

DatZ4020 - Lietiškie algoritmi: Syllabus, 2019.g. rudens

CMU līdzīgu kursu sauc “Algoritmi īstajā pasaulē” (*Algorithms in the Real World*), kas labi izteic kursa būtību, jo dažādi algoritmi var kļūt “lietišķi” un tad to darbināšanā ir svarīgs ne tikai pareizuma pierādījums un asimptotiskā laika sarežģītība, bet arī atmiņas izlietojums, universalitāte, spēja ar heuristikām apstrādāt atsevišķus grūtus gadījumus.

Kurss “Lietiškie algoritmi” stāsta par algoritmiem, kurus ikdienā darbina lietotāju darbstacijās un serveros, kuru veikspēja vistiešāk iespaido datoru lietotāja pieredzi, piemēram, meklējot, pārraidot vai saspiežot un atspiežot dokumentus un multimediju saturu. Kaut arī šo algoritmu “noklusētās” implementācijas ir labi pazīstamas un pieejamas kā gatavas lietojumprogrammas un bibliotēkas visdažādākajās operētājsistēmās un programmēšanas platformās, to variācijas un nestandarta lietojumi joprojām iedvesmo jaunu programmatūras risinājumu radītājus.

Lietiškos algoritmus derīgi saprast dažādām ļaužu grupām un IT pasaulē nodarbinātajiem. *Datorzinātnieki* un arī topošie datorikas magistranti studē algoritmu grāmatas ([?] u.c.), matemātiski analizē un pamato algoritmu darbības pareizumu un ātrdarbību, meklē jaunus algoritmus. *Programmētāji* parasti izvairās no plaši pazīstamu algoritmu atkārtotas kodēšanas, jo tas var būt nevajadzīgs laika patēriņš, kā arī algoritmos, kas uzrakstīti no mācību grāmatas bieži nav izķertas kļūdas, robežgadījumi un reālās pasaules ierobežojumi. Programmētājiem toties jāvar izvēlēties algoritmu bibliotēkas un tās pareizi jāizsauc, jāapzinās, kurās vietās algoritma izvēle atbilst vienam vai otram lietojuma mērķim. Visbeidzot *IT konsultanti* palīdz integrēt dažādu piegādātāju risinājumus; viņiem jāsaprot konfigurējamo parametru sekas.

Lai noderētu minētajām IT profesionāļu lomām, “Lietiškie algoritmi” gan iepazīstina ar algoritmu vispārīgajām īpašībām, gan arī mudina tās izmēģināt praktiski. Dažreiz jāvar teorētiski analizēt algoritma veikspēja, piemēram, atbildot uz jautājumu, vai “stiprāka” datu saspiešana ietaupa pietiekami daudz vietas, lai attaisnotu lielāku CPU resursa patēriņu. Citreiz var praktiski izmēģināt, cik garu “slepeno ziņojumu” var noslēpt noteikta lieluma attēla failā, izmantojot steganogrāfijas metodes; vai šiem failiem var pievienot “ūdenszīmes”, kuru klātesamību var pārbaudīt arī pēc attēla pārveidojumiem.

Kursa mērķi

Lai sagatavotos patstāvīgai algoritmu grāmatu studēšanai, iegūtu praktisku pieredzi izsaukt, konfigurēt un pielāgot dažus bieži lietotus algoritmus kā arī saprastu kontekstu, kurā ar tiem strādā programmatūras produkti un integrēti risinājumi, kursu sadalām 6 tēmās:

1. Bezzudumu saspiešana
2. Zudumradošā saspiešana
3. Kļūdu korekcija
4. Kriptogrāfija
5. Lineārā programmēšana
6. Meklēšana virknēs

Šīs nodaļas nav vienīgā iespējamā izvēle (sal. [?] – šī kursa sākotnējās idejas autora mājaslapu Karnegī-Melona universitātē), tomēr LU piedāvātais kurss domāts bakalauru studijām. Kursā aplūkojamie algoritmi veido savstarpēji saistītu sistēmu, kura sākas ar idejām, kas lietojamās individuālā dokumentu vai mediju failu saspiešanai, turpinās ar to drošu nogādāšanu datortīklos un visbeidzot aplūko svarīgas tehnoloģijas, ko lieto pārsvarā servera pusē, lai galalietotājs varētu efektīvi sameklēt tos dokumentus un mediju failus, ko vēlas atrast. Uz šiem algoritmiem balstās pazīstamu uzņēmumu, piemēram, Google un Netflix populārākie pakalpojumi.

Nodarbību tēmas

- **2019-09-03. Bezzudumu saspiešana – 1.** Informācijas teorijas jēdzieni, entropija. Hofmana saspiešana. Vispārīgā prefiksu saspiešana un prefiksu koki. Aritmētiskā saspiešana. *Pasaka:* Kādas atvērtā koda implementācijas pieejamas šī kursa algoritmiem, program-mēšanas vides, testēšana un veiktspējas profilēšana.
- **2019-09-10. Bezzudumu saspiešana – 2.** LZ77 un Lempela-Ziva-Velča algoritmi. Empīriskā saspiešanas mērīšana (biti uz simbolu). Entropija un saspiešanas teorētiskā robeža. *Pasaka:* Saspiežamības atkarība no failu žanra; Kalgari korpuss. LZW un GIF - kas ir “patentēti” algoritmi un failu formāti, vārda brīvība algoritmu pasaulē.
- **2019-09-17. Bezzudumu saspiešana – 3.** Berouza-Vīlera algoritms. *Pasaka:* Arhivēšanas rīku ‘zip’, ‘bzip2’ u.c. salīdzinājums. Kas ir PNG saspiešanas līmenis (compression level). HTTP sūtāmo failu un biroja programmu formātu (docx, nevis doc) saspiešanas sekas.
- **2019-09-24. Zudumradošā saspiešana – 1.** Diskrētais kosinusu pārveidojums, JPEG algoritms. *Pasaka:* Krāsu kodēšana, acīm atšķiramas atšķirības, Raw un JPEG dati foto un video aparatūrā.
- **2019-10-01. Zudumradošās saspiešana – 2.** Solomona-Rīda algoritms. *Pasaka:* MP3 un video formāti, to DRM (*digital rights management*). Redzamās un neredzamās ūdenszīmes mediju failos (*watermarking algorithms*); to noturība atkarībā no saspiešanas veida.
- **2019-10-08. Kļūdu korekcijas algoritmi – 1.** Heminga kods, piemēri un vispārīgais gadījums. Rīda-Solomona kodi. *Pasaka:* Steganogrāfija jeb informācijas paslēpšana saturīgi nesaistītā failā. Kā aizsargāt uzņēmumus un iestādes pret konfidencialu datu noplūdēm: Data Leak Prevention (DLP).
- **2019-10-15. Kļūdu korekcijas algoritmi – 2.** Galīgu lauku jēdziens. Grafu kodi un Tornado kodi. *Pasaka:* TCP/IP saimes tīklošanās protokoli, kuros izmanto redundanci.
- **2019-10-22. Kriptogrāfija – 1.** Difi-Helmaņa publisku atslēgu apmaiņa. Semestra vidus eksāmens (*Midterm exam*). *Pasaka:* Publisko atslēgu algoritmi HTTP un SMTP (Weba un epasta) lietojumos.
- **2019-10-29. Kriptogrāfija – 2.** TLS 1.3 un PGP standartos izmantotie algoritmi. *Pasaka:* Heartbleed bug un OpenSSL bibliotēka; kriptogrāfijas bibliotēku ievainojamības.
- **2019-11-05. Lineārā programmēšana – 1.** Optimizācijas problēmu veidi. Lineārās programmēšanas vizualizācija zemām dimensijām. Simpleksalgoritma ievads. *Pasaka:* Transporta, preču ražošanas u.c. uzdevumi, kas noved pie lineārās optimizācijas.
- **2019-11-12. Lineārā programmēšana – 2.** Simpleksu algoritms, tā korektums. Primārā un duālā lineārā programma. *Pasaka:* Algoritmu sarežģītība (laika un telpas prasības aprēķina veikšanai atkarībā no ievades datu lieluma); praksē populāri algoritmi, kam ir slikti “sliktākie gadījumi”.
- **2019-11-19. Lineārā programmēšana – 3.** Elipsoīda algoritms. Iekšējā punkta metode un tās varianti (afīnā mērogošana, potenciāla redukcija, centrālā trajektorija). *Pasaka:* Lineāri modeļi Google meklēšanas rezultātu ranžējumā (*Page Rank*), Netflix rekomendācijās utml.

- **2019-11-26. Meklēšana virknēs – 1.** Algoritmisko problēmu definīcijas. Naivais algoritms. Knuta-Morisa-Prata algoritms, tam veidojamās datu struktūras (tabuliņas). Laika sarežģītības salīdzinājums.
Pasaka: Vienkāršs zināšanu pārvaldības (*knowledge management*) uzdevums – kā nepazaudēt pašam savus dokumentus. Teksta dokumentu indeksācijas rīku salīdzinājums un to ātrdarbība.
- **2019-12-03. Meklēšana virknēs – 2.** Bojera-Mūra algoritms. Laika sarežģītības salīdzinājums ar KMP.
Pasaka: Antivīrusu skenēšana, tradicionālās AV produktu vīrusu signatūras. Kā iepakot vīrusus, lai no tām izvairītos.
- **2019-12-10. Meklēšana virknēs – 3.** Dinamiskā programmēšana. Ukkonena algoritms. Edsgera Deikstras (Edsger W. Dijkstra) algoritms.
Pasaka: Regulāru izteiksmju meklēšanas lietojumi DLP produktos (kā pasargāt organizācijas datortīklu no cilvēku vārdu, adresu, telefonu u.c. noplūdes). Kā rakstīt regulāras izteiksmes, lai DLP skenēšana būtu efektīva. Kāpēc kredītkaršu datu noplūdes parasti nemeklē ar regulārām izteiksmēm.
- **2019-12-17. Pārskata nodarbība.** Savācam atgriezenisko saiti, uzklaušām kursa dalībniekus, atbildam uz jautājumiem.
Gala eksāmens (*Final exam*).

Kursa vērtēšana

- 50% no vērtējuma veido 4 mājas darbi, kas vienmērīgi sadalīti semestra laikā.
- 15% no vērtējuma veido Semestra vidus eksāmens (45 minūtes, mājasdarbiem līdzīgi uzdevumi, drīkst izmantot materiālus, bet eksāmena laikā nedrīkst sazināties ar citiem cilvēkiem).
- 25% no vērtējuma veido Gala eksāmens (90 minūtes, drīkst izmantot materiālus).
- 10% no vērtējuma veido nodarbību apmeklējums un līdzdalība lekciju ekspressaptaujās – šis vērtējums pienākas ikvienam, kurš atnācis un aktīvi piedalās nodarbībā. Ekspresaptaujās vērtē godprātīgu piedalīšanos nevis pašas atbildes, jo aptaujas var būt par nupat izstāstīto materiālu.

Nokavēšanās politika. Mājas darbi jāiesniedz paredzētajā termiņā (parasti termiņš ir 13 dienas pēc mājasdarba izplatīšanas nodarbības jeb pusnakts tieši pirms aiznākamās nodarbības). Par mājas darbiem, kas nokavēti par laiku līdz 3 diennaktīm no atbilstošā mājas darba atzīmes tiek atskaitīti 10% no vērtējuma; par laiku līdz 7 diennaktīm –20% no vērtējuma. Abi eksāmeni jāraksta tiem paredzētajā laikā. Studentiem, kuriem ir pamatots neierašanās iemesls (darba nespējas lapa, dekanāta lūgums augstskolas atzītās īpašās situācijās), var paredzēt iespēju tos rakstīt citā laikā, bet ar citu uzdevumu lapu.

Atzīmi par kursu aprēķina, visos vērtējumos iegūtos procentpunktus dalot ar 10 un aprēķinot apakšējo veselo daļu. Vērtējumu "10" ir pat teorētiski sarežģīti iegūt, vācot procentus aprakstītajā veidā. Papildus 10% vērtējumu par kursu piešķir studentiem, kuri izpildījuši mājasdarba neobligāto daļu - piemēram, pierādījuši kādu sarežģītu apgalvojumu par algoritmiem vai uzprogrammējuši kādu neobligāto vingrinājumu.

Akadēmiskā godīguma politika

Mājasdarbu uzdevumus un to risināšanas pieejas studenti drīkst apspriest savā starpā kā arī ar cilvēkiem, kuri paši kursā nepiedalās. Viens no kursa uzdevumiem ir mācīties komunicēt par algoritmiem. Mājasdarba teksts jāraksta patstāvīgi, izmantojot tikai savas piezīmes, Internetā brīvi pieejamus resursus (uz kuriem mājasdarbā pareizi jāatsaucas), paša veidotas datorprogrammas un to izvades datus. Mājasdarba rakstīšanas laikā nedrīkst sazināties ar citiem cilvēkiem vai norakstīt kaut ko no svešām piezīmēm neatkarīgi no saziņas kanāla. Līdz kursa beigām (t.i. nākamā semestra pirmajai mācību dienai) kursa dalībniekiem aizliegts publiskot Internetā savus mājasdarbu atrisinājumus vai citos veidos padarīt tos pieejamus citiem kursa dalībniekiem. Semestra vidus un gala eksāmenā var ņemt līdzi piezīmes, bet nevar sazināties ar citiem cilvēkiem vai lietot elektroniskas ierīces.