

Mezinárodně povinné úlohy v národním kole soutěže Bobřík informatiky 2010

JIŘÍ VANÍČEK

Pedagogická fakulta JU, České Budějovice



Úvod

V listopadu 2010 proběhl třetí ročník soutěže *Bobřík informatiky*, kterou pro žáky základních a středních škol organizuje Pedagogická fakulta Jihočeské univerzity pod patronací Jednoty školských informatiků. V národním kole soutěžilo téměř 15 000 soutěžících z více než 170 škol, z toho (a to si považujeme) 42 % dívek.

Soutěžilo se nově ve čtyřech věkových kategoriích, pro žáky 8. – 9. tříd byla vyčleněna kategorie Kadet. Tím se věkově oddělily kategorie pro základní a střední školy (více o této soutěži v [1]). Soutěž patří do hroznu mezinárodních soutěží, v loňském roce se uskutečnila ve 13 zemích Evropy. V této společnosti naše vlast zaujala 4. místo jak co do absolutního počtu soutěžících, tak podle „hustoty“ soutěžících (počtu zúčastněných na 100 000 obyvatel) [2].

Úlohy do soutěže navrhuji jednotlivé zúčastněné státy, mezinárodní komise je diskutuje, schvaluje a upravuje. Tyto úlohy jsou poté poskytovány národním organizátorům, kteří sestavují své vlastní soutěžní testy, v nichž mohou použít i jiné než nabízené úlohy, tedy testy obsahují jednak úlohy povinné a jednak národní.

Každý stát ze sdružení zemí pořádajících soutěž typu Bebras si sestavuje svůj seznam úloh, které v testování použije. Aby byla zaručena aspoň nějaká obsahová podobnost těchto soutěží, je stanoveno, že určitý počet předem daných úloh musí každý stát do svých testů zařadit povinně. Vloni se jednalo o dvě úlohy pro každou kategorii plus jednu úlohu společnou pro všechny čtyři kategorie (a kterou tedy řešili všichni soutěžící). V českém testu je celkem 15 otázek, povinných tedy bylo v každé kategorii 20 %.

Volba povinných úloh probíhá poměrně demokraticky na společném workshopu, který se koná v některé zúčastněné zemi. Po tzv. „primárkách“, kdy každá pracovní skupina vybere ze seznamu svých zpracovávaných úloh ty nejlepší, jsou tyto úlohy jednotlivě prezentovány a podrobně diskutovány v plénu (cca 45 přítomných). Každý přítomný se může vyjádřit k obsahu, ke znění, ke vhodnosti úlohy, hledají se její slabiny. Na základě připomínek jsou kandidátské úlohy přepracovány a poté vystaveny jako kandidáti na povinné úlohy. Z kandidátských úloh se vybírá hlasováním zástupců každého státu. Úlohy s největším počtem hlasů jsou vybrány jako povinné.

V tomto článku seznámíme čtenáře se zněním povinných úloh, aby česká učitelská veřejnost měla představu, jaké úlohy jsou v Evropě považovány za informatické, kvalitní a přitom průchozí pro všechny státy. V naší české verzi soutěže totiž občas máme signály od učitelů, že jim úlohy nepřipadají informatické, že informatika učí něco jiného.

Výhodou online soutěže je kromě okamžité zpětné vazby i pohodlná možnost kvantitativní analýzy výsledků. Druhým záměrem článku je poskytnout informace o srovnání úspěšnosti řešení některých úloh. Při této analýze vycházíme z vlastních šetření a také z výpočtů, které na základě sběru dat z loňské soutěže v šesti státech provedla Jonaitytė [3] a které zde interpretujeme. Jiná porovnání lze najít ve srovnávací analýze českých a slovenských úloh, publikované v [4], již lze stáhnout z webu soutěže [5].

Soutěžní úlohy

Nejprve uvedeme zadání jednotlivých úloh. Správná řešení s vysvětlením každé zveřejněné úlohy zařazujeme až na konec článku, abychom podpořili čtenářův zájem úlohy si samostatně vyřešit. Je třeba podotknout, že na řešení jedné úlohy má soutěžící v průměru méně než 3 minuty.

Brouk a jeho pouť

Zajímala nás otázka, jak budou stejnou úlohu řešit žáci od nejmladších po nejstarší. Proto jsme tuto úlohu zařadili do všech kategorií.

Zadání: Brouk se pohybuje po hracím plánu (obr. 1) podle těchto pravidel:

- Začíná svoji cestu na libovolném políčku.
- Při jednom tahu popojde o tolik políček, kolik šipek je na políčku, na kterém stojí, ve směru, který šipky ukazují.
- Během tohoto tahu se brouk nezajímá o šipky na políčkách, přes která se pohybuje.
- Brouk opakuje své tahy tak dlouho, dokud nevyleze pryč z hracího plánu nebo nedorazí do některého políčka bez šipek (v sloupci E).

Ve kterém z polí ve sloupci A musí brouk svoji pouť začít, aby skončil v políčku ve sloupci E?

	A	B	C	D	E
1	→ →	→ →	↓ ↓	↓ ↓	
2	↓ ↓	→ →	↓ ↓ ↓	→ →	
3	→ →	↑ ↑	↓ ↓	← ←	
4	→ →	↑ ↑ ↑	→ →	→ →	

Obr. 1 Brouk a jeho pouť

Nabízené odpovědi:

A2, A4 A2, A3, A4 A1, A2 A1, A4 bez odpovědi

Komentář: Tato úloha zdá se být obtížná pro mladší žáky a velmi snadná pro žáky starší. Očekávali jsme, že s rostoucím věkem bude úspěšnost v řešení této úlohy růst. Jednotlivé věkové kategorie dosáhly následujícího podílu správných odpovědí: Benjamin 44 %, Kadet 59 %, Junior 69 %, Senior 70 %. Zatímco u mladších kategorií úspěšnost opravdu výrazně roste, nečekali jsme, že Junioři i Senioři dosáhnou stejné úspěšnosti. Podle této úlohy to vypadá, jako by se vývoj uvažování tohoto typu po 16. roku věku zastavil. Výsledek je ovšem pouhá hypotéza (jednalo se o jedinou takovou úlohu), v příštích letech ji však budeme zkoumat dále.

V kategorii Senior (tedy v podstatě u maturantů) jsme zjišťovali, které úlohy jim činí největší potíže. Ukázalo se, že problematické byly takové,

jejichž podstatou je porozumět nějakému formálnímu zápisu, reprezentaci reality pomocí informatických struktur. Nevalnou úspěšnost měly také logické úlohy, viz dále.

0X (pro kat. Senior, 26 % správných odpovědí)

Zadání: Máme řadu písmen, která obsahuje mezery a jedno X. Kurzor myši (značený |) je umístěn na začátku řady:

|----- X -----

Jsou dány následující instrukce:

Dokud není kurzor před X {piš 0}

Dokud není kurzor na začátku řady {piš X a posuň kurzor o dvě místa doleva}

Jak bude vypadat horní řada po provedení těchto instrukcí? Pozor, při psaní na příslušném místě mezeru nebo znak přepisujete!

Nabízené odpovědi:

A	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	
C		0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-
D		0	X	X	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-

Natírání dveří (pro kat. Senior, úloha nepovinná, 23 % správných odpovědí)

Zadání: Patrik bydlí na internátu, který má dveře pokojů natřeny tmavou nebo světlou barvou (obr. 2). Natěrač přichází natřít některé dveře na bílo. Použije k tomu proceduru nazvanou Natři(patro, dveře):

Natři(patro, dveře) znamená:

Jestliže existuje pokoj(patro, dveře), pak

Jestliže barva dveří pokoje(patro, dveře) je tmavá,
vykonej následujících 5 řádků:

Obarvi dveře pokoje (patro, dveře) na bílo

Natři(patro, pokoj - 1)

Natři(patro, pokoj + 1)

Natři(patro - 1, pokoj)

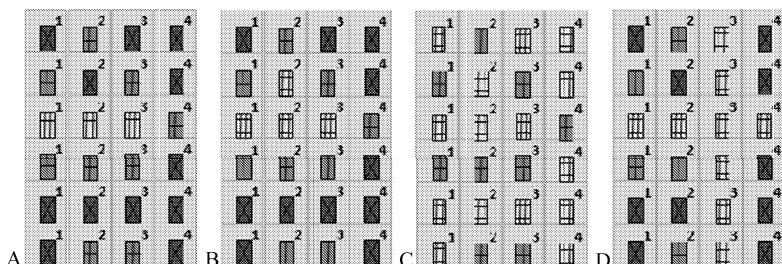
Natři(patro + 1, pokoj)

Jak bude internát vypadat po provedení procedury Natři(4, 3)?

6. patro				
5. patro				
4. patro				
3. patro				
2. patro				
1. patro				

Obr. 2 Natírání dveří

Nabízené odpovědi:



Obr. 2A,

Obr. 2B,

Obr. 2C,

Obr. 2D

Kdo snědl džem (pro kat. Senior, úloha nepovinná, 30 % správných odpovědí)

Zadání: Pavel, Kuba a Mates jeli na prázdniny k babičce. Jedno odpodne našla babička sklenici od jahodového džemu prázdnou. Chlapci byli dotázáni, kdo snědl džem. Řekli:

Pavel: Jsem nevinný!

Kuba: Jsem nevinný!

Mates: Kuba ho snědl.

Jestliže víme, že jen jedno z těchto tvrzení je pravdivé, kdo snědl džem?

Nabízené odpovědi:

Pavel Kuba Mates ani jeden z nich

Funkce KDYŽ (pro kat. Senior, úloha nepovinná, 38 % správných odpovědí)

Zadání: V programovacím jazyce je k dispozici funkce KDYŽ. Tato funkce pracuje takto:

KDYŽ(podmínka; hodnota1; hodnota2) = hodnota1, když podmínka je splněna (pravda), hodnota2, když podmínka není splněna (nepravda).

Dejme tomu, že $A = 3$, $B = 4$ a $C = 5$.

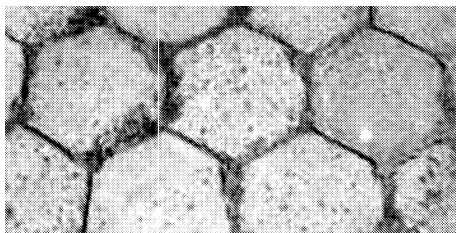
Co bude hodnotou funkce KDYŽ($A > B$; A ; KDYŽ($B < C$; C ; B))?

Nabízené odpovědi: 3 4 5 pravda

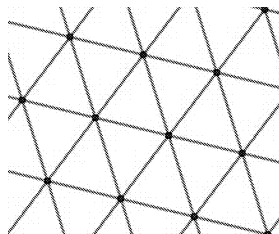
Komentář: Zatímco s předchozími třemi úlohami se typově ani tematicky nemusí žák během školní docházky setkat, poslední úloha patří do tematického celku práce s tabulkami, povinného pro všechny typy středních škol. Zde nemůžeme tvrdit, že se soutěžící setkali s neznámým tématem, dokonce výraz v otázce odpovídá syntaxi v prostředí software MS Excel nebo OpenOffice Calc a lze tedy tuto úlohu považovat za snadnou. Přesto úloha dopadla poměrně neslavně, nicméně možné příčiny (netriviální úloha, odstrašující text první věty zadání, absence vizualizace, zaměření školní výuky spíše na uživatelský přístup, výuka tématu bez řešení netriviálních úloh) si netroufáme posoudit.

Dláždění (česká úloha, pro kat. Junior, 43 % správných odpovědí)

Zadání: Petr si vyfotil dlažbu před domem a sestavil podle ní graf (obr. 3-1 a 3-2).



Obr. 3-1 Dláždění



Obr. 3-2 Graf

Na grafu bod znamená dlaždici a čára, spojující dva body, znamená, že tyto dlaždice spolu sousedí. Petr poté chodil městem a fotil dlažbu na chodnících. Doma zjistil, že na všechny dlažby (s výjimkou jedné) může použít stejný graf, který již nakreslil. Pro kterou dlažbu se nehodí Petrův graf?

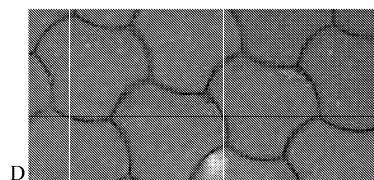
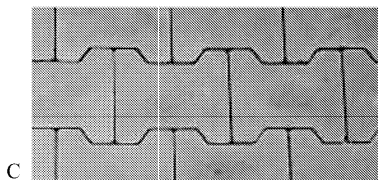
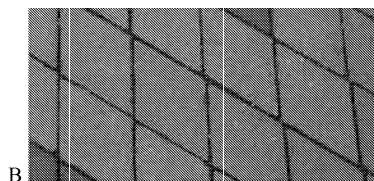
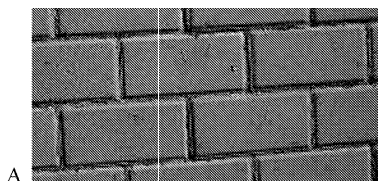
Nabízené odpovědi:

A (obr. 3A)

B (Obr. 3B)

C (Obr. 3C)

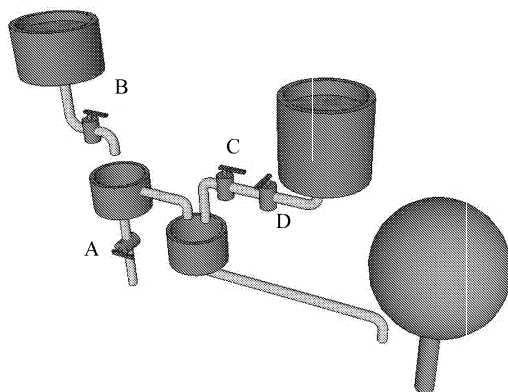
D (Obr. 3D)



Obr. 3A Obr. 3B Obr. 3C Obr. 3D

Zavlažovací systém (pro kat. Benjamin, 69 % správných odpovědí)

Zadání: Petr pro svoji jablň sestrojil systém zavlažování. Kohouty A, B, C, D se mohou otevřít nebo zavřít. *Ve kterém případě jistě dostane strom vodu?*



Obr. 4 Zavlažovací systém

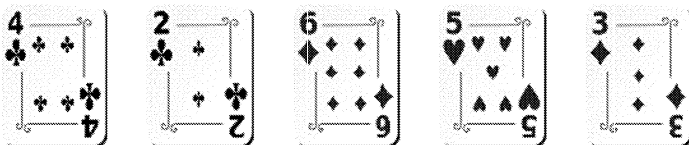
Nabízené odpovědi:

A:	A otevřeno,	B zavřeno,	C zavřeno,	D otevřeno
B:	A otevřeno,	B otevřeno,	C otevřeno,	D zavřeno
C:	A zavřeno,	B otevřeno,	C zavřeno,	D zavřeno
D:	A zavřeno,	B zavřeno,	C zavřeno,	D otevřeno

Řazení karet (pro kat. Kadet, 56 % správných odpovědí)

Zadání: O přestávce mezi vyučováním si žáci hrají karetní hru. Na konci hry musí být karty seřazeny od nejmenší po největší. V jednom tahu lze prohodit pouze dvě sousední karty. Na kartách se počítají pouze hodnoty (na barvě nezáleží).

Kolik nejméně tahů je třeba k seřazení karet na obrázku?



Obr. 5 Počáteční stav

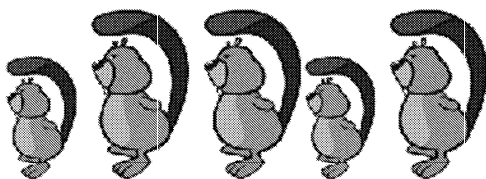
Nabízené odpovědi:

4 5 6 7

Sloupec talířů (pro kat. Benjamin)

Tuto úlohu zařadily letos do kategorie pro žáky od 5. do 7. ročníku ZŠ všechny státy.

Zadání: V bobří školní jídelně jsou dva druhy talířů: světlé pro malé a tmavé pro velké žáky. Jednou měly kuchařky místo jen pro jeden sloupec talířů. Žáci se řadí do fronty na oběd a v kuchyni se musí dát talíře podle fronty do správného pořadí. Například fronta může vypadat dle obr. 6-1 a talíře musí být položeny tak, aby souhlasily s frontou (obr. 6-2) ...



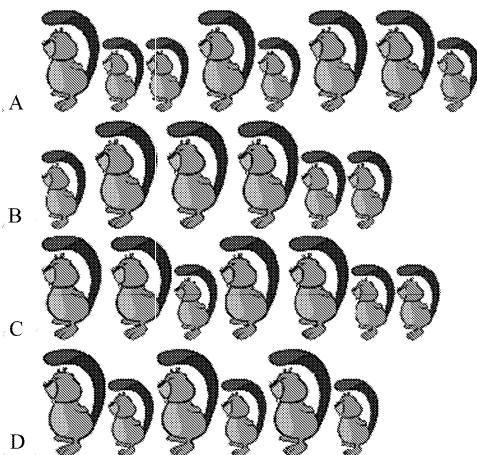
Obr. 6-1 Fronta bobrů



Obr. 6-2 Sloupec talířů

Ve kterém z následujících obrázků je nesouhlas mezi frontou a talíři?

Nabízené odpovědi:

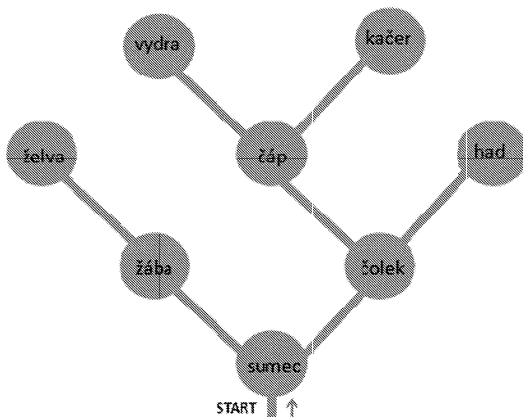


Kanoe na jezeře (pro kat. Senior, 71 % správných odpovědí)

Zadání: Indián Malý Manitou se rozhodl prozkoumat neznámou řeku. Vzal kanoe a pádloval. Řeka byla rozvětvená a byla na ní malá jezera, v každém žilo jedno zvíře (obr. 7). Malý Manitou si podle zvířat jezera pojmenoval a jejich jména si zapisoval. Vyjel z místa *start* a také se tam nakonec vrátil. Aby nezabloudil, rozhodl se dodržovat tři pravidla:

1. Jestliže z jezera vedou dvě řeky, po kterých ještě nejel, pustí se po té, kterou má po své levici.
2. Vede-li z jezera jedna řeka, po které ještě nejel, vydá se po ní.
3. Jestliže z jezera nevede žádná řeka, po které ještě nejel, popluje tím směrem, odkud do jezera připlul poprvé.

V jakém pořadí Malý Manitou zapsal spatřená zvířata?



Obr. 7 Kanoe na jezeře

Nabízené odpovědi:

- A sumec, žába, želva, čolek, čáp, vydra, kačer, had
- B sumec, žába, želva
- C sumec, čolek, had, čáp, kačer, vydra, žába, želva
- D sumec, žába, čolek, želva, čáp, had, vydra, kačer

Ranní běh (pro kat. Kadet, 60 % správných odpovědí)

Zadání: Mírek je baseballista a tak trénuje. Každé ráno běhá kolem domu a má předepsaný tréninkový plán, který přesně dodržuje. Na dnešní den má naplánováno:

Aktivita Běhání

vykonej aktivitu Oběhni_blok
vykonej aktivitu Oběhni_blok
vykonej aktivitu Oběhni_blok

Aktivita Oběhni_blok

vykonej aktivitu Přeběhni_ulici
vykonej aktivitu Přeběhni_ulici
vykonej aktivitu Přeběhni_ulici
vykonej aktivitu Přeběhni_ulici

Aktivita Přeběhni_ulici

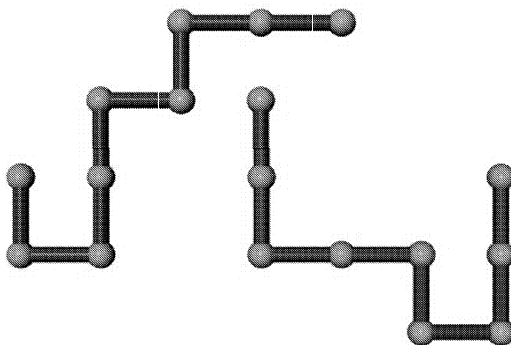
uběhni 100 kroků
zahni doleva

Kolik Mirek uběhne kroků, vykoná-li aktivitu Běhání?

Nabízené odpovědi: 100 300 400 1200

Potrubí (pro kat. Junior)

Zadání: Každé z těchto dvou potrubí je složeno z osmi stejných dílů. Tato potrubí budou položena na sebe tak, aby se co nejvíce překrývala (potrubím se může otáčet). *Jaký je největší možný počet dílů, kterými se potrubí budou překrývat?*



Obr. 8 Potrubí

Nabízené odpovědi: 3 4 5 6

Hraje tu roli i soutěžní taktika?

V loňském roce jsme měli možnost porovnat výsledky u stejných úloh, zadaných v české a slovenské verzi soutěže. Letos jsme takové porovnání mohli učinit u českých a litevských škol. Vystoupil zde jeden zajímavý fakt: čeští soutěžící častěji než jejich slovenští či litevští vrstevníci častěji volili možnost „Žádná odpověď“. Toto se potvrdilo i u srovnání sedmi států v povinných úlohách, kde byli Češi v této „disciplíně“ u většiny úloh nadprůměrní.

Je třeba vysvětlit, že každá úloha umožňuje kromě čtyř nabízených možností odpovědí možnost označit také pátou „já nebudu odpovídat“. Soutěžící musí tuto položku úmyslně zvolit, nejde tedy o to, že by otázku přeskočil nebo nestihl.

Možná, že naši žáci spíše hrají na jistotu a neriskují, nebo jsou pohodlnější, mají méně „bojovné srdce“, nebo jsou prostě lépe seznámeni s pravidly soutěže. Test je totiž hodnocen s handicapem; pokud žák odpoví špatně, odečtou se mu nějaké body, pokud sdělí, že neodpoví, zůstává na svých bodech.

Poznámka redakce: Po dohodě s autorem budou řešení všech úloh a literatura uvedeny až v příštím čísle, aby každý čtenář měl dostatek času si zadání úloh promyslet a úlohy vyřešit, aniž by jej rušilo oficiální „autorské řešení“.

(Pokračování)

Počítačová bezpečnost ve výuce informatiky

(4. část – automatické luštění monoalfabetické šifry)

MICHAL MUSÍLEK – ŠTĚPÁN HUBÁLOVSKÝ,

Přírodovědecká fakulta Univerzity Hradec Králové

V minulém článku (viz [3]) našeho seriálu jsme si vysvětlili pojmy *jednoduchá záměna* (monoalfabetická substituce) a frekvenční analýza šifrovaného textu, kterou jsme si ukázali na příkladu „ručního“ luštění šifrovaného textu.

V tomto pokračování na naše úvahy navážeme, ukážeme si možnost automatického luštění s využitím matematických metod. Součástí článku je i ukázka, jak mechanickou práci přenechat počítači a luštit šifrový text automaticky na základě vhodného algoritmu. Ve své práci jsme vyšli z algoritmu popsaného *Thomase Jacobsenem* [1]. Tento algoritmus jsme modifikovali, na rozdíl od něj jsme jej neaplikovali na anglický text s mezerami, ale na český text bez mezer.

Aby však bylo možné uvedený algoritmus aplikovat obecně na jakýkoliv zvolený jazyk, je naším prvním úkolem získat z vhodného referenčního textu frekvenci bigramů, v našem konkrétním případě bigramů v českém jazyce. Použili jsme text knihy Karla Čapka *Zahradníkův rok*. V textu jsme ignorovali diakritiku, interpunkční znaménka a mezery a pro každou dvojici po sobě jdoucích písmen jsme inkrementovali příslušný prvek čtvercové matice řádu 26 (což je počet písmen mezinárodní abecedy). Nakonec se tato matice celá vynásobí 100 a vydělí počtem písmen celého referenčního textu, takže hodnoty prvků matice nyní představují relativní četnosti bigramů vyjádřenou v procentech. Program, který jsme vytvořili, umožňuje snadnou změnu hodnoty této referenční matice, takže je možné jej použít i pro jiné jazyky než češtinu.

Dříve než přejdeme k vlastnímu výkladu algoritmu automatického luštění, detailněji se zmíníme o možnosti přehledného zápisu frekvence bigramů, protože je to zásadní pro pochopení problematiky automatického luštění.

Bigram, jak bylo uvedeno v předešlém čísle, je dvojice po sobě jdoucích písmen (např. AN, TU atd.). Pro daný jazyk jsou frekvence bigramů, stejně jako frekvence znaků, charakteristické. V dešifrování monoalfabetické substituce je tedy kromě frekvenční analýzy jednotlivých písmen důležité i analyzovat šifrovaný text pomocí bigramů. Frekvenční analýza písmen je totiž málo citlivá v tom smyslu, že řada písmen má přibližně stejnou četnost výskytu, snadno dojde k jejich záměně a nejsme tedy schopni správně automaticky provést luštění. Jedná se například o písmena C, D, K, L, M, P, U, Z, která mají četnost výskytu v českém jazyce okolo 3,5 %. Zapojíme-li do šifrování i porovnávání frekvence bigramů, situace se stane, jak uvidíme v následujícím textu, téměř jednoznačnou.

Vraťme se tedy k tomu, jak co nejefektivněji zaznamenat četnost bigramů. Vzhledem k tomu, že bigram je dvojice po sobě jdoucích písmen, je optimálním způsobem, jak zaznamenat jeho četnost, využití dvourozměrného pole, kde první index pole je určen prvním písmenem a druhý

index druhým písmenem. Hodnoty pole jsou pak čísla, která určují četnost daného bigramu. Například $B[A,N] = 3,6$ znamená že bigram AN má četnost 3,6 %; $B[N,A] = 5,2$ znamená, že bigram NA má četnost 5,2 %, atd. Četnost bigramů vyjádříme maticí, v níž . sloupce matice představují první písmeno bigramu a řádky matice druhé (v pořadí) písmeno digramu; prvek matice pak udává četnost, popř. frekvenci daného bigramu v daném textu. Ukázka části matice bigramů je na obrázku 1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
60	Matice bigramů referenčního textu															
61	Frekvence písmen															
62		11011	9605	7333	6577	6196	5799	5109	4577	4493	3879	3713	3701	3646	3477	3190
63		E	A	O	I	N	T	S	R	K	U	V	L	D	Z	M
64	E	7	20	5	47	1913	1287	1112	625	266	2	853	829	564	727	530
65	A	271	144	65	252	1156	841	244	1188	705	179	702	657	378	874	442
66	O	134	73	47	97	573	903	205	874	765	36	311	468	492	81	186
67	I	30	25	6	39	1192	540	598	656	8	8	374	776	286	348	370
68	N	1315	739	440	715	67	251	188	136	185	247	190	124	518	263	234
69	T	966	911	434	599	132	69	1346	44	252	275	45	36	40	107	88
70	S	900	739	739	570	69	109	34	117	113	478	170	81	66	79	115
71	R	551	434	355	190	7	460	43	6	271	128	93	6	173	57	108
72	K	715	930	298	648	146	135	312	50	41	205	55	84	130	155	65
73	U	105	69	850	64	200	287	102	304	425	19	96	172	181	66	314
74	V	466	629	748	427	29	175	221	104	240	142	3	26	62	121	92

Obr. 1 Matice bigramů referenčního textu v tabulce MS Excelu

Zde je zřejmé, že bigram AN se v daném textu vyskytuje 739 krát, zatímco bigram NA 1156 krát. Tato matice je vytvořena v tabulkovém procesoru MS Excel, ve kterém také, jak uvidíme v dalším textu, vytvoříme program pro automatické luštění. Z obrázku je zároveň zřejmé, že jednotlivá písmena jak v řádcích, tak ve sloupcích, jsou uspořádána dle své frekvence sestupně, což je důležité pro další analýzu textu – porovnání referenčního textu *E* a textu šifrovaného *D*.

Algoritmus automatického luštění

Jak již bylo řečeno v úvodu, v našich úvahách vycházíme z algoritmu publikovaného v [1]. Vlastní algoritmus lze rozdělit do tří samostatných podprogramů, které jsou na sobě relativně nezávislé.

Frekvence

Tato procedura slouží k počátečnímu nastavení převodové tabulky četností. Zjistí frekvence jednotlivých znaků v šifrovém textu a seřadí znaky šifrovaného textu sestupně podle četnosti. Z referenční matice $\mathbf{E}$ zjistí součty řádků, které odpovídají frekvencím jednotlivých písmen v referenčním textu a seřadí znaky referenčního textu sestupně podle četnosti. Nakonec přiřadí prvnímu nejčetnějšímu písmenu šifrovaného textu první nejčetnější písmeno referenčního textu, druhému druhé, třetímu třetí atd.

Procedura dále vytvoří matici $\mathbf{D}$ relativních frekvencí bigramů v šifrovaném textu a ohodnotí shodu s referenčním textem pomocí evaluační funkce

$$f = \sum_{i=1}^{26} \sum_{j=1}^{26} |D_{ij} - E_{ij}| \quad (1)$$

kde D_{ij} jsou prvky matice šifrovaného textu a E_{ij} jsou prvky matice textu referenčního. Jak uvidíme dále, nižší hodnota f znamená „lepší“ shodu v určení šifry.

Bigramy

Tato procedura je zásadní pro určení minima funkce f při luštění šifrovaného textu. Procedura se skládá z následujících kroků.

1. Postupně zaměňujeme vždy 2 sloupce a 2 řádky v matici $\mathbf{D}$ a tak získáváme matici $\mathbf{D}'$. Přitom vyměňujeme sloupce a řádky v pořadí daném frekvencí znaků v šifrovém textu od nejfrekventovanějších k nejméně frekventovaným, tedy

- vyměníme sloupec odpovídající pořadí prvního vyměňovaného znaku za sloupec odpovídající pořadí druhého vyměňovaného znaku
- a pak vyměníme řádek odpovídající pořadí prvního vyměňovaného znaku za řádek odpovídající pořadí druhého vyměňovaného znaku.

Zaměňujeme nejprve sousední písmena, potom ob jedno, ob dvě atd. až nakonec zaměníme první písmeno s posledním (tj. zaměňujeme sloupce a řádky číslo $1 \leftrightarrow 2$, $2 \leftrightarrow 3$, $3 \leftrightarrow 4$, $\dots$ $25 \leftrightarrow 26$, pak $1 \leftrightarrow 3$, $2 \leftrightarrow 4$, $\dots$ $24 \leftrightarrow 26$, pak $1 \leftrightarrow 4$, $\dots$, atd. až nakonec $1 \leftrightarrow 26$). Tím získáme všechny možné výměny písmen, je jich celkem $25 + 24 + 23 + \dots + 1 = 325$.

2. Po každé záměně získáme novou matici $\mathbf{D}'$ a pokaždé znovu vyhodnotíme shodu relativních frekvencí bigramů v šifrovaném textu a referenčním textu pomocí evaluační funkce

$$f' = \sum_{i=1}^{26} \sum_{j=1}^{26} |D'_{ij} - E_{ij}|$$

3. Po každé záměně porovnáme hodnoty f a f' . Pokud $f' > f$, pokračujeme další dvojicí písmen v pořadí. Pokud je ale $f' < f$, okamžitě přerušíme provádění záměn a navrhneme provedení příslušné výměny v převodové tabulce, které povede ke zlepšení shody (snížení hodnoty evaluační funkce f) a k přiblížení se k řešení úlohy správnou převodovou tabulkou – viz procedura *Výměna*.

Jestliže program vyčerpá všechny možnosti záměn znaků, aniž by nastala situace $f' < f$, oznámí to obsluze hlášením. Pokud v tu chvíli není šifrový text zcela rozluštěn, uživatel může navrhnout manuální výměnu.

Výměna

1. Protože ne vždy, zejména ne u krátkých textů (kratších než 500 písmen), navrhne procedura „Bigramy“ optimální výměnu, je možné požadovanou výměnu nastavit také ručně přímo v programem zobrazené převodové tabulce na základě čtení částečně rozluštěného textu a odhadu obsahu zprávy a z něho plynoucích očekávaných slov.

2. Ať již je výměna navržena automaticky, nebo manuálně, procedura *Výměna* provede nové vytvoření matice $\mathbf{D}$ a to tak, že matice $\mathbf{D}$ bude nahrazena novou maticí $\mathbf{D}'$, která má jiné pořadí písmen v záhlaví. Nová matice $\mathbf{D}$ se od tohoto okamžiku stává výchozí pro další analýzu textu. Podíváme-li se například na obrázek 1, který odpovídá matici $\mathbf{D}$, pak, pokud funkce $f' < f$ pro výměnu 23, bude nová matice $\mathbf{D}$ mít pořadí písmen jak ve sloupcích, tak řádcích E, O, A, I atd. oproti původnímu pořadí E, A, O, I, atd. Tomu bude odpovídat také nové umístění relativních frekvencí bigramů šifrovaného textu v rámci tabulky a nové ohodnocení shody s referenčním textem pomocí evaluační funkce (1).

Z popisu uvedených procedur je pozornému čtenáři zřejmé, že proceduru *Frekvence* voláme pouze jednou na začátku výpočtu pro nastavení výhodných počátečních podmínek. Procedury *Bigramy* a *Výměna* pak pravidelně

střídáme, případně místo volání procedury *Bigramy* navrhne manuálně výměnu znaků ve druhém řádku převodové tabulky na základě odhadu otevřeného textu. Odhad je tím úspěšnější, čím blíže jsme správné převodové tabulce a čím větší jsou naše zkušenosti s luštěním substitučních šifer.

Řešení automatického luštění monoalfabetické šifry v aplikaci MS Excel

Jak je uvedeno např. v článku [2], všechny aplikace sady MS Office, jako jsou např. MS Excel či MS Word, mají implementován vlastní programovací jazyk Visual Basic for Application (VBA). Ve zmíněném článku je popsáno integrované vývojové prostředí jazyka, jeho specifikace v aplikaci MS Excel, způsoby spouštění programů a základní příkazy vstupu, výstupu, přiřazení, větvení a cyklů.

Program automatického luštění monoalfabetické šifry jsme záměrně realizovali ve VBA pro Excel, protože tabulkový procesor umožňuje jednoduché zobrazení a uložení bigramových matic E , D a D' (což ovšem lze v jiných jazycích realizovat formou dvourozměrného pole). Další, ještě podstatnější odůvodnění, je to, že celý proces automatického luštění může uživatel přímo na listu Excelu sledovat, což, především z didaktických důvodů, považujeme za velmi užitečné.

Řešení lze rozdělit do několika více méně nezávislých částí.

Analýza referenčního textu:

S ohledem na možnost volby jazyka program nejprve vytvoří matici bigramů E referenčního textu. Referenčním textem by měl být jazykově „čistý“ text (neobsahující cizí slova) o délce cca 50 – 100 tisíc znaků. Jak již bylo řečeno, v naší ukázce jsme jako referenční použili text Karla Čapka *Zahradníkův rok*, který má cca 99 tisíc znaků.

Vlastní procedura je rozdělena na následující části:

- Načtení referenčního textu;
- Korekce referenčního textu;
- Frekvenční analýza textu;
- Uspořádání znaků (písmen) v sestupném pořadí;
- Vytvoření matice bigramů referenčního textu a její vypsání do tabulky Excelu.

Načtení referenčního textu provádíme pomocí objektu `TextBox`, který patří do sady nástrojů využívaných v objektovém programování ve VBA.

Do tohoto dialogového okna referenční text vložíme jednoduchým kopírováním – viz obrázek 2.



Obr. 2 Formulář pro vložení referenčního textu

Následuje korekce textu v tom smyslu, že veškerá písmena s diakritikou jsou nahrazena odpovídajícími písmeny bez diakritiky (např. Č – C, Á – A, atd.).

V rámci dalšího kroku se provede frekvenční analýza jednotlivých písmen a seřazení písmen v sestupném pořadí do řádku a sloupce matice **E** – řádek 63 a sloupec A na obrázku 1. V následujícím kroku program vypočítá četnosti bigramů a jednotlivé frekvence zaznamenaná do buněk v oblasti

B64:AA89. Četnost bigramů se počítá v cyklu s konečným počtem opakování, v rámci kterého procházíme dvojice po sobě jdoucích písmen. Pokud například narazíme na bigram **NA**, hodnota příslušné buňky matice bigramů (buňka F65 na obrázku 1) se navýší o hodnotu 1.

V dalších krocích začne již program provádět analýzu šifrovaného textu. Program se skládá z procedur: *Načtení a analýza šifrovaného textu*, *Frekvence*, *Bigramy* a *Výměna*.

Procedura *Načtení* a analýza šifrovaného textu je podobná předešlé proceduře, která analyzuje referenční text. Je rozdělena na dvě části:

- Načtení šifrovaného textu;
- Vypsání korigovaného textu do listu Excelu.

Načtení probíhá stejně jako u načtení referenčního textu pomocí objektu *TextBox*, do kterého uživatel nakopíruje šifrový text. Předpokládáme, že šifrový text neobsahuje písmena s diakritikou, přesto v rámci této procedury se provádí jednoduchá korekce textu, která převede všechna písmena na velká. Šifrový text je poté uložen a vypsán do buněk Excelu. Do každé buňky se vepíše jedno písmeno šifrovaného textu. Na obrázku 3 je vidět část šifrovaného textu uloženého v buňkách řádků 14, 16, 18, atd. Graficky (pomocí ohraničení buněk) je text rozdělen na pětice. Do lichých řádků se potom pomocí dalších procedur vyplňují (krok za krokem, jak provádíme algoritmus luštění) odpovídající písmena textu dešifrovaného podle aktuálně platné (a v průběhu luštění dynamicky se měnící) převodové tabulky.

Frekvence

Tato procedura, zapsaná do samostatného modulu VBA se v rámci dešifrování spustí pouze jednou. Provede následující kroky:

- Frekvenční analýzu šifrovaného textu;
- Uspořádá písmena sestupně dle četností;
- Vytvoří převodovou tabulku.

Vstupem pro frekvenční analýzu je již korigovaný šifrový text uložený v řádcích 14, 16, atd. Na základě jednoduchého algoritmu se vypočte četnost jednotlivých písmen a ta se zapíše do převodové tabulky. Tato tabulka přiřazuje písmenům šifrovaného textu odpovídající písmeno otevřeného textu. V této fázi algoritmu tato převodová tabulka, určená pouze na základě četností, téměř jistě nezajistí přesné dešifrování zašifrovaného textu. Slouží pouze pro „hrubé“ seřazení šifrovaných písmen a jako vhodný počáteční vstup do procedur *Bigramy* a *Výměna*.

	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	A
1	Převodní tabulka - uspořádání písmen dle frekvence znaků																										
2	E	A	O	I	N	T	S	R	K	U	V	L	D	Z	M	C	P	Y	H	J	B	F	G	X	W	Q	
3	M	A	Y	W	H	P	G	R	D	Z	C	X	B	L	T	K	U	Q	F	E	O	V	N	I	J	S	
4	902	350	170	159	142	143	141	128	125	107	105	102	101	75	73	70	68	66	58	55	45	1	1	1	0	0	
5	Převodní tabulka - uspořádání písmen otevřeného textu dle abecedy																										
6	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	
7	A	O	K	B	M	V	N	F	W	E	D	X	T	H	Y	U	S	R	G	P	Z	C	J	I	Q	L	
8	Převodní tabulka - uspořádání písmen šifrovaného textu dle abecedy																										
9	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	
10	A	D	V	K	J	H	S	N	X	W	C	Z	E	G	B	T	Y	R	Q	M	P	F	I	L	O	U	
11																											
12																											
13	Šifrový text a návrh dešifrovaného textu																										
14	P	M	T	N	A	R	F	G	Q	S	F	A	R	A	E	Z	N	R	Q	S	E	Q	O	A	F	D	Z
15	T	A	M	G	E	R	H	S	Y	Q	H	E	R	E	J	U	G	R	Y	Q	J	Y	B	E	H	K	U
16	P	G	O	W	E	P	M	B	F	Z	Z	R	Z	G	P	Y	E	Q	D	Y	G	F	A	B	M	O	E
17	T	S	B	I	J	T	A	D	H	U	U	R	U	S	T	O	J	Y	K	O	S	H	E	D	A	B	J
18	Q	R	W	O	E	M	X	A	F	A	Y	W	T	M	E	M	F	A	S	Z	P	A	I	M	F	A	I
19	Y	R	I	B	J	A	L	E	H	E	O	I	M	A	J	A	H	E	Q	U	T	E	X	A	H	E	X
20	M	X	A	F	Z	X	K	G	S	F	M	N	L	A	B	L	A	N	Z	S	W	Y	W	O	M	E	N
21	A	L	E	H	U	L	C	S	Q	H	A	G	Z	E	D	Z	E	G	U	Q	I	O	I	B	A	J	A

Obr. 3 Převodní tabulky, šifrový a návrh dešifrovaného textu

Na obrázku 3 jsou zřejmé tři převodové tabulky. První, v řádcích 2 a 3, představuje tabulku uspořádanou dle četnosti písmen otevřeného textu (řádek 2, kde četnosti jsou určené frekvenční analýzou referenčního textu) a četností písmen šifrovaného textu (řádek 3, kde četnost je určena frekvenční analýzou šifrovaného textu). Další dvě převodové tabulky jsou uspořádané jednak dle abecedního pořadí písmen otevřeného textu (řádky 6 a 7) a dle abecedního pořadí písmen šifrovaného textu (řádky 9 a 10). Tyto dvě „abecední“ převodové tabulky jsou pouhým jiným uspořádáním výchozí převodové tabulky a slouží k lepší představě o dané šifře.

Současně tato procedura vytvoří matici bigramů D šifrovaného textu a to zcela analogicky, jako se tvořila matice bigramů E referenčního textu – viz výše. Matice D je uložena opět na listu Excelu v oblasti buněk B33:AA58.

Bigramy

Tuto proceduru lze rozdělit do následujících částí:

- Výpočet funkce f ;
- Tvorba matice D' a výpočet funkce f' .

Funkce f se vypočte pomocí prvků matic E_{ij} a D_{ij} . Výpočet provádí část programu, jehož část kódu zapsaná v jazyce Visual Basic je následující:

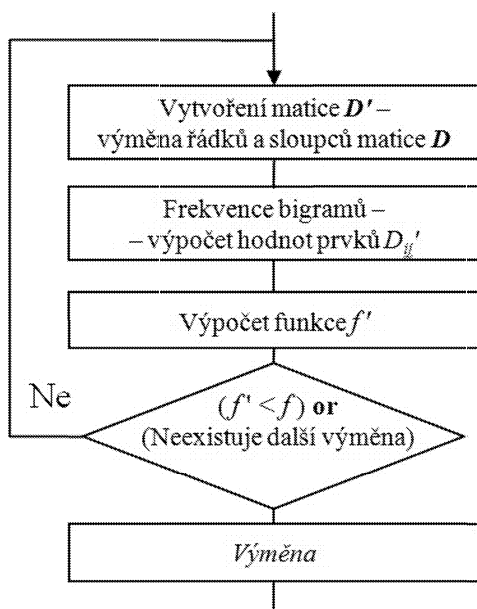
```

F = 0
For I = 1 To 26
  For J = 1 To 26
    Dij = Cells(I + 32, J + 1)
    Eij = Cells(I + 63, J + 1)
    F = F + Abs(Dij - Eij)
  Next J
Next I
Cells(4, 29) = F

```

Tato část programu se využije ještě jednou při výpočtu funkce f' .

Jak již bylo řečeno výše, matici D' dostaneme současnou záměnou dvou řádků a dvou sloupců matice D . Matice D' je uložena na listu Excelu v oblasti buněk B2:AA27. Proceduru *Bigramy* můžeme vyjádřit pomocí vývojového diagramu na obrázku 4.



Obr. 4 Algoritmus procedury *Bigramy*

Základem vývojového diagramu je cyklus s podmínkou na konci. V rámci cyklu se vytvoří matice $\mathbf{D}'$ současnou výměnou řádků a sloupců matice $\mathbf{D}$, výpočet frekvence bigramů matice $\mathbf{D}'$ (prvků matice D_{ij}) a výpočet funkce f' . Cyklus je ukončen, pokud je splněna podmínka $f' < f$.

Výměna

Procedura *Výměna* se provede po ukončení cyklu (obrázek 4). Pokud je cyklus ukončen podmínkou $f' < f$, procedura provede automatickou výměnu písmen šifrovaného textu v převodových tabulkách, tj. v řádcích 3, 7 a 10 na obrázku 3. Konkrétně, když bude cyklus ukončen u výměny písmen $2 \leftrightarrow 3$, v převodových tabulkách se prohodí písmena M a Z. Současně s výměnou se v lichých řádcích od řádku 17 dle obrázku 3 změní i písmena dešifrovaného textu dle aktuálně platné převodové tabulky.

Pokud je cyklus ukončen druhou podmínkou, tj. proběhlo všech 325 výměn a ani v jednom případě nebyla splněna podmínka $f' < f$ vyzve tato procedura uživatele k ručnímu zadání výměny.

Závěr

Popsaný program jsme testovali pomocí dalšího textu Karla Čapka, povídky *Modrá chryzantéma*, zašifrovaného předem monoalfabetickou substituční šifrou. Tento text má 2 483 písmen. Výše uvedený algoritmus dešifroval tento text po 7 955 provedených výměnách. U kratších textů (méně než 500 znaků) automatické dešifrování není stoprocentní, je třeba „ručního“ dokončení.

Literatura

- [1] *Jacobsen, T.*: A Fast Method for the Cryptanalysis of Substitution Ciphers, Cryptologia, Vol. 19, 1995, pp. 265–274.
- [2] *Hubálovský, Š.*: Strukturované programování ve VBA pro Excel. MFI, r. 20 (2010/11), č. 9.
- [3] *Musílek, M. – Hubálovský, Š.*: Počítačová bezpečnost ve výuce informatiky (3. část – luštění jednoduché záměny, frekvenční analýza). MFI, r. 20 (2010/11), č. 9.