

Bobřík informatiky 2009: analýza řešení soutěžních otázek

JIRÍ VANÍČEK

Pedagogická fakulta JU, České Budějovice



Úvod

V listopadu 2009 proběhl druhý ročník soutěže v informatice pro žáky základních a středních škol s názvem *Bobřík informatiky*. Soutěž, která si klade za cíl prohloubit a formovat zájem mládeže o informační a komuni-

kační technologie, byla mezitím registrována MŠMT mezi předmětovými soutěžemi. V národním kole soutěžilo více než 10 000 soutěžících, což mj. u některých soutěžních kategorií přineslo problémy technického rázu s pomalou odezvou soutěžního serveru. Soutěžilo se ve třech věkových kategoriích od žáků 5. tříd po maturanty a díky počtu zúčastněných získali organizátoři bohatý materiál, jehož analýzou je možno zjistit, které typy informatických úloh dělají českým žákům potíže a které jsou naopak nejméně problematické.

Článek seznamuje s výsledky analýzy těchto žákovských řešení; vybírá úlohy, které podle autora poskytly zajímavá nebo překvapivá zjištění, a rozebírá jejich možné příčiny. Závěry této analýzy může využít učitel informatiky v zaměření své výuky nebo při přípravě svých žáků na budoucí kola informatických soutěží.

Zdroj testových úloh

Základní informace o této soutěži byly publikovány v loňském čísle tohoto časopisu [1], případně je zájemce může zjistit na webu soutěže [2]. Proto jen stručně zopakujeme, že se jedná o online vědomostní test, složený z 15 otázek s výběrovými odpověďmi nebo interaktivních úloh. V interaktivních úlohách soutěžící manipulovali s objekty (označovali myši čísla nebo obrázky, případně s nimi tahali po ploše).

Úlohy do soutěže jsou vybírány z mezinárodní databáze úloh pro soutěže Bebras. Tyto úlohy každoročně připravují odborníci z institucí, pořádajících soutěže na národní úrovni. V květnu loňského roku proběhl mezinárodní workshop v Litvě za účasti zástupců 15 zemí, na němž byly všechny navržené úlohy recenzovány, schvalovány a upravovány. Máme tedy jistotu, že úlohy, které řeší naši soutěžící, jsou tematicky a kvalitou srovnatelné s úlohami v dalších evropských zemích.

Úlohy jsou vybírány rovnoměrně z následujících oblastí: D – digitální gramotnost (každodenní práce s počítačem, praktické a technické otázky), A – algoritmizace a programování, I – práce s daty (informační a datové struktury, reprezentace dat), M – řešení problémů a matematické základy informatiky. Podrobněji viz [3], str. 3. Podle našeho názoru je to právě výběr úloh, který ze soutěže činí zajímavou platformu pro srovnávání zaměření školní informatiky a také úspěšnosti účastníků v různých evropských zemích [4], což může mít dopad i na vytváření maturitního standardu v této vzdělávací oblasti.

Analýza výsledků

Nejprve uvedeme celkové výsledky loňského národního kola v tabulce po jednotlivých kategoriích. Dále bude následovat přehled nejobtížnějších a nejsnadnějších úloh, v nichž se pokusíme výsledky komentovat nebo vsadit do souvislostí.

Kategorie	průměrná úspěšnost	neřešené úlohy	počet soutěžících	absolutní vítězové
Benjamin (5. – 8. ZŠ)	49 %	12 %	4045	20
Junior (9. ZŠ – 2. SŠ)	36 %	16 %	4501	5
Senior (3. – 4. SŠ)	45 %	14 %	1805	10

Tabulka srovnání výsledků soutěže po kategoriích. Průměrná úspěšnost znamená, kolik procent odpovědí na všechny otázky bylo správných, neře-

šené úlohy = podíl úloh, které soutěžící přeskočili, nestihli nebo odmítli odpovědět, absolutní vítězové = počet soutěžících, kteří dosáhli plného počtu bodů.

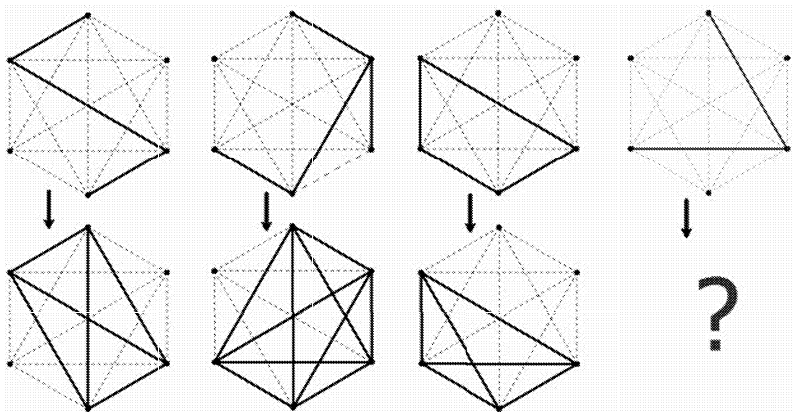
Z tabulky vyplývá, že nejtěžší kategorií byla kategorie Junior, což bylo patrně dáno obtížností vybraných úloh. K této úvaze nás vede fakt, že v otázce *Duhová vejce*, která byla pro všechny kategorie společná, byla s rostoucím věkem zaznamenána vyšší úspěšnost a menší podíl soutěžících, kteří na otázku neodpověděli.

Těžké otázky

V následujícím výčtu je v titulku vždy zapsán název otázky, kategorie, obtížnost a typ otázky (podle typologie uvedené výše). Správná odpověď je vyznačena tučně.

Diagramy – Ben – 2 – I (znamená kategorie Benjamin, obtížnost střední, typ – práce s daty)

Otázka:



Grafy v horním řádku byly doplněny o další čáry podle stejného pravidla a vznikly grafy na spodním řádku. Kolik čar je třeba doplnit v pravém grafu, doplňujeme-li podle stejného pravidla?

Odpovědi: 1, 2, 3, 5

Zdůvodnění správné odpovědi: Původním grafům se doplní všechny spojnice těch bodů, z nichž již nějaká spojnice vede. Mezi čtyřmi body v grafu vede maximálně 6 spojníc, mezi pěti body 10 spojníc (viz spodní

obrázky). Mezi třemi body vedou maximálně 3 spojnice. V grafu napravo jsou spojeny tři body dvěma spojnici, je potřeba dokreslit 1 čáru.

Komentář: Tuto otázku zodpovědělo pouze 27 % soutěžících správně, 21 % ji vynechalo. Odpovědi byly poměrně rovnoměrně rozděleny mezi všechny možnosti, z čehož vyplývá, že žáci si nevěděli s úlohou rady a dosti často volili odpověď náhodně.

Virus – Ben – 3 – M

Otázka: Virus se začal šířit po Internetu z jednoho počítače tak, že po celou dobu šíření se za každou další minutu nakazil dvojnásobek počítačů. Právě v tuto chvíli je nakaženo 1 % všech počítačů připojených k Internetu. Kolik času mají tvůrci antiviru na jeho dokončení, než budou nakaženy všechny počítače připojené k Internetu, pokud bude šíření viru pokračovat stejným tempem?

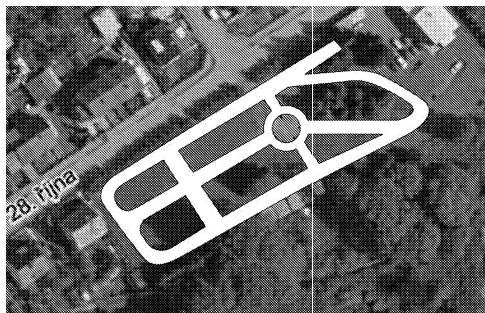
Odpovědi: **nanejvýš 7 minut**, asi 100 minut, více než 2 hodiny, asi 50 minut

Zdůvodnění správné odpovědi: Jestliže se každou minutou zdvojnásobí počet nakažených počítačů, za 1 minutu budou nakažena 2 %, za 2 minuty 4 % a za 6 minut to bude 64 %, tedy více než polovina. Je jisté, že než uplyne sedmá minuta, budou všechny počítače nakaženy.

Komentář: 35 % odpovědí bylo správných, 19 % žáků úlohu vynechalo. Je patrné, že chybějící matematický základ pro geometrickou posloupnost u této věkové kategorie měl dopad na řešení úlohy, i když podle zdůvodnění lze na řešení přijít „selským rozumem“. Značné množství žáků volilo odpověď 50 minut.

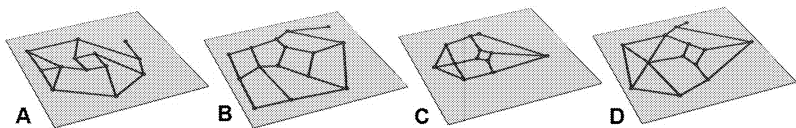
Dopravní hřiště – Jun – 2 – I

Otázka:



Na obrázku je dopravní hřiště v Jindřichově Hradci. Který z následujících grafů správně zobrazuje toto hřiště? – Úsečka grafu představuje ulici, bod představuje křižovatku nebo konec slepé ulice. Kruhový objezd není chápán jako jedna křižovatka.

Odpovědi:



Zdůvodnění správné odpovědi: Správně je A. Z fotografie vidíme, že

- hřiště má jednu slepou ulici (tomu v grafu odpovídá čára s volným koncem);
- hřiště obsahuje 12 křižovek a 1 konec slepé ulice, to je celkem 13 bodů grafu;
- do žádné křižovatky neústí více než čtyři ulice (ze žádného bodu grafu nemůže vycházet více než 4 čáry).

Graf B má místo třinácti 15 bodů. Graf C nemá žádnou čáru s volným koncem. Graf D obsahuje bod, z něhož vychází 5 čar. Jediný správný graf je A.

Komentář: Pouhých 8 % úspěšných, zato variantu B zvolilo 50 % odpovídajících. Je zřejmé, že žáci dali přednost vzhledové podobnosti grafu B s mapou před topologickou. Obrázky v testu byly barevné, tvar hřiště na mapě byl dobře patrný.

Zkopírování vzorce – Jun – 3 – D

Otázka:

fx =A1+2*\$E\$37				
	A	B	C	D
1	1	5		
2	3	3	69	
3	4	2		

V buňce C2 je napsán vzorec (viz obrázek). Jestliže zkopírujeme tuto buňku do buňky D3, jaké číslo bude v buňce D3?

Odpovědi: 71, 69, 73, „z informací, které mám z obrázku, to nelze určit“

Zdůvodnění správné odpovědi: Při kopírování se buňka posune o 1 řádek dolů a 1 sloupec vpravo. Podobně se změní i adresy zapsané ve vzorci, pokud nejsou uzamčeny znaky \$. Ke zjištění výsledku tedy nepotřebujeme znát hodnotu buňky E37, protože ta se nemění.

Vzorec = $A1+2*\$E\37 se kopírováním změní na = $B2+2*\$E\37 . První člen počítaného dvojčlenu se zvětší o 2 (místo 1 se dosadí 3), druhý člen se nezmění. Celkově se tedy celý dvojčlen zvětší o 2, správný výsledek je 71.

Komentář: Toto je úloha, která plně odpovídá školnímu kurikulu příslušné věkové kategorie. Přesto úlohu neřešila plná čtvrtina soutěžících a 12 % je nízká úspěšnost. 33 % žáků odpovědělo, že hodnota vypočítaná vzorcem se po kopírování nezmění; nemají tedy patrně základní představu o mechanismu změny vzorce při kopírování (a to nebereme v potaz uzamykání adres).

Žebřík – Sen – 3 – A

Otázka:

Program Logo kreslí obrázek tak, že kreslicí pero (zvané želva) „leze“ po obrazovce a podle příkazů kreslí:

- *dopředu* n – popoleze vpřed ve směru, do kterého je natočeno, o n pixelů;
- *vpravo* a – otočí se vpravo o a stupňů.

Např. *opaku* 5 [*dopředu* 50 *vpravo* 120] – zopakuje pětkrát tuto akci: popoleze o 50 pixelů a otočí se o 120 stupňů, tímto příkazem nakreslí trojúhelník.

Místo teček doplň chybějící číslo v příkazu:

opaku 10 [*opaku* 4 [*dopředu* 40 *vpravo* 90] *dopředu* ...]
tak, aby želva nakreslila žebřík na obrázku.

Odpovědi: 20, 30, 40, 90



START

Zdůvodnění správné odpovědi: Příkaz *opaku* 4 [*dopředu* 40 *vpravo* 90] nakreslí čtverec o straně 40 pixelů. Když tento příkaz nahradíme slovem *čtverec*, bude původní příkaz vypadat: *opaku* 10 [*čtverec dopředu* ...]. Aby se strany čtverců při překreslování překrývaly, musí se želva po nakreslení čtverce posunout o polovinu délky strany čtverce. Správně je 20.

Komentář: V této úloze byla úspěšnost pouhých 15 %, zato odpověď 40 vybralo 37 % žáků (patrně předpokládali, že želva musí popolezt o délku

strany čtverce, aby obrazce navazovaly). Více než čtvrtina soutěžících se odpovědi úplně vyhnula.

Souběžné výpočty – Sen – 3 – A

Otázka: Počítače mohou vykonávat více programů současně, a to i tehdy, když mají jediný procesor. Trik je v tom, že operační systém nechá chvíli běžet jeden program a pak zase druhý a takto to střídá. Běžící programy se nazývají procesy a operační systém mezi nimi přepíná.

Máme dvě proměnné X a A , obě mohou uchovávat čísla. Na začátku jsou X i A nastaveny na nulu.

Procesy vykonávají postupně příkazy:

<i>Proces 1</i>	<i>Proces 2</i>
zvětši X o 1	zkopíruj hodnotu X do A
zmenši X o 1	zkopíruj hodnotu A do X

Oba procesy startují a běží současně. To znamená, že operační systém může rozhodnout, který z procesů bude startovat jako první, a po dokončení příkazu napsaného v řádku může procesy přepnout. Co bude hodnotou X poté, co oba procesy skončí?

Odpovědi: **může být 0, – 1 nebo 1**; může být 0 nebo 1; vždy 0; může být 0, 1 nebo 2

Zdůvodnění správné odpovědi: Nabízí se provést úplný výčet všech kombinací spouštění obou procesů, zde předvedeme stručnější zdůvodnění: Při spouštění procesů v pořadí Proces2, Proces1, Proces2, Proces1 hodnota X nabude hodnoty -1 . Jestliže je právě jedna odpověď správná, musí to být ta, která obsahuje hodnotu -1 .

Komentář: Toto měla být podle organizátorů nejtěžší úloha, která rozhodne o vítězi v kategorii Senior. Nakonec ale o vítězích nerozhodla; všech 6 soutěžících, kteří měli pouze jednu špatnou odpověď, tuto otázku zodpovědělo správně (a je zajímavé, že všichni udělali chybu u jedné z interaktivních úloh, tedy u relativně lehkých úloh).

Nejlehčí úlohy

V předložském prvním ročníku jsme zařadili úlohu, která vyžadovala místo přemýšlení a znalosti pojmů rozhled ve sféře technologií spíše z uži-

vatelského či komerčního pohledu. Šlo o úlohu, v níž se má mezi jinými zkratkami správně vybrat technologie přenosu dat v síti mobilního operátora, GPRS. Tato úloha měla 92 % úspěšných odpovědí. Letos jsme podobný pokus zopakovali a pro kategorii Senior připravili úlohu, v níž měli studenti podle grafického loga rozpoznat, který produkt není operačním systémem (šlo o loga produktů Windows, Linux, Skype). Také letos to byla nejlehčí úloha v celé soutěži s 81 % správných odpovědí (nejčastější chybná odpověď byla, že Linux není operační systém). Vypadá to, že takovéto úlohy typu „já se vyznám“ jsou snadnější než úlohy na přemýšlení. Z reakcí některých učitelů i studentů však vyplynulo, že úlohu vnímají jako nesystémovou až nevhodnou, což organizátory potěšilo.

Dalším typem snadných úloh byly interaktivní úlohy, letos zařazené poprvé. Interaktivní úlohy měly vysokou účast (vesměs přes 90 %, v kategorii Benjamin interaktivní úlohy řešilo 100 % soutěžících) i dobrou úspěšnost, snad kromě úlohy, v níž měli žáci seskupovat do dvojic odpovídající si konektory. Zde mohla být příčinou nepříliš kvalitní fotografie a použití konektorů navzájem si podobných tvarů.

Nejlehčí úlohou v kategorii Benjamin byla úloha *Školní kroužky*, v níž měli žáci odečítat z grafu rozdíl mezi extrémními hodnotami v jedné datové řadě (73 % úspěšných) a algoritmická úloha *Razítkování*, jejímž úkolem bylo podle výsledného obrázku seřadit razítka podle toho, v jakém pořadí byla použita při jeho tvorbě (71 % úspěšných).

V kategorii Junior byly jako nejlehčí otázky vyhodnoceny obě interaktivní úlohy, *Zámek* (hlavolam s překlápěním 0 a 1 v matici znaků podle určitého pravidla) a *Výpočet* (sestavení početního výrazu s hledaným výsledkem přetahováním číselných objektů).

Závěrem se zmíníme o jedné úloze se společenskou tematikou.

Film na mobilu – Jun – 1 – D

Otázka: Maturantka Lucka dostala od kamarádky CD s filmem, který se právě hraje v kinech. Napadlo ji dát si ten film do mobilu. Stáhla si z Internetu volně šiřitelný (open source) program, který umí převést video do formátu, aby šlo na mobilu spustit. V tomto programu film převedla do formátu vhodného pro mobily a pak si jej na svůj mobil nahrála. Velice si pochvalovala, jak se jí to podařilo. Rozhodla se, že video umístí na svůj web, aby si ho mohly kamarádky stáhnout a také prohlédnout.

Nedopustila se Lucka něčeho nelegálního?

Odpovědi:

Šlo o trestněprávní čin a Lucka za něj může být pohnána před soud a být odsouzena.

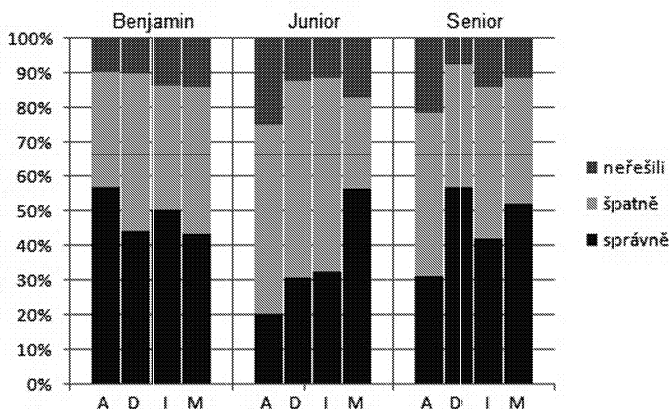
Nedopustila, protože stažený software byl volně šiřitelný a Lucka film neukradla.

No, asi by se to dělat nemělo, ale Luce za to nic nehrozí.

Kdyby Lucka nezveřejnila video na webu, ale posílala jej na mobily kamarádek přes Bluetooth, bylo by vše v pořádku

Komentář: Poměrně potěšitelné bylo, že tato úloha o problematice autorských práv měla nadprůměrnou úspěšnost 41 %, na druhou stranu méně než polovina soutěžících středoškoláků ví, co je a není legální. Nejčastější chybou byl výběr poslední z nabízených odpovědí (Bluetooth). Tento výsledek koresponduje s podobným průzkumem u patnáctiletých žáků, kdy se v problematice neorientovala polovina ze 40 respondentů.

Závěr



Úspěšnost podle jednotlivých typů úloh, řazeno po kategoriích. A – algoritmické úlohy, D – digitální gramotnost, I – práce s informacemi, M – řešení problémů

Z uvedeného přehledu i z analýz ostatních úloh vyplývá, že nejobtížněji se soutěžící vypořádávali s algoritmickými úlohami, kde s nízkou úspěšností koresponduje vysoký počet neřešených úloh. Na druhém místě pomyslného žebříčku skončily úlohy z oblasti práce s daty, jako je čtení z grafů, reprezentace dat, znázornění struktur a vztahů. Tato charakteristika ovšem

neplatí pro všechny kategorie, u žáků základních škol (kde podíl těchto témat ve školní výuce je jistě menší než na středních školách) je situace opačná.

Doufejme, že pro příští ročník soutěž vybědne z technických potíží a bude moci poskytnout všem zájemcům plný přístup k zajímavým informatickým úlohám a problémům.

Plné znění všech úloh, jež byly v textu článku zkráceny nebo pouze zmiňovány, najde čtenář na webu soutěže www.ibobr.cz [2]. V archivu testů je možné si soutěžní testy uplynulých ročníků znovu projít, využít testy ve vyučování, k dispozici jsou i komentáře k řešení všech soutěžních úloh.

L i t e r a t u r a

- [1] *Vaníček, J.*: Bobřík informatiky – soutěž žáků a studentů v informatice. Matematika – fyzika – informatika r. 18 č. 9, s. 548 – 558. ISSN 1210-1761
- [2] Bobřík informatiky, web soutěže [online]. Jihočeská univerzita, Pedagogická fakulta [cit. 12. 1. 2010] Dostupné na [www: <http://www.ibobr.cz>](http://www.ibobr.cz)
- [3] *Kalaš, I.*– *Tomcsányiová, M.*: Students' Attitude to Programming in Modern Informatics [online]. In: 9th IFIP WCCE 2009 – World Conference on Computers in Education, 2009. ISBN 978-3-901882-35-7. Dostupné na [www: <http://www.wcce2009.org/proceedings/papers/WCCE2009_pap82.pdf>](http://www.wcce2009.org/proceedings/papers/WCCE2009_pap82.pdf)
- [4] *Tomcsányi, P.* – *Vaníček, J.*: International comparison of problems from an informatics contest. In: Mechlová, E., Valchař, A. (eds.): Information and communication technology in education '09. Ostrava: Ostravská univerzita, 2008. s. 219 – 223. ISBN 80-7368-459-4

(Autorkou úvodní ilustrace je Mgr. Jaroslava Čermáková ze Suchých Lazců.)

Matematický program Sage

ROBERT MAŘÍK^{*)}

Ústav matematiky, Mendelova univerzita v Brně

Systémy počítačové algebry jsou zajímavým podpůrným prostředkem při výuce matematiky. V současnosti jsou učitelé a žáci středních škol většinou odkázáni buď na komerční programy Derive, Maple, Mathematica,

^{*)} Autor je podporován grantem FRVŠ 131/2010: Počítačová podpora výuky matematiky pomocí volně šiřitelného software.

MuPAD (nyní součást programu MATLAB), MathCAD a TI InterActive!, anebo na nepoměrně méně známé volně šiřitelné programy wxMaxima a Axiom (FriCAS). Cílem tohoto článku je představit další zajímavou alternativu: volně šiřitelný systém počítačové algebry Sage. Výhodou Sage je, že dokáže zdarma nabídnout stejné možnosti jako ostatní výše uvedené programy, je však možno jej používat bez nutnosti instalace na lokální počítač, pouze ve webovém prohlížeči. Díky tomu se jeví vhodný i pro nasazení v počítačem podporované výuce na školách všech stupňů.

Autor tohoto článku již několik let sleduje aktivně dění a vývoj programu Sage a protože reálné nasazení někdy bývá spojeno s jistými problémy či nejasnostmi, snaží se nejdůležitější své zkušenosti shrnout formou tohoto článku. Jedná se o informace pocházející částečně z vlastní zkušenosti a částečně z komunikace na diskusních fórech k programu Sage.¹ Nejprve si však program alespoň stručně představíme.

1. Představení programu Sage

Program Sage je vyvíjen od roku 2004 a na své domovské stránce www.sagemath.org je uveden motem „*vytvoření životaschopné volně šiřitelné alternativy k programům Magma, Maple, Mathematica a Matlab*“.

Program je velmi komplexní – kromě toho, že mnohé věci jsou v programu implementovány nativně (zejména kvůli rychlosti, v některých srovnávacích testech v oblasti teorie čísel Sage předčí komerční alternativy), obsahuje i rozhraní pro další programy a knihovny (program Octave jako volně šiřitelná alternativa Matlabu, program R pro statické výpočty, program Maxima pro manipulaci s algebraickými výrazy a další). Tím je také vysvětleno to, jak je možné, že se na scéně objevil tak rychle podobně obsáhlý software.

Pracovní prostředí programu Sage se zdá chudé kvůli jednoduchému menu a absenci panelů tlačítek známých z grafického prostředí programů Maple a Mathematica. Přesto je práce s programem Sage velmi příjemná, díky přítomnosti podrobně propracovaného systému nápovědy a doplňování příkazů. Například po spuštění příkazu

```
rovnice = x^2+2*x+1 == 0
```

¹<http://groups.google.com/group/sage-support/> (obecně zaměřené fórum), <http://groups.google.com/group/sage-edu/> (zaměřeno na výuku) a <http://groups.google.com/group/sage-devel/> (pro vývojáře)

SŠ – Sage - Mozilla Firefox

Soubor Úpravy Zobrazit Historie Záložky Nástroje nápověda

http://um-bc107.mendelu.cz:8000/home

maw.php System statistics acroweb.log e^xMAW printserver Xerox

Mendelova univerzita v Brně The Sage

robert.marik Toggle Home Published Log Settings Help Report a Problem Sign out

Notebook

Version 4.3.4

SŠ Save Save & quit Discard & quit

last edited on March 02, 2010 12:14 AM by robert.marik

File... Action... Data... sage Typeset

Print Worksheet Edit Text Undo Share Publish

```
implicit_multiplication(True)
automatic_names(True)
```

Budeme řešit kvadratickou rovnici $x^2 + 2x + 1 = 0$. Hledáme kořeny i s jejich násobností a bez použití vzorce $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

```
rovnice = x^2 + 2x + 1 == 0
rovnice
```

$$x^2 + 2x + 1 = 0$$

```
rovnice.solve(x, multiplicities = True)
```

$$([x = (-1)], [2])$$

```
rovnice.
```

evaluate

rovnice.ori	rovnice.full_simplify
rovnice.abs	rovnice.function
rovnice.add_to_both_sides	rovnice.gamma

Hotovo

se do proměnné rovnice uloží rovnice $x^2 + 2x + 1 = 0$. Zadáme-li v dalším vstupním poli

ro

a stiskneme tabelátor, rozbalí se menu s příkazy **round** a **rovnice**, protože tohle jsou jediné funkce a proměnné začínající tímto řetězcem. Napíšeme-li

rov

a stiskneme tabelátor, doplní se text na **rovnice**. Připíšeme-li tečku

rovnice.

a opět stiskneme tabelátor, obdržíme seznam operací, které je možno provádět s algebraickými výrazy a rovnicemi (horní část tohoto seznamu vidíme v dolní části Obrázku 1). Vybereme-li si například funkci **solve**, potom příkazem

rovnice.solve?

zobrazíme nápovědu k této funkci a ukázky použití. Odsud již není těžké zjistit, že příkaz

rovnice.solve(x, multiplicities=True)

slouží pro nalezení všech řešení, včetně násobnosti. Výstupem příkazu je tedy buďto

$([x == -1], [2])$

nebo

$([x = (-1)], [2]),$

podle toho, zda máme vypnuto či zapnuto vykreslování vzorců pluginem **jsMath**².

Tímto způsobem také například snadno prozkoumáme bohaté možnosti, které Sage nabízí při práci s obrázky a grafy funkcí, včetně velmi snadného kombinování více grafických objektů – to je totiž realizováno prostým sčítáním proměnných, ve kterých jsou tyto grafické objekty uloženy.

²<http://www.math.union.edu/~dpvc/jsMath/>

Sage pochopitelně umožňuje vkládat do zápisníku textové komentáře, které obsahují i matematické vzorce. Pro tuto funkci je využit editor **tinymce**³, který žáci a studenti zpravidla dobře znají z mnoha jiných internetových stránek.

Možná jste si pro demonstraci některých pojmů a vztahů ve výuce oblíbili webové applety nebo jejich Maplovskou obdobu, maplety. Tyto objekty umožňují komunikovat s programem ve zjednodušeném prostředí a zejména bez nutnosti ručně zadávat příkazy. Podobné objekty se v programu Sage nazývají sagelety a galerie ukázek sageletů je na webové stránce <http://wiki.sagemath.org/interact>⁴.

S pedagogickou prací v systémech počítačové algebry je nedílně spojeno vytváření vzorových zápisníků a jejich vystavování na Internetu nebo Intranetu. Toto je možno v případě programu Sage realizovat několika způsoby:

- zápisník je možno vytisknout do PDF souboru,
- zápisník je možno exportovat do **sws** souboru a uložit na lokální počítač. Tento soubor je později možno nahrát na jiný Sage server nebo zformátovat do pěkného PDF **L^AT_EX**em a programem **sws2tex**⁵.
- zápisník je možno „publikovat“ – návštěvník našeho serveru si může publikované zápisníky prohlížet a případě potřeby si může vytvořit jejich kopie, se kterými dále pracuje jako se svými zápisníky.

Mnozí učitelé matematiky si k psaní textů oblíbili program **L^AT_EX**. Díky balíčku **SageT_EX** je možné program Sage používat i uvnitř **L^AT_EX**ového dokumentu a díky programu **tex2sws**⁶ je možno **L^AT_EX**ový soubor převést přímo na zápisník programu Sage.

Velká výhoda programu Sage je, že je napsán v programovacím jazyce Python a tento jazyk je v Sage používán ve všech programových konstrukcích. Na rozdíl od komerčních systémů počítačové algebry studenti při používání cyklů nebo rozhodovacích podmínek používají živý a široce rozšířený programovací jazyk, namísto někdy krkolomných konstrukcí, specifických pro jeden jediný konkrétní programový produkt. To je důležité, protože i ve světě programů orientovaných na grafické uživatelské prostředí

³<http://tinymce.moxiecode.com/>

⁴Příspěvky a ukázky na wiki ovšem, na rozdíl od příspěvků do programového kódu programu Sage, nejsou recenzovány a v některých ukázkách se mohou vyskytnout chyby.

⁵<http://bitbucket.org/whuss/sws2tex/> a ukázky na <http://user.mendelu.cz/marik/sage/>

⁶<http://bitbucket.org/rbeezer/tex2sws/>

je stále výhodné využívat silné schopnosti skriptování nebo makroprogramování v jednotlivých aplikacích. A protože pro výuku psaní skriptů je Pyton skutečně velice dobrým kandidátem, je Sage vhodný zejména tam, kde se žáci se základy psaní skriptů seznámí během výuky informatiky.

2. Specifika programu z hlediska střední školy

Jak bylo zmíněno výše, program Sage má velmi široký záběr. Pro nasazení v prostředí základní či střední školy tato vlastnost může být nevýhodou. Například proměnná r , kterou většina uživatelů střední školy použije ve výpočtech pro poloměr kružnice či koule, je implicitně nastavena jako rozhraní pro statistický program R. Chceme-li používat r jako proměnnou, je nutno ji nadeklarovat následujícím příkazem.

```
r = var('r')
```

Jiný efekt souvisejícím s přílišnou komplexností programu Sage je následující: po uložení matice do proměnné a vyvolání nápovědy, jaké metody lze pro tento objekt použít, dostáváme příliš dlouhý seznam, obsahující položky, které jsou z hlediska střední školy nezajímavé (například transformace na Hessenbergův tvar⁷).

Nevýhodu přítomnosti příliš velkého počtu funkcí odstraňuje projekt SageForHighSchool⁸, iniciovaný francouzskými vývojáři. Tento projekt také deklaruje všechny jednopísmenné řetězce jako proměnné a odstraňuje (mimo jiné) výše uvedený problém proměnné r .

Pro začátečníky a mnohdy i pro uživatele jiných systémů počítačové algebry může být v programu Sage obtěžující nutnost deklarovat proměnné⁹, se kterými pracujeme, nutnost explicitního zápisu násobení pomocí znaku $*$, nebo skutečnost, že některé příkazy voláme pomocí funkce (objekt, parametry) a některé pomocí objektového zápisu objekt.funkce(parametry). „Nedostatky“¹⁰ popsané v tomto odstavci je možno odstranit následujícím nastavením.

```
implicit_multiplication(True)
```

⁷http://en.wikipedia.org/wiki/Hessenberg_matrix

⁸<http://wiki.sagemath.org/HighSchoolDesign> a <http://wiki.sagemath.org/SageForHighSchool>

⁹kromě proměnné x , která je zpřístupněna automaticky

¹⁰které se však z jiného úhlu pohledu jeví jako přednosti

`automatic_names(True)`

Toto nastavení je možno provést buď pro daný zápisník, nebo jej může administrátor programu Sage nastavit jako výchozí pro všechny zápisníky na příslušném serveru. Potíže ovšem budou stále přetrvávat u proměnných, které již sice jsou deklarovány, ale jinak, než uživatel potřebuje – viz například výše zmíněná proměnná r , kterou je i v tomto případě nutno předeclareovat příkazem `r=var('r')`, tak jak je to automaticky zařízeno i v projektu SageForHighSchool.

3. Specifika programu z hlediska výuky v počítačové laboratoři

S programem Sage je možno pracovat na všech majoritních operačních systémech a kromě možnosti instalace na svůj lokální počítač můžeme s tímto programem pracovat i vzdáleně prostřednictvím internetového prohlížeče na některém vlastním nebo veřejném volně dostupném serveru (viz níže).

Obsluha grafického rozhraní programu Sage je poměrně náročná na paměť serveru. Toto není příliš svazující, pokud máme program Sage spuštěný pouze pro sebe na svém lokálním počítači a „serverem“ je tedy tento lokální počítač. Paměť 512MB, která byla standardem před několika lety, je pro pohodlnou práci zcela dostačující.

Horší situace však nastane, pokud se rozhodneme program spustit na centrálním serveru a z pracovních stanic se všemi žáky současně přistupovat na tento server. Aby práce nebyla zpomalována swapováním serveru, je nutno pro cca 20 současných uživatelů počítat alespoň s 3GB paměti pro program Sage. Pokud na serveru není spuštěno nic jiného, znamená to počítat s cca 4GB paměti celkem. Z bezpečnostních důvodů je rizikem provozovat program Sage na serveru, který obsahuje další důležité služby nebo data. V praxi to většinou znamená provozování uvnitř virtuálního serveru, což dále poněkud zvyšuje nároky na paměť fyzického stroje. Rozhodně tedy není v případě serveru pro program Sage možné postupovat podobně jako u webových serverů nebo printserverů, kde často stačí starší počítač, na kterém již není možno pohodlně provozovat moderní operační systémy. Jaké mohou tedy být modely práce programem Sage v počítačové laboratoři?

- Práce na dostatečně dimenzovaném vlastním serveru – optimální, avšak nejdražší varianta.

- Práce na některém dostatečně dimenzovaném veřejném serveru¹¹ – existuje riziko, že se během výuky objeví problémy s dostupností služby.
- Každý student pracuje s programem Sage spuštěným na jeho vlastní pracovní stanici – způsob nejnáročnější na správu, poskytuje však zpravidla nejrychlejší odezvu při práci.

O posledním způsobu práce pojednáme poněkud podrobněji. Podpůrné programy pro Sage si vynucují instalaci na Linuxových operačních systémech (Linux, MacOS, Solaris). Pokud máme k dispozici učebnu s Linuxovými pracovními stanicemi a centrálním datovým úložištěm, máme vyhráno. Stačí nainstalovat program Sage na jeden sdílený diskový prostor, tento prostor připojit k pracovním stanicím a spouštět Sage z těchto pracovních stanic. Protože základní adresářová struktura je používána jenom ke čtení, nemůže dojít ke kolizi.

Poněkud složitější situace je v případě učebny s počítači výhradně na platformě Windows. Zde je možno program Sage provozovat dvěma způsoby. Buď necháme nabootovat pracovní stanice z Live CD s programem Sage, nebo v každé pracovní stanici vytvoříme virtuální počítač, kde bude spuštěn program Sage. K tomuto virtuálnímu počítači se již připojujeme klasicky internetovým prohlížečem z Windows.

Každá verze programu Sage je vydávána ve formě zdrojových kódů, ve formě zkompilevané na nejdůležitějších vybraných operačních systémech a platformách a v neposlední řadě ve formě virtuálního počítače pro program VMWare. Díky tomu ani instalace na Windows nevyžaduje od administrátora zkušenosti s provozováním operačních systémů Linuxového typu, protože instalace programu VMWare Player a spuštění virtuálního počítače se příliš neliší například od instalace většího textového editoru a následného otevření textového dokumentu v tomto editoru. Nevýhodou je pouze velký objem stahovaných dat.

4. Závěrečné informace

Zájemcům o systémy počítačové algebry určitě doporučuji systém Sage alespoň vyzkoušet na výše uvedených veřejných serverech¹². Další informace o programu v češtině je možno čerpat v bakalářské práci pana Martina Haupta *Systém počítačové algebry Sage* (online na <http://is.mu->

¹¹<http://www.sagenb.org>, <http://sagenb.kaist.ac.kr>

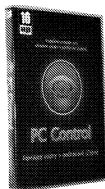
¹²Server <http://sagenb.kaist.ac.kr> je méně známý a proto vykazuje větší nou menší provoz a pohodlnější práci než sice výkonnější, ale více navštěvovaný, <http://www.sagenb.org>.

ni.cz/th/208258 a <http://www.math.muni.cz/~plch/diplomky/Sage.pdf>), která je nicméně v některých ohledech již zastaralá. Další informace jsou dostupné v anglické dokumentaci šířené přímo s programem a dostupné ze zápisníku programu Sage (viz Obrázek 1 a položka „Help“ v horním menu). Zde doporučuji zejména kratičký text vysvětlující ovládání a dále „Tutorial“ a pro případné detailní informace „Reference manual“. Začátečníkům příliš nedoporučuji část „Constructions“, která obsahuje často spíše netypické konstrukce programu Sage a je také v mnoha ohledech zastaralá. Dokumentace je opět jakousi formou zápisníku programu Sage a všechny ukázky v dokumentaci je tedy možno okamžitě zkoušet nebo modifikovat. Protože jsou tyto zápisníky poměrně obsáhlé, je možno v případě pomalé odezvy hledat informace ve statických verzích těchto textů, které jsou dostupné ve formátech html a PDF.

Během července 2010 došlo k zahájení prací na internacionalizaci programu Sage a během několika měsíců je tedy již reálné očekávat možnost práce v českém prostředí. Podrobnosti týkající se internacionalizace programu Sage a lokalizace do jednotlivých jazyků je možno sledovat na adrese <http://wiki.sagemath.org/i18n>.

ZKUŠENOSTI

PC Control



Každý učitel, který někdy vedl hodinu v počítačové učebně, ten problém dobře zná. Zatímco přednáší, vždy se najde několik studentů, které nabídka internetové

zábavy láká víc, než probírané téma: hrají hry, komunikují na sociálních sítích nebo jen bezcílně „surfuji“. Tým softwarových autorů proto vytvořil speciální program, který pedagogům v počítačových učebnách dnes běžně využívaných nejen k výuce IT, ale i mnoha dalším předmětům, umožní z jediného místa kompletně kontrolovat a ovládat všechny ostatní počítače v celé učebně. Současně však tento program nazvaný PC Control nabízí také dlouhou řadu praktických nástrojů, které zjednoduší, výrazně zrychlí a zefektivní jakoukoliv výuku v počítačové učebně.

Unikátní je na programu také to, že ho vytvořil tým autorů složený z trojice středoškolských studentů, úspěšných účastníků programátorských soutěží a olympiád, pod vedením svého bývalého učitele *Libora Kubese* z olomoucké Základní školy