



FIȘA DISCIPLINEI ALGORITMI FUNDAMENTALI

1. Date despre program

| | |
|---------------------------------------|-------------------------------------|
| 1.1 Instituția de învățământ superior | Universitatea OVIDIUS din Constanța |
| 1.2 Facultatea | Matematica și Informatica |
| 1.3 Departamentul | Matematica și Informatica |
| 1.4 Domeniul de studii | Informatica |
| 1.5 Ciclul de studii | Licenta |
| 1.6 Programul de studii/Calificarea | Informatica |
| 1.7 Anul universitar | 2025-2026 |

2. Date despre disciplina

| | | | | | | | |
|--|---------------------------|---------------|---|-----------------------|---|---------------------------------|--------|
| 2.1 Denumirea disciplinei | Algoritmi fundamentali | | | | | | |
| 2.2 Cod disciplina | Info.1.2.14 | | | | | | |
| 2.3 Titularul activităților de curs | Lector dr. Iordache Dorin | | | | | | |
| 2.4 Titularul activităților de seminar | Lector dr. Iordache Dorin | | | | | | |
| 2.5 Anul de studiu | 1 | 2.6 Semestrul | 2 | 2.7 Tipul de evaluare | E | 2.8 Regimul disciplinei */** | DF/DOB |

* DF – disciplină fundamentală, DS – disciplină de specializare, DC – disciplină complementară

** DOB – disciplină obligatorie; DOP – disciplină opțională; DFA – Disciplină facultativă

3. Timpul total estimat

| | | | | | |
|--|----|--------------------|----|-------------------|-----|
| 3. Numarul total estimat | | | | | |
| 3.1 Numarul de ore pe saptamana | 5 | din care: 3.2 curs | 3 | 3.3 laborator *** | 2 |
| 3.4 Total ore din planul de invatamant | 70 | Din care: 3.5 curs | 42 | 3.6 laborator | 28 |
| 3.7 Total ore studiu individual | | | | | 55 |
| Distributia fondului de timp | | | | | ore |
| Studiul cărților, manualelor, suportului de curs, notițelor, bibliografie minimală recomandată | | | | | 10 |
| Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren | | | | | 10 |
| Pregătire seminar / laborator / proiect, teme, referate, portofolii și eseuri | | | | | 15 |
| Pregătire pentru prezentări sau verificări | | | | | 10 |
| Pregătire pentru examinarea finală | | | | | 5 |
| Alte activități: consultații | | | | | 5 |
| 3.8 Total ore pe semestru | | 125 | | | |
| 3.9 Numarul de credite | | 5 | | | |

4. Precondiții (acolo unde este cazul)

| | |
|-------------------|---|
| 4.1 de curriculum | Fundamente programare; Structuri de date |
| 4.2 de competente | <ul style="list-style-type: none">- Capacitatea de a scrie și testa programe simple, utilizând structuri de control și funcții.- Abilitatea de a modela probleme și a le traduce în pași algoritmici.- Înțelegerea modului de organizare și manipulare a datelor în structuri elementare. |

5. Condiții

| | | |
|--|------------------------------------|--|
| 5.1 de desfășurare a cursului | 5.1 de desfășurare a cursului | Sala de curs disponibilă cu videoproiector |
| 5.2 de desfășurare a seminarului/laboratorului | 5.2 de desfășurare a laboratorului | Sala de calculatoare (Limbajul C/C++, Pseudocod) |

*Se alege tipul de aplicație aferent disciplinei



6. Obiectivele disciplinei

| | |
|---------------------------------------|--|
| 6.1 Obiectivul general al disciplinei | Aplicarea metodelor algoritmice fundamentale în rezolvarea problemelor informatice și interdisciplinare. Utilizarea modelelor matematice și a bazelor teoretice pentru descrierea și analiza algoritmilor. Proiectarea, implementarea și optimizarea algoritmilor fundamentali (sortare, căutare, parcurgere) prin integrarea structurilor de date adecvate. Dezvoltarea și testarea aplicațiilor informatice utilizând limbajul de programare C pentru implementarea algoritmilor. |
| 6.2 Obiectivele specifice | Rezolvarea unor sarcini profesionale complexe ce implică proiectarea, analiza și implementarea algoritmilor fundamentali, în condiții de autonomie și responsabilitate. Detectarea și soluționarea problemelor conexe apărute în procesul de optimizare și adaptare a algoritmilor la diverse structuri de date. Utilizarea eficientă a surselor de informare și a resurselor de comunicare pentru documentarea, testarea și validarea algoritmilor. Colaborarea în echipă pentru elaborarea și implementarea soluțiilor algoritmice aplicate unor modele și aplicații informatice complexe. |

7. Rezultatele învățării

| | |
|-------------------------------|---|
| Cunoștințe | <p>Studentul / Absolventul:</p> <ul style="list-style-type: none">- cunoaște și explică principiile de bază ale algoritmilor fundamentali (ex. sortare, căutare, parcurgere de structuri de date) și rolul lor în rezolvarea problemelor informatice.- înțelege relația dintre algoritmi și structurile de date utilizate pentru implementarea și optimizarea acestora.- explică avantajele și limitările diferitelor metode algoritmice în funcție de complexitatea și cerințele problemei.- cunoaște tehnici de analiză a corectitudinii și a complexității algoritmilor, utilizând instrumente matematice și modele formale.- |
| Aptitudini | <p>Studentul / Absolventul:</p> <ul style="list-style-type: none">- identifică și selectează algoritmi fundamentali adecvați pentru rezolvarea unor probleme informatice diverse.- modelează și implementează algoritmi de sortare, căutare și parcurgere, utilizând limbaje de programare de nivel înalt (ex. C).- analizează și compară performanța algoritmilor, evaluând complexitatea lor în funcție de timp și spațiu.- testează și validează corectitudinea și eficiența implementărilor algoritmice pe seturi de date variate.- utilizează instrumente informatice specializate pentru dezvoltarea, simularea și optimizarea algoritmilor fundamentali. |
| Responsabilitate și autonomie | <p>Studentul / Absolventul:</p> <ul style="list-style-type: none">- demonstrează autonomie în alegerea și aplicarea algoritmilor fundamentali, adaptând soluțiile la cerințele problemei.- își asumă responsabilitatea pentru corectitudinea și eficiența algoritmilor implementați, respectând standardele de calitate software.- colaborează eficient în echipe pentru dezvoltarea și testarea aplicațiilor ce implică algoritmi fundamentali.- manifestă inițiativă în explorarea și utilizarea unor metode algoritmice alternative, în vederea optimizării performanței aplicațiilor. |



8. Continuturi

| 8.1 Curs | Metode de predare | Nr ore / Mod de realizare |
|--|---|---------------------------|
| Noțiunea de algoritm și caracteristici. Reprezentare prin pseudocod. Limbajul de programare C – elemente introductive. Exemple de implementare a algoritmilor în C/C++. | Metode de predare-invatare interactive Problematizarea | 4 ore/fizic |
| 2. Complexitatea calculului. Dimensiunea datelor de intrare. Timp de execuție (caz defavorabil, mediu, favorabil). Ordinul de creștere. Analiza asimptotică. Clase de complexitate. | Metode active și interactive; dezvoltarea gândirii critice; interacțiune, problematizare, argumentare | 4 ore/fizic |
| 3. Algoritmi elementari. Structuri elementare de date (variabile, vectori, matrice, stive, cozi). Algoritmul lui Euclid. Numere prime. Maxime și minime. Calculul unor sume. Operații cu numere mari. | Dialogul | 4 ore/fizic |
| 4. Metode de cautare și sortare. Căutare secvențială. Algoritmi de sortare prin interschimbare, inserție, selecție, numărare. Interclasarea vectorilor ordonați. | Problematizarea | 6 ore/fizic |
| 5. Metode avansate de sortare. Sortare topologica. Sortare prin metoda quicksort. Sortare prin metoda merge sort. Heap sort | Dialogul | 4 ore/fizic |
| 6. Algoritmi recursivi. Definiție, implementare și optimizare. Exemple: funcții matematice, parcurgeri de structuri. | Metode de predare-invatare interactive Problematizarea | 4 ore/fizic |
| 7. Metode de elaborare a algoritmilor Divide et Impera (căutare binară, sortare rapidă, subdreptunghi maxim). Metoda Greedy: exemple și contraexemple. Metoda Backtracking: generarea permutărilor, combinațiilor, colorarea grafurilor. | Învățarea independentă și prin cooperare Generalizarea Conversația | 6 ore/fizic |
| 8. Algoritmi de cautare a subsirurilor de caractere Algoritmul Rabin-Karp. Algoritmul Knuth-Morris-Prath. Boyer-Moore | Sintetizarea/ esențializarea informațiilor | 4 ore/ fizic |
| 9. Algoritmi de compresie și de aproximare. Codificare Huffman. Algoritmi de aproximare pentru probleme NP-grele. | Expunerea sistematică a cunoștințelor (deductivă, inductivă și formalizată, expuneri la tablă/ în meet); | 4 ore/fizic |
| 10. Algoritmi paraleli – concepte introductive. Modele de paralelism. Exemple de algoritmi paraleli simpli. Avantaje și limitări. | Expunere interactivă, problematizare | 2 ore/fizic |
| Bibliografie 1. Cormen, T; Leiserson, G; Rive, R.: Introducere în algoritmi, Comp. Libris Agora, Cluj, 2000. 2. Knuth D. E., Tratat de programarea calculatoarelor, vol. I, II, III, Ed. Teora, Bucuresti, 2002. 3. Livovschi, L, Georgescu, H.: Analiza și sinteza algoritmilor, Ed. St. și Enc., Bucuresti, 1986. 4. Andonie R., Gârbacea I., Algoritmi fundamentali, o perspectivă C++, Editura Libris, 1995 | | |
| 8.2 Seminar/laborator | Metode de predare | Nr ore / Mod de realizare |
| Modelare algoritmică de probleme reale. Algoritmi și specificații. Identificare și respectare a principiilor algoritmilor | Conversația frontală; Conversație individuală; Problematizare; | 2 ore/fizic |
| Implementare algoritmi: sir crescator, descrescator, prognoze. Algoritmi de cautare | Studii de caz; Design de proiecte complexe; | 4 ore/fizic |
| Analiza complexității algoritmilor în exemple practice | Modelarea gândirii algoritmice prin exemplificare și paralelizare cu fenomene | 4 ore/fizic |



UOC-PO-10 Anexa 3

| | | |
|---|-------------------|----------------------------|
| Implementare algoritmi simpli de sortare. Analiza complexitatii acestora | general cunoscute | 4 ore/fizic |
| Tehnici de programare: recursivitate si Divide et Impera | | 4 ore/fizic |
| Algoritmi de sortare prin interclasare si pivotare | | 2 ore/fizic |
| Tehnici Greedy: problema fractionara rucsac, colorare harti, permutarea unor multiimi | | 4ore/online |
| Algoritmi grafuri: Sortare topologica, algoritmi Kruskal si Prim | | 2ore/online 2 ore/fizic |
| Bibliografie 1. Cormen, T; Leiserson, G; Rive, R.: Introducere in algoritmi, Comp. Libris Agora, Cluj, 2000. 2. Knuth D. E., Tratat de programarea calculatoarelor, vol. I, II, III, Ed. Teora, Bucuresti, 2002. 3. Livovschi, L.; Georgescu, H.: Analiza si sinteza algoritmilor, Ed. St. si Enc., Bucuresti, 1986. 4. Andonie R., Gârbacea I., Algoritmi fundamentali, o perspectivă C++, Editura Libris, 1995 | | |

9. Evaluare

| Tip de activitate | 9.1 Criterii de evaluare | 9.2 Metode de evaluare | 9.3 Pondere din nota finala |
|---|-------------------------------------|--|-----------------------------|
| 9.4 Laborator | Activitatea din laborator | Participarea activă la activitățile de laborator | 10% |
| | Efectuarea problemelor de laborator | Evaluare orală | 10% |
| | Efectuarea temelor de laborator | Evaluare continuă orală | 10% |
| 9.5 Examen | Examen scris | Nota examinare | 60% |
| Din oficiu | | | 10% |
| 9.6 Standard minim de performanta- Nota 5 din 10. Nota finala se calculează ca medie ponderată a notelor acordate ($9.4 \cdot 0.3, 9.6 \cdot 0.6 + 0.1$). Examenul este promovat dacă media finala este mai mare sau egala cu 5 (este necesar ca nota la Examinare scrisă să fie mai mare decât 5). | | | |
| Implementarea algoritmilor fundamentali studiatii pe suport hartie:utilizand metode recursive, reprezentarea principalilor algoritmi: cautare, sortare Nota finala se calculează ca medie ponderată a notelor acordate (9.4, 9.5). Examenul este promovat dacă media finala este mai mare sau egala cu 5 (este necesar ca nota la Examinare scrisă să fie mai mare decât 5). | | | |

Data completării,
12.09.2025

Titular activităților de curs,
Lector univ. dr. D. Iordache

Titular aplicații,
Lector univ. dr. D. Iordache

Data avizării în Departament,
15.09.2025

Director de Departament,
Conf. univ. dr. E. Pelican

Decan,

Conf. univ. dr. A. Nicola