

"Algoritmid ja andmestruktuurid" (IAS0090)

1. Kursuse eesmärk

Kursuse eesmärgiks on anda üliõpilastele tänapäeva tarkvarainsenerile vajalikud alusteadmised ning praktilised oskused. Kuna kõik näited ja osaliselt ka algoritmide kirjeldused tuginevad C-keelele, siis eeldatakse viimase oskust vähemalt algtasemel.

Kuna programmeerimine on praktiline oskus, mida saab omandada ainult ise tarkvara koostades, on antud kursuses praktikumidele pööratud erilist tähelepanu. Lõpliku hinde määramisel võetakse arvesse niihästi eksami tulemusi kui ka praktikumides tehtud tööde kvaliteeti.

2. Loengutes käsitletavat teemasid

1. Täiendavat materjali C-keelest: viidad, dünaamiline mälujaotus, tekstitöötlus, andmeagregaadid (*struct* ja *union*), viidad funktsioonidele, tehned bittide tasemel.
2. Andmestruktuurid ja andmetöötluse põhilised operatsioonid. Massiiv ja ahelloend kui andmestruktuuride ehitamise alused. Programmeerimisevõtted ja –näited ning nende analüüs.
3. Ahelloendite variatsioone. Mõningad spetsiifilised andmestruktuurid: hõre maatriks, ringpuhver jne. Serialiseerimine.
4. Abstraktsed andmetüübid. Loend, magasin, järjekord, sõnastik. Näiteid magasin kasutamisest ja realiseerimisest.
5. Algoritmide keerukuse hindamine. Keskmise juhu ja halvima juhu analüüs. Hinnang $O(f(n))$.
6. Rekursioon ja rekursiivsed algoritmid. Rekursiivsete algoritmide asendamine iteratiivsetega.
7. Järjestikotsimine ja selle efektiivsuse tõstmise võimalused. Iseorganiseeruv järjestikotsimine. Täiendavate viitude kasutamine järjestikotsimisel. Kahendotsimine.
8. Puud. Kahendpuud ja nende esitamine arvuti mälus. Puu tippude lisamine ja eemaldamine. Puu läbikäigu algoritmid. Paljuharulised puud ja nende taandamine kahendpuuks. Kahendpuust otsimise efektiivsus.
9. Puu tasakaalustatuse probleem. AVL-puud ja nende konstrueerimine. Pööramise operatsioonid.
10. Iseorganiseeruvad puud. Harali puu, tema konstrueerimine ja kasutamine.
11. B-puud. 2-3-4 puu kui B-puu erijuht. Operatsioonid B-puuga: tippude lisamine ja eemaldamine.
12. B*-puud ja operatsioonid nendega.
13. Välismälus olevate andmetega seotud probleemid. Indekseeritud järjestikfailid. B+ puu ja operatsioonid temaga. Konteineri pakkimise ülesanne.
14. Puna-must puu, tema konstrueerimine ja omadused.
15. Digitaalne puu ja *trie*-puu. Konstrueerimine ja kasutamine. Paljuharulised *trie*-puud. Eksistentsitabelid. Kolmikpuu.
16. Prioriteetidega järjekord kui abstraktne andmetüüp. Kuhi kui ülespoole järjestatud täielik puu. Kuhi kui massiiv. Vasakule kaldus puu. Binoompuu ja binomiaalne kuhi.
17. Paiskpaigutuse alused ja põhiline probleemistik. Ideaalne paiskpaigutus. Tabeli pikkuse ja paiskfunktsiooni valik. Paiskfunktsioonide koostamise meetodid: jagamise meetod, ruutkeskmise meetod, voltimise meetod. Kokkupõrke probleemi lahendused: aheldamine ja

avatud adresseerimine. Lineaarne proovimine ja kahekordne paiskpaigutus. Laiendatav paiskpaigutus.

18. Lihtsamad sortimise algoritmid: valikuga, vahelepanekuga, mullmeetodil. Shelli meetod. Sortimine mestimisega. Kiisortimine. Sortimine kuhja abil. Jaotamisega sortimine. Väline sortimine tasakaalustatud paljukanalilise mestimisega.
19. Tekstist otsimise probleem. Elementaarsed algoritmid. Boyer-Moore'i meetod. Rabin-Karpi meetod. Indeksipuu.
20. Andmete pakkimine. Korduste arvu sissehitamine. Huffmani kood. Adaptiivne kodeerimine. Lempel-Zivi meetodi alused.
21. Algoritmide tüübid. "Jaga-ja-valitse" algoritmid. Dünaamilise programmeerimise põhimõtted. Ahned algoritmid. Randomiseeritud algoritmid.

3. Praktikumid

Vt. [siit](#).

4. Õppekirjandus

4.1. Programmeerimine C-keeles

Sobivaid raamatuid on väga palju, neid leidub nii kauplustes kui ka raamatukogudes ja nad on kõik kasutatavad. Kuivõrd C-d saab vaadelda C++ alamhulgana, võib kasutada ka enamikku C++ õpikuid, jättes klasse kirjeldavad peatükid lihtsalt vahele.

4.2. Andmestruktuurid ja algoritmid

Selleteemalist kirjandust on maailmas välja antud küllalt palju kuid paraku ei ole olemas õpikut, mille sisu üks-üheselt kattuks käesolevas kursuses esitatava materjaliga. Paljudel juhtudel on baaskeeleks võetud mitte C vaid C++. See loob täiendavaid raskusi, kuid need ei ole liiga suured.

Nii oma suunitluse kui ka sisu poolest võiks soovitada:

1. Mark Allen Weiss "**Data structures and algorithm analysis in C++**", 2006, Addison-Wesley Pub Co, ISBN 0321397339.
2. Adam Drozdek "**Data structures and algorithms in C++**", 2001, Brooks/Cole, ISBN 0534375979
3. Robert Sedgewick "**Algorithms in C. Fundamentals, data structures sorting, searching**", 1997, Addison-Wesley Pub Co, ISBN 0201314525. Raamat on tõlgitud ka vene keelde.
4. Robert Sedgewick "**Algorithms in C++. Fundamentals, data structures sorting, searching**", 1998, Addison-Wesley Pub Co, ISBN 0201350882. Erineb eelmisest peamiselt näidete poolest: C asemel on kasutatud objekt-orienteeritud lähenemisviisi ja keelt C++.

Täiendavateks allikateks võiks lugeda:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein "**Introduction to algorithms**", 2001, The MIT Press, McGraw-Hill Book Company, ISBN 0262032937.
2. Kyle Loudon "**Mastering algorithms with C**", 1999, O'Reilly Media, ISBN 1565924533.
3. Jüri Kiho "**Algoritmid ja andmestruktuurid**", 2005, Tartu Ülikooli Kirjastus, ISBN 9949110335, täistekst internetis.
4. Rein Jürgenson "**Sortimine ja otsimine**", 1997, TTÜ kirjastus.
5. Rein Jürgenson "**Andmestruktuuride lühikursus**", 1996, TTÜ kirjastus, ISBN 9985591534.

Kõik ülaltoodud allikad on TTÜ raamatukogus olemas, kuid enamasti vaid ühes-kaheks eksemplaris.

4.3. Loengutes kasutatavad abimaterjalid

Vt. [Kõik materjal on slaididel](#). Sealsamast leiab ka linke internetis olevatele lisamaterjalidele.

5. Teadmiste kontroll

Kõik, kellel ÕIS lubab ennast eksamile registreerida, võivad ka tulla. Mitte mingeid lisatingimusi ei ole. Eksam on kirjalik ning vastata tuleb kümnele küsimusele (need on kõigile antud eksamil viibijatele ühised). Kestvus 1.5 tundi. Kokku saab maksimaalselt 30 punkti. Mitte mingisuguste abimaterjalide kasutamine ei ole lubatud. Vastused võivad olla eesti, vene või inglise keeles.

Lõpphinne arvutatakse eksamil ja laboris saadud punktide kogusumma järgmise tabeli alusel:

"0" – alla 30 punkti;

"1" – 30 - 34 punkti;

"2" – 35 - 39 punkti;

"3" – 40 - 44 punkti;

"4" – 45 - 49 punkti;

"5" – 50 ja enam punkti.

Kuid need, kes said eksamil vähem kui 6 punkti, loetakse läbikukkunuks sõltumata laboris saadud punktide arvust.

6. Aeg

Loengud ja praktikumid (kumbagi 2 akadeemilist tundi iga nädal) toimuvad ainult sügissemestril. Eksameid võib teha nii talvisel kui ka kevadisel sessioonil (viimasel juhul on siiski vajalik eelnev kokkulepe).

7. Keel

Õppetöö toimub eesti keeles. Eksamit võib sooritada nii eesti, vene kui ka inglise keeles.

8. Eeldused

Programmeerimine I (IAX0583) ja Programmeerimine II (IAX0584). Need, kel eeldused tegemata, võivad vajalike algteadmiste olemasolul ennast kirja panna, kuid sel juhul on vajalik eraldi kokkulepe õppejõuga. Märkime, et Java ja C# oskajad suudavad enamasti üsna ruttu omandada C alused. Pythoni oskajate puhul see paraku nii ei ole.

8. Konsultatsioonid

Vastavalt eelnevale kokkuleppele. Kirjutage viktor.leppikson@liewenthal.ee

9. Abstracts

This course is intended to give the basic knowledge and skills necessary for software developers. Contents:

1. Programming in C language: advanced problems (memory allocation, pointers, aggregate data types, string processing).
2. Linked lists and arrays. Basic data processing operations (insert, remove, etc.) on linked lists.
3. Abstract data types. Time complexity. Recursion. Strategies (divide and conquer, dynamic programming, randomized algorithms, etc.).
4. Stack. Queue.

5. Sequential search. Binary search. Skip list.
6. Trees. Binary search tree. AVL tree. Splay tree. B tree. B* tree. Red-black tree. ISAM files. B+ tree. Trie. Ternary tree.
7. Priority queue. Heap. Leftist tree. Binomial tree.
8. Hashing: functions, chaining, open addressing.
9. Sorting algorithms: insertion sort, Shell sort, quick sort, merge sort, heap sort, radix sort.
10. String searching: Rabin-Karp and Boyer-Moore methods.
11. Data compression. Run.length encoding. Huffman code. Lempel-Zivi algorithm.