Sveučilišni diplomski studij: Računarstvo i matematika

Robert Manger[1], *Goranka Nogo*[2], *Saša Singer*[3]

Uvod

U ovom članku želimo vam predstaviti sveučilišni diplomski studij *Računarstvo i matematika* na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu. Osnovne informacije o ovom studiju možete naći na mrežnoj stranici

`http://www.math.hr/Default.aspx?art=3523&sec=433`

Nažalost, po prirodi stvari, "službene stranice" moraju biti birokratski suhoparno napisane. Zato nam je dodatni cilj da vam približimo cijelo područje na presjeku računarstva i matematike, te ilustriramo po čemu se naš studij razlikuje od ostalih studija računarstva u Hrvatskoj.

Sasvim ukratko, to je studij računarstva u okviru matematike. Što to znači? Krenimo od naziva i to kronološkim redom.

Matematika je jedna od najstarijih znanosti. Obično ju zamišljamo kao znanost koja se, u osnovi, bavi apstraktnim teorijama, bez neke veće namjere da bude praktično korisna u svakodnevnom životu. To, naravno, nije istina, još od početka razvoja matematike, a posebno ne u modernija vremena. Međutim, ovaj pogled je tipičan primjer matematičkog načina razmišljanja u kojem, polazeći od konkretnog problema, apstrakcijom dolazimo do **općih** zaključaka, s tendencijom da zaboravimo na početnu motivaciju — a to je **rješenje** onog polaznog problema. Sasvim je svejedno je li taj problem bio "čisto" matematički, ili je stigao "izvana", iz nekih drugih znanosti ili struka.

No, u ovom pogledu ima dio istine. Date li matematičarima neki konkretni problem, sva je šansa da će ga oni detaljno proanalizirati, apstraktno izgeneralizirati, dokazati da rješenje postoji i da je jedinstveno (ako to ide), ili dokazati da rješenja nema. Ako rješenje postoji, dobit ćete i razna svojstva koje rješenje zadovoljava. Možda dobijete i metodu kojom si možete **naći** rješenje, ali taj "prljavi" dio posla, donedavno, više nije bio zanimljiv. Tek pojavom prvih računala, dio matematike počeo se ozbiljno baviti ovim "prljavim" dijelom posla — kako što **efikasnije** i **točnije** izračunati rješenje.

Iz te perspektive, sasvim je prirodno da se filozofsko pitanje "*a što se uopće može riješiti, odnosno, izračunati — u najširem smislu pojma izračunati*" pojavilo **prije** prvih računala!

---

[1] Autor je redoviti profesor na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu; e-pošta: `manger@math.hr`.

[2] Autorica je docent na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu; e-pošta: `nogo@math.hr`.

[3] Autor je izvanredni profesor na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu; e-pošta: `singer@math.hr`.

S druge strane, računarstvo je jedna od najmlađih znanosti — staro je tek oko 70 godina. Naime, toliko je prošlo od stvaranja prvog, barem djelomično programabilnog računala. Danas se smatra da je začetnik ili "otac" računarstva engleski matematičar i logičar Alan M. Turing. Sredinom tridesetih godina prošloga stoljeća, Turing je dao ogroman doprinos korektnoj formulaciji navedenog filozofskog pitanja, kao i odgovoru na njega. Naime, predložio je apstraktno računalo — tzv. Turingov stroj, kao univerzalni model izračunljivosti, odnosno, rješivosti nekog problema. U tom smislu, problem je **algoritamski** rješiv, ako se može konstruirati Turingov stroj koji ga rješava. Osim toga, pokazao je da postoje problemi koji **nisu** rješivi — tzv. neodlučivi (engl. undecidable) problemi. Ovim rezultatima Turing je postavio temelje teorijskog računarstva.

Međutim, nakon što je skinut veo tajne s dokumenata o kriptoanalizi iz Drugog svjetskog rata, pokazalo se da je Turing vrlo aktivno sudjelovao i u projektiranju prvih konkretnih računala u Bletchley Parku. Svrha tih računala — Bombe iz 1938. g. i Colossus iz 1943. g., bila je dešifriranje poruka generiranih njemačkim strojevima za šifriranje Enigma i Lorentz SZ40/ 42 (pogledajte članke [1]–[4] na Wikipediji).

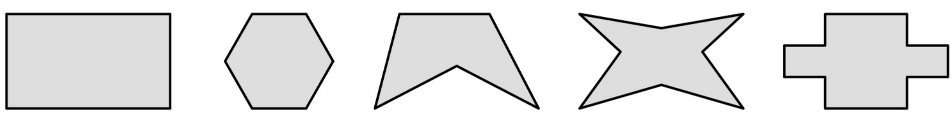
Već i same činjenice da su ove informacije trenutno dostupne skoro svakome i da se lako nalaze mrežnim pretraživačima, pokazuju koliko je računarstvo postalo dio našeg svakodnevnog života. Zadnjih nekoliko desetljeća to je područje znanosti, tehnologije, pa i sociologije, koje se najbrže razvija i mijenja. Pojam "računarstvo" (engl. "computer science") danas uključuje vrlo široki spektar sadržaja i svašta se može podvesti pod taj pojam. Zato i postoji nekoliko studija iz tog područja.

Studij "Računarstvo i matematika" pokriva matematičke aspekte računarstva, baziran je na formalnim matematičkim metodama i teži općim znanjima koja pomažu u praćenju ovako dinamične discipline kao što je računarstvo. Međutim, to **nije** samo studij tzv. teorijskog računarstva, u smislu proučavanja odgovora na ono fundamentalno pitanje "što se sve može izračunati, a što ne", iako je i to katkad korisno za praksu (pogledajte prvi primjer). Pored apstraknih, opća znanja uključuju i vrlo konkretna, praktično primjenjiva znanja na temu "kako se neke stvari mogu što **efikasnije** izračunati". Na primjer, ako vas zanima kako rade web–pretraživači i kako dolaze do poretka relevantnih stranica, dođite na naš studij!

Za ilustraciju razlike između ovog i ostalih studija računarstva, promotrimo sljedeći "programski" zadatak.

Primjer ("Problem popločavanja ravnine", engl. "Tiling problem").

Zadan je neki poligon, koji ne mora biti pravilan i ne mora biti konveksan. Na raspolaganju imamo beskonačno mnogo identičnih podnih pločica tog oblika. Možemo li njima potpuno popločiti cijelu ravninu, bez rupa i razmaka između pločica?



Ako su pločice pravokutne ili, recimo, sukladni pravilni šesterokuti, onda znamo da se to **može** napraviti, tj. odgovor na postavljeno pitanje je "da". Sad bismo smo htjeli napraviti program, kojem zadajemo bilo koji "oblik" pločice (razmislite malo kako bismo zadali oblik), a program treba naći odgovor na postavljeno pitanje.

Zamislite da vam dođe šef neke dizajnerske ili softverske firme i ponudi ovaj posao. Što ćete napraviti? Za očekivati je da će početni stav prema ovom problemu biti ovako nešto: "Hm... Možda i nije baš jednostavno, ali isprogramirat će se". U principu, *spremni* ste ga prihvatiti. Za pregovore o poslu još samo trebate razmisliti o tome koliki je razuman rok za obavljanje tog posla i koliko ga treba (probati) naplatiti. Razmislite..., pa ako poslodavac pristane, potpišite ugovor.

Nažalost, nismo baš "jako fer", jer jedno je sigurno: **nećete** ga isprogramirati! Naime, za ovaj problem je dokazano da je algoritamski **nerješiv**. Drugim riječima, ne postoji algoritam koji bi dao odgovor na ovo vrlo elegantno pitanje.

Dakle, ne potpisujte ugovor! Baš tu je poanta: trebate znati da postoje takvi problemi. Formulacija problema zvuči sasvim razumno i uopće ne upućuje na "podvalu", odnosno, na to da problem nema rješenje. Isto vrijedi i za hrpu sličnih problema, poput Wangovih pločica [5].

Osnovne informacije o studiju

Upis na diplomski studij vrši se prema rang listi kandidata napravljenoj u sklopu razredbenog postupka. Pravo na pristup razredbenom postupku imaju svi kandidati sa završenim preddiplomskim sveučilišnim studijem od najmanje 180 ECTS bodova ili dodiplomskim sveučilišnim studijem (po predbolonjskom sustavu), uz uvjet da su stekli barem 120 ECTS bodova na kolegijima iz matematike, te ostalih struka relevantnih za studij. Za studij Računarstva i matematike relevantne struke su: računarstvo i elektrotehnika.

Razredbeni postupak ne predviđa test provjere znanja, već se rang lista formira na temelju ukupnog broja bodova — to je zbroj bodova zarađenih na temelju ocjena iz položenih kolegija i tzv. dodatnih bodova. Precizna pravila za računanje bodova nalaze se u pravilima upisa. Upis se vrši prema redoslijedu na rang listi, sve do popunjenja upisne kvote. Na primjer, za studij Računarstva i matematike, u akademskoj godini 2011/ 12., upisna kvota je 40 studenata, a upisalo ih se 31.

Osim ovih "birokratskih" uvjeta za upis, postoje i "stvarni" preduvjeti, bez kojih ćete teško pratiti i završiti ovaj studij. Na samom početku, očekuje se da imate dobro i aktivno predznanje iz oba područja — iz matematike i iz računarstva, uključivo i odgovarajuće iskustvo i vještinu u programiranju.

Predviđeno trajanje studija je 2 godine (4 semestra), a standardni nastavni plan studija donosi 120 ECTS bodova. Studij završava izradom i uspješnom obranom diplomskog rada, čime stječete akademski naziv magistra/ magistre računarstva i matematike (mag. inf. et math.).

Koncept studija

Diplomski studij "*Računarstvo i matematika*" je sljedbenik smjera "*Računarstvo*" koji je postojao u sklopu predbolonjskog studija Diplomirani inženjer matematike. Taj program je započeo 1993. godine, bio je atraktivan za studente i pokazao se vrlo uspješnim u formiranju dobrih stručnjaka, pa i znanstvenika, iz područja matematike i računarstva.

Novi program se nastavlja na tu tradiciju, a donosi bitne promjene u dva smjera. Nastavni **sadržaji** su pažljivo ažurirani i osuvremenjeni, tako da prate razvoj računarstva, a uvedeno je i nekoliko sasvim novih tema. Osim toga, jači je naglasak stavljen na matematičke aspekte teorijskog računarstva, čime se popunjava prazina u odnosu na ostale studije računarstva u Hrvatskoj.

Druga bitna promjena je u **režimu studija**, koji je prilagođen modernim europskim i svjetskim standardima. Zahtijeva se prisutnost i aktivno sudjelovanje studenata na nastavi, a posebno se stimulira kontinuirani rad studenata i izvan nastave, kroz sustav domaćih zadaća, kolokvija, studijskih i softverskih projekata. U kasnijoj fazi studija, ovakvi projekti uključuju timski rad i javnu prezentaciju rezultata u obliku seminara, kao bitni dio pripreme za budući posao.

Studij se sastoji od **fiksne** jezgre koja sadrži 16 obaveznih kolegija, **izbornog** dijela od 6 izbornih predmeta koji se biraju između čak 23 ponuđena kolegija i **diplomskog rada**.

Obavezni kolegiji daju temeljno znanje o glavnim relevantnim disciplinama matematike, računarstva i softverskog inženjerstva. Na primjer: *Građa računala*, *Oblikovanje i analiza algoritama*, *Matematička logika*, *Računalna grafika*, *Umjetna inteligencija*, *Baze podataka*, *Izračunljivost*, *Objektno programiranje*, *Softversko inženjerstvo*, *Distribuirani procesi*, *Složenost algoritama*, itd.

Izborni predmeti detaljnije obrađuju pojedine discipline i često se neposredno nastavljaju na obavezne kolegije. Na prvoj godini studija bira se po jedan izborni predmet u svakom semestru, među kojima su: *Kombinatorika*, *Meta-heuristike*, *Uvod u paralelno računanje*, *Matematički softver*, *Operacijska istraživanja*, *Primijenjena statistika*, *Primjena paralelnih računala*.

Na drugoj godini biraju se još po 2 izborna predmeta u svakom semestru, i to iz cijelog skupa ponuđenih kolegija za taj semestar. Na primjer, u ponudi su: *Meko računarstvo*, *Multimedijski sustavi*, *Teorija igara*, *Formalne metode računarstva*, *Računarska statistika*, *Strojno učenje*, *Upravljanje softverskim projektima*, itd.

Takva izbornost daje široke mogućnosti za "profiliranje" i specijalizaciju studija u raznim smjerovima, poput: teorijskog računarstva, umjetne inteligencije, paralelnog i distribuiranog računanja ili modeliranja i analize podataka.

Što dobivate ovom koncepcijom studija? Pored dobrog poznavanja područja matematike bitnih za računarstvo, tj. "teorijskih znanja", upoznat ćete suvremene paradigme i metode za razvoj softvera i steći praktično iskustvo u primjeni modernih tehnologija i alata. Naime, praktični rad je fundamentalni dio većine kolegija na studiju. Osim toga, sadržaj pojedinih kolegija stalno se mijenja u skladu s razvojem tehnologije. Na primjer, prošle godine je u kolegije iz paralelnog računanja uvedeno i programiranje grafičkih kartica, a stigla je i odgovarajuća infrastruktura na kojoj to možete probati.

Tema diplomskog rada je, u principu, vezana s odabirom izbornih predmeta. U većini slučajeva, rad uključuje i izradu softverskog projekta (idealno, za stvarnog korisnika). Naravno, ovisno o sklonostima, rad može biti i teorijskog karaktera. Kao primjer, navodimo nekoliko tema iz prošle akademske godine:

"Crtanje planarnih grafova",
"Rangiranje dokumenata na Internetu",
"Automatsko generiranje programskog koda iz UML dijagrama",
"Distribuirani algoritmi za međusobno isključivanje",
"Meta-heuristike za problem ispunjivosti",
"Heuristika lokalnog pretraživanja za teške probleme",
"Prikaz matematičkih sadržaja u Drupalu",
"Web sustav s jednostavnim praćenjem informacija i korisnika".

Za kraj, kao primjer pristupa rješavanju problema, uzmimo sljedeći zadatak s kolokvija iz Oblikovanja i analize algoritama (ideja je iz [6], Problemi 4.11.5–7, str. 137).

Zadatak ("Većinski element", engl. "Majority element").

Neka je A polje od n elemenata nekog tipa. Na tom tipu **nije** definirana relacija uređaja (poput $\leq$), tako da elemente u polju A smijemo "testirati" samo operatorima jednakosti ili različitosti. Elementarna operacija je test jednakosti ili različitosti jednog para elemenata, a mjera složenosti je **točan broj** takvih testova u ovisnosti o n.

Kažemo da je element x **većinski** ili **dominirajući** element u polju A, ako je strogo preko polovine elemenata u A jednako x, tj. ako je

$$\operatorname{card}\{\, i \mid A[i] = x \,\} > \frac{n}{2}.$$

Koliko različitih većinskih elemenata može biti u polju A?

Sastavite algoritam koji provjerava postoji li dominirajući element u polju A. Ako takav element postoji, algoritam treba vratiti taj element. U protivnom, treba vratiti objekt s imenom `nema`. Zabranjeno je koristiti dodatna polja!

Red veličine složenosti algoritma mora biti $O(n^2)$. Analizirajte složenost vašeg algoritma i pokažite da ona zadovoljava ovaj uvjet.

Napomena: Broj bodova ovisi o složenosti algoritma. Složenost $O(n^2)$ vrijedi najviše 10 bodova. **Bonus**: složenost $O(n \log n)$ vrijedi 10 bodova više!

Razna moguća rješenja. Za početak, očito je da u polju A može postojati najviše jedan većinski element. Dakle, ako postoji, onda je jedinstven, pa je izlaz algoritma korektno definiran. Radi jednostavnosti, uzmimo još da se indeksi elemenata u polju A broje od 1 (a ne od 0, kao u programskom jeziku C).

Već iz teksta zadatka može se zaključiti da različiti pristupi vode do različitih rješenja koja se mogu bitno razlikovati po složenosti.

1. Pristup "grube sile" (ili "iscrpnog traženja") prolazi redom po svim elementima u polju i za svakog od njih broji koliko puta se isti element pojavljuje u ostatku polja. Malo ubrzanje dobiva se ako uočimo da je dovoljno testirati samo elemente koji imaju veće indekse od trenutno izabranog. U oba slučaja, složenost algoritma je reda veličine $O(n^2)$ testiranja.
2. Pristup "sortiranjem polja". Iako među elementima nije zadan uređaj, možemo proizvoljno uvesti neki uređaj u zadani tip. Bitno je samo da različiti elementi ne mogu biti jednaki po tom uređaju. Nakon toga, sortiramo cijelo polje A po tom uređaju, a zatim tražimo najdulji blok jednakih susjednih elemenata u polju. Ako je taj blok dovoljno dug, tj. ima preko $n/2$ elemenata, našli smo većinski element. Složenost ovog rješenja je istog reda veličine kao i složenost izabranog algoritma za sortiranje polja, tj. $O(n^2)$ ili $O(n \log n)$, ali ne i brže od toga. Naime, traženje najduljeg bloka istih susjeda je linearno u n.
3. Pristup "podijeli, pa vladaj". Ako postoji većinski element u cijelom polju, onda on mora biti većinski i u barem jednoj "polovini" polja. Stoga podijelimo polje na dva manja potpolja približno jednake duljine (točne duljine potpolja su $\lceil n/2 \rceil$ i $\lfloor n/2 \rfloor$), rekurzivno pozovemo isti ovaj algoritam na ta dva potpolja, a zatim

provjeravamo dobivene rezultate. Imamo četiri mogućnosti. Ako ni u jednoj "polovini" ne postoji većinski element, onda ga nema ni u cijelom polju. Ako postoji većinski element u jednom ili oba potpolja, onda samo za taj element provjerimo koliko puta se javlja u cijelom polju. To su najviše tri prolaza kroz cijelo "trenutno" polje, a stvar se može i malo ubrzati (razmislite kako). Može se pokazati da je složenost ovog rekurzivnog algoritma reda veličine $O(n \log n)$.

4. Pristup "medijanom polja". Kad bismo sortirali polje A (uz neki proizvoljni uređaj), onda se većinski element (ako postoji) mora nalaziti na "polovini" sortiranog polja, gledano po indeksima. Točnije, mora biti na mjestu $A[\lceil n/2 \rceil]$. To mjesto se obično zove "medijan" polja. No, $\lceil n/2 \rceil$-ti po veličini element u polju možemo naći i bez sortiranja. Postoji i algoritam za to čija je složenost linearna u n, ali je konstanta proporcionalnosti prilično velika (potražite takav algoritam).

5. Pristup "lokalnog popravka" ili "indukcije". Zamislimo sad da elemente obilazimo jednog po jednog i stalno pamtimo samo najboljeg trenutnog "kandidata" za većinski element, s tim da mu brojimo i trenutne "glasove premoći" nad ostalim kandidatima. Zgodna predodžba je obilazak glasača na stranačkoj konvenciji koja bira predsjednika stranke.
   Na samom početku, nema najboljeg kandidata, a broj "glasova" je nula. Onda prvom elementu (onom na mjestu $A[1]$) damo jedan "glas". A zatim, kao u koraku indukcije, popravljamo broj glasova na sljedeći način. Ako je sljedeći element jednak onom kandidatu kojeg trenutno pamtimo, dodamo mu jedan "glas". U protivnom, oduzmemo mu jedan "glas". Ako u nekom trenutku broj glasova padne na nulu, "brišemo" tog kandidata i nastavljamo sa sljedećim elementom, kao da smo na početku posla.
   Kad prođemo kroz cijelo polje, ili nemamo kandidata, pa u polju nema većinskog elementa, ili imamo nekog kandidata (s pozitivnim brojem glasova) i taj je jedini mogući kandidat za većinski element (dokažite da su oba ova zaključka točna). A onda ponovno prođemo kroz cijelo polje i prebrojimo koliko puta se pojavljuje taj element.
   Složenost ovog vrlo elegantnog algoritma je najviše $2n$ testiranja elemenata! Ako uzmete da je minimalni potrebni broj testiranja barem $n - 1$ (treba "pogledati" sve elemente u polju), onda je ovo rješenje skoro optimalno.

Usput, budimo pošteni: na kolokviju se nije ni očekivalo da netko napravi zadnju varijantu rješenja. Ona je pokazana na predavanju, nakon kolokvija.

Zaključak

Diplomski studij Računarstva i matematike na Matematičkom odsjeku zamišljen je tako da bude komplementaran u odnosu na druge studije računarstva i informatike u Hrvatskoj, po tome što se snažno oslanja na matematiku i matematičke aspekte računarstva, dok svi drugi to, dobrim dijelom, nastoje izbjeći.

Naš studij polazi od kompetencija koje su studenti stekli kroz preddiplomski studij matematike i proširuje ih u smjeru računarstva i softverskog inženjerstva, uz naglasak na dugoročna znanja i općenite koncepte, te na apstraktne pristupe rješavanju problema.

Kad ga završite, tj. magistrirate, bit ćete osposobljeni za znanstveni rad u polju računarstva i to u skladu s međunarodnim znanstvenim kriterijima. Vjerovali ili ne, ali naši bivši, pa čak i sadašnji studenti već sudjeluju na konferencijama i objavljuju znanstvene radove! S druge strane, ako vas ne zanima akademska karijera, možete raditi kao softverski inžinjer na raznim poslovima — u znanosti i visokom obrazovanju, softverskoj industriji, gospodarstvu i financijama, državnoj upravi i javnim službama. Vaša osnovna prednost je bezbolno prilagođavanje brzom razvoju tehnologije i znanosti — novim paradigmama, konceptualnim okvirima, metodama i notacijama.

Literatura

[1] —, *Alan Turing*, Wikipedia — The Free Encyclopedia, `http://en.wikipedia.org/wiki/Alan_Turing`

[2] —, *Bletchley Park*, Wikipedia — The Free Encyclopedia, `http://en.wikipedia.org/wiki/Bletchley_Park`

[3] —, *Bombe*, Wikipedia — The Free Encyclopedia, `http://en.wikipedia.org/wiki/Bombe`

[4] —, *Colossus computer*, Wikipedia — The Free Encyclopedia, `http://en.wikipedia.org/wiki/Colossus_computer`

[5] —, *Wang tile*, Wikipedia — The Free Encyclopedia, `http://en.wikipedia.org/wiki/Wang_tile`

[6] G. Brassard and P. Bratley, *Algorithmics*, Prentice–Hall International, Englewood Cliffs, New Jersey, 1988.