



Tematică Examenului de Licenţă – proba scrisă

Sesiunile iunie 2025, februarie 2026

Programul de Studii Informatică

I. ALGORITMI FUNDAMENTALI

1. Analiza complexităţii algoritmilor

- Aspecte generale ale analizei algoritmilor: evaluarea timpului de execuţie, identificarea operaţiilor elementare, determinarea ordinului de creştere şi a termenului dominant.
- Notăţiile asimptotice: Ω , O şi Θ - definiţia şi proprietăţile specifice ale notaţiilor.
- Analiza complexităţii algoritmilor iterativi.
- Analiza complexităţii algoritmilor recursivi: utilizarea Teoremei Master şi a altor metode pentru determinarea complexităţii.
- Tipuri de analiză: determinarea complexităţii în cazul favorabil, nefavorabil şi mediu.

Studentii trebuie să demonstreze capacitatea de a analiza şi deduce complexitatea algoritmilor, inclusiv complexitatea de timp şi spaţiu, pentru secvenţe de cod date, utilizând metode adecvate de evaluare şi justificare.

2. Tehnici de programare

- Tehnica Divide et Impera: descompunerea problemei în subprobleme independente, rezolvarea subproblemelor şi combinarea soluţiilor.
- Tehnica Greedy: definirea funcţiei obiectiv şi stabilirea ordinii de selecţie, identificarea candidaţilor şi verificarea fezabilităţii fiecărei alegeri, justificarea corectitudinii soluţiei.

Bibliografie

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C., *Introduction to algorithms (3rd ed.)*, The MIT Press, 2009, ISBN 9780262046305.
2. Karumanchi Narasimha, *Algorithm Design Techniques: Recursion, Backtracking, Greedy, Divide and Conquer, and Dynamic Programming*, CareerMonk Publications, 2018, ISBN-108193245253

3. Sastry, C.V., Nayak R., Rajaramesh CH, *Data structures and Algorithms*, IK International Publishing House, 2018, ISBN-9789385909849
4. Băicoianu, A., Majercsik, L., *Algoritmi fundamentali – Ghid practic pentru curs si laborator. O perspectivă C++*, Editura Universității Transilvania, 2020, ISBN 978-606-19-1347-3.

II. STRUCTURI DE DATE

1. Tabele de dispersie:

- Proprietăți și mod de funcționare, acces prin cheie cu funcții de dispersie.
- Problema coliziunii și rezolvarea cu liste înlănțuite și adresare deschisă în varianta dublei repartizări (formulele pentru dispersia cheilor se vor da în enunțurile problemei, se cere doar mecanismul în sine).
- Inserție în tabele de dispersie, accesul prin cheie (adică căutarea unui element cu o anumită cheie). Exerciții de inserție și acces prin cheie (căutare) pentru ambele variante menționate mai sus.
- Complexitatea operațiilor (inserție, căutare) și legătura cu factorul de încărcare al tablei.

2. Arbori binari. Arbori binari de căutare. Arbori roșu-negru (ARN)

- Definiții: arbore binar, ce este un arbore binar de căutare, arbore binar echilibrat în particular un ARN.
- Parcurgeri arbore binar (preordine, inordine, postordine și pe nivele) și algoritmi.
- Noțiunile de fiu, frate, părinte, frunză, rădăcină, înălțime, adâncime.
- Calcularea înălțimea minimă și/sau maximă pentru un arbore binar cu n noduri. Determinarea numărul maxim / minim de noduri într-un arbore binar cu o înălțime dată.
- Operațiile de bază și complexitatea acestora pe arbori binari de căutare (inserție, căutare, parcurgere).
- Proprietățile unui ARN și complexitatea operațiilor de bază (inserție, căutare, parcurgere). NU se cere cunoașterea algoritmilor de inserție sau ștergere în ARN.

Bibliografie

1. Cormen T.H, Leiserson C.E, Rivest R., Clifford S., *Intorduction to Algorithms* (fourth Edition), MIT Press, 2022, ISBN 9780262046305
2. Plajer I. C., *Notițe de curs pentru structuri de date* , 2022

III. PROGRAMARE ORIENTATĂ PE OBIECTE (C++)

1. Clase, date și funcții membre, constructori, destructor, constructor de copiere (declarare, utilitate, funcționalitate)
2. Funcții prietene (declarare, utilitate, funcționalitate)
3. Moștenire (fără moștenire multiplă)
4. Funcții virtuale, destructor virtual (declarare, utilitate, funcționalitate)
5. Funcții pur virtuale, clase abstracte (declarare, proprietăți, utilitate, funcționalitate)

Bibliografie

1. Adrian Deaconu, *Programarea în Limbajele C/C++ și aplicații*, Editura Albastra, 2008, ISBN: 978-973-650-211-8
2. Bruce Eckel, *Thinking in C++*, Prentice Hall, 2000, ISBN: 978-0139798092

IV. ALGORITMICA GRAFURILOR

1. Parcurgeri de grafuri
 - Parcurgere în adâncime (parțială, totală)
 - Parcurgere în lățime
 - Componente conexe
 - Componente tare conexe
 - Complexitate în funcție de structurile de date utilizate
2. Drumuri minime
 - Ecuțiile Bellman
 - Algoritmul Dijkstra, complexitate în funcție de structurile de date utilizate
 - Algoritmul Bellman-Ford, complexitate
 - Algoritmul Floyd-Warshall, complexitate

Bibliografie

1. Gabriela Moise, Elena Simona Nicoară, Adrian Deaconu, *Grafuri și fluxuri în rețele. O abordare teoretică și aplicații practice*, Editura MatrixRom, 2020, ISBN: 978-606-25-0592-9
2. Eleonor Ciurea - *Algoritmi. Introducere în algoritmica grafurilor*, Editura Tehnică, 2001

3. Ravindra Ahuja, Thomas Magnanti, James Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993, ISBN: 978-0-13-617549-0

V. PROGRAMARE PARALELĂ, CONCURENTĂ ȘI DISTRIBUITĂ

1. Programare paralelă
 - Legile Amdalh și Gustafsson
 - Comunicare "point to point" și colectivă MPI

2. Programare GPU
 - Ierarhia computațională în GPU
 - Tipuri de memorii în GPU

Bibliografie

1. Barry Wilkinson and Michael Allen, *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*, Prentice-Hall, 2005.
- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, *Introduction to Parallel Computing*, Addison-Wesley, 2003.
3. Jason Sanders and Edward Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley Professional, 2010.
4. Nicholas Wilt, *The CUDA Handbook: A Comprehensive Guide to GPU Programming*, Addison-Wesley Professional, 2013.
5. Pagina de e-learning a cursului "Programare Paralela, Distribuita si Concurenta", 2024-2025.