

Pokroky matematiky, fyziky a astronomie

Donald E. Knuth

Computer science a jej vztah k matematike

Pokroky matematiky, fyziky a astronomie, Vol. 21 (1976), No. 2, 88--96

Persistent URL: <http://dml.cz/dmlcz/139254>

Terms of use:

© Jednota českých matematiků a fyziků, 1976

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://project.dml.cz>

středků nejspíš nepřesvědčí toho, kdo nechce být přesvědčen (o tom, že programování se pomalu stává vědecky fundovanou metodikou hodnou pozornosti matematiků). Případný zájemce však najde další argumenty a zejména detailnější výklad v uvedené literatuře, kde tituly [1] a [2] jsou zpracovány světovými kapacitami s bohatou teoretickou erudiicí i praktickými zkušenostmi; jediná šance článku [3] naproti tomu spočívá v jeho relativně větší dostupnosti.

Literatura

- [1] O. J. DAHL, E. W. DIJKSTRA, C. A. R. HOARE: *Structured Programming*, Academic Press 1972.
- [2] D. E. KNUTH: *Structured Programming with Goto Statements*, Computing Surveys 1974, Nr. 4, p. 261—301.
- [3] J. HOŘEJŠ: *Principy strukturovaného programování I, II*, Informačné systémy 1975, č. 2, str. 97—115, č. 3, str. 227—244.
- [4] F. BACON: *Nové Organon*, ČSAVU 1922.

Computer science a jej vztah k matematice

Donald E. Knuth, Stanford, USA)*

Súčasný trend umožňuje predvídať deň, keď computer science a matematika budú obe existovať ako uctievané disciplíny slúžiace analogickým, ale rozdielnym úlohám vo vzdelaní človeka.

Nová disciplína, nazývaná computer science, sa objavila v programoch väčšiny svetových univerzít. Tento článok opisuje subjektívny pohľad na vztah tohoto predmetu k matematike formou porovnania ich rozdielov a podobností a skúmaním niektorých spôsobov, ktorými si navzájom pomáhajú.

Čo je to computer science?

Pretože computer science je pomerne mladá disciplína, musím najskôr vysvetliť, o čom pojednáva. Aspoň moja manželka vraví, že to musí vysvetľovať, kedykoľvek sa jej niekto pýta, čo robím. Navyše mám pocit, že v súčasnosti väčšina ľudí chápe

*) Reprinted from The American Mathematical Monthly, April 1974, copyright © by the Mathematical Association of America. (Zkrácené znění.)

túto oblasť trochu inak ako ja. Je skutočne málo pravdepodobné, že by ju dvaja ľudia, pracujúci v tejto oblasti, definovali rovnako. Nie je to nič prekvapujúce, pretože rovnako ťažko by bolo nájsť dvoch matematikov, ktorí by dali rovnakú definíciu matematiky. Našťastie v posledných rokoch je módne pociťovať „krízu identity“, takže ľudia od počítačov sú v tomto celkom moderní.

S obľubou charakterizujem computer science ako náuku o algoritmoch. Algoritmus je presne definovaná postupnosť pravidiel, popisujúca ako vyprodukovať určenú výstupnú informáciu z danej vstupnej informácie konečným počtom krokov. Konkrétna reprezentácia algoritmu sa nazýva program, podobne ako sa slovom „dáta“ označuje konkrétna reprezentácia „informácie“ ([14]). Možno, že najdôležitejším objavom indukovaným vznikom počítačov bude poznanie, že algoritmy ako objekty štúdia sú neobyčajne bohaté na zaujímavé vlastnosti a ďalej, že algoritmický pohľad je veľmi užitočný spôsob organizácie vedomostí vôbec. G. E. FORSYTHE poukázal na to, že „otázka, čo možno automatizovať je jedna z najinspirujúcejších filozofických otázok dnešnej civilizácie“ ([8]).

Z týchto poznámok by sme mohli usúdiť, že computer science mala existovať už dlho pred príchodom počítačov. V istom zmysle aj existovala; má hlboké korene v histórii. Napríklad v poslednej dobe ma zaujalo štúdium starých rukopisov, z ktorých som zisťoval, do akej miery boli Babylončania pred 3500 rokov „informatikmi“* ([16]). Ale k tomu, aby sme sa dozvedeli viac o všeobecných vlastnostiach algoritmov sú skutočne potrebné počítače. Ľudia nie sú ani dosť presní ani dosť rýchli na to, aby previedli hociktorú okrem najjednoduchších postupností operácií. Preto si až do príchodu univerzálnych počítačov nikto plne neuvedomil potenciálnu bohatosť štúdia algoritmov.

Mal by som upozorniť na to, že počítače (a algoritmy) nepočítajú len s číslami. Môžu narábať s informáciou akéhokoľvek druhu, keď je presným spôsobom reprezentovaná. Zvykli sme hovoriť, že postupnosť symbolov, napríklad meno, je reprezentovaná vo vnútri počítača ako číslo. Ale v skutočnosti je správnejšie hovoriť, že číslo je reprezentované vo vnútri počítača ako postupnosť symbolov.

Francúzske slovo pre „computer science“ je „informatique“; nemecké „Informatik“ a dánske „datalogi“ ([21]). Všetky tieto termíny múdro naznačujú, že computer science pojednáva o mnohých iných veciach mimo riešenia numerických rovníc. Tieto názvy však zdôrazňujú „materiál“, s ktorým algoritmy narábajú (dáta alebo informáciu), namiesto algoritmov samotných. Nóri na univerzite v Oslo si zvolili o niečo vhodnejší názov – databehandling. Jeho anglický ekvivalent, „spracovanie dát“ (data processing), sa na nešťastie zaužíval v Amerike len v spojení s ekonomickými aplikáciami; a „spracovanie informácie“ (information processing) pripomína skôr knihovnicke aplikácie. Niekoľkí ľudia navrhovali termín „computing science“ (veda o počítaní) ako výstižnejší než „computer science“ (veda o počítačoch)**).

Hľadanie perfektného názvu je samozrejme trochu nepodstatné, pretože pojmy, s ktorými sa narába, sú oveľa dôležitejšie ako názov. Napriek tomu je možno podstatné, že všetky spomínané názvy pre computer science kládli do úzadia úlohu samotných

* Slovom „informatik“ prekladáme v tomto článku autorov termín „computer scientist“. (Pozn. prekl.)

** Radšej ponechávame termín „computer science“ nepreložený. (Pozn. prekl.)

počítačov. Pravdepodobne preto, aby sa tento odbor stal „legitímnejším“ a úctyhodnejším. Názor mnohých ľudí na počítač je, pri najlepšom, že je to nutné zlo. Komplikovaný nástroj, ktorý treba použiť, keď iné metódy skrachujú. Prečo teda klásť taký dôraz na výuku používania počítačov, keď sú iba cennými nástrojmi ako napríklad elektrónový mikroskop?

Informatici, vediac, že počítače sú viac než toto, inštinktívne kladú strojovú stránku do úzadia, keď obhajujú svoj odbor. Netreba sa však príliš odťahovať od počítačov. Výstižne na to poukázali NEWELL, PERLIS a SIMON ([22]), ktorí definovali computer science jednoducho ako náuku o počítačoch, rovnako ako botanika je náuka o rastlinách, astronómia náuka o hviezdach a pod. Deje okolo počítačov sú nesmierne zložité a rôznorodé a vyžadujú popis a vysvetlenie. A podobne ako elektrina, tieto deje majú stránku inžiniersku aj vedeckú.

Keď hovorím, že computer science je náuka o algoritmoch, vyberám tým len jeden z „dejev okolo počítačov“, takže computer science v skutočnosti zahrňuje viac. Zdôrazňoval som algoritmy, pretože sú skutočne jadrom predmetu, spoločným menovateľom, ktorý tvorí základ a zjednocuje rôzne vetvy. Môže sa stať, že technológia sa niekedy ustáli, takže povedzme za 25 rokov sa počítače budú meniť veľmi málo. Nič nenaznačuje takéto ustálenie technológie v blízkej budúcnosti, ba skôr naopak. Verím však, že štúdium algoritmov zostane náročné, stimulujúce a dôležité, aj keď by ostatné stránky počítačov boli niekedy plne preskúmané.

Čitateľ, ktorého zaujímajú ďalšie úvahy o povahe computer science, ich môže nájsť, okrem už citovaných prác, v prácach [17] a [29].

Je computer science časťou matematiky?

Iste sú aspekty počítačov, ktoré sú teraz aktívne skúmané informatikmi a ktoré ťažko nazvať matematickými. Ale ak obmedzíme našu pozornosť na štúdium algoritmov, nie je to iba vetva matematiky? Koniec koncov, ak vôbec niekto študoval algoritmy pred príchodom computer science, tak to boli matematici. Dalo by sa teda dôvodiť, že tento ústredný aspekt computer science je v skutočnosti časťou matematiky.

Verte však, že podobne by sa dalo dokázať tvrdenie, že matematika je časťou computer science. Teda, podľa definície rovnosti množín by sa dalo dokázať, že tieto predmety sú rovnaké alebo prinajmenšom, podľa Schröder-Bernsteinovej vety, ekvipotentné.

Domnievam sa, že žiadna z týchto množinových inklúzií neplatí. Je vždy ťažké určiť presné hranice medzi vedami (porovnajte napríklad chemickú fyziku a fyzikálnu chémiu), ale je možné vydeliť podstatne odlišné prístupy v matematike a v computer science.

Táto pravdivá historka možno najlepšie vysvetlí rozdiel, ktorý mám na mysli. Pred niekoľkými rokmi som sa dozvedel o matematickej vete, ktorej dôsledkom bolo, že ľubovoľné dve celočíselné matice A a B typu $n \times n$ majú „najväčšieho spoločného pravého deliteľa“ D . To znamená, že D je pravým deliteľom A aj B (t. j., $A = A'D$ a $B = B'D$ pre nejaké celočíselné matice A' a B'), a každý spoločný deliteľ A a B je pravým deliteľom D . Rozmýšľal som, ako vypočítať najväčšieho spoločného pravého

deliteľa dvoch daných matic. Niekoľko dní potom som sa zúčastnil konferencie, na ktorej som stretol matematika H. B. MANNU. Zdalo sa mi, že on by mohol poznať riešenie tohoto problému. Spýtal som sa ho, a on skutočne vedel správnu odpoveď. Ale bola to odpoveď matematika a nie informatika. Povedal: „Nech R je okruh celočíselných matic typu $n \times n$. V tomto okruhu je súčet dvoch hlavných ideálov hlavný ideál, takže nech D je také, že $RA + RB = RD$. Potom D je najväčší spoločný pravý deliteľ A a B “. Tento vzorec je určite najjednoduchší možný. Treba iba osem symbolov na jeho zapísanie a opiera sa o presne dokázané vety algebry. Ale z hľadiska informatika je úplne bezcenný, pretože vyžaduje skonštruovanie nekonečných množín RA a RB , ich súčtu, a potom prehľadávanie nekonečne veľa matic D takých, aby sa tento súčet rovnal nekonečnej množine RD . Nedokázal by som prevádzaním takých nekonečných operácií určiť ani najväčšieho spoločného pravého deliteľa matic $\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$ a $\begin{vmatrix} 4 & 3 \\ 2 & 1 \end{vmatrix}$. (Mimochodom, odpoveď informatika na tento problém dal môj študent MICHAEL FREDMAN, viď [15, str. 380].)

Jeden z mojich priateľov matematikov mi povedal, že by bol ochotný uznať, že computer science je dôležitá vedná oblasť, ktorú stojí za to študovať, len čo bude obsahovať 1000 hlbokých viet. Toto kritérium musí byť samozrejme zmenené, aby obsahovalo algoritmy rovnako ako vety. Povedzme, 500 hlbokých viet a 500 hlbokých algoritmov. Ale aj tak je zrejmé, že dnešná computer science neobstojí pred takým kritériom, pokiaľ „hlboký“ znamená, že mimoriadne nadaný človek môže na takú vetu alebo algoritmus prísť po mnohomesačnej práci. Computer science je na to stále primadá a mladostť tu môžem uplatňovať ako nevýhodu. Stále nevieme, ako najlepšie popísať algoritmy, ako im porozumieť alebo ako dokázať ich správnosť, ako ich nájsť a ako analyzovať ich správanie, aj keď sa urobil značný pokrok na všetkých týchto frontoch. Potenciál pre 1000 „hlbokých výsledkov“ je tam, aj keď sa ich dosiaľ objavilo iba zo päťdesiat.

Z toho dôvodu, aby som ilustroval vzájomný vplyv computer science a matematiky a ich vzájomnú úlohu, pozerám sa trochu do budúcnosti, do obdobia, keď computer science bude zrelšia a istejšia sama sebou. Súčasný trend umožňuje predvídať deň, keď computer science a matematika budú obe existovať ako uctievané disciplíny, slúžiace analogickým, ale rozdielnym úlohám vo vzdelaní človeka. Aby som znovu citoval Georgea Forsythea, „Najhodnotnejším prínosom vo vedeckom a technickom vzdelaní sú všeobecné mentálne nástroje, ktoré môžu slúžiť po celý život. Materinský jazyk a matematiku pokladám za najdôležitejšie z týchto nástrojov a computer science radím na tretie miesto“ ([9]).

Ako matematika, computer science bude považovaná za základný predmet všeobecného vzdelania. Ako matematika a iné vedy, computer science zostane nejasnou hranicou rozdelená na dve oblasti, ktoré by sa dali nazvať „teoretická“ a „aplikovaná“. Ako matematika, computer science sa bude trochu líšiť od iných vied tým, že sa zaoberá zákonmi vytvorenými človekom, ktoré môžu byť dokázané, na rozdiel od prírodných zákonov, ktoré nikdy nepoznáme s istotou. Tieto dva predmety budú teda podobné z mnohých stránok. Rozdiel je v objekte skúmania a v prístupe. Matematika sa zaoberá viac-menej vetami, nekonečnými procesmi, statickými vzťahmi a computer science sa zaoberá viac-menej algoritmami, konečnými konštrukciami a dynamickými vzťahmi.

Mnohí informatici robili matematiku, ale oveľa častejšie bola skutočnou náplňou prác matematikov vlastne computer science. Veľmi na mňa zapôsobilo, ako veľa matematických teórií sa v skutočnosti zaoberá konkrétnymi algoritmickými problémami. Tieto teórie sú zvyčajne formulované v matematických termínoch, ktoré sú oveľa ťažkopádnejšie a neprirodzenejšie ako ekvivalentné algoritmické formulácie, ktoré by dnes použili informatici. Napríklad väčšina obsahu 35stránkového článku od ABRAHAMA WALDA sa dá predniesť na zhruba dvoch stránkach, keď sa preformuluje do algoritmických témov ([15, str. 142–144]): a dajú sa uviesť mnohé ďalšie príklady. Ale tým sa bude zaoberať iný článok.

Metodologický prínos

Človek dobre vyškolený v computer science vie ako pracovať s algoritmami, ako ich konštruovať, ako s nimi manipulovať, rozumieť im, analyzovať ich. Tieto vedomosti ho pripravujú pre oveľa viac, ako je písanie dobrých programov pre počítač. Sú mentálnym nástrojom, ktorý mu bude nepochybne nápomocný pri porozumení iným predmetom, či už je to chémia, lingvistiká, hudba alebo iné. Dôvod, prečo to možno chápať takto je tento: často sa hovorí, že človek nerozumie niečomu dokonale, kým to nenaučí niekoho iného. V skutočnosti človek nerozumie niečomu dokonale, kým to nedokáže naučiť počítač, t. j., vyjadriť to vo forme algoritmu. „Počítač si skutočne vynucuje takú presnosť myslenia, aká sa považuje za produkt štúdia matematiky“ ([7]). Pokus formalizovať veci ako algoritmy vedie k postatne hlbšiemu pochopeniu, ako keby sme sa pokúsili pochopiť tieto veci tradičným spôsobom.

Lingvisti si mysleli, že rozumejú jazykom až dovedy, kým sa pokúsili vysvetliť jazyky počítačom. Skoro prišli na to, ako veľa sa o nich ešte treba dozvedieť. Často sa stáva, že pri vytváraní počítačového modelu pre nejaký jav sa o tomto jave dozvieme viac ako z prípadných numerických výsledkov programu, ktorý sme zostavili.

Tri roky som učil abstraktnú algebru v druhom ročníku špecializácie matematika na California Institute of Technology a najťažšou partiou bol vždy „Jordanov kánonický tvar“ matíc. Tretí rok som sa pokúsil o iný prístup, pozeraúc sa na problém algoritmicky, a naraz bolo všetko jasné. To isté sa stalo pri preberaní konečných grúp definovaných generátormi a reláciami a v inej prednáške s redukčnou teóriou binárnych kvadratických foriem. Algoritmickým vyjadrením problému sa účel a význam matematických viet sprehľadnil.

Neskôr, keď som písal knihu o aritmetike počítačov [15], som prišiel na to, že prakticky každá veta v elementárnej teórii čísel sa objavuje prirodzene, s motiváciou, v súvislosti s problémom vykonávania veľmi rýchlych numerických výpočtov v počítačoch. Preto verím, že tradičné prednášky z elementárnej teórie čísel by sa dali celkom dobre adaptovať pre tento prístup pridaním praktickej motivácie k už beztak prekrásnej teórii.

Tieto príklady a mnohé ďalšie ma presvedčili o pedagogickej hodnote algoritmického prístupu. Pomáha pri porozumení pojmov najrôznejších druhov. Verím, že študent, ktorý je správnym spôsobom školený v computer science sa učí niečo, čo mu implicitne pomôže zvládnuť mnohé iné predmety. Preto budeme môcť čoskoro odôvodnene ho-

voríť, že poslucháči computer science dostali dobré všeobecné vzdelanie, tak ako to teraz veríme o poslucháčoch matematiky. Na druhej strane, dnešné prednášky z computer science zatiaľ nespĺňujú tento cieľ. Aspoň sa mi zdá, že mnohí absolventi computer science boli vyškolení v oveľa užšom merítku, ako by som si predstavoval. Informatici samozrejme pracujú na tom, aby odstránili tento dnešný nedostatok, ktorý – mám taký dojem – vznikol vďaka priveľkému zdôrazňovaniu programovacích jazykov na úkor algoritmov.

Niektoré vzájomné vplyvy

Computer science vplývala na matematiku mnohými spôsobmi. Pokúsim sa poukázať tu na tie dobré. V prvom rade, samozrejme, počítače sa dajú použiť k počítaniu a boli často používané v matematickom výskume, keď ručné výpočty boli príliš ťažké. Počítače generovali údaje, ktoré dali vznik mnohým hypotézam a práve tak mnohé vyvrátili. Napríklad GAUSS povedal ([10]), že prvý raz pomyslel na vetu o prvočíslach, keď sa pozeral na tabuľku prvočísel menších ako milión. V mojej vlastnej dizertácii sa mi podarilo vyriešiť hypotézu súvisiacu s nekonečne veľa prípadmi pozorným prehliadnutím výpočtov počítača pre najmenší prípad ([13]). Iným príkladom je pokrok, ktorý dosiahol MARSHALL HALL pri určovaní všetkých jednoduchých grúp rádu menšieho ako milión.

V druhom rade je tu prirodzený súvis medzi computer science a matematikou v numerickej matematike ([30]), logike a teórii čísel. Je natoľko známe, že sa tu ním nemusím zaoberať. Rád by som však spomenul najmä prácu D. H. LEHMERA, ktorý kombinoval použitie počítačov a klasickú matematiku niekoľkými pozoruhodnými spôsobmi. Napríklad dokázal, že každá množina šiestich po sebe idúcich celých čísel väčších ako 285 obsahuje násobok prvočísla väčšieho alebo rovného 43.

Ďalším vplyvom computer science bol zvýšený dôraz na konštrukcie vo všetkých odvetviach matematiky. Nahrádzanie existenčných dôkazov algoritmami, ktoré konštruujú matematické objekty, často viedlo k vylepšeniu abstraktnej teórie, Napríklad E. C. DADE a H. ZASSENHAUS poznamenali v závere článku*) napísanom v r. 1963: „Tento pojem genu sa už ukázal dôležitý v matematickej teórii ‚modules over orders‘. Takže sa ukázalo, že matematická myšlienka zavedená čiste v súvislosti s vypočítateľnosťou má svoju vlastnú vnútornú teoretickú hodnotu.“ Navyše, ako už bolo spomínané, konštruktívny algoritmický prístup má často pedagogický význam.

Iný spôsob, ktorým algoritmický prístup ovplyvňuje matematiku nájdeme pri konštrukci jedno-jednoznačných priradení. Často sa uvádzali nepriame dôkazy, že isté typy matematických objektov sú rovnako početné. V tom prípade priama konštrukcia jedno-jednoznačného priradenia ukazuje, že v skutočnosti platí dokonca viac.

Rozvoj computer science spôsobil nový rozmach diskkrétnej matematiky, najmä kombinatorickej analýzy, rovnako ako aj ostatných oblastí, v ktorých sa momentálne diskrétna matematika rozšiahlo aplikuje.

*) Jedná sa pravdepodobne o článok *How programming difficulties can lead to theoretical advances*, Proc. Sympos. Appl. Math., Vol. XV., pp. 87–94. Amer. Math. Soc., Providence, R. I., 1963. (Pozn. prekl.).

Zmienky o týchto vplyvoch použitia počítačov na matematiku a mnohé ďalšie príklady sa nájdu v nasledujúcom výbere kníh, z ktorých každá obsahuje viacero závažných článkov: [1], [2], [4], [5], [20], [24], [27]. PETER LAX sa vo svojom článku [19] zaoberá vplyvom použitia počítačov na matematickú fyziku.

Ale podľa môjho názoru najdôležitejší vplyv computer science na matematiku je trochu odlišný od všetkých predošlých. Pre mňa je najdôležitejším faktom, že štúdium algoritmov sa stalo živnou pôdou pre nové, zaujímavé matematické problémy. Je to oživenie mnohých oblastí matematiky, ktoré trpeli nedostatkom nových myšlienok. CHARLES BABBAGE, jeden z „otcov“ počítačov, to predvídal už v roku 1864: „Len čo bude existovať analytický stroj (t. j. univerzálny počítač), bude nutne ovplyvňovať ďalší rozvoj vedy. Kedykoľvek sa dosiahne jeho pomocou výsledku, vznikne otázka – Akým výpočtovým postupom sa tieto výsledky dajú dosiahnuť v čo najkratšom čase?“ ([3]). A znovu, GEORGE FORSYTHE v roku 1958: „Použitím prakticky ktorejkoľvek výpočtovej metódy vzniká mnoho matematických problémov. Výpočty majú preto veľký vplyv na samotnú matematiku a dá sa očakávať, že to bude v čoraz väčšej miere ovplyvňovať výskum v matematike ([26])“. GARRETT BIRKHOFF ([4], str. 2) poznamenal, že takéto vplyvy nie sú ničím novým. Hrali dôležitú úlohu v rozvoji matematiky už v starom Grécku.

Zistil som, že keď sa pokúšame o kvantitatívnu analýzu algoritmu, aby sme vedeli ako rýchlo sa bude počítať na počítači, vynára sa spústa pútavých matematických problémov. Iná skupina veľmi zaujímavých problémov sa týka hľadania najlepšieho možného algoritmu z danej triedy (viď nedávny REINGOLDOV prehľad [25]). A jedna z prvých matematických teórií inšpirovaných computer science je teória jazykov, ktorá už obsahuje veľa prekrásnych výsledkov (viď [11], [12]). Vzrušenie, ktoré poskytujú tieto nové teórie je dôvodom, pre ktorý som sa stal informatikom.

Obrátene, matematika má hlboký vplyv na computer science. Prakticky každé odvetvie matematiky tu našlo svoje uplatnenie. Nedávno som pracoval s diskretnými objektami nazývanými „binárne stromy“, ktoré sa často vyskytujú pri niektorých počítačových reprezentáciách. V riešení problému, na ktorý som tu narazil, sa skutočne použila komplexná gama funkcia vynásobená druhou mocninou Riemannovej zeta funkcie ([6]). Takže výsledky klasickej matematiky sa často ukážu užitočnými na veľmi prekvapujúcich miestach.

Mojou najprekvapujúcejšou skúsenosťou z aplikácií matematiky do computer science bol fakt, že sa tak veľa používa diskretná matematika. Taká matematika prakticky úplne chýbala v mojom školskom vzdelaní napriek tomu, že som prešiel veľmi dobrým univerzitným štúdiom matematiky. Temer všetky moje stretnutia s takýmito technikami počas mojich študentských rokov sa vyskytli pri riešení problémov z *American Mathematical Monthly*. Je pochopiteľné, že by som rád vedel, či tradičné osnovy (prednášky z matematickej analýzy a pod.) majú byť revidované, aby zahrňovali viac týchto diskretných manipulácií, alebo či je computer science taká výnimočná v ich častom použití.

Literatúra

- [1] Amer. Math. Society and Math. Assoc. of America, co-sponsors of conference, *The Influence of Computing on Mathematical Research and Education*, August 1973.
- [2] A. O. L. ATKIN and B. J. BIRCH, eds., *Computers in Number Theory*, Academic Press, New York, 1971.
- [3] CHARLES BABBAGE, *Passages from the Life of a Philosopher*, (London, 1864). Reprinted in *Charles Babbage and His Calculating Engines*, by PHILIP and EMILY MORRISON, Dover, New York, 1961; najmä str. 69.
- [4] GARRETT BIRKHOFF and MARSHALL HALL, JR., eds., *Computers in Algebra and Number Theory*, SIAM-AMS Proceedings, 4 (Amer. Math. Soc., 1971).
- [5] R. F. CHURCHHOUSE and J. C. HERTZ, eds., *Computers in Mathematical Research*, North-Holland, Amsterdam, 1968.
- [6] N. G. DE BRUIJN, DONALD E. KNUTH, and S. O. RICE, *The average height of planted plane trees*, in *Graph Theory and Computing*, ed. by RONALD C. READ, Academic Press, New York, 1972, 15–22.
- [7] GEORGE E. FORSYTHE, *The role of numerical analysis in an undergraduate program*, *The American Mathematical Monthly*, 66 (1959), 651–662.
- [8] GEORGE E. FORSYTHE, *Computer Science and Education*, *Information Processing* 68, 1025–1039.
- [9] GEORGE E. FORSYTHE, *What to do till the computer scientist comes*, *The American Mathematical Monthly*, 75 (1968), 454–462.
- [10] K. F. GAUSS, Letter to Enke, *Werke*, vol. 2, 444–447.
- [11] SEYMOUR GINSBURG, *The Mathematical Theory of Context-Free Languages*, McGraw-Hill, New York, 1966.
- [12] SEYMOUR GINSBURG, SHEILA GREIBACH, and JOHN HOPCROFT, *Studies in abstract families of languages*, *Amer. Math. Soc. Memoirs*, 87 (1969), 51 pp.
- [13] DONALD E. KNUTH, *A class of projective planes*, *Trans. Amer. Math. Soc.*, 115 (1965), 541–549.
- [14] DONALD E. KNUTH, *Algorithm and program; information and data*, *Comm. ACM*, 9 (1966), 654.
- [15] DONALD E. KNUTH, *Seminumerical Algorithms*, Addison-Wesley, Reading, Mass., 1969.
- [16] DONALD E. KNUTH, *Ancient Babylonian algorithms*, *Comm. ACM*, 15 (1972), 671–677.
- [17] DONALD E. KNUTH, *George Forsythe and the development of computer science*, *Comm. ACM*, 15 (1972), 721–726.
- [18] DONALD E. KNUTH, *Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.
- [19] PETER D. LAX, *The impact of computers on mathematics*, Chapter 10 of *Computers and Their Role in the Physical Sciences*, ed. by S. FERNBACH and A. TAUB, Gordon and Breach, New York, 1970, 219–226.
- [20] JOHN LEECH, ed., *Computational Problems in Abstract Algebra*, Pergamon, Long Island City, 1970.
- [21] PETER NAUR, „Datalogy“, *the science of data and data processes, and its place in education*, *Information Processing* 68, vol. 2, 1383–1387.
- [22] ALLEN NEWELL, ALLAN J. PERLIS, and HERBERT A. SIMON, *Computer Science*, *Science*, 157 (1967), 1373–1374.
- [23] W. W. PETERSON, *Addressing for random-access storage*, *IBM Journal of Res. and Dével.*, 1 (1957), 130–146.
- [24] Proc. Symp. Applied Math. 15, *Experimental Arithmetic, High-Speed Computing, and Mathematics*, Amer. Math. Soc., 1963.
- [25] E. REINGOLD, *Establishing lower bounds on algorithms - A survey*, *A FIPS Conference Proceedings*, 40 (1972), 471–481.
- [26] PAUL C. ROSENBLUM and GEORGE E. FORSYTHE, *Numerical Analysis and Partial Differential Equations*, *Surveys in Applied Math.* 5, Wiley, New York, 1958.
- [27] *Computers and Computing*, Slaughter Memorial Monograph No. 10, supplement to the *American Mathematical Monthly*, 72, (February 1965), 156 pp.

- [28] J. D. ULLMAN, *A note on the efficiency of hashing functions*, J. ACM, 19 (1972), 569—575.
[29] PETER WEGNER, *Three computer cultures*, Advances in Computers, 10 (1970), 7—78.
[30] J. H. WILKINSON, *Some comments from a numerical analyst*, J. ACM, 18 (1971), 137—147.

Preložil Branislav Rován

vyučování

Problémy výuky fyziky na strojních fakultách

Ivan Šantavý, Brno

V období prudkých kvantitativních i kvalitativních změn ve výrobě jsou kladeny na techniky a zvláště na inženýry zvýšené nároky. Vysoká škola nemůže dnes inženýra vybavit souhrnem poznatků a vědomostí, jež by mu stačily, aby plnil dobře své profesionální úkoly po celou dobu svého produktivního věku. Spíše ho má dobře vybavit po stránce metodologické a rozvinout racionální a morální složky jeho osobnosti tak, aby byl schopen i ochoten doplňovat si své vzdělání paralelně s rozvojem vědy i techniky a v souladu s potřebami společnosti a aby byl adaptibilní, tj. aby se dovedl přizpůsobit požadavkům vznikajícím v jeho profesi. Měl by být schopen změnit případně i své zaměření a naučit se pracovat i v oblastech, pro něž nebyl na vysoké škole připravován. Není pochyby, že tyto požadavky lépe splní inženýr se širokým teoretickým základem než inženýr úzce specializovaný. Rozšíření všeobecného teoretického základu budoucích inženýrů je jedním z hlavních cílů přestavby studia na technických vysokých školách.

V této situaci je paradoxní, že rozsah předmětu Fyzika nejen že nemá být podstatně zvětšen, nýbrž že bude stěží udržen dosavadní stav. Nemyslím si, že tato situace vzniká jen snahou předmětů specializací udržet si počty hodin. Vyplývá i z názorů části učitelů inženýrských předmětů, a to zejména učitelů starších, kteří studovali fyziku před několika desítkami let. Ti jsou přesvědčeni, že fyzikální erudice, kterou by absolventi středních škol při příchodu na vysokou školu technickou měli, kdyby uměli to, co jim střední škola podle učebních osnov měla dát, by byla téměř dostatečná. Myslí si, že je zbytečné, aby se fyzikální vzdělání posluchačů (a mám nyní na mysli především strojní fakulty) v prvních ročnících prohlubovalo v dosavadním, natož pak zvětšeném rozsahu. Ve fyzice by se podle nich měly probírat převážně jen ty její části, které nejsou v jiných předmětech zařazených do programu inženýrského studia dále rozvíjeny, jako např. akustika, optika a atomová fyzika. Domnívám se, že úvahy profesionálních fyziků o nutnosti posílit fyziku pro budoucí inženýry jsou za současného stavu akademické a nevedou k žádoucímu výsledku. Akademickými zůstanou, pokud nebude proveden důkladný rozbor současné úrovně fyzikálního vzdělání uchazečů o inženýrské studium, pokud nebude provedena analýza poslání fyziky při