UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA

BAKALÁŘSKÁ PRÁCE

Aplikace lineární algebry

Katedra algebry a geometrie
Vedoucí bakalářské práce: **Doc. RNDr. Petr Emanovský Ph.D.**
Vypracoval: **Richter Marek**
Studijní program: B1101 Matematika
Studijní obor: Matematika a informatika se zaměřením na vzdělávání
Forma studia: Prezenční
Rok odevzdání: 2020

BIBLIOGRAFICKÁ IDENTIFIKACE

Autor: Richter Marek

Název práce: Aplikace lineární algebry

Typ práce: Bakalářská práce

Pracoviště: Katedra algebry a geometrie

Vedoucí práce: Doc. RNDr. Petr Emanovský Ph.D.

Rok obhajoby práce: 2020

Abstrakt: Bakalářská práce se zabývá aplikací lineární algebry, především v oblasti teorie grafů, teorie kódování a kryptografie. Práce je rozdělena na dva větší celky. V první části je pouze teoretická část, která je nedílnou součástí pro aplikaci. V druhé části jsou řešeny některé příklady z již zmíněné teorie grafů, teorie kódování a kryptografie pomocí lineární algebry.

Klíčová slova: matice, determinant, teorie grafů, teorie kódování, kryptografie, kód, graf, aplikace lineární algebry, sbírka úloh z lineární algebry

Počet stran: 76

Počet příloh: 3

Jazyk: Český

BIBLIOGRAPHICAL IDENTIFICATION

Author: Richter Marek

Title: Application of linear algebra

Type of thesis: Bachelor's

Department: Department of Algebra and Geometry

Supervisor: Doc. RNDr. Petr Emanovský Ph.D.

The year of presentation: 2020

Abstract: Bachelor thesis deals with application of linear algebra, especially in the field of graph theory, coding theory and cryptography. Bachelor thesis is devided into two major parts. The first part contains just a theoretical part, which is an important part for aplication. In the second part, there are solved some examples from the already mentioned graph theory, coding theory and cryptography by linear algebra.

Key words: matrix, determinant, graph theory, coding theory, cryptography, code, graph, application of linear algebra, collection of tasks from linear algebra

Number of pages: 76

Number of appendices: 3

Language: Czech

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně pod vedením pana Doc. RNDr. Petra Emanovského, Ph.D. a všechny použité zdroje jsem uvedl v seznamu literatury.

V Olomouci dne ....

podpis

Obsah

Poděkování

Děkuji panu Doc. RNDr. Petru Emanovskému Ph.D. za cenné rady, připomínky a čas, který mi věnoval při zpracování této bakalářské práce. Rovněž děkuji rodině, přítelkyni a přátelům za obrovskou podporu.

Úvod

Cílem této práce je vytvořit řešenou sbírku úloh speciálních aplikací lineární algebry v teoretické informatice. Na první pohled se může zdát, že teoretická informatika a lineární algebra nemají nic společného, avšak opak je pravdou. Názorným příkladem je takový obrázek, (předpokládejme pro jednoduchost šedotónový obrázek formátu .png) který je většinou v počítačové grafice reprezentován dvourozměrnou maticí, kde každá hodnota reprezentuje právě jeden pixel daného obrázku, respektive jeho barevnou složku. V této práci nebudeme zkoumat význam lineární algebry v počítačové grafice, ale podíváme se, dle mého názoru, na mnohem zajímavější aplikace.

Tento text je rozdělen na dva celky, konkrétněji tedy na teoretickou část a řešené příklady. V teoretické části zavedeme algebraické struktury, nad kterými posléze budeme definovat vektorové prostory, matice a determinanty. Dále se v této části podíváme na operaci modulo a také na množinu $\mathbb{Z}_n$. V závěru teoretické části si uděláme stručný úvod do teorie grafů, teorie kódování a kryptografie, jež mají blízký vztah k lineární algebře. Tyto znalosti a poznatky využijeme k řešení příkladů právě z oné teorie grafů, teorie kódování a kryptografie v následujícím celku.

V této práci se počítá s tím, že vážený čtenář má jistou zkušenost s teoretickou informatikou, a tedy především s lineární algebrou. Předpokládá se například, že čtenář ví, jaké číslo je prvočíslo, co je to zobrazení (spec. tedy bijekce), zná pojem Gaussovy eliminace, umí řešit homogenní i nehomogenní soustavy lineárních rovnic a další. Tento text by měl sloužit k prohloubení znalostí lineární algebry, konkrétněji tedy v oblasti teorie grafů, teorie kódování a kryptografie.

Seznam použitých symbolů

V této práci budeme používat následující symboly:

$\mathbb{N}$	množina přirozených čísel
$\mathbb{Z}$	množina celých čísel
$\mathbb{Z}[x]$	množina polynomů s celočíselnými koeficienty
$\emptyset$	množina prázdná
π	Ludolfovo číslo
$\forall x \in M : V(x)$	pro každé x z množiny M platí vlastnost (předpis) V
$\exists x \in M : V(x)$	existuje alespoň jedno x z množiny M takové, že pro něj platí vlastnost (předpis) V
$M \times N$	kartézský součin množin M a N
$M \to N$	množina M se zobrazí na množinu N
$x \mapsto y$	prvek x se zobrazí na prvek y
$x \in M$	prvek x leží v množině M
$M \subseteq N$	množina M je podmnožinou množiny N
$\|x\|$	absolutní hodnota z čísla x
$min\{x, y\}$	minimální prvek z množiny $\{x, y\}$
$sgn\, x$	znaménko čísla x (funkce *signum*)
$x < y$	prvek x je ostře menší než y
$x > y$	prvek x je ostře větší než y
$x \leq y$	prvek x je menší nebo roven y
$x \geq y$	prvek x je větší nebo roven y

V teoretické části tohoto textu budeme definovat další matematické symboly.

Teoretická část

1 Úvod do lineární algebry

1.1 Algebraické struktury

Definice 1.1.1. Je-li G neprázdná množina, potom *binární operací* na G rozumíme libovolné zobrazení $\circ : G \times G \to G$. [1]

Definice 1.1.2. *Algebraickou strukturou* $\mathcal{G}$ rozumíme některou neprázdnou množinu G spolu s neprázdným systémem $\{f_\alpha; \alpha \in I\}$ n-árních algebraických operací na G. (Číslo n může být pro různé operace různé.) Značíme $\mathcal{G} = (G, f_\alpha; \alpha \in I)$. [1]

Definice 1.1.3. Je-li „$\circ$“ binární operace na množině $G \neq \emptyset$, potom algebraická struktura $\mathcal{G} = (G, \circ)$ se nazývá *grupoid*. Je-li operace „$\circ$“ komutativní, tj. platí-li

$$\forall a, b \in G : a \circ b = b \circ a,$$

pak se $\mathcal{G}$ nazývá *komutativní grupoid*. [1]

Definice 1.1.4. *Pologrupou* rozumíme libovolný grupoid $\mathcal{G} = (G, \circ)$, ve kterém je operace „$\circ$“ asociativní, tj. platí-li

$$\forall a, b, c \in G : a \circ (b \circ c) = (a \circ b) \circ c.$$

Komutativní pologrupa se obvykle nazývá *abelovská*. [1]

Definice 1.1.5. Řekneme, že grupoid $\mathcal{G} = (G, \circ)$ má *neutrální prvek*, je-li pravdivý výrok

$$\exists n \in G \, \forall a \in G : a \circ n = a = n \circ a.$$

Každý takový prvek n nazýváme neutrálním prvkem grupoidu $\mathcal{G}$. [1]

Definice 1.1.6. Nechť grupoid $\mathcal{G} = (G, \circ)$ má neutrální prvek n a nechť $a \in G$. Pak $a^* \in G$ nazýváme *symetrickým prvkem* k prvku a, platí-li

$$a \circ a^* = n = a^* \circ a. \text{ [1]}$$

Definice 1.1.7. Pologrupa $\mathcal{G} = (G, \circ)$ se nazývá *grupa*, obsahuje-li neutrální prvek a existuje-li v ní ke každému prvku symetrický prvek, tj. jsou-li pravdivé výroky

$$\exists n \in G \, \forall a \in G : a \circ n = a = n \circ a,$$
$$\forall a \in G \, \exists a^* \in G : a \circ a^* = n = a^* \circ a.$$

Komutativní grupa se také nazývá *abelovská*. [1]

Definice 1.1.8. *Okruhem* nazýváme algebraickou strukturu $\mathcal{M} = (M, +, \cdot)$ se dvěma binárními operacemi „+“ a „$\cdot$“ takovou, že $(M, +)$ je komutativní grupa, $(M, \cdot)$ je pologrupa a operace násobení je zleva i zprava distributivní vzhledem k operaci sčítaní, tj.

$$\forall a, b, c \in M : a \cdot (b + c) = a \cdot b + a \cdot c,$$
$$\forall a, b, c \in M : (a + b) \cdot c = a \cdot c + b \cdot c.$$

Je-li pologrupa $(M, \cdot)$ komutativní, pak se okruh nazývá komutativní. [1]

Poznámka 1.1.1. V komutativní grupě $\mathcal{M} = (M, +)$ budeme neutrální prvek značit symbolem „o“ a nazývat jej *nulovým prvkem* okruhu $\mathcal{M} = (M, +, \cdot)$.

Definice 1.1.9. Prvek $a \neq o$ okruhu $\mathcal{M} = (M, +, \cdot)$ se nazývá *levý* (*pravý*) *netriviální dělitel nuly*, existuje-li $b \neq o$, $b \in M$ takový, že $a \cdot b = o \, (b \cdot a = o)$. [1]

Definice 1.1.10. Řekneme, že $a \neq o$ je *netriviálním dělitelem nuly*, je-li a současně levým i pravým netriviálním dělitelem nuly. [1]

Definice 1.1.11. *Oborem integrity* budeme rozumět každý okruh $\mathcal{J} = (J, +, \cdot)$, který je komutativní, obsahuje neutrální prvek (vzhledem k násobení) $n \neq o$ a ve kterém neexistují netriviální dělitelé nuly. [1]

Definice 1.1.12. Okruh $\mathcal{T} = (T, +, \cdot)$ se nazývá *těleso*, jestliže T obsahuje alespoň dva navzájem různé prvky a je-li $(T \setminus \{o\}, \cdot)$ grupou. Je-li $(T, \cdot)$ komutativní, pak se $\mathcal{T}$ nazývá komutativní těleso. [1]

1.2 Vektorové prostory

Definice 1.2.1. Jsou-li A a B neprázdné množiny, potom *levou vnější operací* nad množinami A a B (v tomto pořadí) rozumíme každé zobrazení $\circ : A \times B \to B$. (Jsou-li $a \in A$, $b \in B$, pak prvek $\circ(a, b)$ budeme označovat $a \circ b$). [1]

Definice 1.2.2. Nechť $(V, +)$ je komutativní grupa (její prvky budeme označovat např. $\vec{u}, \vec{v}$, nulový prvek $\vec{o}$), $\mathcal{T}$ číselné těleso, $\circ : T \times V \to V$ levá vnější operace nad T a V. Potom systém $\mathcal{V} = (V, +, \mathcal{T}, \circ)$ nazveme *vektorový prostor* nad tělesem $\mathcal{T}$, platí-li:

1. $\forall c \in T, \vec{u}, \vec{v} \in V : c \circ (\vec{u} + \vec{v}) = c \circ \vec{u} + c \circ \vec{v}$,
2. $\forall c, d \in T, \vec{u} \in V : (c + d) \circ \vec{u} = c \circ \vec{u} + d \circ \vec{u}$,
3. $\forall c \in T, \vec{u} \in V : (c \cdot d) \circ \vec{u} = c \circ (d \circ \vec{u})$,
4. $\forall \vec{u} \in V : 1 \circ \vec{u} = \vec{u}$.

Prvky z komutativní grupy $(V, +)$ vektorového prostoru nazýváme *vektory* a čísla z tělesa $\mathcal{T}$ *skaláry*. Vektorový prostor se někdy také nazývá *lineární prostor*. [1]

Poznámka 1.2.1. Operace „+“ v předchozí definici bude symbolizovat sčítaní v grupě $(V, +)$ a také sčítání čísel v tělese $\mathcal{T}$. Levá vnější operace „$\circ$“ bude symbolizovat násobení vektorů skalárem. Místo symbolu „$\circ$“ budeme používat symbol „$\cdot$“. [1]

Definice 1.2.3. Je-li $\mathcal{V}$ vektorový prostor nad číselným tělesem $\mathcal{T}$ a jsou-li $\vec{v}, \vec{u_1}, \ldots, \vec{u_k} \in V$, pak řekneme, že vektor $\vec{v}$ *je lineární kombinací vektorů* $\vec{u_1}, \ldots, \vec{u_k}$, existují-li čísla $c_1, \ldots, c_k \in T$ taková, že platí

$$\vec{v} = \sum_{i=1}^{k} c_i \cdot \vec{u_i}. \text{ [1]}$$

Poznámka 1.2.2. Jsou-li všechny koeficienty c_i, kde $i = 1, 2 \ldots, k$ v předchozí definici nulové, je lineární kombinace označována jako *triviální*. Je-li alespoň jeden z koeficientů $c_i \neq 0$, pak říkáme že lineární kombinace je *netriviální*.

Definice 1.2.4. Vektory $\vec{u_1}, \ldots, \vec{u_k}$ z vektorového prostoru $\mathcal{V}$ se nazývají *lineárně závislé*, existuje-li alespoň jedna jejich netriviální kombinace, která je rovna nulovému vektoru $\vec{o}$ (nazýváme tzv. *nulová kombinace*). V opačném případě se

vektory $\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}$ nazývají *lineárně nezávislé*. [1]

Věta 1.2.1. Jsou-li $\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}$ vektory z vektorového prostoru $\mathcal{V}$, pak jsou tyto vektory lineárně závislé právě tehdy, je-li alespoň jeden z těchto vektorů lineární kombinací ostatních vektorů. [1]

Definice 1.2.5. Řekneme, že vektorový prostor $\mathcal{W} = (W, \oplus, \mathcal{T}, \otimes)$ je *podprostorem* vektorového prostoru $\mathcal{V} = (V, +, \mathcal{T}, \cdot)$, platí-li:

1. $W \subseteq V$,
2. $\forall \overrightarrow{u}, \overrightarrow{v} \in W : \overrightarrow{u} \oplus \overrightarrow{v} = \overrightarrow{u} + \overrightarrow{v}$,
3. $\forall c \in T, \overrightarrow{u} \in W : c \otimes \overrightarrow{u} = c \cdot \overrightarrow{u}$. [1]

Poznámka 1.2.3. Vektorový podprostor se někdy také nazývá *lineární podprostor*. Dle předchozí definice lineární podprostor zachovává operace, neboli jinak řečeno, lineární podprostor je uzavřený na sčítaní v grupě $(V, +)$ a také na násobení vektorů skalárem.

Definice 1.2.6. Je-li M podmnožina vektorového prostoru $\mathcal{V}$, pak *lineárním obalem* množiny M ve $\mathcal{V}$ rozumíme průnik všech podprostorů prostoru $\mathcal{V}$ obsahujících množinu M. Lineární obal množiny M budeme značit symbolem $[M]$. [1]

Definice 1.2.7. Nechť $\mathcal{V}$ je vektorový prostor nad číselným tělesem $\mathcal{T}$. Platí-li pro podmnožinu $M \neq \emptyset$ prostoru $\mathcal{V}$, že $[M] = \mathcal{V}$, pak M se nazývá *množina generátorů* prostoru $\mathcal{V}$. Říkáme také, že množina M *generuje* prostor $\mathcal{V}$. [1]

Definice 1.2.8. Řekneme, že vektorový prostor $\mathcal{V}$ je *konečné dimenze*, jestliže existuje alespoň jedna jeho konečná množina generátorů. [1]

Definice 1.2.9. *Bází vektorového prostoru* $\mathcal{V}$ konečné dimenze rozumíme libovolnou lineárně nezávislou množinu $\{\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}\}$ jeho generátorů. [1]

Věta (Steinitzova) 1.2.2. Nechť $\{\overrightarrow{u_1}, \ldots, \overrightarrow{u_n}\}$ je množina generátorů vektorového prostoru $\mathcal{V} \neq \{\overrightarrow{o}\}$ a $\overrightarrow{v_1}, \ldots, \overrightarrow{v_k}$ jsou lineárně nezávislé vektory z $\mathcal{V}$. Potom platí, že $k \leq n$ a že po vhodném očíslování vektorů $\overrightarrow{u_1}, \ldots, \overrightarrow{u_n}$ je množina $\{\overrightarrow{v_1}, \ldots, \overrightarrow{v_k}, \overrightarrow{u_{k+1}}, \ldots, \overrightarrow{u_n}\}$ množinou generátorů prostoru $\mathcal{V}$. [1]

Věta 1.2.3. Je-li $\mathcal{V} \neq \{\overrightarrow{o}\}$ vektorový prostor konečné dimenze, potom každé dvě jeho báze mají stejný počet prvků. [1]

Definice 1.2.10. Je-li $\mathcal{V} \neq \{\vec{o}\}$ vektorový prostor konečné dimenze, pak počet prvků jeho libovolné báze nazýváme *dimenze* prostoru $\mathcal{V}$ a značíme $\dim\mathcal{V}$. Je-li $\mathcal{V} = \{\vec{o}\}$, pak $\dim\mathcal{V} = 0$. [1]

1.3 Operace modulo

Definice 1.3.1. Řekneme, že celé číslo a *dělí* celé číslo b (tento fakt značíme $a \,|\, b$), pokud existuje celé číslo n takové, že $b = n \cdot a$. [3]

Poznámka 1.3.1. Jestliže číslo a dělí číslo b (v oboru celých čísel), pak číslo a nazýváme *dělitelem* čísla b. [3]

Definice 1.3.2. *Prvočíselným rozkladem* kladného celého čísla označujeme zápis

$$p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r},$$

kde $r \geq 1$ je celé číslo, $p_1 < p_2 < \ldots < p_r$ jsou navzájem různá prvočísla a $n_1, n_2, \ldots, n_r$ jsou kladná celá čísla. [3]

Věta 1.3.1. Pro každé celé číslo $x \geq 2$ existuje až na pořadí činitelů a asociovanost jednoznačně prvočíselný rozklad. [3]

Definice 1.3.3. Řekneme, že kladné celé číslo d je *největším společným dělitelem* kladných celých čísel a, b (značíme $d = NSD(a, b)$), pokud jsou splněny následující dvě podmínky:

1. Číslo d je společným dělitelem čísel a, b, tj. platí $d \,|\, a$ a zároveň $d \,|\, b$ (v oboru kladných celých čísel).

2. Číslo d je největším ze všech společných dělitelů čísel a, b, tj. platí následující: je-li c takové kladné celé číslo, pro které platí $c \,|\, a$ a zároveň $c \,|\, b$, potom $c \,|\, d$. [3]

Věta 1.3.2. Nechť a, b jsou libovolná celá čísla, $b \neq 0$. Pak existují jednoznačně určená celá čísla q a r taková, že jsou splněny následující dvě podmínky:

1. Platí rovnost $a = q \cdot b + r$.

2. Číslo r splňuje nerovnost $0 \leq r < |b|$. [3]

Definice 1.3.4. Jednoznačně určené číslo r z předchozí věty se nazývá *zbytek po dělení čísla a číslem b* a toto číslo značíme $a \,mod\, b$. [3]

Definice 1.3.5. Nechť $n \in \mathbb{N}$. Řekneme, že čísla $a, b \in \mathbb{Z}$ jsou *kongruentní modulo* n, jestliže $n \mid (a - b)$. Tuto skutečnost značíme $a \equiv b \,(mod\, n)$. [4]

Poznámka 1.3.2. V předchozí definici relace „*být kongruentní modulo*“ značí, že čísla a, b mají stejné zbytky po dělení modulem n. Tento fakt zachycuje následující věta. [4]

Věta 1.3.3. Nechť $n \in \mathbb{N}$. Pro čísla $a, b \in \mathbb{Z}$ jsou následující podmínky ekvivalentní:

1. $a \equiv b \,(mod\, n)$,
2. existuje $k \in \mathbb{Z}$ takové, že $a = b + k \cdot n$,
3. $a \,mod\, n = b \,mod\, n$, tj. jsou si rovny zbytky po dělení číslem n. [4]

Věta 1.3.4. Nechť $n \in \mathbb{N}$. Pak platí:

1. Pro každé $c \in \mathbb{Z}$ je $c \equiv c \,(mod\, n)$,
2. Nechť $a \in \mathbb{Z}$. Potom platí, že $a \equiv 0 \,(mod\, n)$ právě tehdy, když n dělí a. [4]

Definice 1.3.6. Nechť $n \in \mathbb{N}$. Symbolem $\mathbb{Z}_n$ značíme množinu $\mathbb{Z}_n = \{0, 1, \ldots, n - 1\}$. Pro všechna $a, b \in \mathbb{Z}_n$ definujeme operace

$$a \oplus b = (a + b) \,mod\, n,$$

$$a \odot b = (a \cdot b) \,mod\, n. \,[4]$$

Poznámka 1.3.3. Takto zavedenou množinu někdy také nazýváme *množinou zbytkových tříd*. V celém tomto textu (pokud nebude řečeno jinak) budeme pracovat s množinou $\mathbb{Z}_p$, kde p je prvočíslo, čímž se vyhneme jistým problémům, neboť množina $\mathbb{Z}_p$ je tělesem. Symbolem $\mathbb{Z}_p^k$ budeme rozumět množinu všech uspořádaných k-tic nad $\mathbb{Z}_p$. [4]

1.4 Matice a jejich základní vlastnosti

Definice 1.4.1. Nechť $\mathcal{T} = (T, +, \cdot)$ je číselné těleso, $m, n \in \mathbb{N}, a_{ij} \in T, i = 1, 2, \ldots, m$, $j = 1, 2, \ldots n$. Potom schéma

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

se nazývá *matice typu* $m \times n$ nad $\mathcal{T}$. Tuto matici A budeme zapisovat ve zkráceném tvaru $A = (a_{ij})$ typu $m \times n$. Je-li $r = min\{m, n\}$, pak řekneme, že prvky $a_{11}, a_{22}, \ldots, a_{rr}$ tvoří *hlavní diagonálu* matice A. [1]

Poznámka 1.4.1. Pokud matice bude typu $n \times n$ nad $\mathcal{T}$, budeme hovořit o tzv. *čtvercové matici řádu n.* Předchozí definice zavádí matice nad číselným tělesem. V tomto textu se objeví i matice, které místo čísel budou mít znaky a symboly. S takovými maticemi nebudeme dělat žádné operace.

Definice 1.4.2. Matici $O = (o_{ij})$ typu $m \times n$ nazýváme *nulovou maticí*, platí-li $o_{ij} = 0$ pro každé $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots n$. [1]

Definice 1.4.3. Čtvercovou matici $E = (e_{ij})$ řádu n nazýváme *jednotkovou maticí*, jestliže všechny prvky na její hlavní diagonále jsou rovny 1 a všechny prvky mimo hlavní diagonálu jsou rovny 0.

Definice 1.4.4. Je-li $A = (a_{ij})$ matice typu $m \times n$, potom *maticí transponovanou* k matici A nazýváme matici $A^T = (a_{ji})$ typu $n \times m$, která vznikne z A vzájemnou záměnou řádků a sloupců. [1]

Definice 1.4.5. Nechť $A = (a_{ij})$ je čtvercová matice řádu n. *Symetrickou maticí* k matici A rozumíme každou matici, která splňuje rovnost $A = A^T$. [1]

Definice 1.4.6. Nechť $A = (a_{ij}), B = (b_{ij})$ jsou matice typu $m \times n$ nad $\mathcal{T}$. Potom *součtem matic* A a B rozumíme matici $A + B = (c_{ij})$ typu $m \times n$ nad $\mathcal{T}$ takovou, že $c_{ij} = a_{ij} + b_{ij}$ pro každé $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. [1]

Definice 1.4.7. Nechť $A = (a_{ij})$ je matice typu $m \times n$ nad $\mathcal{T}$ a nechť $c \in \mathcal{T}$. Potom *součinem skaláru* c a *matice* A rozumíme matici $c \cdot A = (c \cdot a_{ij})$ typu $m \times n$ nad $\mathcal{T}$. [1]

Definice 1.4.8. Nechť $A = (a_{ij})$ je matice typu $m \times n$ nad $\mathcal{T}$, $B = (b_{jk})$ je matice typu $n \times p$ nad $\mathcal{T}$. Potom *součinem matic* A a B (v tomto pořadí) rozumíme matici $A \cdot B = (c_{ik})$ typu $m \times p$ nad $\mathcal{T}$ takovou, že

$$c_{ik} = \sum_{j=1}^{n} a_{ij} \cdot b_{jk},$$

pro každé $i = 1, 2, \ldots, m$, $k = 1, 2, \ldots, p$. [1]

Definice 1.4.9. *Elementárními řádkovými transformacemi matice* A nazýváme následující operace:

1. výměna libovolných dvou řádků v A,
2. vynásobení některého řádku v A prvkem z $\mathcal{T}$ různým od nuly,
3. přičtení libovolného násobku některého řádku z A k jinému řádku v A. [1]

Definice 1.4.10. Jsou-li A, B matice typu $m \times n$ nad $\mathcal{T}$, pak řekneme, že matice B je *řádkově ekvivalentní* s maticí A (značíme $A \sim B$), může-li B vzniknout z A pomocí konečného počtu elementárních řádkových transformací. [1]

Definice 1.4.11. *Hodností matice* $A = (a_{ij})$ typu $m \times n$ nad $\mathcal{T}$ rozumíme číslo $h(A)$, které je rovno maximálnímu počtu lineárně nezávislých řádků matice A. [1]

Věta 1.4.1. Řádkově ekvivalentní matice mají stejnou hodnost. [1]

Definice 1.4.12. Čtvercovou matici $A = (a_{ij})$ řádu n nazveme *regulární*, jestliže je její hodnost $h(A) = n$. V opačném případě nazveme matici A *singulární*.

Definice 1.4.13. Pro každé $n \in \mathbb{N}$ budeme (*řádkovým*) *n-rozměrným aritmetickým vektorovým prostorem* nad $\mathcal{T}$ (značíme $\mathcal{T}^n$) rozumět komutativní grupu $(M_{1\times n}(\mathcal{T}), +)$ všech matic typu $1 \times n$ nad $\mathcal{T}$ uvažovanou spolu s násobením matic z $M_{1\times n}(\mathcal{T})$ skaláry z $\mathcal{T}$. Každou matici z $M_{1\times n}(\mathcal{T})$ pak nazveme *n-rozměrný* (*řádkový*) *aritmetický vektor* nad $\mathcal{T}$. [1]

Definice 1.4.14. Je-li $\vec{u} = (u_1, \ldots, u_n)$ n-rozměrný aritmetický vektorový vektor, pak prvky $u_1, \ldots, u_n \in \mathcal{T}$ nazýváme *souřadnice vektoru* $\vec{u}$. [1]

Poznámka 1.4.2. V tomto textu budeme uvažovat *řádkové vektory*, tj. vektory, které mají souřadnice zapsány v řádku. Vektory, které budou mít souřadnice zapsány ve sloupci, budeme nazývat *vektory sloupcové*. *Transponovaný vektor* k řádkovému vektoru je vektor sloupcový a naopak.

1.5 Pořadí, permutace, determinant

Definice 1.5.1. Je-li $A = \{a_1, a_2, \ldots, a_n\}$, kde $n \geq 1$, konečná množina, potom *pořadím* množiny A nazveme libovolnou posloupnost

$$\pi = (a_{k_1}, a_{k_2}, \ldots, a_{k_n})$$

prvků z A takovou, že každý prvek z množiny A se v π vyskytuje právě jednou. [1]

Definice 1.5.2. *Permutací* P na množině A rozumíme každou bijekci A na A. Je-li P permutace na A, pak ji budeme psát ve tvaru schématu

$$P = \begin{pmatrix} i_1 & i_2 & \ldots & i_n \\ P(i_1) & P(i_2) & \ldots & P(i_n) \end{pmatrix}. [1]$$

Definice 1.5.3. Je-li $\pi = (k_1, k_2, \ldots, k_n)$ pořadí, pak řekneme, že prvky k_i a k_j tvoří v pořadí π *inverzi*, platí-li $i < j$ a $k_i > k_j$. [1]

Poznámka 1.5.1. Počet inverzí pořadí π budeme v tomto textu označovat symbolem $[\pi]$. [1]

Definice 1.5.4. *Znaménkem pořadí* π rozumíme číslo $sgn\,\pi = (-1)^{[\pi]}$. [1]

Definice 1.5.5. *Znaménkem permutace*

$$P = \begin{pmatrix} i_1 & i_2 & \ldots & i_n \\ k_1 & k_2 & \ldots & k_n. \end{pmatrix} = \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}$$

rozumíme číslo $sgn\,P$, které se rovná $+1$, platí-li $sgn\,\pi_1 = sgn\,\pi_2$, a rovná se -1, platí li $sgn\,\pi_1 = -sgn\,\pi_2$. [1]

Definice 1.5.6. Nechť $A = (a_{ij})$ je čtvercová matice řádu n nad číselným tělesem $\mathcal{T}$. *Determinantem* matice A rozumíme číslo $detA$ z tělesa $\mathcal{T}$ takové, že

$$detA = \sum_P sgnP \cdot a_{1k_1} \cdot a_{2k_2} \cdot \ldots \cdot a_{nk_n},$$

kde sčítáme přes všechny permutace

$$P = \begin{pmatrix} 1 & 2 & \ldots & n \\ k_1 & k_2 & \ldots & k_n. \end{pmatrix} = \begin{pmatrix} 1 & 2 & \ldots & n \\ P(1) & P(2) & \ldots & P(n) \end{pmatrix}$$

množiny $\{1, 2, \ldots, n\}$. Každý ze součinů $a_{1k_1} \cdot a_{2k_2} \cdot \ldots \cdot a_{nk_n}$ nazýváme *člen determinantu detA*. [1]

Věta 1.5.1. Má-li čtvercová matice A řádu n nad $\mathcal{T}$ v některém řádku samé nuly, pak $detA = 0$. [1]

Věta 1.5.2. Má-li čtvercová matice A řádu n nad $\mathcal{T}$ všechny prvky pod hlavní diagonálou rovny nule, potom $detA$ je roven součinu $a_{11}, a_{22}, \ldots, a_{nn}$ prvků na hlavní diagonále. [1]

Definice 1.5.7. Nechť $A = (a_{ij})$ je matice typu $m \times n$ nad $\mathcal{T}$. Potom každou matici, která vznikne z matice A vynecháním některých řádku a některých sloupců, nazýváme *submatice* matice A. Je-li submatice matice A čtvercová, potom její determinant nazýváme *subdeterminant* matice A. [1]

Definice 1.5.8. Je-li $A = (a_{ij})$ čtvercová matice řádu n nad $\mathcal{T}$, potom subdeterminant submatice A_{ij} řádu $(n-1)$ vzniklé vynecháním i-tého řádku a j-tého sloupce A nazýváme *minor* matice A příslušný k prvku a_{ij} a značíme jej $\mathcal{M}_{ij}$. *Algebraickým doplňkem* prvku a_{ij} rozumíme prvek $\mathcal{A}_{ij} = (-1)^{i+j} \cdot \mathcal{M}_{ij}$. [1]

Věta (Laplaceova) 1.5.3. Nechť $A = (a_{ik})$ je čtvercová matice řádu n nad $\mathcal{T}$. Potom pro každé $i = 1, 2, \ldots, n$ platí

$$\sum_{k=1}^{n} a_{ik} \cdot \mathcal{A}_{ik} = detA,$$

a pro každé $i, j = 1, 2, \ldots, n$, $i \neq j$ platí

$$\sum_{k=1}^{n} a_{ik} \cdot \mathcal{A}_{jk} = 0. \, [1]$$

Definice 1.5.9. Nechť $A = (a_{ij})$ je matice řádu n nad $\mathcal{T}$. *Inverzní maticí* k matici A, budeme rozumět matici A^{-1}, pro kterou platí $A \cdot A^{-1} = E = A^{-1} \cdot A$. Matice A, ke které existuje inverzní matice A^{-1}, se nazývá *invertibilní*.

Věta 1.5.4. Je-li $A = (a_{ij})$ čtvercová matice řádu n nad $\mathcal{T}$, potom k ní existuje inverzní matice A^{-1} tehdy a jen tehdy, jeli $detA \neq 0$. [1]

Poznámka 1.5.2. Inverzní matici A^{-1} nalezneme tak, že vytvoříme matici $(A \,|\, E)$, kterou se snažíme elementárními řádkovými transformacemi dostat do

tvaru $(E \mid A^{-1})$. Tento postup platí pro všechny matice, které splňují kritérium předchozí věty, tedy pro všechny regulární matice.

1.6 Lineární algebra nad množinou $\mathbb{Z}_n$

Poznámka 1.6.1. Pojmy z lineární algebry, které byly zavedeny v předchozích kapitolách, jako např. vektory, matice, determinanty a další, se dají analogicky jako pro reálná čísla zavést i pro množinu $\mathbb{Z}_n$. Sčítání a násobení na této množině jsme si definovali v kapitole 1.3. Rozdíl mezi $\mathbb{Z}$ a $\mathbb{Z}_n$ nastává v případě, kdy chceme nalézt inverzní matici A^{-1}. Tuto skutečnost zachycuje následující věta. [4]

Věta 1.6.1. Ke čtvercové matici A nad $\mathbb{Z}_n$ existuje inverzní matice A^{-1} právě tehdy, když determinant matice A je invertibilní prvek $\mathbb{Z}_n$. Pak

$$A^{-1} = (detA)^{-1} \cdot (\mathcal{A}_{ij})^T,$$

kde $(\mathcal{A}_{ij})$ je matice algebraických doplňků matice A. [4]

Definice 1.6.1. Nechť $A = (a_{ij}), B = (b_{ij})$ jsou matice typu $m \times n$ nad $\mathcal{T}$ a nechť $n \in \mathbb{N}$. Řekneme, že matice A a B jsou *kongruentní modulo n*, jestliže pro každý prvek a_{ij} matice A a každý prvek b_{ij} matice B platí $n \mid (a_{ij} - b_{ij})$ pro všechna $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots n$. Tento fakt značíme $A \equiv_n B$. [4]

Poznámka 1.6.2. Problém v $\mathbb{Z}_n$ nastává, pokud chceme nalézt hodnost $h(A)$ matice A. Dá se dokázat, že v $\mathbb{Z}_n$ neplatí rovnost $h(A^T) = h(A)$. Tomuto problému se v tomto textu vyhneme tak, že budeme pracovat nad množinou $\mathbb{Z}_p$, kde p je prvočíslo (pokud nebude řečeno jinak), jak již bylo zmíněno v kapitole 1.3. Tato množina je tělesem, díky čemuž se vyhneme jistým nepříjemnostem při hledání hodnosti $h(A)$ matice A. Dále bude možné díky tomuto tělesu hledat inverzní matici převodem popsaným na konci kapitoly 1.5. [4]

Poznámka 1.6.3. Další nepříjemností v $\mathbb{Z}_n$ může být *Gaussova eliminační metoda*. Tato metoda v $\mathbb{Z}_n$ přestává být spolehlivým nástrojem pro výpočet determinantu či řešení *soustav lineárních rovnic*. Vzhledem k faktu, že v tomto textu jsme Gaussovu eliminační metodu ani soustavy lineárních rovnic nezaváděli, není nutné tuto skutečnost řešit, ale je dobré ji alespoň zmínit. [4]

2 Teorie grafů

2.1 Stručný úvod do teorie grafů

Teorie grafů nám dává v matematice a informatice velmi užitečnou a praktickou pomůcku k řešení nejrůznějších druhů problémů. Graf je poměrně obecný pojem a vyskytuje se v různých významech. Zde bude tento pojem představovat grafický způsob vyjádření vztahů mezi nějakými objekty. Objekty budou v grafu reprezentovány vrcholy. Vztahy budou v grafu reprezentovány hranami.

Definice 2.1.1. *Jednoduchý graf* G je uspořádaná dvojice $(V(G), E(G))$, kde $V(G)$ je neprázdná množina vrcholů a $E(G)$ je nějaká množina dvouprvkových podmnožin množiny $V(G)$. Prvkům $E(G)$ říkáme *hrany*. [5]

Poznámka 2.1.1. Pokud z kontextu bude zřejmé, o který graf se jedná, budeme pro množinu vrcholů používat symbol V a pro množinu hran symbol E. Symbolem $|V(G)|$ budeme rozumět počet vrcholů a symbolem $|E(G)|$ počet hran grafu G. Vrcholy budeme značit symbolem v_i a hrany symbolem e_i, přičemž $e_i = \{v_i, v_j\}$. V tomto textu budeme pracovat výhradně s konečnými grafy. Vrcholy v_i a v_j nazýváme *koncové vrcholy* hrany e_i. Je-li vrchol v_i koncovým vrcholem hrany e_i, říkáme, že vrchol v_i *je incidentní* s hranou e_i. *Násobnou hranou* budeme rozumět více hran spojujících stejné vrcholy. [5]

Poznámka 2.1.2. Takto zavedená definice jednoduchého grafu nedovoluje, aby oba koncové vrcholy hrany byly stejné, protože by se nejednalo o dvouprvkovou podmnožinu množiny $V(G)$. Takovým hranám se říká *smyčky*. V celém tomto textu budeme předpokládat, že graf G neobsahuje smyčky. Dále si z definice můžeme všimnout, že graf G nemůže být prázdný. Graf G s množinou vrcholů $V(G) = \{v_1, \ldots, v_n\}$, pro $n \geq 3$ a množinou hran $E(G) = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots,$

$\{v_{n-1}, v_n\}, \{v_n, v_1\}\}$ se nazývá *cyklus* (někdy také *kružnice*). [5]

Definice 2.1.2. *Stupeň vrcholu* v_i je počet hran, se kterými je vrchol v_i incidentní. Stupeň vrcholu budeme značit symbolem $deg(v_i)$. [5]

Definice 2.1.3. *Orientovaným grafem* rozumíme uspořádanou dvojici $G = (V, E)$, kde $V(G)$ je množina vrcholů a $E(G) \subseteq V(G) \times V(G)$ je množina orientovaných hran. [5]

Poznámka 2.1.3. Každý graf G, na kterém není zavedená orientace se nazývá *neorientovaný graf G. Orientace* je zde chápaná jako směr hrany e_i od vrcholu v_i do vrcholu v_j.

Definice 2.1.4. Řekneme, že graf G je *ohodnocený*, jestliže jsou hranám nebo vrcholům přiřazena nenulová čísla. [5]

Definice 2.1.5. Mějme dán graf $G = (V, E)$. Řekneme, že graf $H = (V', E')$ je *podgrafem* grafu G, jestliže $V' \subseteq V$ a současně $E' \subseteq E$. [5]

Poznámka 2.1.4. Speciálním případem podgrafu je podgraf, který obsahuje všechny vrcholy původního grafu (tj. $V = V'$). Takovému podgrafu říkáme *faktor*. [5]

Definice 2.1.6. *Sled* v grafu G je taková posloupnost vrcholů a hran

$$(v_o, \{v_0, v_1\}, v_1, \{v_1, v_2\}, v_2, \ldots, \{v_{n-1}, v_n\}, v_n),$$

že hrana $e_i = \{v_{i-1}, v_i\}$ má koncové vrcholy v_{i-1} a v_i pro všechna $i = 1, 2, \ldots, n$. [5]

Poznámka 2.1.5. Sled uvedený v předchozí definici nazýváme (v_o, v_n)-sled. [5]

Definice 2.1.7. Graf je *souvislý*, jestliže pro každou dvojici vrcholů $v_i, v_j \in V(G)$ existuje (v_i, v_j)-sled. [5]

Definice 2.1.8. Graf se nazývá *acyklický*, jestliže žádný jeho podgraf není cyklus. Souvislý acyklický graf se nazývá *strom*. [5]

Definice 2.1.9. Faktor grafu, který je stromem, se nazývá *kostrou grafu*. [5]

Definice 2.1.10. Nechť G je souvislý graf spolu s ohodnocením hran c, tj. pro každou hranu $e_i \in E(G)$ je dáno číslo $c(e_i)$ (číslo $c(e_i)$ nazýváme *cenou* hrany e_i). *Minimální kostra grafu* $G = (V, E)$ je taková kostra grafu $K = (V, E')$, že $\sum_{e_i \in E'} c(e_i)$ je nejmenší možná (mezi všemi kostrami grafu G). [8]

Definice 2.1.11. *Matice incidence* $I(G) = (g_{ij})$ orientovaného grafu $G = (V, E)$, kde $|V(G)| = m$ a $|E(G)| = n$ je matice typu $m \times n$ definovaná vztahem

$$g_{ij} = \begin{cases} +1 & \text{pokud hrana } e_j \text{ vychází z vrcholu } v_i, \\ -1 & \text{pokud hrana } e_j \text{ vchází do vrcholu } v_i, \\ 0 & \text{jinak. [6]} \end{cases}$$

Definice 2.1.12. *Matice sousednosti* $S(G) = (s_{ij})$ orientovaného grafu $G = (V, E)$, kde $|V(G)| = n$ je čtvercová matice řádu n definována vztahem

$$s_{ij} = \begin{cases} 1 & \text{jestliže } e_i \in E(G) \\ 0 & \text{jinak.} \end{cases}$$

Pro neorientovaný graf platí, že matice sousednosti $S(G)$ je symetrická. [6]

Definice 2.1.13. *Laplaceova matice sousednosti* $L(G) = (l_{ij})$ neorientovaného grafu $G = (V, E)$ s množinou vrcholů $V(G) = \{v_1, v_2, \ldots, v_n\}$, kde G je bez smyček a násobných hran je čtvercová matice řádu n definovaná vztahem

$$l_{ij} = \begin{cases} \deg(v_i) & \text{pokud } i = j, \\ -1 & \text{pokud } \{v_i, v_j\} \in E(G), \\ 0 & \text{jinak. [6]} \end{cases}$$

Definice 2.1.14. Počet koster neorientovaného grafu $G = (V, E)$ je roven determinantu matice $L'(G)$, která vznikne vypuštěním posledního řádku a sloupce z matice $L(G)$. [6]

2.2 Grafové algoritmy

Kruskalův algoritmus

Kruskalův algoritmus je jeden z algoritmů využívaných k nalezení minimální kostry grafu.

Definice 2.2.1. Mějme dán souvislý ohodnocený graf G s nezáporným ohodnocením hran c. Počet hran grafu G označíme m.

1. Seřadíme hrany grafu G do neklesající posloupnosti podle jejich ohodnocení:

$$c(e_1) \leq c(e_2) \leq \ldots \leq c(e_m).$$

2. Začneme s prázdnou množinou hran $E(G) = \emptyset$ pro kostru.

3. Pro $i = 1, 2, \ldots, m$ vezmeme hranu e_i a pokud přidáním této hrany nevznikne cyklus (v grafu s množinou hran $E(G) \cup \{e_i\}$), tak přidáme hranu e_i do $E(G)$. Jinak hranu e_i „zahodíme“.

4. Pro zpracovaní všech hran obsahuje $E(G)$ hrany minimální kostry ohodnoceného grafu G. [5]

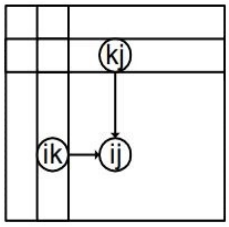
Poznámka 2.2.1. Jednotlivé kroky Kruskalova algoritmu budeme značit symbolem $\rightsquigarrow$.

Floydův–Warshallův algoritmus

Floydův–Warshallův algoritmus slouží především k vyhledání vzdáleností (vzdálenost = délka minimální cesty) v ohodnocených grafech. Algoritmus je založen na porovnání hodnot přímých a nepřímých vzdáleností. Využívá se toho, že hrana $\{v_i, v_j\}$ patří do minimální cesty tehdy, pokud nevede minimální cesta jinudy. Zapsáno matematicky:

$$c(\{v_i, v_j\}) > c(\{v_i, v_k\}) + c(\{v_k, v_j\}). \text{ [2]}$$

Ukázáno schématicky:



Obrázek 1: Schéma Floydova-Warshallova algoritmu

Definice 2.2.2. Mějme dán souvislý ohodnocený graf G s nezáporným ohodnocením hran c. Počet hran grafu G označíme m.

1. Sestavíme matici přímých vzdáleností F, přičemž pro prvky f_{ij} této matice platí:

a) $f_{ij} = 0$ pokud $i = j$,

b) $f_{ij} = c(e_i)$ pro $i = 1, 2, \ldots, m$ pokud $i \neq j$ a hrana spojující vrcholy v_i, v_j existuje,

c) $f_{ij} = \infty$ pokud $i \neq j$ a hrana spojující vrcholy v_i, v_j neexistuje.

2. Zavedeme pomocnou proměnnou k a položíme $k = 1$. Tato proměnná představuje index vrcholu, přes který provádíme přepočet.

3. Provedeme přepočet jednotlivých prvků f_{ij} matice F podle pravidla $f_{ij} = min\{f_{ij}, f_{ik} + f_{kj}\}$, přičemž nepřepočítáváme prvky matice, pro které platí $i = j$ (hlavní diagonála matice), prvky, pro které platí $i, j = k$ (leží v řádku či sloupci s indexem k), a prvky $i \neq k$ a $j \neq k$, pro které $f_{ik} = \infty$ a $f_{kj} = \infty$.

4. Pokud $k < n$ (n je počet vrcholů grafu G), potom položíme $k = k+1$ a vracíme se zpět ke kroku 3. Je-li $k = n$ je výpočet ukončen a poslední získaná matice je hledanou maticí vzdáleností. [2]

Poznámka 2.2.2. Jednotlivé kroky Floydova-Warshallova algoritmu budeme značit symbolem $\rightsquigarrow$.

Algoritmus prohledávání do šířky

Občas tento algoritmus můžeme zahlédnout pod anglickým názvem „Breadth-first search“ (zkráceně BFS). Algoritmus prohledávání do šířky je grafový algoritmus, který postupně prochází všechny vrcholy v daném maximálním souvislém podgrafu. Tento algoritmus postupuje systematickým prohledáváním grafu přes všechny vrcholy. Nepoužívá při svém prohledávání žádnou heuristickou analýzu. Pouze prochází všechny vrcholy a pro každý vrchol projde jejich všechny následovníky. Přitom si poznamenává předchůdce jednotlivých vrcholů a tím je poté vytvořen strom nejkratších cest k jednotlivým vrcholům z kořene (vrchol, ve kterém jsme začínali). [11]

3 Teorie kódování

3.1 Stručný úvod do teorie kódování

Teorie kódování se zabývá nejen konstrukcemi kódů, ale i studiem jejich vlastností. Úlohou teorie kódování je tvorba postupů a metod, které nám zajistí bezpečný přenos zpráv komunikačním systémem. Teorie kódování se dělí na minimální kódování (komprese dat), na bezpečnostní kódování (samoopravné kódy) a na kryptografické kódování (kryptografie). V tomto textu se nebudeme věnovat kompresi dat, ale zaměříme se na samoopravné kódy a kryptografii (v další kapitole). Nyní si připomeňme množinu zbytkových tříd $\mathbb{Z}_2$.

Definice 3.1.1. *Těleso* $\mathbb{Z}_2$ je dvouprvková množina $\{0, 1\}$, na které je definováno sčítání $+ : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$ a násobení $\cdot : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$ takto:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Tedy $0+0=0$, $0+1=1+0=1$, $1+1=0$, $0\cdot 0=0\cdot 1=1\cdot 0=0$, $1\cdot 1=1$. [7]

Definice 3.1.2. Nechť A je konečná množina (tzv. *abeceda*). Pak *slovo* je libovolná konečná posloupnost prvků z A. *Kódování* v obecném smyslu zahrnuje algoritmus, kterým informace převádíme do posloupnosti slov (tzv. *kodér*) a algoritmus, kterým zpětně z těchto slov získáváme původní informaci (tzv. *dekodér*). Slova, která vytváří kodér, se nazývají *kódová slova*. Množina všech kódových slov se nazývá *kód*. Je-li kód množinou slov stejné délky (každé kódové slovo má stejný počet znaků abecedy), mluvíme o tzv. *blokovém kódu*. Blokový kód *délky n* značí, že všechna kódová slova mají n znaků abecedy. [7]

Poznámka 3.1.1. *Algoritmus* je přesný postup, kterým se řeší daný typ úlohy.

Definice 3.1.3. Řekneme, že kód K je *binární*, jestliže všechna jeho slova jsou zapsána ve dvojkové soustavě (tj. všechna slova patří do $\mathbb{Z}_2^n$). [8]

Definice 3.1.4. Binární blokový kód K délky n je *lineární*, pokud kód K tvoří lineární podprostor lineárního prostoru $\mathbb{Z}_2^n$. Jestliže dimenzi tohoto podprostoru označíme k, pak mluvíme o *lineárním* (n, k) *kódu*. [7]

Definice 3.1.5. Pro vektory (slova) $\vec{v} = (v_0 v_1 \dots v_{n-1}), \vec{w} = (w_0 w_1 \dots w_{n-1})$ nad $\mathbb{Z}_2^n$ definujeme *operaci sčítaní* následovně: $\vec{v} + \vec{w} = (v_0 + w_0 \;\; v_1 + w_1 \;\; \dots \;\; v_{n-1} + w_{n-1}) \; mod \; 2$ a *operaci násobení skalárem* $c \in \mathcal{T}$ následovně: $c \cdot \vec{v} = (c \cdot v_0 \;\; c \cdot v_1 \;\; \dots \;\; c \cdot v_{n-1}) \; mod \; 2$.

Poznámka 3.1.2. Pro vektor (slovo) $\vec{v} = (v_0 v_1 \dots v_{n-1})$ nad $\mathbb{Z}_2^n$ platí následující rovnosti: $1 \cdot \vec{v} = \vec{v}, 0 \cdot \vec{v} = \vec{o}, \vec{v} \cdot \vec{v} = \vec{o}$ a $\vec{v} = -\vec{v}$.

Definice 3.1.6. *Opakovací kód* K délky n je kód složený ze všech slov tvaru $\vec{v} = (v_0 v_1 \dots v_{n-1})$, kde $v_i = 1$ nebo $v_i = 0$ pro všechna $i = 0, 1, \dots, (n-1)$. [8]

Definice 3.1.7. *Generující matice* G lineárního (n, k) kódu K je po řádcích zapsaná báze tohoto kódu. *Kontrolní matice* lineárního (n, k) kódu K je taková matice H s lineárně nezávislými řádky, pro kterou platí: množina řešení homogenní soustavy $H \cdot (\vec{x})^T = \vec{o}$ je rovna kódu K. [7]

Definice 3.1.8. Je-li počet prvků některé báze kódu K roven k, pak říkáme, že kód K má k *informačních bitů* a $(n-k)$ *kontrolních bitů*. [8]

Věta 3.1.1. Nechť G je generující matice a H kontrolní matice lineárního (n, k) kódu. Pak $H \cdot G^T = O_1$ a také $G \cdot H^T = O_2$, kde O_1 je nulová matice s $(n-k)$ řádky a k sloupci a platí $O_2 = O_1^T$. [7]

Poznámka 3.1.3. Matice G je často ve tvaru $G = (E \,|\, C)$, kde E je jednotková matice řádu n. Matice H je pak často ve tvaru $H = (C^T \,|\, E)$, kde E je jednotková matice řádu $(k-n)$. Navíc platí, že matice H je odvozena z matice G. [7]

Definice 3.1.9. Kód K nazveme *systematický*, právě když existuje generující matice G lineárního (n, k) kódu, která je ve tvaru $G = (E \,|\, C)$, kde E je jednotková matice řádu n. [7]

3.2 Cyklické kódy

Definice 3.2.1. *Cyklický* (n, k) *kód* K délky n nad $\mathbb{Z}_2^n$ je lineární (n, k) kód, který je uzavřený na cyklický posun písmen (prvků z abecedy). Pro každé $\vec{v} \in \mathbb{Z}_2^n$ platí: Je-li $\vec{v} = (v_0 v_1 \dots v_{n-1}) \in K$, pak $c(\vec{v}) = (v_1 \dots v_{n-1} v_0) \in K$. [8]

Věta 3.2.1. Opakovací kód délky n nad $\mathbb{Z}_2^n$ je cyklický. [8]

Poznámka 3.2.1. Cyklický (n, k) kód je uzavřený na cyklické posuny písmen doleva i doprava, neboť posun o i míst doleva je vlastně posunem o $(n - i)$ míst doprava, pro $0 \leq i \leq (n - 1)$. [8]

Poznámka 3.2.2. Pro popis cyklických posunů slov se ukazuje výhodné pracovat s n-ticemi znaků jako s polynomy. Proto zavádíme zobrazení

$$\vec{v} = (v_0 v_1 \dots v_{n-1}) \mapsto v(x) = v_0 + (v_1 \cdot x) + \dots + (v_{n-1} \cdot x^{n-1}).$$

Cyklický posun je pak realizován vynásobením polynomu *v(x)* s proměnnou x s tím, že $x^n = 1$ (neboť pracujeme nad $\mathbb{Z}_2^n$), přičemž $x \neq 0$. Tedy

$$x \cdot v(x) = (v_0 \cdot x) + (v_1 \cdot x^2) + \dots + (v_{n-2} \cdot x^{n-1}) + (v_{n-1} \cdot x^n)$$

$$= v_{n-1} + (v_0 \cdot x) + (v_1 \cdot x^2) + \dots + (v_{n-2} \cdot x^{n-1}). \text{ [8]}$$

Definice 3.2.2. Každý cyklický (n, k) kód K obsahující více než jedno slovo obsahuje právě jeden (až na nenulový násobek) polynom $g(x)$ stupně $(n - k)$. Polynom $g(x)$ se nazývá *generující polynom kódu* K a má následující vlastnosti:

1. Kód K se skládá právě ze všech násobků polynomu $g(x)$.
2. Polynomy $g(x), x \cdot g(x), x^2 \cdot g(x), \dots, x^{k-1} \cdot g(x)$ tvoří bázi kódu K.
3. Polynom $g(x)$ dělí polynom $x^n - 1$ beze zbytku v okruhu $\mathbb{Z}_p[x]$. [8]

Poznámka 3.2.3. Každý cyklický (n, k) kód K je jednoznačně určen svým generujícím polynomem. [8]

Poznámka 3.2.4. Generující polynom cyklického (n, k) kódu K je polynom nejmenšího stupně mezi všemi nenulovými polynomy cyklického (n, k) kódu K. [8]

Definice 3.2.3. Nechť je dán generující polynom $g(x)$ cyklického (n, k) kódu K. Pak *kódování generované polynomem* probíhá takto: máme-li dány informační bity

$(v_0 v_1 \dots v_{k-1})$. Vytvoříme polynom $v(x) = v_0 + (v_1 \cdot x) + \dots + (v_{k-1} \cdot x^{k-1})$. Pak

$$v(x) \mapsto v(x) \cdot g(x).$$

Jestliže polynom $v(x) \cdot g(x)$ je roven $u_0 + (u_1 \cdot x) + \dots + (u_{n-1} \cdot x^{n-1})$, pak kódování bude

$$(v_0 v_1 \dots v_{k-1}) \mapsto (u_0 u_1 \dots u_{n-1}).$$ [8]

3.3 Hammingův kód, rozšířený Hammingův kód

Poznámka 3.3.1. Existují lineární (n, k) kódy, které opravují některá přijatá slova lišící se pouze v jednom znaku od kódového slova (nazýváme tzv. *jednoduchá chyba*) při přenosu dat. Aby lineární (n, k) kód opravoval všechny jednoduché chyby ve slově, je nutné, aby počet sloupců byl $n = 2^c - 1$, kde $c = n - k$ (c je v tomto případě počet řádků) a dále je nutné, aby kontrolní matice tohoto kódu neměla žádný sloupec nulový a všechny sloupce byly od sebe navzájem různé. Z toho hned vyplývá jediný možný tvar kontrolní matice (až na pořadí sloupců). Pro $c = 2, 3, 4, \dots$ dostáváme tedy lineární $(3, 1), (7, 4), (15, 11), \dots$ kódy. [7]

Definice 3.3.1. Pro přirozené číslo $c \geq 2$ definujeme *Hammingův kód* délky $n = 2^c - 1$, kde $c = n - k$ jako lineární (n, k) kód, jestliže má kontrolní matici H, jejíž sloupce jsou všechna nenulová slova dané délky a žádné z nich se neopakuje. [9]

Poznámka 3.3.2. V jednotlivých sloupcích kontrolní matice H Hammingova kódu jsou všechna čísla $1, 2, 3, \dots, c$ zapsána ve dvojkové soustavě. [7]

Věta 3.3.1. Hammingův kód délky $n = 2^c - 1$ dokáže opravit jednu chybu. [9]

Věta 3.3.2. Nechť G je generující matice Hammingova $(7, 4)$ kódu

$$G = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,0 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix},$$

pak kontrolní matice H Hammingova $(7, 4)$ kódu je ve tvaru

$$H = \begin{pmatrix} 1\ 1\ 0\ 1\ 1\ 0\ 0 \\ 1\ 0\ 1\ 1\ 0\ 1\ 0 \\ 0\ 1\ 1\ 1\ 0\ 0\ 1 \end{pmatrix} \sim \begin{pmatrix} 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \end{pmatrix}. [9]$$

Poznámka 3.3.3. *Kontrola parity* je metoda, která spočívá v tom, že k přenášené informaci přidáme další bit, tzv. *paritní bit*. Hodnotu tohoto bitu určíme tak, že v nově vzniklém slově bude vždy dohromady buď sudý počet jedniček (tzv. *sudá parita*) nebo lichý počet jedniček (tzv. *lichá parita*). [9]

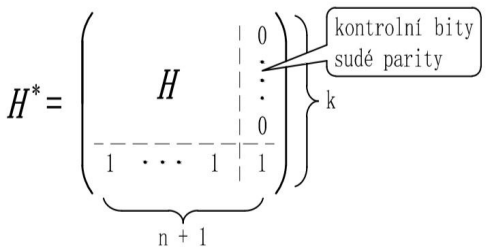
Definice 3.3.2. *Samoopravný rozšířený Hammingův kód* je lineární (n, k) kód, který vznikne rozšířením Hammingova kódu o prvek z abecedy celkové kontroly parity. Je to tedy $(2^m, 2^m - m - 1)$ kód všech slov $(v_0 v_1 \dots v_{2^m})$ takových, že $(v_0 v_1 \dots v_{2^m - 1})$ je kódové slovo Hammingova kódu a platí

$$v_{2^m} = (v_0 + v_1 + \dots + v_{2^m - 1})\ mod\ 2. [9]$$

Poznámka 3.3.4. Princip rozšíření Hammingova kódu spočívá v přidání prvku z abecedy celkové kontroly parity ke každému řádku kontrolní matice H a dále doplnění řádku jedniček. Tzn., že z lineárního (n, k) kódu K vytvoříme $(n+1, k)$ kód všech slov $(v_0 v_1 \dots v_{n-1} v_n)$ takových, že platí

$$(v_0 v_1 \dots v_{n-1}) \in K \text{ a } v_n = (v_0 + v_1 + \dots + v_{n-1})\ mod\ 2. [9]$$

Nově vzniklá kontrolní matice H^* rozšířeného Hammingova kódu má tvar:



Obrázek 2: Rozšíření kontrolní matice H Hammingova kódu

Definice 3.3.3. Nechť H je kontrolní matice lineárního (n, k) kódu. *Syndrom* slova w je vektor $\vec{s}$, pro který platí $\vec{s}^T = H \cdot w^T$. [7]

Definice 3.3.4. Nechť v je slovo vyslané kodérem a w je slovo přijaté dekodérem. Pak $e = w - v$ je *chybové slovo*. [7]

Poznámka 3.3.5. Slovo chybové i přijaté má stejný syndrom. Platí tedy $H \cdot w^T = H \cdot e^T$. Je-li $H \cdot w^T = \vec{o}$, pak při přenosu dat nedošlo k žádné chybě. Je-li $H \cdot w^T \neq \vec{o}$, pak při přenosu dat došlo alespoň k jedné chybě. Došlo-li k jediné chybě na pozici j, tak syndrom chybového slova je roven j-tému sloupci kontrolní matice H. Tuto chybu opravíme tak, že změníme hodnotu na pozici j na opačnou. Jestliže tedy na pozici j byla 1, pak ji nastavíme na 0 a opačně. Došlo-li k více než jedné chybě při přenosu dat, pak chybové slovo není možné opravit (resp. zpětně zkonstruovat). [9]

4 Kryptografie

4.1 Stručný úvod do kryptografie

Kryptografie je vědní disciplína, která se zabývá metodami ochrany dat před neautorizovaným přístupem. Cílem kryptografie je poslat někomu zprávu tak, aby zprávě kromě odesílatele už rozuměl pouze její příjemce a nikdo jiný. Samotné slovo *kryptografie* je odvozeno ze dvou slov - *kryptos* = skrytý, *graphein* = text. Teď je hned jasné, o čem kryptografie je. Nyní tedy formálně.

Definice 4.1.1. *Otevřenou abecedou* rozumíme konečnou množinu A znaků, které používáme k zápisu nešifrovaných zpráv. *Otevřeným textem* rozumíme zprávu určenou k zašifrovaní, tj. konečný řetězec $c = c_1 \dots c_n$, kde $c_i \in A$ (n je délka řetězce). *Prostorem otevřených textů* nazýváme množinu všech otevřených textů a značíme C. [10]

Definice 4.1.2. *Šifrovací abecedou* rozumíme konečnou množinu B znaků, které používáme k zápisu zašifrovaných zpráv. V případě, kdy $B = \{0, 1\}$, pak mluvíme o *binárním šifrování*. *Šifrovaným textem* rozumíme konečný řetězec $d = d_1 \dots d_n$ znaků šifrovací abecedy, který vznikl zašifrováním některého otevřeného textu $c \in C$. *Prostorem šifrovaných textů* nazýváme množinu všech šifrovaných textů a značíme D. [10]

Definice 4.1.3. *Klíčem* rozumíme uspořádanou dvojici $k = (r, s)$, kde r je *šifrovací klíč* (parametr šifrovací metody) a s je *dešifrovací klíč* (parametr dešifrovací metody). Množina všech klíčů se nazývá *prostor klíčů* a značí se Q. Navíc platí, že $(r, s) \in Q$. [10]

Definice 4.1.4. *Konkatenací vektorů* rozumíme zřetězení neboli spojování dvou či více vektorů do jednoho. Konkatenaci vektorů budeme značit symbolem $\circledast$.

Definice 4.1.5. *Substituční tabulka* je tabulka, která ke každému znaku dané abecedy přiřadí číselnou hodnotu (resp. jiný znak).

Poznámka 4.1.1. Znaky v substituční tabulce jsou většinou umístěny v prvním řádku. Číselné hodnoty pak v dalších řádcích. Běžná substituční tabulka má znaky $A - Z$, které odpovídají hodnotám $1 - 26$. V tomto textu budeme pracovat s vlastními substitučními tabulkami.

Poznámka 4.1.2. Šifrovacím klíčem může být obecně cokoliv, co vytvoří odesílatel (slovo, čísla, tabulka, ...). V tomto textu se omezíme na šifrovací klíče, které budou mít charakter konkatenace vektorů a substitučních tabulek, a dále na šifrovací klíče, které budou reprezentovány šifrovací maticí R. Dešifrovací matice pak bude ve tvaru R^{-1}. U šifrovacích (resp. dešifrovacích) matic se ještě navíc omezíme na čtvercové matice.

Definice 4.1.6. *Šifrováním* rozumíme proces transformace otevřeného textu do zašifrovaného textu (neboli „nesrozumitelného“ textu). *Dešifrovaní* je inverzní proces k šifrovaní, tedy jde o proces převedení zašifrovaného textu do podoby otevřeného textu. [10]

Definice 4.1.7. *Šifrovací transformací* (*funkcí*) rozumíme vzájemně jednoznačné zobrazení $R_r : C \to D$ definované pro všechny šifrovací klíče z prostoru klíčů Q. *Dešifrovací transformací* (*funkcí*) rozumíme zobrazení $S_s : D \to C$, které je inverzní k zobrazení $R_r : C \to D$, kde $(r, s) \in Q$. [10]

Poznámka 4.1.3. Vzájemná jednoznačnost zobrazení R_r je nutnou podmínkou pro možnost zpětného dešifrovaní. [10]

Definice 4.1.8. Uspořádaná trojice $(\mathcal{R}, \mathcal{S}, Q)$, kde
$Q = \{(r, s)\}$ je prostor klíčů,
$\mathcal{R} = \{R_r \,|\, (r, s) \in Q\}$ je množina šifrovacích transformací,
$\mathcal{S} = \{S_s \,|\, (r, s) \in Q\}$ je množina dešifrovacích transformací,
tvoří *šifrovací systém*, jestliže

$$\forall k = (r, s) \in Q \, \forall c \in C : S_s(R_r(c)) = c,$$

tedy každý klíč (r, s) jednoznačně definuje dvojici transformací R_r a S_s (šifrovací a jí příslušnou dešifrovací), které jsou navzájem inverzní. [10]

Poznámka 4.1.4. Šifrování se obvykle dělí na dva typy. Prvním typem je tzv. *asymetrické šifrování*. Metody asymetrického šifrování jsou takové šifrovací metody, kde dešifrovací klíč je výpočetně složité odvodit ze šifrovacího klíče. Asymetrické šifrování není předmětem této práce. Druhým typem je tzv. *symetrické šifrování*. Metody symetrického šifrování jsou šifrovací metody, kde dešifrovací klíč je výpočetně snadné odvodit ze šifrovacího klíče. V tomto textu si ukážeme některé aplikace symetrického šifrování. [10]

4.2 Symetrické šifry

Transpoziční šifra

Definice 4.2.1. *Transpoziční šifra* je bloková šifra délky n, tj. šifra, která nejprve rozdělí otevřený text na bloky délky n po sobě jdoucích znaků a poté každý blok zašifruje jako celek. [10]

Poznámka 4.2.1. Pokud délka otevřeného textu není násobkem čísla n, můžeme doplnit text libovolnými znaky na délku rovnou prvnímu násobku čísla n většímu než d, kde d je délka celého řetězce. V tomto textu bude jeden blok reprezentován jedním vektorem. [10]

Princip šifrování: Šifrovací klíč π náleží množině permutací přirozených čísel neboli $\pi \in S_n$, kde $n \in \mathbb{N}$. Šifrování poté probíhá tak, že nejprve rozdělíme otevřený text na bloky délky n po sobě jdoucích znaků, tj. $c = c^{(1)} \dots c^{(k)}$, kde $c^{(i)} = c_1^{(i)} \dots c_n^{(i)}$ je i-tý blok. Následně každý blok $c^{(i)}$ zašifrujeme pomocí transformace:

$$R_\pi(c_1^{(i)} \dots c_n^{(i)}) = c_{\pi(1)}^{(i)} \dots c_{\pi(n)}^{(i)}\,,\ i = 1, 2, \dots, k. \text{ [10]}$$

Princip dešifrování: Dešifrovací klíč $\pi^{-1} \in S_n$, kde π^{-1} označuje inverzní permutaci k π. Dešifrování poté probíhá tak, že nejprve zašifrovaný text rozdělíme na bloky délky n po sobě jdoucích znaků, tj. $d = d^{(1)} \dots d^{(k)}$, kde $d^{(i)} = d_1^{(i)} \dots d_n^{(i)}$ je i-tý blok. Následně každý blok $d^{(i)}$ dešifrujeme pomocí transformace:

$$S_{\pi^{-1}}(d_1^{(i)} \dots d_n^{(i)}) = c_{\pi^{-1}(1)}^{(i)} \dots c_{\pi^{-1}(n)}^{(i)}\,,\ i = 1, 2, \dots, k. \text{ [10]}$$

Afinní šifra

Definice 4.2.2. *Afinní šifra* je šifra, která při šifrování nejprve převede otevřený text $c = c_1 \dots c_n$ na číselný řetězec $c = x_1 \dots x_n$ a následně tento řetězec $c = x_1 \dots x_n$ zašifruje. [10]

Poznámka 4.2.2. Šifrování se provede například tak, že každý znak nahradíme jeho pořadím v rámci uvažované otevřené abecedy - substituční tabulky. Důležitým faktorem pro afinní šifru je šifrovací klíč, který je ve tvaru (a, b), kde $a, b \in \mathbb{Z}_p$, přičemž $NSD(a, p) = 1$ (p v tomto případě nemusí být prvočíslo). [10]

Princip šifrování: Šifrovací klíč je tvaru (a, b), kde $a, b \in \mathbb{Z}_p$ a $NSD(a, p) = 1$. Šifrování je poté výsledkem aplikace šifrovací funkce

$$R_{(a,b)}(x_1 \dots x_n) = d_1 \dots d_n,$$

kde x_i je číselná reprezentace i-tého znaku otevřeného textu a

$$d_i = ((a \cdot x_i + b) \, mod \, p)$$

je číselná reprezentace i-tého znaku šifrovaného textu. [10]

Princip dešifrování: Dešifrovací klíč je ve tvaru (a^{-1}, b), kde a^{-1} je inverzní prvek k $a \; mod \; p$. Dešifrování je pak výsledkem aplikace dešifrovací funkce

$$S_{(a^{-1},b)}(d_1 \dots d_n) = x_1 \dots x_n, \text{ kde } x_i = (a^{-1} \cdot (d_i - b) \, mod \, p). \text{ [10]}$$

Hillova šifra

Definice 4.2.3. *Hillova šifra* je substituční bloková šifra délky n, která je navíc polygrafická, tj. nahrazuje m-tice znaků za jiné m-tice. [10]

Poznámka 4.2.3. *Substituční šifra* je taková šifra, ve které znaky otevřeného textu mění svou identitu, ale nemění svou pozici. Hillova šifra pracuje s maticemi. Pokud délka otevřeného textu není násobkem čísla n, můžeme doplnit text libovolnými znaky na délku rovnou nejbližšímu většímu násobku čísla n. [10]

Princip šifrování: Šifrovací klíč je ve tvaru matice $R = (r_{ij})$ řádu n, kde $r_{ij} \in \mathbb{Z}_p$ (p v tomto případě nemusí být prvočíslo). Šifrování poté probíhá tak, že nejprve rozdělíme otevřený text c na bloky délky n po sobě jdoucích znaků, tj.

$c = c^{(1)} \dots c^{(k)}$, kde $c^{(i)} = c_1^{(i)} \dots c_n^{(i)}$. Následně každý blok $c^{(i)}, i = 1, 2, \dots, k$ převedeme na číselný řetězec $x^{(i)} = (x_1^{(i)}, \dots, x_n^{(i)})$, který zašifrujeme pomocí transformace:

$$R_R(y^{(i)}) = x^{(i)} \cdot R \,(mod\, p),$$

kde $y^{(i)} = (y_1^{(i)}, \dots, y_n^{(i)})$ je číselný vektor reprezentující i-tý blok zašifrovaného textu $d = (y^{(1)}, \dots, y^{(k)})$. [10]

Princip dešifrování: Dešifrovací klíč je ve tvaru R^{-1}, tj. inverzní matice R modulo p. Dešifrování probíhá zcela analogicky jako šifrování. Tedy zašifrovaný text rozdělíme na číselné bloky (vektory) $y^{(i)}$, $i = 1, 2, \dots k$, délky n, které dešifrujeme pomocí inverzní transformace:

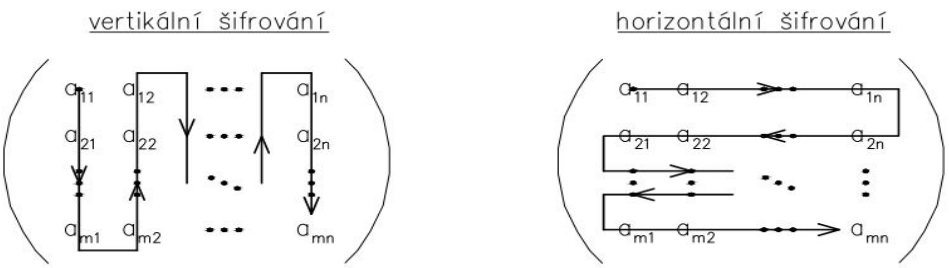
$$S_{R^{-1}}(x^{(i)}) = y^{(i)} \cdot R^{-1} \,(mod\, p). \,[10]$$

Poznámka 4.2.4. Existence inverzní matice R^{-1} je nezbytnou podmínkou pro jednoznačné dešifrování. Lze dokázat, že nutnou a postačující podmínkou je $NSD(detR, p) = 1$. Navíc platí vztah $R \cdot R^{-1} \equiv E \,(mod\, p)$. [10]

Věta 4.2.1. Transpoziční šifra, afinní šifra a Hillova šifra patří mezi symetrické šifry.

Poznámka 4.2.5. Existují i další symetrické šifry, například tzv. *jednoduchá substituce*, *Vigenèrova šifra*, *binární bloková šifra* a další. V tomto textu si vystačíme pouze s výše definovanými šiframi a jejich kombinacemi.

Poznámka 4.2.6. Otevřený nebo šifrovaný text v šifrovacích nebo dešifrovacích maticích budeme v tomto textu číst vertikálně nebo horizontálně. Čtení textu vertikálně nazýváme *vertikální šifrovaní*. Čtení textu horizontálně nazýváme *horizontální šifrovaní*.



Obrázek 3: Vertikální a horizontální šifrování

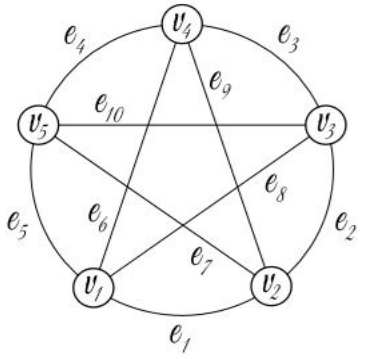
Řešené příklady

1 Příklady z teorie grafů

Příklad 1

Zadání: Dva kamarádi Martin a Petr se vsadili o čokoládu, že Martin nedokáže spočítat počet koster v neorientovaném grafu připomínající pentagram. Petr Martinovi vysvětlil, co to kostra grafu je, přičemž obratem na to Martin začal jednu po druhé počítat. Po chvíli počítaní se Martin zastavil a přemýšlel, zda danou kostru již nezapočítal. Pomozte Martinovi spočítat počet koster.

Řešení: V první řadě vytvoříme neorientovaný graf, který bude připomínat pentagram a označíme vrcholy a hrany.



Obrázek 4: Pentagram

Nyní sestavíme Laplaceovu matici sousednosti $L(G)$ tohoto neorientovaného grafu, která bude mít tvar:

$$L(G) = \begin{pmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}.$$

Počet koster budeme počítat jako determinant Laplaceovy matice sousednosti $L(G)$ bez posledního řádku a posledního sloupce, tedy

$$det \begin{pmatrix} 4 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & 4 \end{pmatrix} = 125.$$

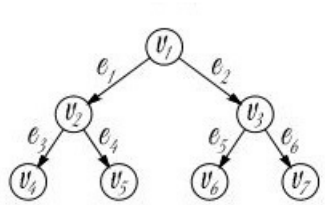
Zjistili jsme, že neorientovaný graf připomínající pentagram má přesně 125 koster a tím pádem jsme pomohli Martinovi vyhrát vsázku.

Příklad 2

Zadání: Určete hodnost matice incidence $I(G)$ orientovaného grafu $G = (V,E)$ s $|V(G)| = n$ vrcholy a $|E(G)| = m$ hranami.

Nápověda: Začněte se stromem, pak uvažujte souvislý graf a nakonec zobecněte na jednoduchý graf.

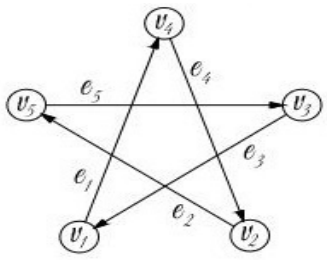
Řešení: V první řadě vytvoříme libovolný strom a rovnou k němu určíme matici incidence $I(G_1)$ a vypočteme hodnost $h(I(G_1))$ této matice.



$$I(G_1) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \; h(I(G_1)) = 6.$$

Obrázek 5: Strom s maticí incidence a hodností této matice

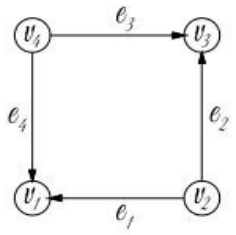
Dále vytvoříme libovolný souvislý graf. Opět k němu rovnou určíme matici incidence $I(G_2)$ a vypočteme hodnost $h(I(G_2))$ této matice.



$$I(G_2) = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{pmatrix}, \; h(I(G_2)) = 4.$$

Obrázek 6: Souvislý graf s maticí incidence a hodností této matice

V poslední řadě vytvoříme libovolný jednoduchý graf a postupujeme analogicky jako v předchozích případech.



$$I(G_3) = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \; h(I(G_3)) = 3.$$

Obrázek 7: Jednoduchý graf s maticí incidence a hodností této matice

Docházíme k závěru, že hodnost matice incidence $I(G)$ orientovaného grafu je $h(I(G)) = n - 1$.

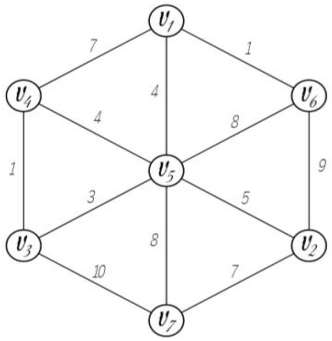
Poznámka: Kdybychom v tomto textu uvažovali smyčky v grafech, pak bychom hodnost matice incidence $I(G)$ mohli zobecnit i na obecný graf. Tento graf v tomto textu nebyl definován, avšak výsledek by byl stejný.

Příklad 3

Zadání: Mladý inženýr se rozhodl, že založí jednu nejmenovanou internetovou společnost. Jako správný začínající podnikatel si v první řadě našel potenciální zájemce a následně si zmapoval cesty k jednotlivým zájemcům. Všechny tyto hodnoty zaznamenal do matice sousednosti *S(G)*. Nalezněte nejkratší možnou trasu ke všem potenciálním zájemcům, tak aby je všechny propojila a určete nejvhodnější místo pro založení společnosti.

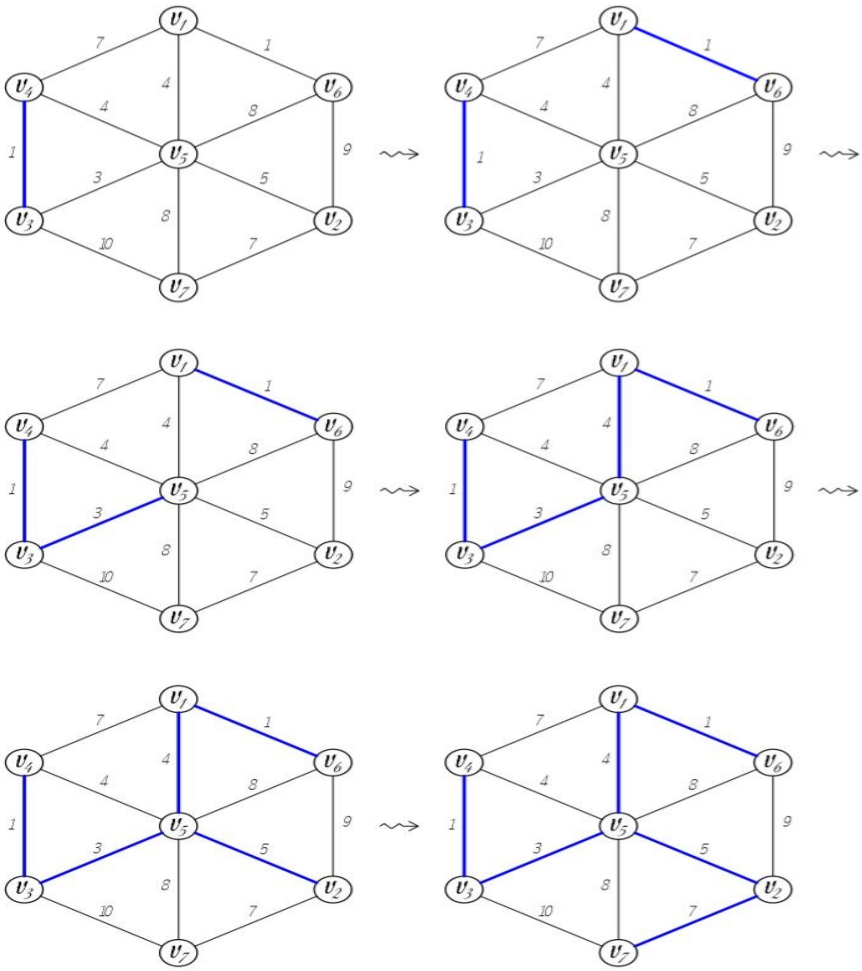
$$S(G) = \begin{pmatrix} 0 & 0 & 0 & 7 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 5 & 9 & 7 \\ 0 & 0 & 0 & 1 & 3 & 0 & 10 \\ 7 & 0 & 1 & 0 & 4 & 0 & 0 \\ 4 & 5 & 3 & 4 & 0 & 8 & 8 \\ 1 & 9 & 0 & 0 & 8 & 0 & 0 \\ 0 & 7 & 10 & 0 & 8 & 0 & 0 \end{pmatrix}$$

Řešení: V matici sousednosti *S(G)* hodnoty na místě s_{ij} určují, zda dva dané vrcholy v_i a v_j jsou spojeny hranou či nikoli. Pokud je graf ohodnocený, pak každý prvek s_{ij} je roven ohodnocení hrany e_i. Dále si můžeme všimnout, že tato matice sousednosti je symetrická, takže se bude jednat o ohodnocený neorientovaný graf, který může mít např. následující tvar:



Obrázek 8: Ohodnocený neorientovaný graf č.1

Nyní budeme hledat minimální kostru grafu. K nalezení této kostry grafu využijeme Kruskalova algoritmu a rovnou uděláme první krok algoritmu, tj.



Obrázek 9: Kruskalův algoritmus

Po dokončení Kruskalova algoritmu vidíme, že nejvhodnějším místem pro založení nejmenované internetové společnosti je místo (vrchol) s označením v_5.

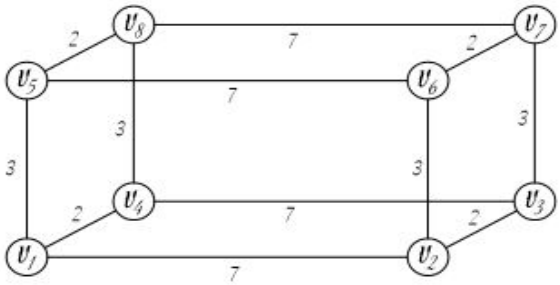
Příklad 4

Zadání: Nakreslete souvislý graf zadaný maticí sousednosti *S(G)*. Pokud při konstrukci tohoto souvislého grafu vznikne nějaké trojrozměrné těleso, vypočítejte jeho objem a povrch.

$$S(G) = \begin{pmatrix} 0 & 7 & 0 & 2 & 3 & 0 & 0 & 0 \\ 7 & 0 & 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 2 & 0 & 7 & 0 & 0 & 3 & 0 \\ 2 & 0 & 7 & 0 & 0 & 0 & 0 & 3 \\ 3 & 0 & 0 & 0 & 0 & 7 & 0 & 2 \\ 0 & 3 & 0 & 0 & 7 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 0 & 2 & 0 & 7 \\ 0 & 0 & 0 & 3 & 2 & 0 & 7 & 0 \end{pmatrix}$$

Nápověda: Jistě se bude jednat o trojrozměrné těleso. Při sestrojování tohoto trojrozměrného tělesa dbejte na to, aby hrany se stejným ohodnocením byly stejně dlouhé a měly stejné směry.

Řešení: Analogicky jako v předchozím příkladě vytvoříme neorientovaný graf, neboť matice *S(G)* je symetrická. V matici sousednosti *S(G)* hodnoty na místě s_{ij} odpovídají ohodnocení hrany vedoucí z vrcholu v_i do vrcholu v_j. Dále je potřeba si uvědomit, že se bude jednat o nějaké trojrozměrné těleso, které má 8 vrcholů. Důležitým faktorem při konstruování tohoto grafu je, aby hrany se stejným ohodnocením byly stejně dlouhé a měly stejné směry (viz nápověda). Z tohoto důvodu bude neorientovaný graf vypadat následovně:

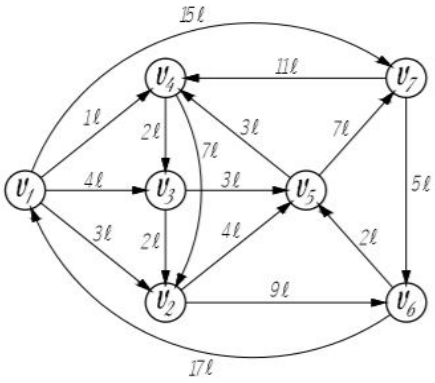


Obrázek 10: Ohodnocený neorientovaný graf č.2

Vidíme, že naše hledané trojrozměrné těleso odpovídá kvádru, jehož strany mají 2, 3 a 7 jednotek. Objem kvádru je tedy $V = a \cdot b \cdot c = 2 \cdot 3 \cdot 7 = 42$ jednotek krychlových a povrch kvádru je $S = 2 \cdot (a \cdot b + a \cdot c + b \cdot c) = 2 \cdot (2 \cdot 3 + 2 \cdot 7 + 3 \cdot 7) = 82$ jednotek čtverečných.

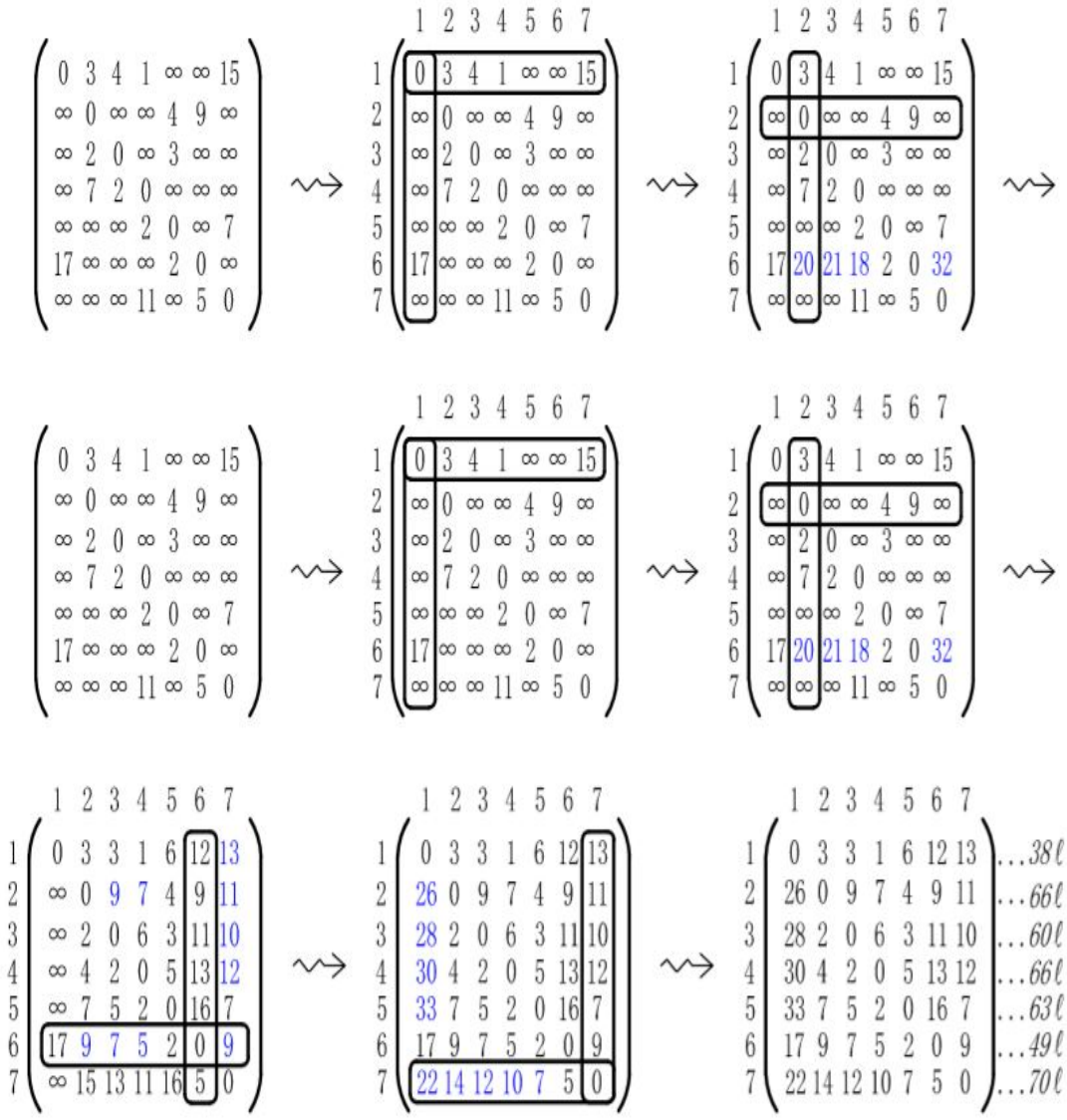
Příklad 5

Zadání: Poštovní kurýr každý pátek pravidelně rozváží reklamní letáky do sedmi navzájem různých rodin. Žádné dvě rodiny nežijí ve stejném městě. Poštovnímu kurýrovi se poslední dobou zdá, že prodělává peníze na benzínu za cestu. Z tohoto důvodu si nakreslil mapu všech měst, ve kterých rodiny žijí. Dále na mapě vyznačil cesty, kterými kdy projel a připsal ke každé z nich počet litrů benzínu, které za cestu spotřeboval. Pomozte poštovnímu kurýrovi nalézt nejlevnější cestu a určete, kolik tato cesta bude stát Kč. Pro výpočet předpokládejte, že cena za jeden litr benzínu je 35 Kč.



Obrázek 11: Mapa všech měst

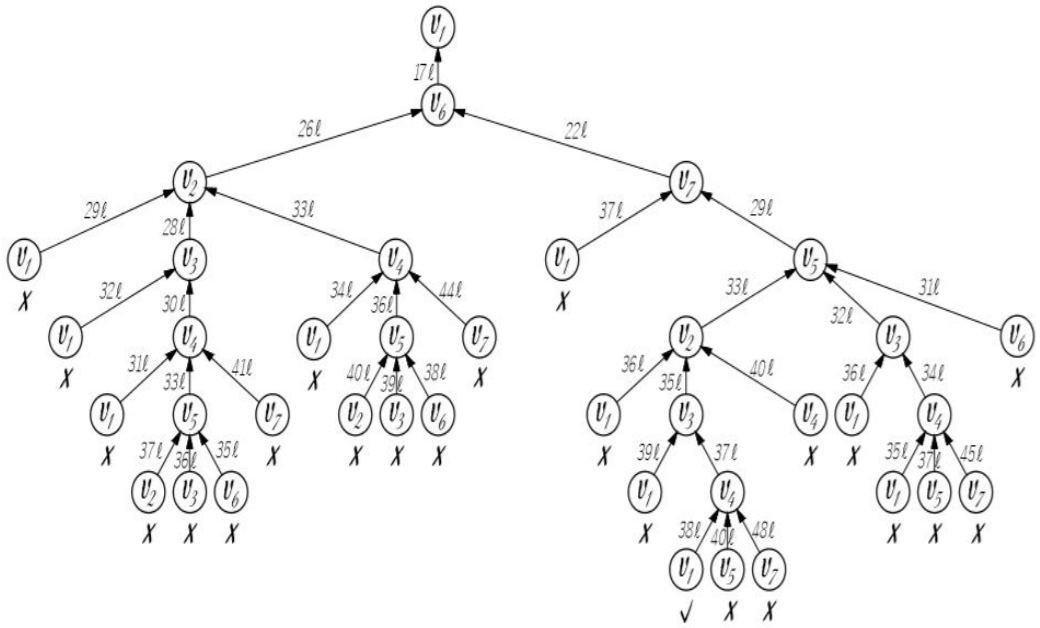
Řešení: Jedním z možných způsobů nalezení této cesty je ji náhodně zvolit a doufat, že je to nejkratší, a tudíž nejlevnější možná cesta. Tento způsob není jistě vůbec efektivní, a proto tuto úlohu vyřešíme jinak. K řešení tohoto příkladu nám pomůže Floydův–Warshallův algoritmus. Díky tomuto algoritmu zjistíme limit nejkratší možné cesty od vrcholu v_i po vrchol v_j. V první řadě sestrojíme matici a následně budeme přepočítávat hodnoty v této matici dle Floydova-Warshallova algoritmu.



Obrázek 12: Floydův–Warshallův algoritmus

Vidíme, že limit vyšel 38 litrů. Tento limit je nejmenší možný v tomto případě. Nejlevnější cesta je tedy za $38 \cdot 35 = 1330$ Kč. Dále si můžeme z výsledné matice všimnout, že tato cesta začíná ve vrcholu s označením v_1. Nyní budeme vytvářet strom všech možných cest a hledat tu nejmenší možnou (tedy do limitu *38l*). Začneme u vrcholu v_1 (pozn.: nezáleží na tom, kterým vrcholem začneme, protože musíme objet všechny vrcholy a vrátit se do vrcholu původního). Strom budeme konstruovat „odzadu“ a budeme se držet koncepce algoritmu prohledávání do šířky. Navíc při konstruování tohoto stromu budeme sčítat hodnoty hran, kterými

se lze dostat do vrcholu v_i z vrcholu v_1 a budeme ověřovat, zda jsme nevytvořili cyklus, anebo jsme nepřekročili limit. Pro ilustraci: Chceme se dostat do vrcholu v_2 „odzadu“. Vytvoříme sled mezi vrcholy $v_1 - v_6 - v_2$. Ohodnocení hran sečteme a dostaneme $17 + 9 = 26l$. Takhle budeme pokračovat dál, až nakonec vytvoříme následující strom:



Obrázek 13: Strom vytvořený prohledáváním do šířky „odzadu“

Vidíme, že díky tomuto stromu jsme tedy nalezli nejkratší okružní cestu s limitem $38l$, kterou tvoří sled vrcholů $v_1 - v_4 - v_3 - v_2 - v_5 - v_7 - v_6 - v_1$.

Poznámka: Obecně neplatí, že Floydův-Warshallův algoritmus určí nejkratší okružní cestu. To, že nám to v tomto případě vyšlo stejně, je čistě náhodné! Tento příklad byl ilustrací tzv. problému nalezení Hamiltonovského cyklu s limitem K. Tento problém patří do třídy NPTIME. Předchozí algoritmus má exponenciální časovou složitost, a proto není v praxi použitelný. Všechny pojmy uvedené v tomto příkladu, jako např. „Hamiltonovský cyklus“ atd., jsou podrobně popsány v odborné literatuře. Doporučují např. učební text Teoretická informatika, který je dostupný na tomto odkazu: http://phoenix.inf.upol.cz/~jancarp/VAS/jancar-ti-vsb.pdf.

2 Příklady z teorie kódování

Příklad 1

Zadání: Ověřte, zda množina kódových slov tvoří lineární (n, k) kód a nalezněte bázi a dimenzi tohoto kódu, pokud existuje.

a) $K = \{(11111),(11110),(11101),(11100),(00011),(00010),(00001),(00000)\}$

b) $K = \{(11111111),(10101010),(11011011),(01010101),(01110001),$
$(10001110),(10001110),(01110001)\}$

Řešení a:

$K = \{(11111),(11110),(11101),(11100),(00011),(00010),(00001),(00000)\}$

- z teorie víme, že kód je lineární, pokud tvoří lineární podprostor lineárního prostoru $\mathbb{Z}_2^n$. Množina kódových slov musí tedy být uzavřená na operaci sčítaní vektorů a násobení vektorů skalárem v $\mathbb{Z}_2^n$. Ověříme.

Sčítání:

(11111) + (11110) = (00001) ✓ (11110) + (11101) = (00011) ✓

(11111) + (11101) = (00010) ✓ (11110) + (11100) = (00010) ✓

(11111) + (11100) = (00011) ✓ (11110) + (00011) = (00001) ✓

(11111) + (00011) = (11100) ✓ (11110) + (00010) = (00000) ✓

⋮ ⋮

Až bychom nakonec ověřili všechny možné kombinace a zjistili, že tato množina K je uzavřená na operaci sčítaní vektorů.

Násobení:

$1 \cdot$ „kódové slovo z K“ = „kódové slovo z K“ ✓

$0 \cdot$ „kódové slovo z K“ = (00000) ✓

Vidíme, že množina K tvoří lineární (n, k) kód. Nyní můžeme nalézt bázi a následně určit dimenzi tohoto lineárního (n, k) kódu.

Báze:

$$\begin{pmatrix} 1&1&1&1&1\\ 1&1&1&1&0\\ 1&1&1&0&1\\ 1&1&1&0&0\\ 0&0&0&1&1\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 0&0&0&0&0 \end{pmatrix} \sim \begin{pmatrix} 1&1&1&1&1\\ 1&1&1&1&0\\ 1&1&1&0&1\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{pmatrix} \sim \begin{pmatrix} 1&1&1&0&1\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{pmatrix}$$

Báze lineárního kódu je například množina {(11101),(00010),(00001)}. Dimenze tohoto kódu je 3. Jedná se tedy o lineární $(5, 3)$ kód.

Poznámka: Skutečně se jedná o bázi tohoto lineárního (n, k) kódu. Každá lineární kombinace těchto tří kódových slov (vektorů) nám dá některé kódové slovo (vektor), které leží v množině K. Ověřte.

Řešení b:

$K = \{(11111111),(10101010),(11011011),(01010101),(01110001),(10001110),$
$(10001110),(01110001)\}$

- analogicky jako v případě *a)* ověříme uzavřenost na operaci sčítaní vektorů a násobení vektorů skalárem v $\mathbb{Z}_2^n$.

Sčítaní:

(11111111) + (10101010) = (01010101) ✓

(11111111) + (11011011) = (00100100) ✓

(11111111) + (01110001) = (10001110) ✓

(10101010) + (11011011) = (01110001) ✓

$\vdots$

A tak pokračujeme dál, až bychom opět prověřili všechny možnosti. Pojďme se nyní podívat na uzavřenost násobení.

Násobení:

1 · „kódové slovo z K“ = „kódové slovo z K“ ✓

0 · „kódové slovo z K“ = (00000000) ✗

Vidíme, že kódové slovo (00000000) nenáleží množině K, a tudíž se nejedná o lineární (n, k) kód. Báze a dimenze tohoto kódu neexistují. Ověřte.

Poznámka: Na začátku tohoto příkladu v části *b)* jsme si mohli všimnout, že v množině K není nulový vektor $\vec{o}$, resp. kódové slovo (00000000) a tím pádem automaticky tato množina K nemůže být lineárním (n, k) kódem.

Příklad 2

Zadání: Nalezněte některou kontrolní matici H, když víte, že generující matice G lineárního (n, k) kódu nad $\mathbb{Z}_5$ má tvar:

$$G = \begin{pmatrix} -4 & 5 & 1 & 3 & -1 & 2 & -2 & -2 & 3 \\ 1 & 1 & 3 & 1 & 1 & 3 & 3 & 2 & 2 \\ -4 & 5 & -3 & -1 & 2 & 3 & -4 & -1 & 2 \\ 1 & 5 & -2 & 1 & -4 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Řešení: V první řadě se budeme snažit o to, abychom tuto generující matici G dostali do tvaru $(E \,|\, C)$, kde $(E \,|\, C)$ je matice vzniklá z G pomocí elementárních řádkových transformací modulo 5.

$$\begin{pmatrix} -4 & 5 & 1 & 3 & -1 & 2 & -2 & -2 & 3 \\ 1 & 1 & 3 & 1 & 1 & 3 & 3 & 2 & 2 \\ -4 & 5 & -3 & -1 & 2 & 3 & -4 & -1 & 2 \\ 1 & 5 & -2 & 1 & -4 & 1 & 0 & 1 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 1 & 3 & 4 & 2 & 3 & 3 & 3 \\ 1 & 1 & 3 & 1 & 1 & 3 & 3 & 2 & 2 \\ 1 & 0 & 2 & 4 & 2 & 3 & 1 & 4 & 2 \\ 1 & 0 & 3 & 1 & 1 & 4 & 0 & 1 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 1 & 3 & 4 & 2 & 3 & 3 & 3 \\ 0 & 1 & 2 & 3 & 2 & 1 & 0 & 4 & 4 \\ 0 & 0 & 1 & 1 & 3 & 1 & 3 & 1 & 4 \\ 0 & 0 & 2 & 3 & 2 & 2 & 2 & 3 & 2 \end{pmatrix} \sim$$

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

Nyní již není složité nalézt některou kontrolní matici H, tedy

$$H = \begin{pmatrix} -4 & -2 & -2 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -3 & 0 & -2 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & -2 & 0 & -4 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 3 & 3 & 4 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 3 & 4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 1 & 0 \\ 4 & 3 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Poznámka: Tento příklad lze řešit i Gaussovou eliminační metodou.

Příklad 3

Zadání: Nalezněte některou generující matici G, když víte, že kontrolní matice H lineárního (n, k) kódu nad $\mathbb{Z}_{11}$ má tvar:

$$H = \begin{pmatrix} 2 & 1 & -6 & 10 & -7 & 5 \\ 8 & 2 & -4 & 2 & -3 & 6 \end{pmatrix}$$

Řešení: Budeme postupovat téměř analogicky jako v předchozím příkladě. Jediná změna bude ta, že kontrolní matici H se budeme snažit upravit do tvaru $(C^T \,|\, E)$, kde $(C^T \,|\, E)$ je matice vzniklá z H pomocí elementárních řádkových transformací modulo 11.

$$\begin{pmatrix} 2 & 1 & -6 & 10 & -7 & 5 \\ 8 & 2 & -4 & 2 & -3 & 6 \end{pmatrix} \sim \begin{pmatrix} 2 & 1 & 4 & 10 & 4 & 5 \\ 8 & 2 & 7 & 2 & 8 & 6 \end{pmatrix} \sim \begin{pmatrix} 2 & 1 & 5 & 10 & 4 & 5 \\ 4 & 0 & 8 & 4 & 0 & -4 \end{pmatrix} \sim \begin{pmatrix} 2 & 1 & 5 & 10 & 4 & 5 \\ -1 & 0 & -2 & -1 & 0 & 1 \end{pmatrix} \sim$$

$$\sim \begin{pmatrix} 7 & 1 & 4 & 4 & 4 & 0 \\ -1 & 0 & -2 & -1 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 10 & 3 & 1 & 1 & 1 & 0 \\ -1 & 0 & -2 & -1 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 10 & 3 & 1 & 1 & 1 & 0 \\ 10 & 0 & 9 & 10 & 0 & 1 \end{pmatrix}$$

Některá generující matice G má tedy tvar:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & -10 & -10 \\ 0 & 1 & 0 & 0 & -3 & 0 \\ 0 & 0 & 1 & 0 & -1 & -9 \\ 0 & 0 & 0 & 1 & -1 & -10 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 8 & 0 \\ 0 & 0 & 1 & 0 & 10 & 2 \\ 0 & 0 & 0 & 1 & 10 & 1 \end{pmatrix}$$

Poznámka: V tomto i předchozím příkladě skutečně platí vztah $G \cdot H^T = O$. Ověřte.

Příklad 4

Zadání: Určete, zda daný kód je cyklický.

a) $K = \{(111111111)\}$ *d)* $K = \{(111011001)\}$

b) $K = \{(100010101)\}$ *e)* $K = \{(000000000)\}$

c) $K = \{(100100100)\}$ *f)* $K = \{(111000000)\}$

Řešení:

a) Ano. Tento kód je cyklický. Cyklickým posunem doprava či doleva dostáváme tentýž kód. Jedná se o opakovací kód délky 9.

b) Ne. Uvažujme například generující matici G, která má tvar:

$$G = \begin{pmatrix} 1\,0\,0\,0\,1\,0\,1\,0\,1 \\ 1\,1\,0\,0\,0\,1\,0\,1\,0 \\ 0\,1\,1\,0\,0\,0\,1\,0\,1 \end{pmatrix}.$$

Vidíme, že žádnou elementární řádkovou transformací se nedostaneme do tvaru $(E \,|\, C)$, tj. kód není cyklický.

c) Ano. Generující matice G tohoto kódového slova je ve tvaru:

$$G = \begin{pmatrix} 1\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,1 \end{pmatrix}.$$

Jedná se o systematický kód $K = \{(abcabcabc)\,;\, a, b, c \in \mathbb{Z}_2\}$.

d) Ne. Uvažujme generující matici G ve tvaru:

$$G = \begin{pmatrix} 1\,1\,1\,0\,1\,1\,0\,0\,1 \\ 1\,1\,1\,1\,0\,1\,1\,0\,0 \\ 0\,1\,1\,1\,1\,0\,1\,1\,0 \end{pmatrix} \sim \begin{pmatrix} 1\,0\,0\,1\,0\,1\,1\,1\,1 \\ 1\,0\,0\,0\,1\,1\,0\,1\,0 \\ 0\,1\,1\,1\,1\,0\,1\,1\,0 \end{pmatrix}.$$

Stejně jako v případě *b)* se nedostaneme do námi požadovaného tvaru $(E \,|\, C)$.

e) Analogicky jako v případě *a).*

f) Ne. Generující matice G tohoto kódového slova je ve tvaru:

$$G = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1 \end{pmatrix}.$$

Tento kód se nazývá tzv. *koktavý kód* $K = \{(aaabbbccc)\,;\, a, b, c \in \mathbb{Z}_2\}$ délky n. V tomto případě kódové slovo nabývá délky 9. Permutací znaků v tomto kódovém slově lze vytvořit systematický kód, viz. příklad *c).*

Poznámka: Koktavé kódy jsou obecně lineární, ale postrádají systematické kódování, a proto nejsou cyklické.

Příklad 5

Zadání: Ověřte, že binární kód kontroly parity délky 3, $K = \{(000),(011),(101),(110)\}$, generovaný polynomem $g(x) = x + 1$ je cyklický a navíc určuje toto kódování:

$$\vec{v} = (00)$$
$$\vec{v} = (01)$$
$$\vec{v} = (10)$$
$$\vec{v} = (11)$$

Řešení: V první řadě určíme bázi tohoto kódu, tj.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Kód K je ekvivalentní množině polynomů $K = \{0, x+1, x^2+1, x^2+x\}$. Zadané kódování můžeme rovněž přepsat jako polynomy, tj.

$$\vec{v} = (00) \mapsto 0$$
$$\vec{v} = (01) \mapsto 1$$
$$\vec{v} = (10) \mapsto x$$
$$\vec{v} = (11) \mapsto x + 1$$

Nyní hledáme polynom $u(x) = v(x) \cdot g(x)$. Hodnoty polynomu $u(x)$ by měly být ekvivalentní hodnotám kódu K (v případě, že se jedná o cyklický kód generovaný polynomem $g(x)$). Ověříme.

$$u(x) = 0 \cdot (x+1) = 0 \Leftrightarrow (000)$$
$$u(x) = 1 \cdot (x+1) = x + 1 \Leftrightarrow (011)$$
$$u(x) = x \cdot (x+1) = x^2 + x \Leftrightarrow (110)$$
$$u(x) = (x+1) \cdot (x+1) = x^2 + 2x + 1 \equiv x^2 + 1 \Leftrightarrow (101)$$

Což je očekávaný výsledek vzhledem k tomu, jak nám vyšla báze tohoto kódu. Jedná se tedy o cyklický kód generovaný polynomem $g(x) = x + 1$.

Příklad 6

Zadání: Uvažujme Hammingův (7,4) kód a přijatá slova

a) (1010101)

b) (1101110)

c) (0010100)

Jaké slovo je chybné? Pokud při přenosu došlo k jedné chybě, pokuste se tuto chybu najít a zkonstruovat původní slovo.

Řešení: Kontrolní matice Hammingova (7,4) kódu má tvar:

$$H = \begin{pmatrix} 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \end{pmatrix}.$$

Hledáme syndrom kódového slova w, tj. vektor $\vec{s} = H \cdot w^T$.

a)

$$\begin{pmatrix} 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Vidíme, že nám syndrom vyšel nulový, tj., při přenosu nedošlo k žádné chybě.

b)

$$\begin{pmatrix} 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Syndrom slova vyšel nenulový, takže při přenosu došlo k chybě. Syndrom $\vec{s}$ je ekvivalentní čtvrtému sloupci kontrolní matice Hammingova (7,4) kódu, z čehož vyplývá, že je potřeba opravit čtvrtý bit přijatého slova na kódové slovo $w =$ (1100110).

c)

$$\begin{pmatrix} 0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Analogicky jako v předchozím případě nám vyšel nenulový syndrom, takže opět došlo při přenosu k chybě. Syndrom odpovídá šestému sloupci kontrolní matice Hammingova (7,4) kódu, tj. opravíme šestý bit. Původní kódové slovo bylo ve tvaru $w = (0010110)$.

Příklad 7

Zadání: Uvažujme samoopravný rozšířený Hammingův (8,4) kód a přijatá kódová slova

a) (10110110)
b) (10011001)
c) (11100111)

Které ze slov bylo chybně přijato? Pokud je možné, chybně přijatá slova opravte a napište původní kódová slova.

Řešení: Kontrolní matici samoopravného rozšířeného Hammingova (8,4) kódu vytvoříme z kontrolní matice Hammingova (7,4) kódu přidáním jednotkového řádku a nulového sloupce, tj.

$$H^* = \begin{pmatrix} 0&0&0&1&1&1&1&0 \\ 0&1&1&0&0&1&1&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&1&1&1&1&1&1 \end{pmatrix}.$$

Analogicky jako v předchozím příkladě hledáme syndrom kódového slova w, tj. vektor $\vec{s} = H^* \cdot w^T$.

a)

$$\begin{pmatrix} 0&0&0&1&1&1&1&0 \\ 0&1&1&0&0&1&1&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&1&1&1&1&1&1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Při přenosu došlo k jedné chybě. Je potřeba opravit sedmý bit kódového slova. Původní kódové slovo bylo $w = (10110100)$.

b)

$$\begin{pmatrix} 0&0&0&1&1&1&1&0 \\ 0&1&1&0&0&1&1&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&1&1&1&1&1&1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Přijaté slovo odpovídá odeslanému slovu. Při přenosu tedy nedošlo k žádné chybě.

c)

$$\begin{pmatrix} 0&0&0&1&1&1&1&0 \\ 0&1&1&0&0&1&1&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&1&1&1&1&1&1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Při přenosu došlo k více než 1 chybě, a proto původní kódové slovo nelze zkonstruovat.

3 Příklady z kryptografie

Poznámka: U příkladů kombinace afinní a Hillovy šifry budeme pracovat nad množinou $\mathbb{Z}_n$. U těchto příkladů budeme využívat operaci modulo n. Tuto operaci použijeme pokud budeme hledat dešifrovací matici ve tvaru R^{-1}, anebo výsledná čísla v maticích překročí velikost substituční tabulky.

Příklad 1

Zadání: Převeďte do binární reprezentace a zašifrujte následující text:

„Co jsem vás opustil, mám oči v duši jen,
a skutečný můj zrak, který mé kroky řídí,
už přestal pracovat a zpola oslepen,
po pravdě nevidí, i když se zdá, že vidí.“

(úryvek se Shakespearovy knihy Sonety, verš 113), kde každý symbol z otevřené abecedy zašifrujte přesně do 7 bitů.

Řešení: V první řadě zvolíme vhodné kódování, např. klasické ASCII (může být klidně i UTF-8, Windows-1250, atd.). Nyní převedeme všechny symboly od dolních uvozovek po horní uvozovky do symboliky čísel pomocí námi zvolené ASCII tabulky. (ASCII tabulka neuvažuje diakritiku, takže ani my ji zde nebudeme uvažovat). Dostáváme tedy tvar:

„67 111 32 106 115 101 109 32 118 97 115 32 111 112 117 115 116 105 108 44
32 109 97 109 32 111 99 11 32 118 32 100 117 115 105 32 106 101 110 44 97
32 115 107 117 116 101 99 110 121 32 109 117 106 44 32 107 116 101 114 121
32 109 101 32 107 114 111 107 121 32 114 105 100 105 44 117 122 32 112 114
101 115 116 97 108 32 112 114 97 99 111 118 97 116 32 97 32 122 112 111 108

97 32 111 115 108 101 112 101 110 44 112 111 32 112 114 97 118 100 101 32 110 101 118 105 100 105 44 32 105 32 107 100 121 122 32 115 101 32 122 100 97 44 32 122 101 32 118 105 100 105 46“

Zde máme přesně 153 čísel, které bychom mohli vyjádřit jako 51 vektorů, kde každý vektor má právě 3 souřadnice (nebo také např. 17 vektorů o 9 souřadnicích, atd.). Takže bychom měli např. vektor $\overrightarrow{u_1} = (67, 111, 32)$, $\overrightarrow{u_2} = (106, 115, 101)$, atd. Každou složku každého vektoru převedeme do dvojkové soustavy. Tj. dostaneme $\overrightarrow{u_1} = (1000011, 1101111, 0100000)$, $\overrightarrow{u_2} = (1101010, 1110011, 1100101)$, atd. Když chceme vektory zašifrovat, musíme si vytvořit nějaký šifrovací klíč. Z tohoto důvodu zvolíme šifrovací klíč např. takový:

$$r = u_1 \circledast u_2 \circledast u_3 \circledast \ldots \circledast u_{51}$$

Výsledný zašifrovaný text má následující tvar:

„1000011,1101111,0100000,1101010,1110011,1100101,1101101,0100000,1110110, 1100001,1110011,0100000,1101111,1110000,1110101,1110011,1110100,1101001, 1101100,0101100,0100000,1101101,1100001,1101101,0100000,1101111,1100011, 0001011,0100000,1110110,0100000,1100100,1110101,1110011,1101001,0100000, 1101010,1100101,1101110,0101100,1100001,0100000,1110011,1101011,1110101, 1110100,1100101,1100011,1101110,1111001,0100000,1101101,1110101,1101010, 0101100,0100000,1101011,1110100,1100101,1110010,1111001,0100000,1101101, 1100101,0100000,1101011,1110010,1101111,1101011,1111001,0100000,1110010, 1101001,1100100,1101001,0101100,1110101,1111010,0100000,1110000,1110010, 1100101,1110011,1110100,1100001,1101100,0100000,1110000,1110010,1100001, 1100011,1101111,1110110,1100001,1110100,0100000,1100001,0100000,1111010, 1110000,1101111,1101100,1100001,0100000,1101111,1110011,1101100,1100101, 1110000,1100101,1101110,0101100,1110000,1101111,0100000,1110000,1110010, 1100001,1110110,1100100,1100101,0100000,1101110,1100101,1110110,1101001, 1100100,1101001,0101100,0100000,1101001,0100000,1101011,1100100,1111001, 1111010,0100000,1110011,1100101,0100000,1111010,1100100,1100001,0101100, 0100000,1111010,1100101,0100000,1110110,1101001,1100100,1101001,0101110“

Poznámka: Jedná se o transpoziční šifru. Pokud by bylo potřeba zašifrovanou zprávu dešifrovat, museli bychom znát šifrovací klíč a zvolené kódování (v tomto případě ASCII). Dešifrování by poté probíhalo tak, že bychom zašifrovaný text převedli do dekadické soustavy, následně vytvořili 51 vektorů o 3 souřadnicích a poté každou souřadnici každého vektoru převedli do symboliky znaků pomocí ASCII tabulky. Tato ASCII tabulka je obsažena v příloze na konci tohoto textu.

Příklad 2

Zadání: Zašifrujte zprávu:

„Jsme jako kočka a myš“

pomocí Morseovy abecedy, kde šifrovací klíč R je ve tvaru:

$$R = \begin{pmatrix} 1\ 2\ 0\ 1\ 2\ 1\ 0 \\ 2\ 1\ 1\ 0\ 1\ 2\ 1 \\ 3\ 2\ 1\ 1\ 3\ 2\ 1 \\ 0\ 1\ 0\ 1\ 2\ 3\ 0 \\ 1\ 1\ 2\ 1\ 2\ 1\ 1 \\ 1\ 0\ 2\ 1\ 0\ 1\ 2 \\ 3\ 1\ 0\ 1\ 2\ 3\ 1 \end{pmatrix}.$$

Pro výpočet volte následující substituci:

$$0 = .$$
$$1 = -$$
$$2 = |$$

Řešení: Text přeložený do Morseovy abecedy je ve tvaru:

„. − − − |...| − −|.||. − − − |. − | − . − | − − − || − . − | − − − | − . − .| − . − |. − ||. − || − −| − . − −|...|“.

Text po použití substituce je ve tvaru:

„0 1 1 1 2 0 0 0 2 1 1 2 0 2 2 0 1 1 1 2 0 1 2 1 0 1 2 1 1 1 2 2 1 0 1 2 1 1 1 2 1 0 1 0 2 1 0 1 2 0 1 2 2 0 1 2 2 1 1 2 1 0 1 1 2 0 0 0 2“.

Šifrovací matice R je řádu 7. Abychom mohli text zašifrovat, musíme nalézt matici, která v sobě bude obsahovat všech 69 čísel, a navíc musí hledaná matice splňovat kritérium pro násobení, tj. hledáme matici typu $7 \times m$. Nejbližší hledané m, které obsahuje všechna čísla je $m = 10$, takže dostáváme matici 7×10 (70. číslo v matici bude reprezentovat prázdný symbol „|“, který je substituován za hodnotu 2, a tudíž nepřijdeme o žádnou informaci). Nyní již můžeme zprávu zašifrovat. Čísla do hledané matice zapíšeme tak, jak jdou za sebou a vynásobíme s maticí R.

$$\begin{pmatrix} 1&2&0&1&2&1&0 \\ 2&1&1&0&1&2&1 \\ 3&2&1&1&3&2&1 \\ 0&1&0&1&2&3&0 \\ 1&1&2&1&2&1&1 \\ 1&0&2&1&0&1&2 \\ 3&1&0&1&2&3&1 \end{pmatrix} \cdot \begin{pmatrix} 0&1&1&1&2&0&0&0&2&1 \\ 1&2&0&2&2&0&1&1&1&2 \\ 0&1&2&1&0&1&2&1&1&1 \\ 2&2&1&0&1&2&1&1&1&2 \\ 1&0&1&0&2&1&0&1&2&0 \\ 1&2&2&0&1&2&2&1&1&2 \\ 1&0&1&1&2&0&0&0&2&2 \end{pmatrix} = \begin{pmatrix} 7&9&6&5&12&6&5&6&10&9 \\ 5&9&10&6&12&6&7&5&12&11 \\ 10&14&14&9&21&10&9&9&20&16 \\ 8&10&9&2&10&10&8&7&9&10 \\ 7&9&11&6&12&8&8&7&13&11 \\ 5&7&10&5&8&6&7&4&10&11 \\ 9&13&13&6&18&10&8&7&17&15 \end{pmatrix}$$

Pokud bychom nyní očíslovali otevřenou abecedu např. od indexu 1, tj. $1 = A$, $2 = B$, $3 = C$ atd. a následně čísla ve výsledné matici tímto způsobem substituovali, dostali bychom matici ve tvaru:

$$\begin{pmatrix} G&I&F&E&L&F&E&F&J&I \\ E&I&J&F&L&F&G&E&L&K \\ J&N&N&I&U&J&I&I&T&P \\ H&J&I&B&J&J&H&G&I&J \\ G&I&K&F&L&H&H&G&M&K \\ E&G&J&E&H&F&G&D&J&K \\ I&M&M&F&R&J&H&G&Q&O \end{pmatrix}.$$

Odkud tedy šifrovaná zpráva může mít např. tvar:

„GIFELFEFJIEIJFLFGELKJNNIUJIITP
HJIBJJHGIJGIKFLHHGMKEGJEHFGDJKIMMFRJHGQO“.

Poznámka: Jedná se o afinní šifru se šifrováním pomocí matice a substituční tabulky. Dešifrování by probíhalo tak, že zašifrovaný text by byl vložen do nalezené matice typu 7×10 dle schéma horizontálního šifrování. Tuto matici označme

např. X. Následně by znaky v matici X byly substituovány za čísla (pomocí substituční tabulky). Dále by bylo potřeba nalézt dešifrovací matici, která by byla ve tvaru R^{-1} (předpokladem je, že známe šifrovací matici R). Tyto matice bychom poté spolu vynásobili, tj. $R^{-1} \cdot X$, čímž bychom nalezli matici Y. Hodnoty z této matice Y by bylo potom potřeba vypsat na řádek (opět dle schéma horizontálního šifrování) a následně tento šifrovaný text substituovat dle tabulky Morseovy abecedy. Tato tabulka Morseovy abecedy je obsažena v příloze na konci tohoto textu.

Příklad 3

Zadání: Zašifrujte text:

„Podstata matematiky spočívá v její svobodě“ (Georg Cantor).

K šifrování použijte klíč R, který je ve tvaru:

$$R = \begin{pmatrix} 3 & 2 & 7 \\ 5 & 1 & 4 \\ 2 & 3 & 1 \end{pmatrix}$$

a substituční tabulka má tvar:

A	Á	B	C	Č	D	E	Ě	F	G	H	I	Í	J	K	L
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	_
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Řešení: Text, který budeme šifrovat, bude ve tvaru:

„PODSTATA_MATEMATIKY_SPOČÍVÁ_V_JEJÍ_SVOBODĚ“,

neboť tabulka nezavádí malá písmena. Nyní můžeme písmena substituovat za čísla, tj.

„19 18 5 22 23 0 23 0 30 16 0 23 6 16 0 23 11 14 28 30 22 19 18 4 12 25 1 30 25 30 13 6 13 12 30 22 25 18 2 18 5 7“.

Odtud dostáváme 12 matic typu 3×1 (resp. 12 transponovaných vektorů, které mají právě 3 souřadnice). Nyní tyto matice vynásobíme šifrovacím klíčem R.

$$\begin{pmatrix}3&2&7\\5&1&4\\2&3&1\end{pmatrix}\cdot\begin{pmatrix}19\\18\\5\end{pmatrix}=\begin{pmatrix}128\\133\\97\end{pmatrix}\equiv_{30}\begin{pmatrix}8\\13\\7\end{pmatrix}=\begin{pmatrix}F\\J\\\check{E}\end{pmatrix}$$

$$\begin{pmatrix}3&2&7\\5&1&4\\2&3&1\end{pmatrix}\cdot\begin{pmatrix}22\\23\\0\end{pmatrix}=\begin{pmatrix}112\\133\\113\end{pmatrix}\equiv_{30}\begin{pmatrix}22\\13\\23\end{pmatrix}=\begin{pmatrix}S\\J\\T\end{pmatrix}$$

$$\begin{pmatrix}3&2&7\\5&1&4\\2&3&1\end{pmatrix}\cdot\begin{pmatrix}23\\0\\30\end{pmatrix}=\begin{pmatrix}279\\235\\76\end{pmatrix}\equiv_{30}\begin{pmatrix}9\\25\\16\end{pmatrix}=\begin{pmatrix}G\\V\\M\end{pmatrix}$$

$$\vdots$$

Až bychom dostali zašifrovanou zprávu:

„FJĚSJTGV M ZSVQMAGBCY O OÁGECZH Q V L SCXH Y MDÁHTCY“.

Poznámka: Jedná se o kombinaci afinní a Hillovy šifry. Dešifrování by v tomto případě probíhalo tak, že zašifrovaný text by bylo potřeba rozdělit na 12 transponovaných vektorů o 3 souřadnicích. Tyto vektory bychom následně převedli na čísla (pomocí substituční tabulky). V dalším kroku by bylo nutné nalézt dešifrovací matici R^{-1}. Následně by pak každý vektor byl násoben s touto maticí R^{-1}, tj. $R^{-1}\cdot(\vec{x})^T$, čímž bychom dostali 12 matic typu 3×1, které by bylo potřeba pomocí substituční tabulky převést do symboliky znaků.

Příklad 4

Zadání: Mějme zašifrovaný text *„XKXTDXBTFGSN“*. Dále uvažujme dešifrovaný text *„KRYPTOGRAFIE“* a následující substituční tabulku:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
11	18	4	22	6	3	24	12	2	9	10	1	8	7	13	25	14	5	19	0	20	17	15	16	23	21

Nalezněte šifrovací klíč a následně určete dešifrovací klíč.

Řešení: V první řadě substituujeme znaky šifrovaného i dešifrovaného textu pomocí substituční tabulky za čísla.

$$\text{„}XKXTDXBTFGSN\text{“} = \text{„}16\ 10\ 16\ 0\ 22\ 16\ 18\ 0\ 3\ 24\ 19\ 7\text{“}$$
$$\text{„}KRYPTOGRAFIE\text{“} = \text{„}11\ 5\ 23\ 25\ 0\ 13\ 24\ 5\ 11\ 3\ 2\ 6\text{“}$$

Obě slova mají 12 znaků, takže mají po substituci 12 čísel. Šifrovací matice bude jistě typu 2×2, neboť $12 = 2 \cdot 2 \cdot 3$ (kdyby šifrovací matice byla typu 3×3, zbyla by nám 3 čísla, a kdyby matice byla 4×4, tak by nám 4 čísla chyběla). Nyní si musíme uvědomit, že šifrovací matice je vždy „nalevo“ a také to, že čísla můžeme zapsat jako transponované vektory, anebo jako čtvercové matice. Tímhle způsobem bychom tedy dostali 6 transponovaných vektorů o 2 souřadnicích (jiné být nemohou), anebo 3 čtvercové matice řádu 2. Pokud bychom uvažovali čtvercové matice, museli bychom čísla zapisovat do matice po řádcích. Nyní je už jedno, co zvolíme. Takže např. vezmeme čtvercové matice řádu 2, tím tedy dostaneme

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} \cdot \begin{pmatrix} 11 & 23 \\ 5 & 25 \end{pmatrix} = \begin{pmatrix} 16 & 16 \\ 10 & 0 \end{pmatrix},$$

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} \cdot \begin{pmatrix} 0 & 24 \\ 13 & 5 \end{pmatrix} = \begin{pmatrix} 22 & 18 \\ 16 & 0 \end{pmatrix},$$

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} \cdot \begin{pmatrix} 11 & 2 \\ 3 & 6 \end{pmatrix} = \begin{pmatrix} 3 & 19 \\ 24 & 7 \end{pmatrix}.$$

Nyní využijeme definice součinu matic a díky tomu získáme 3 soustavy lineárních rovnic o 4 neznámých. Nám stačí vzít pouze jednu soustavu (zbylé 2 musí dát stejný výsledek), tj.

$$11a + 5c = 16,$$
$$23a + 25c = 16,$$
$$11b + 5d = 10,$$
$$23b + 25d = 0.$$

Tuto soustavu lineárních rovnic vyřešíme, čímž získáme řešení, které bude ve tvaru $\{a, b, c, d\} = \{2; \frac{25}{16}; -\frac{6}{5}; -\frac{23}{16}\}$. Tím jsme získali původní šifrovací matici a nyní se budeme zabývat hledáním dešifrovací matice. Tu získáme tak, že k této šifrovací matici vypočítáme matici inverzní.

$$\left(\begin{array}{cc|cc} 2 & -\frac{6}{5} & 1 & 0 \\ \frac{25}{16} & -\frac{23}{16} & 0 & 1 \end{array}\right) \sim \cdots \sim \left(\begin{array}{cc|cc} 1 & 0 & \frac{23}{16} & -\frac{6}{5} \\ 0 & 1 & \frac{25}{16} & -2 \end{array}\right).$$

Příklad 5

Zadání: Mějme zašifrovaný text:

„ÍOC.KQITIŽ P GOŘŽ Í GŽNHMŽWFDPJIÍ Č ÍŽU_ D“.

Šifrovací matice R je ve tvaru:

$$R = \begin{pmatrix} 2 & 1 & 3 & 5 & 1 \\ 1 & 2 & 5 & 1 & 3 \\ 4 & 7 & 0 & 2 & 4 \\ 5 & 8 & 1 & 6 & 2 \\ 1 & 0 & 3 & 5 & 0 \end{pmatrix}$$

a substituční tabulka má následující charakter:

A	B	C	Č	D	E	F	G	H	I	Í	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

O	P	Q	Ř	S	T	U	V	W	X	Y	Ž	_	.	:
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Rozšifruje tuto zprávu.

Řešení: Dešifrovací matici šifrované zprávy budeme hledat ve tvaru R^{-1}.

$$\left(\begin{array}{ccccc|ccccc} 2 & 1 & 3 & 5 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 5 & 1 & 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 7 & 0 & 2 & 4 & 0 & 0 & 1 & 0 & 0 \\ 5 & 8 & 1 & 6 & 2 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \sim \left(\begin{array}{ccccc|ccccc} 1 & 0 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & 2 & 26 & 3 & 0 & 1 & 0 & 0 & 29 \\ 0 & 7 & 18 & 12 & 4 & 0 & 0 & 1 & 0 & 26 \\ 0 & 8 & 16 & 11 & 2 & 0 & 0 & 0 & 1 & 25 \\ 0 & 1 & 27 & 25 & 1 & 1 & 0 & 0 & 0 & 28 \end{array}\right) \sim$$

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 27 & 25 & 1 & 1 & 0 & 0 & 0 & 28 \\ 0 & 0 & 9 & 17 & 27 & 23 & 0 & 1 & 0 & 10 \\ 0 & 0 & 10 & 21 & 24 & 22 & 0 & 0 & 1 & 11 \\ 0 & 0 & 8 & 6 & 1 & 1 & 28 & 0 & 0 & 3 \end{array}\right) \sim \left(\begin{array}{ccccc|ccccc} 1 & 0 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 27 & 25 & 1 & 1 & 0 & 0 & 0 & 28 \\ 0 & 0 & 1 & 11 & 26 & 25 & 29 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 & 4 & 12 & 10 & 20 & 1 & 1 \\ 0 & 0 & 0 & 8 & 3 & 8 & 9 & 22 & 0 & 7 \end{array}\right) \sim$$

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 27 & 25 & 1 & 1 & 0 & 0 & 0 & 28 \\ 0 & 0 & 1 & 11 & 26 & 25 & 29 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 & 4 & 12 & 10 & 20 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 19 & 12 & 22 & 29 \end{array}\right) \sim \left(\begin{array}{ccccc|ccccc} 1 & 0 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 27 & 25 & 0 & 29 & 11 & 18 & 8 & 29 \\ 0 & 0 & 1 & 11 & 0 & 3 & 15 & 19 & 28 & 3 \\ 0 & 0 & 0 & 1 & 0 & 4 & 24 & 2 & 3 & 5 \\ 0 & 0 & 0 & 0 & 1 & 2 & 19 & 12 & 22 & 29 \end{array}\right) \sim$$

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 3 & 0 & 0 & 10 & 0 & 20 & 15 & 6 \\ 0 & 1 & 27 & 0 & 0 & 19 & 11 & 28 & 23 & 24 \\ 0 & 0 & 1 & 0 & 0 & 19 & 21 & 27 & 25 & 8 \\ 0 & 0 & 0 & 1 & 0 & 4 & 24 & 2 & 3 & 5 \\ 0 & 0 & 0 & 0 & 1 & 2 & 19 & 12 & 22 & 29 \end{array}\right) \sim \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 13 & 27 & 29 & 0 & 12 \\ 0 & 1 & 0 & 0 & 0 & 16 & 14 & 19 & 8 & 18 \\ 0 & 0 & 1 & 0 & 0 & 19 & 21 & 27 & 25 & 8 \\ 0 & 0 & 0 & 1 & 0 & 4 & 24 & 2 & 3 & 5 \\ 0 & 0 & 0 & 0 & 1 & 2 & 19 & 12 & 22 & 29 \end{array}\right)$$

Dešifrovací klíč je tedy ve tvaru:

$$R^{-1} = \begin{pmatrix} 13 & 27 & 29 & 0 & 12 \\ 16 & 14 & 19 & 8 & 18 \\ 19 & 21 & 27 & 25 & 8 \\ 4 & 24 & 2 & 3 & 5 \\ 2 & 19 & 12 & 22 & 29 \end{pmatrix}.$$

Zašifrovaný text substituujeme za čísla dle substituční tabulky, čímž získáme šifrovaný text ve tvaru:

„10 16 2 29 12 18 9 21 9 27 17 7 16 19 27 10 7 27 15 8 14 27 24 6 4 17 11 9 10 3 10 27 22 28 4“.

Tato čísla můžeme zapsat jako 7 transponovaných vektorů o 5 souřadnicích, tedy

$$\begin{pmatrix} 10 \\ 16 \\ 2 \\ 29 \\ 12 \end{pmatrix}, \begin{pmatrix} 18 \\ 9 \\ 21 \\ 9 \\ 27 \end{pmatrix}, \begin{pmatrix} 17 \\ 7 \\ 16 \\ 19 \\ 27 \end{pmatrix}, \begin{pmatrix} 10 \\ 7 \\ 27 \\ 15 \\ 8 \end{pmatrix}, \begin{pmatrix} 14 \\ 27 \\ 24 \\ 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 17 \\ 11 \\ 9 \\ 10 \\ 3 \end{pmatrix}, \begin{pmatrix} 10 \\ 27 \\ 22 \\ 28 \\ 4 \end{pmatrix}.$$

Nyní každý tento vektor vynásobíme s dešifrovací maticí R^{-1}, tj.

$$\begin{pmatrix} 13 & 27 & 29 & 0 & 12 \\ 16 & 14 & 19 & 8 & 18 \\ 19 & 21 & 27 & 25 & 8 \\ 4 & 24 & 2 & 3 & 5 \\ 2 & 19 & 12 & 22 & 29 \end{pmatrix} \cdot \begin{pmatrix} 10 \\ 16 \\ 2 \\ 29 \\ 12 \end{pmatrix} = \begin{pmatrix} 764 \\ 870 \\ 1401 \\ 575 \\ 1334 \end{pmatrix} \equiv_{30} \begin{pmatrix} 14 \\ 0 \\ 21 \\ 5 \\ 14 \end{pmatrix} = \begin{pmatrix} M \\ A \\ T \\ E \\ M \end{pmatrix}$$

$$\vdots$$

Až nakonec dostaneme dešifrovaný text:

„MATEMATIKA_UČÍ:_NEPŘEHLÍŽEJTE_NULY.“ (Gabriel Laub).

Poznámka: Jedná se o kombinaci afinní a Hillovy šifry. Šifrování zprávy bylo názorně předvedeno v příkladu 3.

Příklad 6

Zadání: Profesionální zvědi už nějakou dobu mapují terén ve válečné zóně. Jejich cílem je prolomit zašifrované zprávy nepřítele. Jeden zvěd zjistil, že nepřátelé k šifrování používají jako klíč slovo *„TRANSPORT“* a substituční tabulku, která u počátečního symbolu A má index *3* a u koncového *prázdného* symbolu má index *29*. Dále zvěd zjistil, že indexy jsou uspořádány vzestupně a že tabulka má 26 znaků. Při posledním pátráním se zvědům podařilo najít lístek s tímto vzkazem:

„984 307 1305 855 602 874 573 862 353
1072 405 1595 912 820 1125 490 1115 432
1148 425 1711 959 871 1225 520 1249 456“

Obrázek 14: Získaný lístek se vzkazem

Pokuste se dešifrovat zprávu pomocí zjištěných informací od profesionálních zvědů.

Řešení: Jistě budeme pracovat s tabulkou, kde počáteční symbol je písmeno A s indexem *3* a tabulka končí prázdným symbolem s indexem *29* a navíc indexy jsou uspořádány vzestupně, tj.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

P	Q	R	S	T	U	V	W	X	Y	Z	
18	19	20	21	22	23	24	25	26	27	28	29

Šifrovací klíč po substituci bude ve tvaru *„22 20 3 16 21 18 17 20 22“*, z čehož můžeme vytvořit šifrovací matici R, kde

$$R = \begin{pmatrix} 22 & 20 & 3 \\ 16 & 21 & 18 \\ 17 & 20 & 22 \end{pmatrix}.$$

Z této šifrovací matice R získáme dešifrovací matici ve tvaru R^{-1}, tedy

$$R^{-1} = \frac{1}{1213} \cdot \begin{pmatrix} 102 & -380 & 297 \\ -46 & 433 & -348 \\ -37 & -100 & 142 \end{pmatrix}.$$

Šifrovaný text vložíme do matice tak, jak je zapsaný a vynásobíme jej s dešifrovací maticí R^{-1}, tj.

$$\frac{1}{1213} \cdot \begin{pmatrix} 102 & -380 & 297 \\ -46 & 433 & -348 \\ -37 & -100 & 142 \end{pmatrix} \cdot \begin{pmatrix} 984 & 307 & 1305 & 855 & 602 & 874 & 573 & 862 & 353 \\ 1072 & 405 & 1595 & 912 & 820 & 1125 & 490 & 1115 & 432 \\ 1148 & 425 & 1711 & 959 & 871 & 1225 & 520 & 1249 & 456 \end{pmatrix}$$

$$= \begin{pmatrix} 28 & 3 & 29 & 21 & 7 & 21 & 22 & 29 & 6 \\ 16 & 11 & 29 & 18 & 20 & 17 & 4 & 7 & 10 \\ 16 & 7 & 29 & 11 & 16 & 24 & 3 & 28 & 7 \end{pmatrix}$$

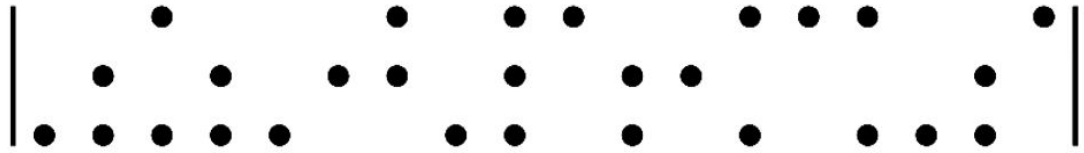
Čísla ve výsledné matici můžeme dle substituční tabulky substituovat zpět a dostaneme dešifrovaný text:

„ZA SEST DNI PROBEHNE INVAZE“,

čímž jsme prolomili šifru a zjistili, že profesionální zvědi měli správné informace.
Poznámka: Jedná se o afinní šifru se šifrováním pomocí matice a substituční tabulky. Šifrovací matice R je „schována“ ve slově „TRANSPORT“ (díky čemuž dokážeme vytvořit šifrovací matici řádu 3). Pokud bychom chtěli zprávy šifrovat, bylo by potřeba převést otevřený text na matici $3 \times m$ (popř. na sloupcové vektory o 3 souřadnicích). Šifrovací matici R bychom následně s touto maticí $3 \times m$ (popř. s těmito sloupcovými vektory) vynásobili, čímž bychom dostali zašifrovanou zprávu v číselné podobě. Pokud by bylo potřeba zprávu reprezentovat znaky, mohli bychom využít substituční tabulky. Důležitým faktorem je, aby příjemce znal klíčové slovo, díky kterému lze vytvořit dešifrovací matici R^{-1}.

Příklad 7

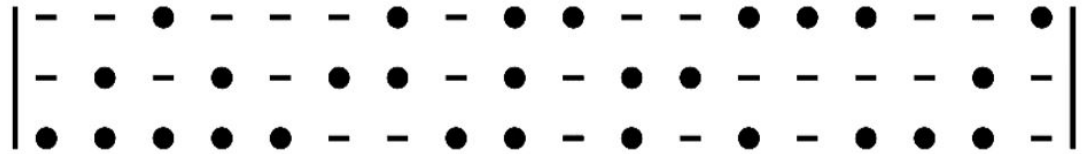
Zadání: Jednoho dne manžel v poštovní schránce našel dopis, který byl pro manželku a po neoprávněném otevření našel tento vzkaz:



Obrázek 15: Vzkaz v dopise

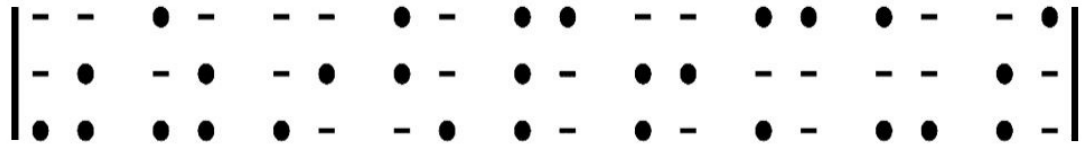
Manželovi to přišlo zvláštní, a proto jako každý žárlivý muž vše manželce vyčetl. Žena mu poté řekla, že má tajného milence, se kterým si nějakou dobu dopisuje a schází se. Dále mu prozradila, že zašifrovaný vzkaz nemá s Braillovým písmem nic společného, avšak samotná zpráva je tímto písmem psána. Manžel zabrblal, že je mu tato informace k ničemu, když nezná dešifrovací klíč. Vytočená žena omylem prozradila, že dešifrovací klíč v sobě ukrývá 3 stejná čísla. Na závěr dodala, že s milencem skončí, pokud manžel šifru prolomí a vzkaz rozluští. Jak má manžel postupovat, aby zachránil manželství?

Řešení: V první řadě si doplníme prázdná místa např. pomlčkou, takže tajný vzkaz bude vypadat následovně:



Obrázek 16: Vzkaz doplněný o pomlčky

Dále je potřeba si uvědomit, že mezi Braillovým písmem a touto šifrou bude určitě nějaký vztah. Každý znak v Braillově písmě je zapsán do tabulky 3×2, většinou symbolikou pomocí tečky a pomlčky. Takže bychom text měli trochu přeuspořádat a to následovně:



Obrázek 17: Přeuspořádaný vzkaz

Tak, teď už to nevypadá tak strašidelně. Nyní můžeme zavést substituci, např. *tečka = 1*, *pomlčka = 0*. Pokud bychom každou šestici uzávorkovali, dostali bychom následující matice:

$$\left| \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \right|$$

Obrázek 18: Vzkaz jako matice

Nyní si stačí uvědomit, že dešifrovací klíč musí být ve tvaru čtvercové matice, která splňuje následující podmínky:

1. Tato matice je řádu 3 a je regulární.
2. Tato matice je dešifrovací (tudíž je inverzní k šifrovací matici).
3. Touto maticí lze vynásobit všechny matice, které jsme dostali po substituci.
4. Tato matice nemůže obsahovat jiná čísla než 0 a 1 (tedy tečky a pomlčky po substituci) - kdyby obsahovala, nikdy bychom nedostali výsledné matice, které by obsahovaly jen 1 a 0, takže by nebyl možný překlad Braillova písma.
5. Tato matice obsahuje 3 stejná čísla.

Když všech těchto 5 podmínek dáme dohromady, pak je jasné, že dešifrovací matice musí být čtvercová matice řádu 3, která má na všech místech buď 1 nebo 0, přičemž platí, že v matici budou buďto tři jedničky a šest nul, anebo tři nuly a šest jedniček. Možnost, kdy v matici bude šest jedniček a tři nuly, můžeme hned zavrhnout, protože stačí uvažovat např.

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}, \ldots$$

Vidíme, že i kdybychom vzali libovolnou čtvercovou matici řádu 3 se šesti jedničkami, která by splňovala všech 5 podmínek uvedených výše, potom by výsledná matice v sobě obsahovala číselné hodnoty 2 (nebo i 3), a to nemůže nastat (viz bod 4). Z tohoto důvodu nám zbylo pouze šest dešifrovacích matic a to tyto:

$$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}, \begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}, \begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}, \begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}, \begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}, \begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix}.$$

Nyní postupně každou z nich vynásobíme s výslednými maticemi (text po substituci). Pro první čtvercovou matici 3×3 dostaneme stejný text. Jak to vypadá s další čtvercovou maticí?

$$\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix} \cdot \begin{pmatrix}0&0\\0&1\\1&1\end{pmatrix} = \begin{pmatrix}1&1\\0&1\\0&0\end{pmatrix}$$

$$\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix} \cdot \begin{pmatrix}1&0\\0&1\\1&1\end{pmatrix} = \begin{pmatrix}1&1\\0&1\\1&0\end{pmatrix}$$

$$\vdots$$

Až bychom dostali tyto matice:

$$\begin{pmatrix}1&1\\0&1\\0&0\end{pmatrix}, \begin{pmatrix}1&1\\0&1\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\0&1\\0&0\end{pmatrix}, \begin{pmatrix}0&1\\1&0\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\1&0\\1&1\end{pmatrix}, \begin{pmatrix}1&0\\1&1\\0&0\end{pmatrix}, \begin{pmatrix}1&0\\0&0\\1&1\end{pmatrix}, \begin{pmatrix}1&1\\0&0\\1&0\end{pmatrix}, \begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix},$$

což bychom převedli na tečky a pomlčky (návrat do substituce). Kdybychom měli po ruce tabulku Braillova písma, podívali bychom se, zda náhodou nějaké znaky neodpovídají příslušným „maticím“. Zjistili bychom, že takovýto zápis nám dá odpověď:

„DNES V 8 U MĚ“.

Tím jsme prolomili šifru a smutnému manželovi zachránili manželství.

Poznámka: Na konci tohoto příkladu je slovo *matice* v uvozovkách, protože se nejedná o takové matice, jaké je známe, nýbrž o matice, ve kterých se vyskytují jen pomlčky a tečky. Tabulka Braillova písma je obsažena v příloze na konci tohoto textu.

Příklad 8

Zadání: Dešifrujte tuto zprávu:

„671 6721 671 6732 671 792 803 6721 6765 825 220 770 550 6765 6721 825 704 220 6710 803 814 715 220 704 6743 220 737 792 671 814 825 6732 825 6710 759 2354 220 814 671 6721 220 6710 803 814 715 220 704 6765 792 671 693 759 220 671 220 6710 671 220 616 671 6765 803 6732 759 220 671 7810 220 803 715 6743 2332 693 715 814 6732 759 220 6743 6765 6710 759 220 682“,

když víte, že k zašifrování zprávy byl použit jedenáctinásobek vektorů. Dále víte, že čísla byla zapsána hexadecimálně a byla použita ASCII tabulka pro šifrování. Bohužel ale nevíte, v jakém pořadí byly vektory uspořádány.

Řešení: V první řadě si spočítáme počet čísel v kódované zprávě a zjistíme, že jich je 84. Těchto 84 čísel můžeme vyjádřit jako 7 vektorů o 12 souřadnicích (nebo také např. 14 vektorů o 6 souřadnicích, atd.). Budeme tedy uvažovat kombinaci 7 vektorů o 12 souřadnicích (i když nevíme v jakém pořadí byly tyto vektory uspořádány). Takže dostáváme

$$\begin{aligned}
\overrightarrow{u_1} &= (671\ 6721\ 671\ 6732\ 671\ 792\ 803\ 6721\ 6765\ 825\ 220\ 770),\\
\overrightarrow{u_2} &= (550\ 6765\ 6721\ 825\ 704\ 220\ 6710\ 803\ 814\ 715\ 220\ 704),\\
\overrightarrow{u_3} &= (6743\ 220\ 737\ 792\ 671\ 814\ 825\ 6732\ 825\ 6710\ 759\ 2354),\\
\overrightarrow{u_4} &= (220\ 814\ 671\ 6721\ 220\ 6710\ 803\ 814\ 715\ 220\ 704\ 6765),\\
\overrightarrow{u_5} &= (792\ 671\ 693\ 759\ 220\ 671\ 220\ 6710\ 671\ 220\ 616\ 671),\\
\overrightarrow{u_6} &= (6765\ 803\ 6732\ 759\ 220\ 671\ 7810\ 220\ 803\ 715\ 6743\ 2332),\\
\overrightarrow{u_7} &= (693\ 715\ 814\ 6732\ 759\ 220\ 6743\ 6765\ 6710\ 759\ 220\ 682).
\end{aligned}$$

Nyní můžeme každý vektor vydělit 11, tj.

$$\begin{aligned}
\overrightarrow{u_1} &= (61\ 611\ 61\ 612\ 61\ 72\ 73\ 611\ 615\ 75\ 20\ 70),\\
\overrightarrow{u_2} &= (50\ 615\ 611\ 75\ 64\ 20\ 610\ 73\ 74\ 65\ 20\ 64),\\
\overrightarrow{u_3} &= (613\ 20\ 67\ 72\ 61\ 74\ 75\ 612\ 75\ 610\ 69\ 214),\\
\overrightarrow{u_4} &= (20\ 74\ 61\ 611\ 20\ 610\ 73\ 74\ 65\ 20\ 64\ 615),\\
\overrightarrow{u_5} &= (72\ 61\ 63\ 69\ 20\ 61\ 20\ 610\ 61\ 20\ 56\ 61),
\end{aligned}$$

$$\vec{u_6} = (615\ 73\ 612\ 69\ 20\ 61\ 710\ 20\ 73\ 65\ 613\ 212),$$
$$\vec{u_7} = (63\ 65\ 74\ 612\ 69\ 20\ 613\ 615\ 610\ 69\ 20\ 62).$$

Víme, že čísla byla zakódována hexadecimálně a tato čísla jsou ve tvaru: *0 1 2 3 4 5 6 7 8 9 10 A B C D E F*, díky čemuž můžeme vytvořit tabulku:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Dále víme, že hexadecimální čísla v ASCII tabulce jsou zapsána jako dvouciferná čísla, takže nyní budeme substituovat poslední 2 číslice (u tříciferných čísel) za znaky A-F.

$$\vec{u_1} = (61\ 6B\ 61\ 6C\ 61\ 72\ 73\ 6B\ 6F\ 75\ 20\ 70),$$
$$\vec{u_2} = (50\ 6F\ 6B\ 75\ 64\ 20\ 6A\ 73\ 74\ 65\ 20\ 64),$$
$$\vec{u_3} = (6D\ 20\ 67\ 72\ 61\ 74\ 75\ 6C\ 75\ 6A\ 69\ 2E),$$
$$\vec{u_4} = (20\ 74\ 61\ 6B\ 20\ 6A\ 73\ 74\ 65\ 20\ 64\ 6F),$$
$$\vec{u_5} = (72\ 61\ 63\ 69\ 20\ 61\ 20\ 6A\ 61\ 20\ 56\ 61),$$
$$\vec{u_6} = (6F\ 73\ 6C\ 69\ 20\ 61\ 7A\ 20\ 73\ 65\ 6D\ 2C),$$
$$\vec{u_7} = (63\ 65\ 74\ 6C\ 69\ 20\ 6D\ 6F\ 6A\ 69\ 20\ 62).$$

Díky ASCII tabulce dostaneme následující vektory:

$$\vec{u_1} = (\mathrm{a\ k\ a\ l\ a\ r\ s\ k\ o\ u\ \_\ p}),$$
$$\vec{u_2} = (\mathrm{P\ o\ k\ u\ d\ \_\ j\ s\ t\ e\ \_\ d}),$$
$$\vec{u_3} = (\mathrm{m\ \_\ g\ r\ a\ t\ u\ l\ u\ j\ i\ .}),$$
$$\vec{u_4} = (\mathrm{\_\ t\ a\ k\ \_\ j\ s\ t\ e\ \_\ d\ o}),$$
$$\vec{u_5} = (\mathrm{r\ a\ c\ i\ \_\ a\ \_\ j\ a\ \_\ V\ a}),$$
$$\vec{u_6} = (\mathrm{o\ s\ l\ i\ \_\ a\ z\ \_\ s\ e\ m\ ,}),$$
$$\vec{u_7} = (\mathrm{c\ e\ t\ l\ i\ \_\ m\ o\ j\ i\ \_\ b}).$$

A při vhodném přeuspořádání těchto vektorů dostáváme text:

„Pokud_jste_dosli_az_sem,_tak_jste_docetli
_moji_bakalarskou_praci_a_ja_Vam_gratuluji.“

Poznámka: Jedná se o transpoziční šifru. Šifrování bylo názorně předvedeno v příkladu 1. Šifrovací klíč je ve tvaru $r = \vec{u_2} \circledast \vec{u_6} \circledast \vec{u_4} \circledast \vec{u_7} \circledast \vec{u_1} \circledast \vec{u_5} \circledast \vec{u_3}$.

Závěr

Během realizace předkládané práce jsem nastudoval tři oblasti matematiky, jež mají blízký vztah k teoretické informatice. Na základě těchto získaných teoretických poznatků jsem vytvořil tuto práci. Nejdříve jsem vymyslel zadání příkladů z kryptografie, poté z teorie grafů a nakonec z teorie kódování. Během výpočtů jsem bádal různě po internetu a inspiroval se v odborné literatuře. Příklady jsem postupně jeden po druhém po čase vyřešil. Při výpočtech jsem velmi hojně využíval internetovou stránku https://matrixcalc.org/cs/, která je velmi silným nástrojem, např. k výpočtu determinantu matice vyššího řádu. Po dokončení praktické části této bakalářské práce jsem začal zpracovávat teoretickou část, která by měla pomoct pochopit řešené příklady a postupy k jejich řešení. Po sepsání všech poznámek jsem text vysázel v prostředí LaTeX.

Výsledkem této bakalářské práce je řešená sbírka úloh speciálních aplikací lineární algebry v teoretické informatice. Konkrétně tedy úloh z již zmíněné teorie grafů, teorie kódování a kryptografie. Tato řešená sbírka úloh by měla prohloubit znalosti váženého čtenáře z lineární algebry ve třech uvedených oblastech. Předpokladem pro porozumění tomuto textu byla jistá zkušenost čtenáře s teoretickou informatikou a především tedy s lineární algebrou.

Literatura

[1] Hort, D., Rachůnek, J.: *Algebra I*, 1. vyd. Olomouc: Univerzita Palackého, 2003, s. 172, ISBN: 80-244-0631-4 (brož.)

[2] Holčík, J., Komenda, M. *Matematická biologie*: e-learningová učebnice [online]. 1. vyd. Brno: Masarykova univerzita, 2015. ISBN 978-80-210-8095-9

[3] Velebil, J.: *Diskrétní matematika*, Praha, 2007. Dostupné z: https://docplayer.cz/13616502-Diskretni-matematika.html

[4] Doc. Mgr. Habala, P. Ph.D.: *Diskrétní matematika*, Praha, 2012. Dostupné z: https://math.feld.cvut.cz/habala/teaching/dma/dmknih07.pdf

[5] Kovář, P.: *Teorie Grafů*, Paskov, 2012. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/skriptum_teorie_grafu_interaktivne.pdf

[6] Brousek, J., Čada R., Kaiser R., Ryjáček Z.: *Diskrétní matematika*, Plzeň, 2003. Dostupné z: http://martin.lipinsky.cz/skola/dma/prednasky/

[7] Olšák, P.: *Lineární algebra*, Praha, 2000-2007. Dostupné z: http://petr.olsak.net/ftp/olsak/linal/linal.pdf

[8] Demlová, M.: *Algebra pro výpočetní techniku*, ČVUT, 2010. Dostupné z: http://math.feld.cvut.cz/demlova/teaching/avt/celek-a9.pdf

[9] Ing. Matoušek, R. Ph.D.: *Metody kódování*, Brno, 2006. Dostupné z: http://www.uai.fme.vutbr.cz/~matousek/TIK/TIKv19.pdf

[10] Doc. RNDr. Koucký, M. CSc.: *Matematika pro informatiky II*, Liberec, 2016. Dostupné z: https://kap.fp.tul.cz/attachments/article/279/Matematika%20pro%20informatiky%20II.pdf

[11] Wikipedia (CZ), *Algoritmus prohledávání do šířky*, Internetový odkaz: https://bit.ly/2Qex4o4

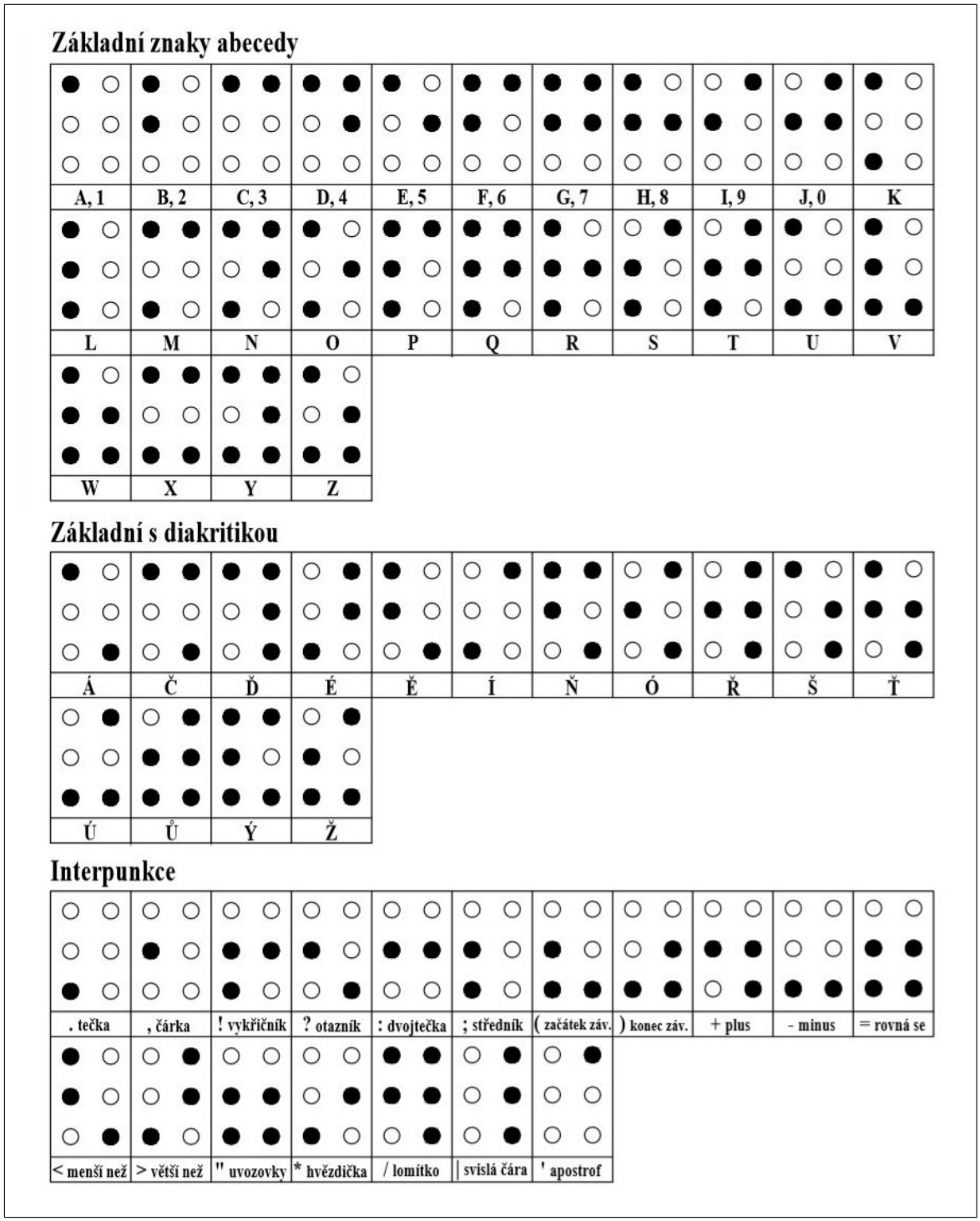
Seznam obrázků

Přílohy

Příloha 1: ASCII tabulka

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Příloha 2: Tabulka Braillova písma



Příloha 3: Tabulka Morseovy abecedy

Písmeno	Kód	Písmeno	Kód
A	• –	S	• • •
B	– • • •	T	–
C	– • – •	U	• • –
D	– • •	V	• • • –
E	•	W	• – –
F	• • – •	X	– • • –
G	– – •	Y	– • – –
H	• • • •	Z	– – • •
CH	– – – –	1	• – – – –
I	• •	2	• • – – –
J	• – – –	3	• • • – –
K	– • –	4	• • • • –
L	• – • •	5	• • • • •
M	– –	6	– • • • •
N	– •	7	– – • • •
O	– – –	8	– – – • •
P	• – – •	9	– – – – •
Q	– – • –	0	– – – – –
R	• – •	Oddělení znaku	|