Võistlusprogrammeerimine

ehk kuidas kirjutada kiiret koodi

Targo Tennisberg, Katrin Gabrel

0 SISSEJUHATUS

0.1 MIS ON VÕISTLUSPROGRAMMEERIMINE?

Programmeerimise eesmärk on panna arvuti tegema midagi kasulikku. Enamasti on see mingi tegevus, mille muidu oleks ära teinud inimesed. Nemad saavad nüüd aga teha midagi meeldivamat – las masin töötab, tema on rauast. Distsipliinina on programmeerimine üsna noor, kuid ometi läbi teinud pead pööritama paneva arengu, eelkõige kasutatavate vahendite (arvutid ja programmeerimist abistav tarkvara) osas. Kaasaegne riistvara võimaldab „rauast masinal“ sooritada teatud ülesandeid miljardeid kordi kiiremini kui ükski inimene suudaks. Samas mõeldakse pidevalt välja uusi ja keerulisemaid ülesandeid, mis lükkavad tagant ka programmeerimise sisemist võimekust ehk algoritmide ja meetodite arengut.

Uute algoritmide ja meetodite väljamõtlemine loob aga viljaka pinnase programmeerijatele üksteisega mõõduvõtmiseks. Kes kirjutab kiiremini efektiivsema programmi etteantud ülesande lahendamiseks? Selles vallas toimub palju võistlusi, kus pannakse proovile osalejate teadmised algoritmide alal, võimekus oma mõtteid kiiresti programmideks realiseerida ning oskus kirjutada veavaba koodi.

Eestis on selles vallas peamine võistlus Eesti Informaatikaolümpiaad, mis on suunatud eelkõige kooliõpilastele, kuid mille raames toimub ka kõigile avatud üritusi. Edukamatel kooliõpilastel on võimalus osaleda regionaalsel Balti Informaatikaolümpiaadil (kus osalevad Läänemere äärsed riigid) ning Rahvusvahelisel Informaatikaolümpiaadil (IOI).

Ülikoolitasemel on peamine võistlus ACM International Collegiate Programming Contest (ACM-ICPC). Neil, kes on koolid juba lõpetanud, on võimalik osaleda erafirmade korraldatud võistlustel, nagu Google Code Jam või Facebook Hacker Cup. Peale selle on olemas hulk internetipõhiseid võistluskeskkondi, kus toimuvad regulaarsed üritused. 2017 alguse seisuga on neist tuntumad CodeForces, TopCoder, HackerRank ja CodeChef.

Programmeerimisvõistlustel on üldjuhul ajapiirang (näiteks 2-5 tundi), mille jooksul on antud lahendamiseks hulk ülesandeid (enamasti 3-6). Ülesannetele antakse sisendandmed, mille põhjal tuleb leida väljundandmed, mille õigsust kontrollitakse.

Lahendus võib olla iseseisev programm (sel juhul loeb ta sisendit ja väljundit failist või standardsisendist ja kirjutab väljundi teise faili või standardväljundisse) või siis lihtsalt funktsioon, mis lingitakse testprogrammi külge ja mida testprogramm välja kutsub, andes sisendi ette funktsiooni parameetritena.

Suurim väljakutse on sageli mitte lihtsalt vastuse leidmiseks sobiva lahenduse leidmine, vaid sobiva **kiire** lahenduse leidmine. Lahendustel on antud ajalimiit, mille jooksul tuleb vastus väljastada, tänapäeval tavaliselt sekund või paar.

Siin on üks lihtne näide võistlusstiilis ülesandest:

> Malelaual peab ratsu liikuma N käiguga ühelt etteantud väljalt teisele. Mitu erinevat teekonna võimalust tal selleks on?

Kui võistleja on lahenduse valmis programmeerinud, esitab ta selle testimiseks. Olenevalt võistluse formaadist võib testimine toimuda kas kohe või lahendamisaja lõpus. Kohese testimise puhul

öeldakse võistlejale, kas programm läbis testid või mitte, nii et tal on võimalus oma lahendust parandada.

Testimisel on üldjuhul neli võimalikku tulemust:

- OK – kõik töötas õigesti.
- Vale vastus – programm ei väljastanud oodatud tulemust.
- Ajalimiit ületatud – programm ei lõpetanud ajapiiri jooksul oma tööd.
- Käivitusaja viga – programmi töös juhtus viga (näiteks mittelubatud mälupiirkonnast lugemine või nulliga jagamine), mis ei lasknud tal tööd lõpetada.

Mõnedel võistlustel on ülesande eest punktide saamiseks vaja korrektselt läbida kõik testid, teistel saab vastavalt läbitud testide arvule osalise arvu punkte. Suurima punktide arvuga võistleja saab auhinna või siis lihtsalt au ja kuulsust.

Selle teksti kirjutamise ajal on maailma parim võistlusprogrammeerija Gennadi Korotkevitš Valgevenest, kes praegu esindab St.Peterburgi ITMO ülikooli. Korotkevitš hakkas olulisi võistlusi võitma juba 11-aastaselt, sai IOI-lt kokku kuus kuldmedalit ning on hetkel olulisematel võistlussaitidel maailma kõrgeima reitinguga.

Eesti seni edukaim võistlusprogrammeerija on olnud Martin Pettai, kes sai aastatel 1999-2002 IOI-lt kolm kuld- ja ühe hõbemedali.

0.2 MIDA SIIT ÕPIKUST LEIDA VÕIB?

Käesoleva õpiku peamine eesmärk on ehitada sild programmeerimisoskuse ja teoreetilise arvutiteaduse vahele. Eesti keeles on mitmesuguseid programmeerimist õpetavaid raamatuid, kus keskendutakse programmeerimistehnikale. Samuti on ilmunud arvutiteaduse sissejuhatavaid õpikuid, kus räägitakse mitmesugustest teoreetilistest küsimustest.

Ka isiklikult olen ma elus kohanud paljusid inimesi, kes jäävad kas ühte või teise serva: ühel pool on puhtad praktikud, kes pole põhjalikumat teooriat omandanud või on selle unustanud kui „mittevajaliku“. Teisel pool on teoreetikud, kes tunnevad paljusid algoritme, kuid kellele kuluks ära rohkem praktilist kogemust nende realiseerimisel.

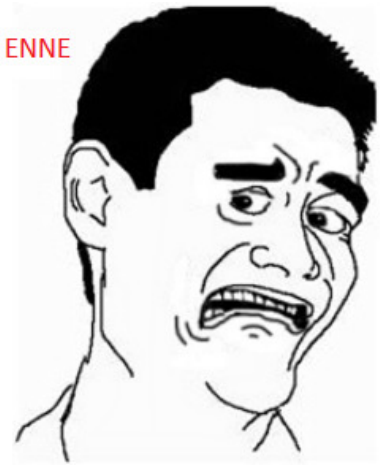
Siin raamatus tuuakse teoreetiline ja praktiline pool kokku ja räägitakse mõlemast. Käsitletakse nii võtteid efektiivsemaks programmeerimiseks kui ka algoritme, mis võimaldavad esmapilgul võimatuid ülesandeid mõistliku ajaga lahendada.

Tutvustatakse mitmeid algoritme ja andmestruktuure, millest mõned kuuluvad arvutiteaduse parimate saavutuste sekka, kuid teoreemide ja tõestuste asemel on siin hulk praktiliselt läbiprogrammeeritavaid näiteid. Raamatuga kaasas olevas lisas on võimalik tutvuda paljude näidisprogrammidega ja lahendada mitmesuguse raskusega katseülesandeid enesekontrolliks.

Õpiku ülesehitus võimaldab seda kasutada struktureeritud kursustel, kus õpetatakse programmeerimist või arvutiteadust üldiselt või siis tegeletakse spetsiifiliselt programmeerimisvõistlusteks ettevalmistamisega. Teiselt poolt võiks see olla kasutatav praktilise käsiraamatuna, kust on vajadusel võimalik ühe või teise algoritmi kohta infot leida.

Tegemist ei ole programmeerimise algkursusega, eelduseks on, et lugejad on algtasemel programmeerimist juba õppinud ning oskavad oma lemmikkeeles põhiasjadega toime tulla.

Raamatust võivad kasu saada mitmesugused huvilised:



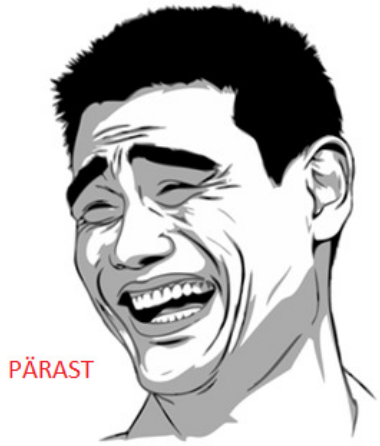
Esiteks **kooliõpilased**, kellel on olemas teadmised programmeerimise põhialustest ja kes soovivad teistega mõõtu võtta. Raamat sisaldab suurel hulgal teadmisi, mida on vaja informaatikaolümpiaadil. 2016. aasta lõpu seisuga on raamatu esimese osa läbitöötamine piisav, et olla edukas Eesti olümpiaadidel ja teine osa võimaldaks koju tuua medali rahvusvahelistelt võistlustelt.

Teiseks **üliõpilased ja kraadiõppurid**, kes soovivad täiendada oma teadmisi arvutiteaduses, neid samal ajal praktiliste oskustega seostades. Õpik pakub mitmeid kasulikke lähenemisviise, mis võimaldavad uurimistöödes vajalikke programme efektiivsemalt kirjutada.

Kolmandaks **professionaalsed programmeerijad**, kes soovivad omandada algoritmiliselt keerulisemate ülesannete lahendamise oskust. Võistlustel ja olümpiaadidel programmeerimine on nagu rallisõit, kus läheb vaja spetsiifilisi oskusi. Igapäevane tavaprogrammeerimine mõnes ettevõttes on selle kõrval nagu liinibussi või takso juhtimine – sarnaste elementidega, aga siiski mitte päris sama. On aga olukordi, kus rallioskused kuluvad ka igapäevaelus marjaks ära. Olen ise oma karjääri jooksul nii mõnelgi korral suutnud „võimatut“ teha ja mõne asja kümneid või sadu kordi kiiremini tööle panna, samal ajal kui teised olid juba valmis massiivseks täiendava riistvara ostuks. Ja kes ei tahaks vahel kangelane olla :)

Neljandaks **õpetajad ja õppejõud**, kes soovivad oma õpilasi olümpiaadideks ja võistlusteks ette valmistada. Raamat on üles ehitatud struktuurselt, iga järgnev peatükk kasutab ja täiendab eelmistes õpitut. Tekstis on palju ülesandeid, mida võib anda lahendamiseks nii grupitööna kui individuaalselt.

Viiendaks **ambitsioonikad spetsialistid**, kes soovivad minna tööle mõnesse maailma „kõrgliiga“ tehnoloogiaettevõttesse. Ettevõtetes nagu Google, Facebook või Palantir on tavaline, et tööintervjuudel antakse kandidaatidele ülesandeid näiteks graafiteooria või dünaamilise planeerimise vallast. Seega on intervjuu läbimiseks vaja ka võistlusprogrammeerimise alaseid teadmisi. Eestis pole nii põhjalik valmistumine tööintervjuudeks veel väga levinud, aga maailma tipptasemel juhtub sageli, et kandidaadid õpivad intervjuudeks nädalaid või isegi kuid.



Ja lõpuks kõik ülejäänud, kellele meeldib programmeerimine ning teadmiste omandamine.

Raamat on jagatud kaheks osaks ja neist kumbki kuueks peatükiks, millest igaüks käsitleb konkreetset teemat.

Iga peatükk sisaldab järgmisi osi:

1. Tekst, mis avab teema olemust ja teoreetilist tausta.
2. Näidisülesanded, kus põhjalikult selgitatakse, kuidas kirjutada programme vastavate algoritmide realiseerimiseks.
3. Veel mõned ülesanded, kus on antud lahenduse idee ja viited näidislahendustele.
4. Ülesanded enesekontrolliks (või kontrolltöödeks akadeemilises keskkonnas). Kõigi ülesannete lahendusi saab esitada *online* võistluskeskkonnas, kus neid automaatselt testitakse.

Kõik näited on saadaval kolmes programmeerimiskeeles:

- C++ kui võistlustel ja olümpiaadidel enim kasutatav keel
- Java kui Eesti tarkvaratööstuses enim kasutatav keel
- Python kui hariduses ja programmeerimise õppimisel populaarne keel.

Tekstis toodud näited kasutavad C++ varianti, teistes keeltes kirjutatud vasted on saadaval lisamaterjalina.

0.3 KONTROLLÜLESANNETE LAHENDAMINE

Iga peatüki lõpus on toodud rida kontrollülesandeid. Neid on võimalik lahendada omal käel või kui raamatut kasutatakse õppetöös, saab kursuse juhendaja korraldada nende põhjal kontrolltöö.

Ülesandeid saab lahendada online võistluskeskkonnas CodeForces, kus:

- Igale ülesandele on üles seatud automaatsed testid.
- Kasutajad saavad veebibrauseri kaudu esitada oma koodi ning saavad kohest tagasisidet selle kohta, kas programm läbis testid või ei.

Ülesannete lahendamiseks:

- Loo CodeForces'i konto (või logi sisse Google'i kontoga)
- Ühine Eesti Võistlusprogrammeerimise grupiga aadressil http://codeforces.com/group/jODaK73gf0/
- Kontrolltööde jaoks on loodud võistlused, vali neist asjakohane

Lahendused peavad lugema andmed standardsisendist ja kirjutama vastuse standardväljundisse. Kui väljundisse kirjutada veel midagi muud, on tulemuseks tõenäoliselt *Wrong Answer*. Kõik testid vastavad ülesande kirjelduses toodud spetsifikatsioonile ja nende korrektsust eraldi kontrollima ei pea.

0.4 KASUTATUD ÜLESANDED JA PILDID

Näidisülesanded kuuluvad enamikus arvutiteaduse klassikasse loodud spetsiaalselt käesoleva raamatu jaoks, osad on ka taaskasutatud mõnelt Eesti olümpiaadilt. Teistest allikatest pärit ülesanded on vastavalt viidatud.

Kontrolltööde ülesanded on loodud kas selle raamatu jaoks või kohandatud teiste riikide võistlustelt. Kuna enamasti on teiste võistluste ülesanded toodud koos lahendustega, pole neid raamatust otse viidatud, viite saab küsimise peale autoritelt.

Kasutatud pildid on võetud vabadest pildipankadest nagu Pixabay, Pexels, Wikimedia Commons jt, kus nad on antud maailmale kasutamiseks CC0 litsentsi alusel.

0.5 AUTORITEST

Targo õppis programmeerima aastal 1990, mil tal oli ainult tekstirežiimi võimaldav kasutatud IBM XT arvuti, millel polnud mänge, mida mängida. Lahenduseks oli ise mängud kirjutada, mis eeldas omakorda programmeerimise äraõppimist. Pärast seda on ta programmeerinud igasuguseid asju klaasilõikepinkidest ja lennuki elektrisüsteemidest dokumendihalduse ja mobiilirakendusteni. Praegu

juhib Targo ka Eesti Informaatikaolümpiaadi. Kuna talle meeldib rohkem endast rääkida, tähistab tekstis „mina“ enamasti Targot.

Katrin huvitus mängudest ise vähem, kuid kirjutas neid oma nooremale vennale. Hiljem on ta õppinud arvutuslingvistikat ja töötanud süsteemianalüütiku ning programmeerijana, samuti tegelenud võistlusprogrammeerimise õpetamisega Eesti Informaatikaolümpiaadi raames.

0.6 TÄNUAVALDUSED

Tahame tänada paljusid inimesi, kes raamatu valmimisele kaasa aitasid:

- Mihkel Kree ja Jaak Vilo organisatsioonilise toetuse eest,
- Härmel Nestra, Tähvend Uustalu, Liisa Jõgiste, Oliver-Matis Lill, Tiina Kull ja Heno Ivanov sisuliste soovituste ja veaparanduste eest,
- Toomas Tennisberg ja Oliver Tennisberg ülesannete läbilahendamise ja testide koostamise ning kontrollimise eest.

0.7 VEAPARANDUSED JA SOOVITUSED

Tegemist on elava dokumendiga, millest loodame aja jooksul uusi versioone luua. Valdkond areneb kiiresti, samuti võib esile tulla vajadusi materjali täpsustamiseks, mitmesuguseid vigu jne.

Kõik parandused ja soovitused võib saata aadressil targot@gmail.com.