które przekazują informacje, źródeł informacji i sposobów mierzenia ilości przekazywanych danych po przetwarzanie sygnałów z analogowych na cyfrowe i odwrotnie. Po drodze zgłębisz takie zagadnienia jak podstawy kodowania (w tym kodowanie Huffmana), bezstratna kompresja i digitalizacja danych, grafiki i dźwięku, a także konstrukcja kodów liniowych i cyklicznych.

Dr hab. inż. Wojciech Kordecki — absolwent Wydziału Elektroniki Politechniki Wrocławskiej, uzyskał doktorat z matematyki na tej samej uczelni i habilitację z matematyki na Uniwersytecie im. Adama Mickiewicza w Poznaniu. W latach 1971–2008 pracował na Politechnice Wrocławskiej, obecnie jest profesorem uczelni w Collegium Witelona Uczelni Państwowej w Legnicy. Jego główna specjalizacja naukowa to matematyka dyskretna — grafy i matroidy losowe, ale również badania operacyjne, informatyka, teoria niezawodności i inżynieria biomedyczna.

Helion

helion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej! ▶

ISBN 978-83-289-0088-2

9 788328 900882

Cena: 59,00 zł