

ANALIZA ȘI PROIECTAREA ALGORITMILOR
1. Date despre disciplină/modul

Facultatea	Calculatoare, Informatică și Microelectronică				
Catedra/departamentul	Ingineria Software și Automatică				
Ciclul de studii	Studii superioare de licență, ciclul I				
Programul de studii	0613.1 Tehnologia Informației				
Anul de studii	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
II (învățământ cu frecvență); II (învățământ cu frecvență redusă)	3	E	S – unitate de curs de specialitate	O - unitate de curs obligatorie	4

2. Timpul total estimat

Total ore în planul de învățământ	Din care				
	Ore auditoriale		Lucrul individual		
	Curs	Lucrări practice	Proiect de an	Studiul materialului teoretic	Pregătire aplicații
120	30	30		10	50
120 f/r	30	15/15		10	50

3. Precondiții de acces la unitatea de curs/modul

Conform planului de învățământ	Matematică superioară; Matematica discretă; Structuri de date și algoritmi; Programare în limbajul C.
Conform competențelor	Abstractizarea situațiilor reale pentru a formula probleme și modele matematice pentru acestea și cunoașterea unui limbaj de programare structurat

4. Condiții de desfășurare a procesului educațional pentru

Curs	Pentru prezentarea materialului teoretic în sala de curs este nevoie tablă sau tablă electronică. Nu vor fi tolerate întârzierile studenților, precum și convorbirile telefonice în timpul cursului.
Laborator/seminar	Studenții vor perfecta rapoarte conform condițiilor impuse de indicațiile metodice. Termenul de predare a lucrării de laborator – două săptămâni după finalizarea acesteia. Pentru predarea cu întârziere a lucrării aceasta se depunțează cu 1pct./s. de întârziere.

5. Competențe specifice acumulate

Competențe profesionale	<p>CP2. Proiectarea și dezvoltarea aplicațiilor Acționează creativ pentru a dezvolta aplicații și a selecta opțiunile tehnice adecvate. Participă la alte activități de dezvoltare. Optimizează dezvoltarea, întreținerea și performanța aplicațiilor prin utilizarea modelelor de design și prin reutilizarea soluțiilor testate.</p> <p>CP3. Integrarea Componentelor Ia în considerare propriile acțiuni și cele ale terților în procesul de integrare. Respectă standardele și procedurile de control adecvate pentru a menține integritatea funcționalității și fiabilitatea generală a sistemului.</p> <p>CP4. Testarea aplicațiilor Asigură expertiza pentru a supraveghea programele complexe de testare. Asigură documentarea testelor și a rezultatelor pentru a furniza informații managerilor relevanți ai procesului/proceselor cum ar fi proiectanții, utilizatorii sau tehnicienii de întreținere. Asigură conformitatea cu procedurile de testare, inclusiv trasabilitatea documentată.</p> <p>CP6. Elaborarea documentației Adaptează nivelul de detaliere în funcție de obiectivul documentației și publicul vizat.</p> <p>CP7. Ingineria sistemelor Asigură interoperabilitatea componentelor sistemului. Valorifică o gamă largă de cunoștințe de specialitate pentru a crea un sistem complet care să îndeplinească cerințele sistemului și să răspundă așteptărilor clientului.</p> <p>CP8. Managementul problemelor Valorifică cunoștințele de specialitate și înțelegerea aprofundată a infrastructurii TIC și a procesului de gestionare a problemelor pentru identificarea defecțiunilor și rezolvarea acestora cu cele mai</p>
--------------------------------	---

mici întreruperi posibile. Ia decizii informate în situații tensionate emoțional cu privire la acțiunile adecvate necesare pentru a minimiza impactul asupra afacerii. Identifică rapid componentele defecte, selectează alternative privind modul de reparare, înlocuire sau reconfigurare.

CP9. Îmbunătățirea proceselor

Valorifică cunoștințele de specialitate pentru a studia procesele și soluțiile existente în domeniul TIC în vederea definirii posibilelor inovații. Face recomandări bazate pe argumente științifice.

6. Obiectivele unității de curs/modulului

Obiectivul general	Formarea abilităților de utilizare a algoritmilor pentru rezolvarea problemelor din diferite domenii
Obiectivele specifice	Să obțină abilități de analiză a complexității algoritmilor Să selecteze eficient algoritmi și metodele pentru rezolvarea problemelor. Să folosească tehnicile de proiectare a algoritmilor pentru crearea noilor algoritmi.

7. Conținutul unității de curs/modulului

Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
Tematica prelegerilor		
T1. ÎNȚIERE ÎN ALGORITMICĂ Aspectele cuprinse în studiul algoritmilor. Noțiunea de problemă și algoritm. Date. Tipuri de prelucrări. Descrierea algoritmilor. Limbajul algoritmic (Pseudocod). Specificarea datelor.	4	4
T2. ANALIZA COMPLEXITĂȚII ALGORITMILOR Scopul analizei. Timpul de execuție. Timpul în cazul cel mai defavorabil. Ordin de creștere. Notății asimptotice. Complexitatea asimptotică. Etapele analizei complexității algoritmilor. Analiza empirică a algoritmilor	4	4
T3. ANALIZA ALGORITMILOR RECURSIVI Funcții recursive. Metode de determinare a valorii funcțiilor recursive. Metoda ecuațiilor caracteristice. Metoda schimbării variabilei. Metoda master. Teorema master.	6	6
T4. TEHNICI DE PROIECTARE A ALGORITMILOR. TEHNICA REDUCERII ȘI TEHNICA DIVIDE ET IMPERA Algoritmi recursivi. Tehnica reducerii. Tehnica divizării. Aplicații ale tehnicii divizării. Algoritmi de sortare. Algoritmi de înmulțire a matricelor.	4	4
T5. TEHNICI DE PROIECTARE A ALGORITMILOR. TEHNICA ALEGERII LOCAL OPTIMALE Principiul tehnicii greedy. Minimizarea timpului mediu de așteptare. Arbori binari de interclasare. Coduri Huffman. Arbori de acoperire minimali. Algoritmii lui Kruskal și algoritmul lui Prim. Cele mai scurte drumuri care pleacă din același punct. Algoritmii lui Dijkstra.	6	6
T6. TEHNICI DE PROIECTARE A ALGORITMILOR. TEHNICA PROGRAMĂRII DINAMICE Principiul tehnicii și etapele aplicării. Determinarea celor mai scurte drumuri între fiecare pereche de vârfuri. Parantezarea optimală a unui produs de matrice. Problema rucsacului. Cel mai lung subșir comun. Caracterizarea celui mai lung subșir comun. O soluție recursivă a subproblemelor. Calculul lungimii unui CMLSC. Construirea unui CMLSC. Triangularea optimală a poligoanelor.	6	6
Total prelegeri:	30	8
Tematica lucrărilor practice (seminarii/lucrări de laborator)		
LP1. Exemple de probleme rezolvate algoritmic. Descrierea algoritmilor cu ajutorul Pseudocodului. Studiarea diferitor algoritmi, care rezolvă aceeași problemă și implementarea lor în limbajul C.	6	6
LP2. Analiza complexității asimptotice a algoritmilor. Exemple. Analiza empirică a algoritmilor.	4	4
LP3. Analiza și implementarea algoritmilor recursivi. Metode de rezolvare a recurențelor.	4	4
LP4. Analiza și implementarea algoritmilor de sortare proiectați în baza tehnicii divide et impera. Tehnica divide et impera.	4	4
LP5. Analiza și implementarea algoritmilor greedy pentru determinarea arborilor de acoperire minimali. Aplicații ale tehnicii greedy.	6	6
LP6. Compararea tehnicii greedy cu tehnica programării dinamice prin implementarea algoritmilor FLOYD și DIJKSTRA. Aplicații ale tehnicii programării dinamice. Exemple.	6	6
Total lucrări practice:	30	15/15

8. Referințe bibliografice

Principale	<ol style="list-style-type: none"> 1. T. Cormen, Ch. Leiserson, R. Rivest. <i>Introducere în algoritmi</i>. Computer Libris Agora, Cluj-Napoca, 2000. 2. Oltean Mihai. <i>Proiectarea și implementarea algoritmilor</i>. Computer Libris Agora, Cluj-Napoca, 1999. 3. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ.- М.: МЦНМО, 2001.- 960с. 4. Donald E. Knuth. <i>Fundamental Algorithms</i>, volume 1 of <i>The Art of Computer Programming</i>. Addison-Wesley, 1973. 5. Donald E. Knuth. <i>Seminumerical Algorithms</i>, volume 2 of <i>The Art of Computer Programming</i>. Addison-Wesley, 1981. 6. Donald E. Knuth. <i>Sorting and Searching</i>, volume 3 of <i>The Art of Computer Programming</i>. Addison-Wesley, 1973. 7. А. Ахо, Д. Ульман, Д. Хопкрофт. Структуры данных и алгоритмы. Издательский дом «Вильямс», 2000 – 348 с. 8. Analiza și proiectarea algoritmilor. Ciclu de prelegeri. M. Catruc, format digital. 9. https://moodle.ati.utm.md/course/view.php?id=41
Suplimentare	<ol style="list-style-type: none"> 1. Хэзфилд Р. Кирби Л. и др. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста. К.: Издательство “ДиаСофт”, 2001. – 736 с. 2. Bălănescu T.; Gavrilă Ș.; Nicolescu R.; Sofonea L.; “Metodologii și tehnici moderne de proiectare și scriere a programelor”, Universitatea din București, 1981. 3. Грис Д. Наука программирования. – М.: Мир, 1984

9. Evaluare

Periodică		Curentă	Studiu individual	Proiect/teză	Examen
EP 1	EP 2				
Învățământ cu frecvență					
15%	15%	15%	15%		40%
Învățământ cu frecvență redusă					
25%		25%		50%	
Standard minim de performanță: Prezența și activitatea la prelegeri și lucrări de laborator; Obținerea notei minime de „5” la fiecare dintre atestări și lucrări de laborator; Demonstrarea în lucrarea de examinare finală a cunoașterii conținutului cursului.					

10. Criterii de evaluare

Activitate	Componente evaluare	Metodă de evaluare, Criterii de evaluare	Pondere în nota finală a activității	Ponderea în evaluarea disciplinei
Învățământ cu frecvență				
Evaluare periodică I	Conținut teoretic, teme 1-3	Test pe MOODLE	100%	15%
Evaluare periodică II	Conținut teoretic, teme 4-6	Test pe MOODLE	100%	15%
Evaluare curentă	Activitatea practică	Discuții în cadrul lucrărilor practice	50%	15%
		Dosar completat cu Rapoarte pentru fiecare Studiu de caz în discuție	50%	
Studiul individual	Cercetare la temă	Prezentare/discurs public	100%	15%
Evaluarea finală	Conținut teoretic și practic	Examen scris.	100%	40%
Învățământ cu frecvență redusă				
Evaluare periodică I	Conținut teoretic, teme 1-3	Test pe MOODLE	100%	25%
Evaluare periodică II	Conținut teoretic, teme 4-6	Test pe MOODLE	100%	
Evaluare curentă	Activitatea practică	Discuții în cadrul lucrărilor practice. Dosar completat cu Rapoarte pentru fiecare Studiu de caz în discuție	100%	
Studiul individual	Cercetare la temă	Prezentare/discurs public	100%	25%
Evaluarea finală	Conținut teoretic și practic	Examen scris.	100%	50%