



MAGAZÍN

KATEDRY INFORMATIKY

číslo 5 | červen 2016

Univerzita Palackého v Olomouci

Úvodní slovo

Vážení čtenáři,

tímto, v pořadí již pátým a poněkud obsáhlejším vydáním našeho magazínu, bychom se s Vámi opět chtěli podělit o aktuální informace z katedry informatiky a ze světa informatiky vůbec.

Nejdříve zavzpomínáme na nedávno zesnulého profesora Klira, který spolupracoval s členy naší katedry, a na jehož počest bude naše katedra udělovat stipendium pro nejlepší studenty jednotlivých oborů. Připomeneme si také našeho kolegu Jirku Hronka a běžecký závod nesoucí jeho jméno, který se pomalu stává významnou sportovní událostí.

Hlavním tématem tohoto vydání je informatika jako vědní obor. Možná se také občas setkáváte s dotazy, co to vlastně je ta informatika, co informatik umí, či proč vůbec informatiku studovat, když s počítačem si dnes rozumí i děti na základní škole. Odpovědi na tyto a další otázky nastínil v širších souvislostech profesor Bělohávek na stránkách časopisu Matematika – Fyzika – Informatika určeném středoškolským studentům a učitelům. Jelikož tento článek představuje zajímavý pohled na informatiku jako vědní disciplínu, na její úlohu, možnosti a důvody, proč se informatice věnovat, věříme, že zaujme i Vás, a proto jsme se jej rozhodli přetisknout i v našem magazínu.

Závěrem se jako obvykle dostane na hádanku, která tentokrát prověří Vaše znalosti bitových operací.

Za všechny, kteří se na přípravě tohoto magazínu podíleli, Vám příjemné čtení přeje

Petr Krajča

redaktor Magazínu Katedry informatiky Univerzity Palackého v Olomouci



OBSAH

- 2 **KATEDRA**
Vzpomínka na profesora George J. Klira
Opustila nás významná osobnost
- 3 **KATEDRA**
Klirovo stipendium
O novém stipendijním programu
- 4 **KATEDRA**
Třetí ročník Memoriálu Jiřího Hronka
Sportovní událost pořádaná naší katedrou
- 5 **TÉMA**
Informatika jako obor
Zamyšlení o informatice
- 15 **HÁDANKA**
Záhadná funkce
Co počítá neznámá funkce?

Vzpomínka na profesora George J. Klira

Radim Bělohávek

Dne 27. května 2016 zemřel ve věku 84 let profesor George J. Klir. Řada z nás ho měla možnost osobně poznat, mj. během jeho několika návštěv v Olomouci. Kdo ho znal blíže, poznal ho jako pozorného a laskavého kolegu a učitele.

Profesor Klir vždy žil vědou. Do její historie se významně zapsal svými vědeckými pracemi, úspěšnými knihami a učebnicemi v inženýrských, matematických a informatických oblastech i organizační činností. Je autorem více než 300 odborných prací zveřejněných v mezinárodních časopisech a sbornících mezinárodních konferencí, 20 knih publikovaných předními nakladatelstvími a také několika patentů. Zabýval se postupně několika vědeckými oblastmi. Začínal u zakladatele české školy počítačů profesora Antonína Svobody, pod jehož vedením se věnoval návrhu počítačů. Poté se dlouhé roky věnoval teorii systémů. V posledních 30 letech se hlavním bodem jeho zájmu stala neurčitost, teorie informace a fuzzy logika.

Působil jako prezident čtyř mezinárodních organizací: Society for General Systems Research, International Federation for Systems Research, North American Fuzzy Information Processing Society a International Fuzzy Systems Association. V roce 1974 založil a pak přes 40 let jako šéfredaktor vedl časopis International Journal of General Systems. Jeho výzkum byl podporován řadou organizací, mj. National Science Foundation, U. S. Office of Naval Research, U.S. Air Force, NASA, NATO, Sandia Laboratories a některými firmami. Za svůj přínos vědě získal mnohá ocenění včetně několika čestných doktorátů a prestižní Bolzanovy medaile Akademie věd České republiky.

Byl člověkem vysokého morálního standardu. Činil to, co považoval za správné. To ho přinutilo opusit v še-



desátých letech Československo. Nakonec se usadil v Binghamtonu ve státě New York, kde prožil více než čtyřicet let. Mnoha lidem pomohl v jejich vědecké kariéře. I po odchodu do důchodu v roce 2007 zůstal na poli vědy aktivní.

S profesorem Klirem jsem měl možnost spolupracovat od roku 1999, kdy mě do Binghamtonu pozval na 15měsíční pobyt. V Binghamtonu jsem nakonec strávil více než pět let. Binghamton a profesora Klira později poznali i někteří další členové naší katedry. Obdivovali jsme jeho psychickou i fyzickou kondici a neutuchající optimismus. Setkání s ním byla inspirativní a nabíjela člověka dobrou náladou.

Zemřel nečekaně, v plné síle. Jeho odchod je velkou ztrátou pro jeho ženu Milenu, jejich dceru a další jeho blízké. Je také ztrátou pro vědeckou obec.

Klirovo stipendium

Eduard Bartl

Již delší dobu připravujeme pro studenty bakalářských a magisterských prezenčních oborů vyučovaných na katedře informatiky možnost získání mimořádného stipendia. Bude se jednat o stipendium prospěchové, udělováno tedy bude za vynikající studijní výsledky. Nedávná smutná událost – úmrtí profesora G. J. Klira – iniciovala nápad pojmenovat toto ocenění Klirovo stipendium.

Spojení osoby profesora Klira s názvem stipendia má hlubší význam. Jak bylo uvedeno v předchozím článku, profesor Klir byl významným vědcem, který spolupracoval s některými členy katedry informatiky. Kromě vědecké a organizační činnosti se také intenzivně věnoval výuce a svým studentům. Bylo mi ctí, že jsem mohl na podzim roku 2007 absolvovat kurz Introduction to Systems Science, který na Binghamton University přednášel těsně před svým odchodem do důchodu. Zodpovědně tedy mohu říci, že jeho přednášky byly poutavé, písanky obtížné a jeho přístup ke studentům nadobytě příjemný a vstřícný. Velmi oceňoval píli studentů a trápil ho odliv nadaných studentů z technických a přírodovědných oborů na amerických univerzitách. Název stipendia po tak významném odborníkovi s vynikajícími pedagogickými schopnostmi by měl tomuto ocenění přinést určitou prestiž.

Klirovo stipendium bude vypláceno každý semestr dvěma nejlepším studentům podle prospěchu v každém ročníku standardní doby studia napříč všemi obory (prezenční bakalářské obory Aplikovaná informatika, Informatika a Informatika pro vzdělávání, navazující magisterské obory Aplikovaná informatika, Informatika a Učitelství informatiky pro střední školy). Celkem tedy bude každý semestr odměněno deset nejlepších studentů.

Finanční prostředky na vyplácení stipendia budeme získávat zejména od firem, se kterými dlouhodobě spolupracujeme. V současnosti jednáme s několika společnostmi o jejich podpoře a jsme otevřeni novým nabídkám. Klirovo stipendium bude poprvé uděleno na konci zimního semestru akademického roku 2016/2017.



Třetí ročník Memoriálu Jiřího Hronka

Radim Bělohávek

Asi před šesti lety jsme se s Jirkou Hronkem bavili o tom, že bychom mohli jako jednu z akcí univerzitního sportovního dne, který se koná vždy v polovině května, uspořádat běh. Vymýšleli jsme trať, měla vést podél Moravy a měla mít dvě varianty, kratší a delší. Stejnou debatu jsme pak měli další rok i ten rok následující. Na jaře 2014 ale Jirka – vášnivý amatérský sportovec – náhle zemřel. Toho roku jsme tedy běh uspořádali a pojmenovali ho po Jirkovi. Trať je jiná, běží se olomouckými parky, i vzdálenosti jsou jiné. Myšlenka je ale stejná – dát běžcům příležitost se proběhnout, trochu poměřit síly a někoho třeba i k běhu přivést.

Dne 11. května 2016 proběhl už třetí ročník Memoriálu Jiřího Hronka. Počty zúčastněných (zapsaných) v jednotlivých letech byly následující: 2014: 48 (68), 2015: 81 (104), 2016: 75 (100). Tyto počty nás velmi těší. V roce 2014 jsme zařadili tratě 5 km, 10 km, 20 km a 30 km; poslední trať nese název „Jirkova výzva“. K nim od roku 2015 přibýly tratě 16 km a maratonská trať 42 km. V roce 2015 i 2016 byl memoriálový maraton jediným maratonem v Olomouci. V obou těchto letech byl maraton po 5 km a 10 km třetí nejobsazenější tratí. Na memoriálu jde hlavně o to si zaběhat, nepadají tady

rekordní časy. Za pozornost ale některé výkony stojí. Třeba vítězný maratonský čas v letošním roce je 3 hodiny 14 minut. Jeden rekord ale přece nejspíš držíme: Věkový rozdíl nejstaršího a nejmladšího účastníka byl letos 91 let.

Máme v plánu příští rok informace o závodě ještě lépe rozšířit a trochu i vylepšit zázemí pro závodníky. Memoriálu se daří a patří mezi nejpoblábnější události Sportovního dne Univerzity Palackého. Poděkování patří všem, kdo odvádějí mravenčí pořadatelskou práci. Jirka by měl radost.



Informatika jako obor

Radim Bělohlávek

Proč se zabývat otázkou, co je informatika?

Denně čteme a slyšíme o tom, že žijeme v informační společnosti, že dnešnímu světu vládnou informační technologie (IT) a že bez jejich znalosti se nelze obejít.¹ Ve firmách i ve státních organizacích existují oddělení informatiky nebo IT, v letech 2003–2007 v České republice dokonce existovalo Ministerstvo informatiky. Předměty s informatickou náplní se staly součástí učiva na základních a středních školách.

Všeobecně rozšířená představa o tom, co informatika vlastně je, je ale chybná. Tuto představu bohužel velmi často získávají i žáci základních a studenti středních škol. Základní důvody jsou přitom prosté: za prvé, nesprávnou představu mají jejich učitelé, kteří ve většině případů informatiku nevystudovali; za druhé, výuka vychází ze špatných učebních osnov.² Žáci a studenti jsou tímto stavem pochopitelně ochuzeni. V mnoha evropských zemích včetně České republiky se pro výuku informatiky na základních a středních školách chystají změny.³ Zatímco takové změny jsou během na dlouhou trať, pokusit se uvést nesprávnou představu o informatice na správnou míru lze hned. Pokusíme se o to v tomto článku.

Článek je určen zejména středoškolským studentům a učitelům. Studentům může pomoci v rozhodování

vání o dalším studiu. Zatímco představa o podstatě matematiky, fyziky nebo chemie získaná na střední škole v zásadě umožňuje kompetentně se rozhodnout, zda tento obor studovat dále, případný zájemce o vysokoškolské studium informatiky je na tom hůř – střední škola mu totiž mnohdy nabízí představu matoucí. Učitelům, zvláště těm, kteří informatiku nestudovali, snad článek poskytne osvětlující pohled na informatiku jako obor, dá chuť dovědět se o některých oblastech více a třeba i začít vyučovat informatiku jinak.

Chceme ukázat, že informatika je samostatný vědní obor, který má kromě číselných, viditelných a dnes a denně všemi využívaných aplikací také – jako jiné obory – své hodnotné teoretické jádro, a že povědomí o tomto jádře – stejně jako povědomí o elementárních fyzikálních, chemických, matematických či ekonomických principech a zákonech – člověku umožňuje lépe se orientovat ve světě a zvládat problémy, se kterými se potýká.

Co informatika není

Informatik je podle rozšířené představy ten, kdo „rozumí počítačům, vyzná se ve všech těch programech a aplikacích, hlavně od Microsoftu, jako třeba Word, Excel a Internet Explorer, umí je nainstalovat,

¹Zajímavým aspektem zmíněné nabídnuté představy o dnešní společnosti jako společnosti informační, popřípadě o dnešní době jako o éře informací, je – přes nesporný zásadní význam, který informační technologie v dnešní společnosti mají – mnohdy zcela nekritické přijímání této představy i představ podobných, jako je představa o tzv. znalostní společnosti. Jako by informace a znalosti dříve zásadní roli neměly.

²Státní kurikulární dokumenty, tj. dokumenty předepisující obsah vzdělávacích oblastí a oborů v jednotlivých etapách vzdělávání, jsou pro informatiku neuspokojivé. Nejde o počty hodin těchto předmětů, ale o jejich náplň. Jako symbolický doklad uvedme, že v současném rámcovém vzdělávacím programu pro gymnázia [7] je vzdělávací obor „Informatika a informační a komunikační technologie“ uveden za obory „Výtvarný obor“, „Výchova ke zdraví“ a „Tělesná výchova“. V podobně neutěšené situaci je i ekonomie.

³Situace v jednotlivých evropských zemích analyzuje studie [1]. Vláda ČR přijala 12. listopadu 2014 jako usnesení vlády ČR č. 927/2014 strategii digitálního vzdělávání [8].

umí spravit, když něco na počítači nefunguje, umí připojit počítač k síti, rozšířit mu paměť, když je pomalý, zprovoznit tiskárnu, vyzná se v internetu, umí vytvářet webové stránky, ví, jak synchronizovat počítač a mobil s cloudem, umí třeba i pracovat s databázemi ...a ještě spoustu dalších věcí“. Vyzná se prostě v informačních technologiích, je to tedy „ajták“. Patří k němu i to, že pořád sedí u počítače, což dělat musí, protože IT se tak rychle vyvíjí, že je pořád něco nového a úplně jiného. Sledovat ten vývoj je tedy jediná možnost, jak se v tom vyznat.⁴ Tuto nesprávnou představu dobře ilustruje následující rozhovor mezi prarodiči a vnukem z populárního seriálu Ulice, který vysílá TV Nova:⁵

Babička: „Kdybys šel na informatiku, tak ji budeš dělat levou zadní. Počítače máš přece v malíku.“

Vnuk: „Hm, přesně, tohle je jenom ztráta času. Navíc, ty věci jdou tak rychle dopředu, že to, co se voni dneska naučej, už je za dva roky zastaralý.“

Babička: „No jestli je to tak, tak co tam pak lidi studují pět let?“

Děda: „A neměly by se vysoké školy zavřít, když jsou tak zbytečné?“

Vnuk: „Ale já neříkám, že jsou zbytečné. Pro někoho jsou třeba užitečné.“

Děda: „Ale pro tebe ne, že?“

Vnuk: „Nevím, no, teď mám pocit, že ne.“

Co tedy informatika je?

Definice informatiky

⁴Výše popsaná představa vychází z toho, že většina lidí sama s počítačem a IT pracuje, a setkává se tedy například s prodejci počítačů, správci počítačů nebo správci sítí. Nechceme zde nijak snižovat tyto náročné profese. Odráží však jen zlomek toho, co informatika představuje. Je to podobné, jako kdyby člověk na základě toho, že zná, co dělá ekonomické oddělení jeho zaměstnavatele, odvodil, že ekonomie je vědou o účetnictví.

⁵ 2975. díl, 19. února 2016. Děkuji paní Aleně Kolovrátníkové, že mě na něj upozornila a dala mi jeho přepis.

⁶ Americký informatik a bývalý prezident hlavní informatické společnosti Association for Computing Machinery (ACM).

⁷ Jako první tento termín použil německý průkopník informatiky Karl Steinbuch v článku „Informatik: Automatische Informationsverarbeitung“, který vyšel v roce 1957 v SEG-Nachrichten, Heft 4.

Informatika je obor mladý a k tomu navíc značně různorodý. Pokusíme-li se ho charakterizovat stručnou definicí, bude to nutně definice abstraktní, a tedy možná hůře srozumitelná. Pokusíme-li se ho charakterizovat výčtem oblastí nebo problémů, kterými se zabývá, bude to výčet poměrně dlouhý, navíc vzhledem k mládí tohoto oboru a jeho stále ještě rychlému vývoji výčet nestabilní (objevují se nové problémy, členění do oblastí se mění). Přesto se pokusíme o obojí. Začneme stručnou definicí, která míří k jádru věci:

Informatika se zabývá studiem procesů zpracovávajících informace, jejich teoretickými základy, analýzou, návrhem, efektivitou, implementací a aplikacemi, ať už jde o informace uložené ve formě bitů v paměti počítače, nacházející se v dokumentech na internetu nebo zapsané v genech živých organismů. Základní otázkou, která se promítá do všech oblastí informatiky, je: Co vše lze efektivně mechanicky spočítat?

Existuje celá řada jiných, více či méně podobných definicí. Ta naše vychází z klasického článku [4]. Za pozornost stojí také [6], zřejmě první článek, který se pokusil informatiku definovat, a dále pak celá řada zajímavých článků o informatice jako oboru a profesi, viz například [2, 3], jejichž autorem nebo spoluautorem je Peter J. Denning.⁶

V angličtině se pro pojem informatika používají dva termíny: „computer science“, který je používán zejména v Severní Americe, a „informatics“, který je běžněji užíván v evropských zemích, protože – stejně jako v češtině – je bližší místně používanému termínu („Informatik“ v němčině, „informatique“ ve francouzštině, „informatica“ v italštině, „informática“ ve španělštině, „informatyka“ v polštině).⁷ Doslovný překlad termínu „computer science“, tedy „věda o počítačích“, nás

přivádí k důležité otázce: jaký je vztah informatiky a počítačů?

Už v práci [6] autoři správně poznamenali, že chceme-li uvažovat o „computer science“ jako o oboru, který studuje počítače, je třeba „počítač“ chápat jako „žijící počítač“, tj. nejen hardware, ale i počítačové programy, algoritmy a vše, co s tím souvisí. Spíše než o počítače jde tedy o fenomény, které počítače doprovázejí a které je díky nim možné studovat. Popisovaný vztah pěkně vyjadřuje citát, který je připisován jednomu z průkopníků informatiky Edgaru Dijkstrovi:

*Informatika není o počítačích o nic víc,
než je astronomie o dalekohledech.⁸*

Počítač je pro informatiku zejména nástrojem, i když samozřejmě nástrojem nezbytným a zásadním. Právě na počítačích se totiž implementují a realizují zmíněné procesy zpracovávající informace, které informatika studuje. Prvořadé jsou ale ty procesy, ne počítače.

Jak informatici pracují

Počítače konstruují a jejich návrhem se zabývají elektroinženýři. Je ovšem pravdou, že poznatky informatiky návrh počítačů ovlivňují (procesory se navrhují tak, aby určité typy výpočtů byly rychlé) a naopak, že souboré možnosti počítačů ovlivňují metody zpracování informací, kterými se informatika zabývá (zkoumají se ty metody, které jsou na dostupných počítačích realizovatelné). Ostrá hranice mezi informatikou a elektroinženýrstvím neexistuje. Někteří by řekli, že návrh počítačů a hardware je jednou z oblastí informatiky, někteří, že to už není informatika, ale elektroinženýrství. Většina by se však shodla na tom, že to je hraniční oblast, kde se informatika s elektroinženýrstvím prolíná.

Oblastí, které představují prolínání informatiky s jinými obory, je celá řada. Tak například s matematikou se informatika prolíná ve výpočetní matematice, při studiu konečných a diskrétních struktur nebo v logice a automatickém dokazování, s fyzikou v oblasti kvantových

výpočtů a kvantové kryptografie, s chemií v oblasti výpočetní chemie a počítačového modelování chemických procesů, s biologii v oblasti bioinformatiky a biologicky inspirovaných výpočetních procesů, s psychologií v oblasti umělé inteligence a HCI (human-computer interaction, interakce člověka a počítače) a mohli bychom uvést mnoho dalších příkladů. Typické také je, že v mnohých z těchto oblastí se prolíná oborů více. Například v umělé inteligenci je to kromě informatiky a psychologie podstatnou měrou také matematika a elektroinženýrství. Tyto oblasti prolínání existují i u jiných oborů, než je informatika (prolíná se například matematika a fyzika, fyzika a chemie, matematika a psychologie nebo ekonomie a psychologie). Pro informatiku jsou ale takové oblasti prolínání typické a četné, byť tyto oblasti informatiku zdaleka nevyčerpávají.

Stejně jako prolínání matematiky s fyzikou neznamená, že by fyzika byla jen odvětvím matematiky nebo naopak, neznamená prolínání informatiky s matematikou, že by informatika byla odvětvím matematiky.⁹ S touto představou se nicméně setkáváme, stejně jako s představou, že informatika je odvětvím elektroinženýrství. Obě prameny z toho, že v počátcích informatiku rozvíjeli lidé, kteří byli původem zejména matematici nebo elektroinženýři, a že teprve postupem času se informatika etablovala jako samostatný obor. Dnešní informatici – výzkumníci i lidé z praxe – jsou již většinou absolventy informatických studijních oborů a takto neuvažují. Informatika je prostě jejich oborem a za samostatný obor ji samozřejmě považují. Setrvačnost a také nedostatek povědomí o tom, čím se vlastně informatika zabývá, však vedou k tomu, že představa o informatice jako odnoži matematiky nebo elektroinženýrství u neinformatiků zatím přetrvává.

Jak tedy informatika jako obor vypadá? Jakému z klasických oborů se nejvíc podobá? Každý má základní představu o tom, že matematikova práce spočívá zejména v tom, že definuje pojmy (například prvočíslo, kvadratická rovnice apod.) a o těchto pojmech dokazuje teoremy (tj. matematická tvrzení, například že prvočísel je nekonečně mnoho nebo že řešení kvadratické rovnice je dáno určitým vzorcem). Fyzik, chemik nebo bio-

⁸Computer science is no more about computers than astronomy is about telescopes. Dijkstra je známý i jinými podobně trefnými výroky, například: Nemusím ztrácet svůj čas prací u počítače jen proto, že jsem informatik (I don't need to waste my time with a computer just because I am a computer scientist); překlady vlastní.

⁹Poznamenejme, že toto prolínání je „obousměrné“ a nespočívá jen v tom, že informatika využívá matematické metody. Například v numerické matematice se metody informatiky používají při řešení matematických problémů.

log pozoruje přírodu (každý na jiné úrovni), formuluje o chování přírody hypotézy a tyto hypotézy pak experimentálně ověřuje. Inženýři pak zejména navrhují a vyvíjejí umělé systémy (televize, letadla, domy, komunikační sítě), tyto systémy konstruuji, analyzují a uvádějí do provozu. To jsou určující rysy tří základních vědeckých paradigmat – matematického, přírodovědeckého a inženýrského. V informatice nalézáme kombinaci těchto paradigmat. Některé oblasti mají – stylem a metodami práce – povahu matematiky: teoretici informatici definují pojmy a dokazují teoremy; tyto teoremy ale nejsou jen o obvyklých matematických pojmech jako celé číslo, kvadratická funkce nebo hyperbola, ale o přesných, formálně definovaných pojmech informatiky jako je časová složitost výpočtu, uvážnutí v distribuovaném systému nebo suboptimální řešení problému. Některé oblasti mají povahu inženýrskou: návrh operačních systémů, vývoj softwarových aplikací a softwarové inženýrství, návrh komunikačních protokolů pro počítačové sítě. Některé mají povahu přírodních věd: experimentální algoritmika formuluje a ověřuje hypotézy o chování složitých algoritmů, které nelze analyzovat jinak než pomocí experimentů; vědci v umělé inteligenci se zabývají hypotézami o tom, jak řeší problémy člověk, a testují je na počítači; testují se hypotézy o tom, jak funguje životní cyklus softwaru a jak vznikají chyby. Dalo by se říci, že metody informatiky jsou směsí metodických přístupů matematických, inženýrských a přírodovědeckých. Ne každý informatik všechny tyto přístupy uplatňuje. Například teoretici informatici si velmi často vystačí s tužkou a papírem, a tedy neexperimentují ani nepoužívají inženýrské metody; vývojáři software naopak téměř nikdy nedokazují teoremy. Tedy:

Informatika používá metody matematické, inženýrské i přírodovědecké. V některých oblastech převažují matematické, v jiných inženýrské, v některých přírodovědecké. Typická pro informatiku je kombinace těchto přístupů.

Existuje ale nějaký obecný rys společný všem oblastem informatiky? Rys, který by informatiku dobře charakterizoval? Vzhledem k rozmanitosti informatiky to

je obtížná otázka. Jednu možnou odpověď dává hnutí *Great Principles of Computing* Petera Denninga, které definuje několik základních principů společných různým oblastem informatiky.¹⁰

Algoritmus jako základní pojem

Druhá možná odpověď vychází ze skutečnosti, že základním konceptem, přítomným v každé oblasti informatiky, je pojem *algoritmus*. Stručně řečeno, *algoritmus je konečná posloupnost instrukcí pro řešení nějakého problému, které lze vykonávat mechanicky, tj. vykonávání nevyžaduje důvtip nebo dodatečný vhled do problému*. Slovo „algoritmus“ pochází ze jména významného perského matematika al-Chvárizmího, které bylo v latině přetvořeno na Algorismi.¹¹

Příkladem algoritmu, který zná každý už od základní školy, je algoritmus pro sčítání čísel v desítkové soustavě: dvě sčítaná čísla napíšeme pod sebe a sčítání provádíme po jednotlivých číslicích zprava s přenášením jedničky, pokud součet přesáhne 10. V informatice mají algoritmy zásadní roli: všechny výdobytky informatiky mají někde uvnitř dobře navržený algoritmus. Informační technologie, které v dnešní době často považujeme za samozřejmé, nám pomáhají jen díky tomu, že někde uvnitř „běží“, neboli je procesorem vykonáván, vhodný algoritmus. Říkáme-li „chytrý telefon“, „chytrý internetový vyhledávač“, „dobrý kompresní program“, „inteligentní robotický vysavač“, „bezpečná komunikace“, „dobrý textový procesor“, „kvalitní filtr spamových e-mailů“, pak – jdeme-li k jádru věci – to vždy znamená dobře navržený algoritmus. Například algoritmus pro vyhledávání na internetu, který prohledá vše, ale vrátí jen to významné a relevantní; algoritmus komprese dat, který má velký kompresní poměr a je rychlý; algoritmus, který umožňuje odesílateli data spolehlivě zašifrovat a příjemci je pak dešifrovat; algoritmus, který rozpozná, že e-mail je spam, a podobně.

Aby mohly být algoritmy vykonávány počítačem, musí být zapsány v programovacím jazyce, tj. v symbolické formě, které počítač rozumí. Tvorba algoritmů je ale něco jiného než programování. Programování je

¹⁰<http://greatprinciples.org/>

¹¹Žil přibližně v letech 780–850. Český překlad jeho díla je nyní k dispozici s rozsáhlým historickým komentářem: Al-Chvárizmí, *Aritmetický a algebraický traktát* (Nymburk: OPS, 2009).

jen zápis algoritmů ve formě vhodné pro počítač, i když „jen“ by mělo být v uvozovkách. Samo o sobě je programování náročným oborem a s tvorbou algoritmů je velmi úzce provázáno. Složitější programové systémy, například rozsáhlé informační systémy, software v automatických pilotech dopravních letadel nebo operační systémy počítačů, obsahují celou řadu algoritmů – algoritmy tvoří základní stavební bloky takových systémů.

Algoritmy může pochopitelně vykonávat člověk, tak jako když na papíře sčítáme dvě čísla. Fascinující užitečnost algoritmů je ale dána tím, že je můžeme vykonávat na počítačích, tj. velmi rychle (nejrychlejší počítače umí vykonávat desítky biliard aritmetických operací za sekundu¹²), opakovaně (počítač se totiž neunaví) a bez zásahu člověka.

Prvním zařízením pro vykonávání algoritmů byl abakus, kuličkové počítadlo, které se používalo už v Sumeru před více než čtyřmi tisíci roky. Myšlenka sestrojít samočinný počítací stroj prošla dlouhou cestou. Je znákována milníky v podobě mechanické kalkulačky Blaise Pascala z roku 1642, na kterou navázal Gottfried Leibniz pokročilejší kalkulačkou Step Reckoner sestrojenou v roce 1673. V roce 1837 popsal Charles Babbage první univerzálně programovatelný počítač, mechanický Analytical Engine. V roce 1843 napsala Ada Lovelace první program pro takový počítač. Trvalo však dalších sto let, než byl univerzálně programovatelný počítač sestrojen. Byl jím první programovatelný elektronický počítač Colossus Mark 1, který byl uveden do provozu v prosinci 1943. Byl pak používán v Bletchley Park v Anglii ve známém projektu britské vlády, jehož cílem bylo luštit šifrované zprávy německé armády. Poznamenejme také, že již v roce 1937 ale americký fyzik Howard Aiken navrhl první elektromechanický počítač Harvard Mark 1. Ten byl pak v roce 1944 sestrojen firmou IBM v americkém Endicottu. Ohromující užitek, který počítače přinesly zejména pro vojenské operace druhé světové války, vedl k nevídaně rychlému vývoji v oblasti počítačů a počítačové techniky, který trvá dodnes.

Lidé se zabývají algoritmy již po dlouhá staletí. Na-

příklad velký řecký matematik Eukleides (cca 325–260 př. Kr.) popsal známý algoritmus pro nalezení největšího společného dělitele dvou čísel ve svých *Základech* už kolem roku 300 př. Kr.¹³ Lze říci, že teprve moderní počítače a jejich dostupnost ukázaly, jak mocným vynálezem algoritmus je. Bez nadsázky lze říct, že pojem algoritmus patří mezi nejvýznamnější plody lidského ducha. Pěkně to vystihují následující citáty:¹⁴

*Na sametu klenotníka září dvě ideje.
První je kalkulus,¹⁵ druhý je algoritmus.
Kalkulus a bohatý repertoár matematické
analýzy, který byl díky němu vytvořen,
umožnil vznik moderní vědy; ale byl to
algoritmus, díky němuž mohl vzniknout
moderní svět.*

–D. Berlinski, *The Advent of the Algorithm*, 2000

*Člověk s informatickým vzděláním ví, jak
nakládat s algoritmy: jak je vytvářet,
zacházet s nimi, rozumět jim, analyzovat
je. Tato schopnost nás připravuje na
mnohem více než jen na psaní dobrých
počítačových programů; je to obecný
myšlenkový nástroj, který nám pomáhá
porozumět jiným oblastem, ať už chemii,
lingvistice nebo třeba hudbě. Důvod této
skutečnosti je následující. Často se říká, že
člověk dané věci nerozumí, dokud ji
nevysvětlí jinému. Ve skutečnosti ale
člověk dané věci nerozumí, dokud ji
nevysvětlí počítači, tj. nevyjádří ji formou
algoritmu...Přístup spočívající ve
formalizaci prostřednictvím algoritmů
vede k mnohem hlubšímu porozumění,
než když se věcem snažíme porozumět
tradičním způsobem.*

–D. E. Knuth, *Selected Papers on
Computer Science*, 1996

¹² Přesně řečeno desítky biliard FLOPS (floating-point operations per second, tj. operací v tzv. plovoucí řádové čarce za sekundu). Jedna biliarda je 10^{15} , tj. 1 000 000 000 000 000. Viz také seznam 500 nejrychlejších počítačů na světě na <https://en.wikipedia.org/wiki/TOP500>.

¹³ Český překlad tohoto významného díla evropské vědy vyšel s podrobnými komentáři Petra Vopěnky nedávno: Eukleides, *Základy. Kniha I–XIII* (Nymburk: OPS, 2008–2012).

¹⁴ Překlady vlastní.

¹⁵ „Kalkulus“ zde znamená diferenciální a integrální počet, tj. nauku o derivování a integrování funkcí.

Druhý citát explicitně upozorňuje na všeobecnou užitečnost algoritmů. Již v padesátých letech 20. století se začal objevovat termín *algorithmic thinking*, tedy *algoritmické myšlení*. Vlivem práce [9] se později začal používat, prakticky pro totéž, termín *computational thinking*, což lze přeložit jako výpočetní nebo výpočetně-orientované myšlení. Tento pojem představuje přístup k řešení problémů používaný v informatice: formulaci problému jako transformaci přesně popsáných vstupů na přesně popsané výstupy a hledání algoritmů provádějících takovou transformaci. Důsledné uplatnění tohoto v principu jednoduchého přístupu zahrnuje řadu koncepčně důležitých a technicky někdy i značně netriviálních kroků, které jsou jádrem způsobu práce informatiků. Musíme si při něm uvědomit mimo jiné toto: jak je vlastně problém přesně formulován, včetně případných dodatečných omezení, a jak ho vhodným způsobem přesně, tedy formálně reprezentovat; co vlastně je správným řešením problému, zda a do jaké míry je přijatelné řešení přibližné, suboptimální; jak problém strukturovat, rozložit na podproblémy a zda a jak je řešení těchto podproblémů navzájem provázáno; že v mnoha případech lze díky vhodné abstrakci převádět tyto podproblémy na problémy, pro něž řešení už známe; jaká je složitost daného problému a jak ji kvantifikovat; zda dostupné zdroje pro řešení problému vůbec umožňují daný problém vyřešit; jaká je složitost jednotlivých kroků navrhovaného výpočtu, jak ji kvantifikovat (např. vyjádřit, kolik času kroky zaberou); zda je tato složitost přijatelná, popř. zda je potřebné a možné jednotlivé kroky dělat jinak, efektivněji.

Snad je patrné, že algoritmické myšlení zahrnuje soubor všeobecně prospěšných dovedností, jejichž osvojení je užitečné nejen informatikům, ale každému, kdo chce být dobře vzdělaný, a tedy dobře připravený pro život. Význam algoritmického myšlení pro člověka se odráží ve snahách – pozorovatelných v posledních deseti letech v Evropě i v USA – změnit způsob výuky informatiky na základních a středních školách. Základní myšlenka zní: ze stejného důvodu, pro který jsou do výuky zařazovány matematika a přírodní vědy, by do ní měly být zařazeny základy algoritmického myšlení. Ne proto, že bychom chtěli v první řadě vychovat více informatiků, ale proto, že algoritmické myšlení je dobrou přípravou pro život (stejně tak matematiku neučíme

v první řadě proto, že bychom chtěli vychovat více matematiků). Takový záměr je obsažen i ve strategii Ministerstva školství, mládeže a tělovýchovy [8], ve které se termín „computational thinking“ překládá souslovím „informatické myšlení“. Tuto vítanou snahu lze charakterizovat slovy, která by nejspíš podepsala naprostá většina zasvěcených: více výuky algoritmického myšlení, méně výuky soudobých informačních technologií a počítačové gramotnosti, které mají z podstaty věci menší hodnotu a jen dočasnou platnost.¹⁶

Informatika jako vědní obor

V mnohých kruzích je informatika chápána jako užitečný pomocník ostatních oborů, jako nástroj, spíše než jako plnohodnotný, samostatný vědní obor. Že jde o samostatný obor, jsme do jisté míry objasnili výše. Pojďme se ale na věc podívat podrobněji a začneme důležitou otázkou: Má informatika – podobně jako matematika, fyzika, chemie nebo biologie – nějaké fascinující penzum poznatků, které má neoddiskutovatelnou intelektuální hodnotu, dokáže uchvátit laika nebo zlákat pro obor studenta? Má tedy informatika něco podobně poutavého jako je teorie relativity nebo kvantová mechanika ve fyzice, geometrie vícerozměrných prostorů nebo teorie nekonečných množin v matematice či evoluce a genetiky v biologii? Má. Takových oblastí je řada. Příkladem jsou šifrování (kryptografie), umělá inteligence, strojové učení, teorie informace nebo teorie výpočtů a výpočetní složitost. O všech těchto oblastech se pro jejich atraktivitu píšou populární knížky. Pro úplnost uvedme i některé další – zda je považujeme za zajímavé, je koneckonců otázka vkusu. Mezi další oblasti informatiky tedy patří: programovací jazyky a překladače, programování, vývoj softwaru a softwarové inženýrství, architektura počítačů, operační systémy, počítačové sítě, paralelní a distribuované systémy, databázové systémy, informační systémy, dolování znalostí z dat (data mining), numerické a symbolické výpočty, zpracování signálů, počítačová grafika a počítačové vidění, komprese dat a další. V tomto článku se všem oblastem podrobněji věnovat nemůžeme. Omezíme se proto na jedinou z nich, teorii výpočtů a výpočetní složitosti, která představuje teoretický základ celé informatiky.

¹⁶Kromě toho, jak každý rodič ví, děti se používání informačních technologií učí snadno a rychle samy.

Základní otázky, kterými se teorie výpočtů zabývá, jsou: co je to algoritmus, co je to výpočet a co vlastně lze pomocí algoritmů na počítačích řešit? Bez důkladného porozumění těmto otázkám bychom přísně vzato nevěděli, o čem mluvíme, když se o informatice bavíme. Skeptik řekne: „Nejsou ale přeci jen tyto otázky zbytečné? Vždyť co je algoritmus, jsme už řekli. Co je to výpočet, je taky vcelku jasné. A co že lze na počítačích spočítat? Má nás vůbec tato otázka zajímat? Vždyť v praxi to je jinak: dostaneme problém a my prostě navrhneme algoritmus a napíšeme program, který tento problém vyřeší. A tím se také dostáváme k odpovědi na tuto poněkud teoretickou otázku, pokud na ní tedy tazatel trvá: jakýkoli výpočetní problém, který je přesně zadán – to znamená, že je přesně popsáno, jak vypadají vstupy a odpovídající výstupy – je v principu počítačem zvládnutelný. Může to být pracné, ale požadovaný program je možné napsat.“

Byť se tyto skeptikovy úvahy zdají být rozumné, jsou zásadně chybné. Za prvé, výše uvedená definice pojmu algoritmus není žádná definice. Definice má být přesná, naše však přesná není. Naše „definice“ je jen intuitivní vysvětlení, se kterým se nedá přesně pracovat. Toho si byli vědomi průkopníci moderní informatiky, zejména Alonzo Church a Alan Turing, který vytvořil přesnou definici toho, čemu se dnes říká Turingův stroj. Turingův stroj je přesně definovaný matematický model jednoduchého počítače.¹⁷ Churchova-Turingova teze je tvrzení, které říká, že problémy intuitivně řešitelné pomocí algoritmů na běžných počítačích jsou právě problémy řešitelné na Turingových strojích. Tuto tezi nelze dokázat, protože spojuje intuitivní pojem (výpočet podle algoritmu) s pojmem přesně definovaným (výpočet na Turingově stroji). Jinými slovy, teze říká, že pojem Turingova stroje bychom měli brát za přesnou definici pojmu algoritmus, a informatici tuto tezi skutečně přijímají.

Kromě přesné definice algoritmu přináší pojem Turingova stroje možnost konečně uchopit naši otázku, co

všechno lze pomocí algoritmů řešit. Ta je už teď totiž přesně definovaná. Čeká nás překvapení, které koncem třicátých let objevili nezávisle na sobě výše zmínění Church a Turing a které v té době vědce šokovalo: existují jednoduché, přesně formulované výpočetní problémy, které nejsou Turingovými stroji řešitelné, tedy nejsou na počítači řešitelné žádným algoritmem. Těchto takzvaně *algoritmicky neřešitelných problémů* existuje celá řada, ve skutečnosti více než algoritmicky řešitelných.¹⁸ Zmíníme se o dvou z nich.

Prvním je *problém zastavení* (halting problem). Jedna z jeho variant je tato:¹⁹ na vstupu je zdrojový kód algoritmu (v nějakém předem zvoleném programovacím jazyce); úkolem je rozhodnout, zda se tento algoritmus vždy, tj. pro libovolná vstupní data, zastaví.²⁰ To je ryze praktický problém. Představme si, že softwarová firma ví, že její programátoři dělají chyby často vedoucí k tomu, že se program zacyklí. Vedení firmy se rozhodne, že je třeba vyvinout testovací program, který takové chyby pomůže odhalovat. Takový program, označme ho T , má pracovat následovně. Na vstup dostane zdrojový kód testovaného programu P . Testovací program T má odpovědět „ano“, pokud je testovaný program P v pořádku, tj. P by se zastavil pro jakákoli vstupní data, a „ne“ v opačném případě, tj. pokud se testovaný program P pro nějaká vstupní data zacyklí.²¹ Takový program T by jistě bylo užitečné mít. Háček je v tom, že žádný takový program neexistuje, tj. nejen že ho nikdo dosud nevytvořil, ale ani ho nikdy nevytvoří, protože je matematicky dokázáno, že neexistuje. Tento poznatek je jedním z elementárních výsledků teoretické informatiky.

Druhý algoritmicky neřešitelný problém souvisí s dávným snem o strojích, které by byly schopné samostatně myslet. Zjednodušená podoba tohoto mnohovrstevného snu může znít následovně. Máme dána nějaká tvrzení, ze kterých chceme vycházet a která považujeme

¹⁷Není to tedy fyzický počítač. Je to abstrakce, přesně popsané schéma, podle kterého by se fyzický počítač dal postavit. Church navrhl místo pojmu Turingův stroj jiný teoretický pojem, takzvaný λ -kalkul. Oba stavěli na tehdy čerstvých výsledcích fenomenálního logika, brněnského rodáka Kurta Gödela.

¹⁸Algoritmicky neřešitelných problémů je nekonečně mnoho, dokonce nespočetně mnoho. Algoritmicky řešitelných problémů existuje také nekonečné množství, toto množství je ale spočetné. Neřešitelných je tedy více než řešitelných: pokud bychom se pokusili řešitelné problémy s neřešitelnými spárovat, nějaké neřešitelné vždy zbydou.

¹⁹V původní formulaci je na vstupu zdrojový kód algoritmu a data pro tento algoritmus; úkolem je rozhodnout, zda se algoritmus spuštěný s těmito daty na vstupu zastaví.

²⁰Mohl by se totiž takzvaně zacyklit, a tedy nikdy neskončit. Jako příklad uveďme algoritmus, který obsahuje po sobě následující instrukce „do proměnné i vlož hodnotu 3“ a „pokud $i > 3$, pokračuj další instrukcí, jinak se vrať k předchozí instrukci“.

²¹S touto úlohou se na nás na Katedře informatiky Přírodovědecké fakulty Univerzity Palackého v Olomouci skutečně firmy obracejí.

za pravdivá. Takovým tvrzením říkáme axiomy. Množinu daných axiomů označme S . Označme dále symbolem t nějaké další dané tvrzení. S může být systém axiomů popisujících vlastnosti přirozených čísel, operací sčítání a násobení a podobně, t může být tvrzení „když x je dělitelem y a y je dělitelem z , pak x je dělitelem z “. V principu ale může jít o jakékoli axiomy a tvrzení, které můžeme zapsat pomocí klasické logiky. Úkolem je rozhodnout, zda tvrzení t z axiomů S plyne.²² Tento problém, takzvaný *Entscheidungsproblem* (problém rozhodnutí), předložil v roce 1928 známý německý matematik David Hilbert jako zásadní matematický problém. Pokud by existoval algoritmus, který by *Entscheidungsproblem* řešil, znamenalo by to mimo jiné, že značnou část práce matematiků, ale i jiného formalizovaného usuzování a odvozování, by bylo možné provádět mechanicky pomocí počítače. Bylo ovšem dokázáno, opět k velkému překvapení Hilberta i ostatních, že takový algoritmus neexistuje. V určitých případech mechanické odvozování možné je, obecně ale ne.

Příběh má ovšem zajímavé pokračování a vede k jedné z nejdůležitějších otevřených otázek současné informatiky, otázce *P versus NP*. Pokud se ukáže, že problém, který řešíme, je algoritmicky řešitelný, nemáme ještě vyhráno. Důležité totiž je, jak rychlý je algoritmus, který pro řešení tohoto problému máme, tj. jak je náročný na čas. Tato časová náročnost, nazývaná v teorii výpočtů *časová složitost algoritmu*, se – zhruba řečeno – měří v počtech instrukcí, které musí algoritmus pro vyřešení problému vykonat.

Řekneme-li, že časová složitost algoritmu je n^2 , znamená to, že bude-li tento algoritmus zpracovávat vstup velikosti n , pak v nejhorším možném případě bude k provedení výpočtu zapotřebí provést n^2 instrukcí. Tedy pokud problém spočívá v setřídění vstupní posloupnosti n čísel od nejmenšího po největší, pak pro posloupnost, pro kterou trvá výpočet nejdéle, algoritmus vykoná n^2 instrukcí. Předpokládáme-li, že výpočet probíhá na superpočítači, který provede 10^{15} instrukcí za sekundu, pak i setřídění posloupnosti milionu

čísel, tj. $n = 10^6$, je velmi rychlé – trvá jednu tisícinu sekundy. Potřebný čas je totiž $\frac{\text{počet instrukcí}}{10^{15}} = \frac{n^2}{10^{15}} = \frac{(10^6)^2}{10^{15}} = \frac{1}{10^3} = 0,001$ sekundy. Pokud by ale algoritmus měl časovou složitost $n!$,²³ byla by situace katastrofická. Předpokládejme, že takový algoritmus budeme chtít použít pro setřídění mnohem kratší posloupnosti, která obsahuje deset tisíc čísel, tj. vstup bude mít velikost $n = 10^4$. Potřebný čas pak je $\frac{\text{počet instrukcí}}{10^{15}} = \frac{n!}{10^{15}} \approx \frac{2,85 \cdot 10^{35659}}{10^{15}} = 2,85 \cdot 10^{35644}$ sekundy, což je asi $9 \cdot 10^{35636}$ let. Uvědomíme-li si, že planeta Země existuje asi $4,54 \cdot 10^9$ let, je jasné, že výpočtu tímto algoritmem by se nikdo nedočkal a vůbec o něm nemá smysl uvažovat. Podobně by to dopadlo s algoritmem o časové složitosti 2^n , tedy exponenciální: výpočet pro vstup velikosti $n = 10^2$, tj. setřídění pouhého sta čísel, by trval asi $4 \cdot 10^7$ let, tedy asi 40 milionů let (před 65 miliony let vyhynuli dinosauři). Že máme pro daný problém algoritmus, tedy ještě nic neznamená. Důležité je, jakou má tento algoritmus časovou složitost.

Díky výzkumu v teorii výpočtů dnes víme mnoho zajímavých věcí. Jsou například známy problémy, pro které je dokázáno, že k nim žádný algoritmus s menší než exponenciální časovou složitostí neexistuje. Takové problémy jsou tedy sice algoritmicky řešitelné, ale prakticky vlastně neřešitelné, protože výsledků výpočtu pro větší vstupní data by se nikdo nedočkal. Mnoho zajímavých otázek je ale zatím nevyřešeno. Je například známa celá množina problémů, nazývají se NP úplné, která obsahuje stovky prakticky velmi významných problémů, pro něž se lidé snažili najít efektivní, rychlé algoritmy dlouhé desítky let.²⁴ Nikomu se to ale zatím nepodařilo. Jedním z nich je známý *problém obchodního cestujícího*: je dáno n míst a vzdálenosti mezi nimi; je možné navštívit všechna tato místa tak, aby celková vzdálenost, kterou při tom urazíme, byla nejvýše d ? S tímto problémem nebo jeho variantami se potýkají například dopravní společnosti, které mají rozvést do určených míst zboží a přitom spotřebovat co nejméně paliva. Ví se, že pokud by se podařilo najít rychlý algoritmus pro jeden

²²Přesněji: zda z nich vyplývá ve smyslu klasické predikátové logiky.

²³ $n!$ je takzvaný faktoriál přirozeného čísla n a je definován takto: $n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1$. Algoritmem se složitostí $n!$ by byl například algoritmus, značně naivní, který by zkoušel všechny možné permutace vstupní n -prvkové posloupnosti a pro každou z nich zjistil, zda je setříděná; pokud ano, skončí, pokud ne, vezme další permutaci. Protože všech takových permutací je, jak známo, $n!$, algoritmus potřebuje v nejhorším případě (zhruba) $n!$ instrukcí.

²⁴Rychlým algoritmem se zde rozumí algoritmus, jehož časová složitost je polynomiální, tj. například n^2 , n^3 nebo n^4 . Množina problémů, pro které existují tyto rychlé algoritmy, se značí P. Odtud název „P versus NP“.

jediný z těchto problémů z množiny NP úplných (libovolný z nich), znamenalo by to, že každý z těchto problémů je řešitelný rychlým algoritmem – tyto problémy jsou jeden na druhý redukovatelné, tj. převoditelné. Nikomu se však zatím takový algoritmus pro žádný z těchto problémů najít nepodařilo. Co když ale takový rychlý algoritmus prostě neexistuje? To je sice možné, ale ani to se zatím matematicky dokázat nepodařilo. Přitom je této otázce věnováno velké úsilí již více než 40 let.²⁵ Uvedená otázka, známá jako otázka P versus NP, tj. otázka, zda pro problém obchodního cestujícího a další problémy z množiny NP úplných existují rychlé algoritmy, je považována za snad nejdůležitější otevřenou otázku informatiky a za její vyřešení je vypsána odměna jeden milion dolarů.²⁶ Jak tedy vidíme, informatika nabízí jak fascinující poznatky, tak důležité, zatím nezodpovězené otázky.

Informatika jako obor ke studiu

Proč informatiku studovat

O vysokoškolské studium informatiky je dlouhodobě velký zájem. Je to jednak tím, že všude kolem nás jsou počítače. Práce s nimi je zajímavá a student, ať už pronikl či nepronikl hlouběji než k běžným uživatelským dovednostem, si prostě může říct: „To by mě asi bavilo.“ Druhým faktorem je skutečnost, že vystudovaný informatik velmi dobře najde uplatnění. Tak je to v pořádku, dodejme ale jednu poznámku. Vzhledem k nesprávné představě široké veřejnosti o tom, co informatika obnáší, a tedy nesprávné představě většiny studentů, dochází – možná ve větší míře než u jiných oborů – k tomu, že studovat informatiku se rozhodnou i studenti, kteří si představují, že to bude snadné, a pak studium nezvládnou. Stejný důvod ale vede studenty, kteří se chtějí naučit „něco pořádného“, k tomu, že se na informatiku nepřihlásí, protože si myslí, že informatika není „pořádný“, plnohodnotný obor a že už stejně většinu informatiky znají. Že to tak není, jsme vysvětlili výše. Podtrhněme následující:

Studiem informatiky získá student nejen konkrétní znalosti a dovednosti, které bude potřebovat v praxi informatika, ale i schopnosti obecného matematického, logického i inženýrského myšlení. Získá tak nejen předpoklady pro výkon profese informatika, ale i pro výkon povolání vyžadujících analytické schopnosti a racionální, fakty a daty podložené rozhodování.

Kde lze informatiku studovat

V České republice se vysoké školy tradičně dělí na univerzity a technické školy. Je ale třeba říct, že toto dělení se postupně stírá. Univerzity poskytují vzdělání zejména v přírodovědných, lékařských a humanitních oborech, techniky pak v inženýrských oborech a v ekonomii. Vzhledem k povaze informatiky není divu, že se u nás dá tento obor studovat jak na univerzitách, tak na technikách. Na univerzitách je více než na technikách zastoupena matematická složka, ve studiu je více předmětů o principech informatiky, například o algoritmech a jejich analýze. Na technikách obvykle převládá inženýrská složka informatiky, ve studiu jsou více zastoupeny informační technologie. Na technikách se informatika obvykle studuje na fakultách, které mají slovo „informatika“ v názvu, na univerzitách obvykle na fakultách přírodovědně zaměřených. Z obou typů škol však vycházejí informatici a více než na typu školy záleží na její kvalitě. Kvalitu školy, resp. pracoviště (fakulty, katedry) zabezpečujícího matematické obory, však nelze posuzovat podle žebříčků, které vycházejí v českých denících. Ty jsou nespolehlivé a některá pracoviště se jich už odmítají účastnit. Spolehlivější jsou různá hodnocení zahraniční nebo – což je velmi přínosné – návštěva dne otevřených dveří na dané škole, popřípadě návštěva webových stránek, které obvykle obsahují podrobné informace o studiu, o zaměření i o vědeckých výsledcích daného pracoviště.

²⁵Neznamená to ale, že bychom při setkání s těmito problémy byli bezbranní. Byly vyvinuty různé *heuristické algoritmy*, pomocí kterých je možné hledat přibližná řešení. Neumíme sice rychle spočítat délku optimální, tj. nejkratší cesty obchodního cestujícího, ale umíme poměrně spolehlivě rychle spočítat délku téměř optimální cesty.

²⁶<http://www.claymath.org/millennium-problems/p-vs-np-problem>.

²⁷Posuzováno tradicí a mezinárodními žebříčky kvality.

Na předních českých univerzitách²⁷ lze informatické obory studovat na Univerzitě Karlově v Praze (Matematicko-fyzikální fakulta), Masarykově univerzitě v Brně (Fakulta informatiky) a Univerzitě Palackého v Olomouci (Přírodovědecká fakulta). Na předních technikách pak například na Českém vysokém učení technickém v Praze (zejm. Fakulta informačních technologií a Fakulta elektrotechnická), Vysokém učení technickém v Brně (Fakulta informačních technologií) a Vysoké škole báňské – Technické univerzitě Ostrava (Fakulta elektrotechniky a informatiky).

Poznámka: Článek vyjde v časopise *Matematika – fyzika – informatika*.

Reference

- [1] *Computing Our Future. Computer Programming and Coding: Priorities, School Curricula and Initiatives*. European Schoolnet, Brusel, 2015.
- [2] Denning, P. 2005. Is computer science science? *Communications of ACM* 48 (4): 27–31.
- [3] Denning, P. 2010. The great principles of computing. *American Scientist* 98: 369–372.
- [4] Denning, P. et al. 1989. Computing as a discipline. *Communications of ACM* 32 (1): 9–23.
- [5] National Research Council of the National Academies. 2004. *Computer Science: Reflections on the Field, Reflections from the Field*. Washington, DC: National Academies Press.
- [6] Newell, A., Perlis, A. J., Simon, H. A. 1967. Computer science. *Science* 157: 1373–1374.
- [7] *Rámcový vzdělávací program pro gymnázia*. Výzkumný ústav pedagogický v Praze, Praha, 2007. http://www.msmt.cz/uploads/Vzdelavani/Skolska_reforma/RVP/RVP_gymnazia.pdf
- [8] *Strategie digitálního vzdělávání do roku 2020*. MŠMT, 2014. <http://www.msmt.cz/ministerstvo/strategie-digitalniho-vzdelavani-do-roku-2020>
- [9] Wing, J. M. 2006. Computational thinking. *Communications of ACM* 49 (3): 33–35.

Hádanka

Petr Osička

Zjistěte, co vrací následující funkce, a zdůvodněte proč. Správné řešení hádanky naleznete na webové stránce magazínu.

```
int mystery(unsigned int x)
{
    int count = 0;
    while(x) {
        count++;
        x = x & (x - 1);
    }
    return count;
}
```

Redakce:
Petr Krajča, Petr Osička
Katedra informatiky PřF UP
17. listopadu 12
771 46 Olomouc

web: www.inf.upol.cz/magazin
email: magazin@inf.upol.cz
atom: <http://www.inf.upol.cz/atom/magazin>
telefon: 585634701
GPS: 49.591870, 17.262336

