FIŞA DISCIPLINEI

1. Date despre program

Facultatea	Facultatea de Inginerie Electrică şi Ştiinţa Calculatoarelor
Departamentul	Calculatoare, Electronică și Automatică
Domeniul de studii	Ingineria Sistemelor
Ciclul de studii	Licenţă
Programul de studii	Automatică și Informatică Aplicată

2. Date despre disciplină

Denumirea disciplinei		**PROIECTAREA ALGORITMILOR**				
Anul de studiu	3	Semestrul	6	Tipul de evaluare	C	
Regimul disciplinei	Categoria formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară					DC
	Categoria de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă					DFA

3. **Timpul total estimat** (ore alocate activităţilor didactice)

I a) Număr de ore pe săptămână	5	Curs	3	Seminar	-	Laborator/lucrări practice	2	Proiect	-
I b) Totalul de ore pe semestru din planul de învăţământ	70	Curs	42	Seminar	-	Laborator/lucrări practice	28	Proiect	-

II Distribuţia fondului de timp pe semestru:	ore
II. Studiu individual	52
III. Examinări	3
IV. Alte activități (precizați):	

Total ore studiu individual (II+III)	55
Total ore pe semestru (I.b+II+III+IV)	125
Numărul de credite	5

4. Competenţe specifice acumulate

Competenţe profesionale	
Competenţe transversale	

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale din sisteme automate, sisteme încorporate și inteligente, știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete.	Studentul/absolventul utilizează limbaje, medii şi tehnologii de programare şi instrumente specifice (algoritmi, scheme, modele, protocoale etc.) în rezolvarea de probleme bine definite din ingineria sistemelor.	Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.

6. **Obiectivele disciplinei** (reieşind din grila competenţelor specifice acumulate)

Obiectivul general al disciplinei	Introducere în proiectarea și analiza algoritmilor prin prezentarea, exemplificarea, analiza și discutarea diverselor metode de proiectare a algoritmilor, evaluare a complexității teoretice și implementare de soluții algoritmice pentru probleme specifice.
	- Însuşirea de metode de proiectare a algoritmilor (Divide et Impera, programarea dinamică) ca fundamente științifice, inginerești și ale informaticii. - Însuşirea metodelor de proiectare a algoritmilor pentru prelucrarea datelor în diverse reprezentări (grafuri, date geometrice) și diverse aplicații (*e.g.*, sisteme inteligente) - Însuşirea metodelor de analiză a complexității algoritmilor în vederea evaluării și îmbunătățirii performanţelor sistemelor software.

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observaţii
1. Modele de calcul şi analiză a complexităţii algoritmilor 1.1. Instrumente de măsurare a complexităţii unui algoritm. Complexitatea temporală. Complexitatea spațială. Ordine de complexitate. Notaţiile asimptotice O, Θ şi Ω. Exemple. 1.2. Demonstrarea modului de analiză a complexității algoritmilor pentru probleme specifice.	3	Expunerea, prelegerea, conversația, exemplificarea, demonstraţia	
2. Metoda Divide et Impera: introducere 2.1. Prezentarea principiilor metodei Divide et Impera. 2.2. Introducerea metodei Divide et Impera prin discutarea și analiza unor probleme exemplu (determinarea valorii extreme dintr-o mulţime de numere reale, algoritmul căutării binare) și realizarea de comparații cu alte soluții algoritmice pentru rezolvarea acestor probleme. 2.3. Analiza complexităţii algoritmilor Divide et Impera. Teorema Master. Exemple de aplicare a teoremei Master pentru diverse probleme.	3		
3. Metoda Divide et Impera: aplicații 3.1. Algoritmul sortării prin interclasare. 3.2. Algoritmul sortării rapide. 3.3. Analiza comparativă a performanţelor diverselor metode de sortare a datelor. Realizarea de conexiuni cu alți algoritmi studiați în cadrul unor discipline anterioare. Identificarea algoritmilor de sortare implementați în cadrul diverselor platforme de dezvoltare a aplicațiilor software. 3.4. Demonstrarea aplicării metodei Divide et Impera pentru diverse probleme specifice, *e.g.*, problema determinării numarului de inversiuni ale unei permutări, problema tăieturilor, probleme de partiționare a figurilor geometrice, etc.	3		
4. Metoda programării dinamice: introducere 4.1. Prezentarea principiilor metodei programării dinamice. Rezolvarea problemelor algoritmice folosind principiile programării dinamice. 4.2. Exemplificarea metodei prin analiza unor probleme specifice, *e.g.*, șirul lui Fibonacci, calculul combinărilor $C_{n,k}$. Realizarea de comparații cu alte abordări pentru rezolvarea acestor probleme și analiza complexităţii diverselor abordări.	3		
5. Metoda programării dinamice: aplicații (1) 5.1. Problema subsecvenţei de sumă maximă. Discutarea diverselor soluții algoritmice și analiza complexității acestora. 5.2. Rezolvarea eficientă a problemelor de natură algoritmică folosind structuri de tip Lookup Table (LUT) sau Summed Area Table (SAT).	3		
6. Metoda programării dinamice: aplicații (2) 6.1. Problema celei mai lungi subsecvențe crescătoare (LIS/LAS) și aplicații privind rezolvarea de probleme asemănătoare întâlnite în practică. 6.2. Problema celei mai lungi subsecvențe de elemente comune (LCS) și aplicații privind rezolvarea de probleme asemănătoare întâlnite în practică. Analiza complexității algoritmice a soluțiilor problemelor prezentate.	3	Expunerea, prelegerea, conversația, exemplificarea, demonstraţia	

7. Metoda programării dinamice: aplicații (3) 7.1. Algoritmi eficienți pentru determinarea drumurilor optime în probleme de parcurgere a unui labirint. 7.2. Rezolvarea problemelor de drumuri în labirint și prezentarea de aplicații privind probleme asemănătoare întâlnite în practică. Analiza complexității algoritmice a soluțiilor prezentate.	3		
8. Algoritmi în grafuri: introducere 8.1. Modalităţi de reprezentare a grafurilor: matricea de adiacenţă și liste de adiancenţă. Exemple. 8.2. Parcurgerea informației dintr-un graf: algoritmul parcurgerii în lăţime (BFS) și algoritmul parcurgerii în adâncime (DFS).	3		
9. Algoritmi în grafuri: componente conexe 9.1. Probleme practice rezolvate folosind arborele parcurgerii în adâncime. 9.2. Determinarea componentelor conexe ale unui graf neorientat. Analiza complexității algoritmice. Rezolvarea de probleme exemplu.	3		
10. Algoritmi în grafuri: drumuri 10.1. Drumuri în graf. Drumuri de cost minim. 10.2. Algoritmul Roy-Floyd-Warshall pentru determinarea tuturor drumurilor de cost minim dintre nodurile unui graf. 10.3. Algoritmul lui Disjkstra. 10.4. Aplicaţii ale determinării drumurilor într-un graf pentru rezolvarea de probleme specifice.	3		
11. Algoritmi în grafuri: cicluri 11.1. Cicluri în graf. 11.2. Cicluri Hamiltoniene. Cicluri Euleriene. 11.3. Analiza performanţelor algoritmilor de determinare a ciclurilor unui graf.	3		
12. Algoritmi în grafuri: arbori parțiali 12.1. Arbori parțiali. Arbori parţiali de cost minim. 12.2. Algoritmul lui Prim. 12.3. Algoritmul lui Kruskal.	3		
13. Algoritmi geometrici 13.1. Introducere în geometria computaţională. Reprezentarea datelor geometrice. 13.2. Problema intersecţiei segmentelor de dreaptă. 13.3. Problema intersecţiei figurilor geometrice. 13.4. Calculul ariilor figurilor geometrice. 13.5. Reprezentarea poligoanelor convexe și concave. 13.6. Ordonarea punctelor în plan. Algoritmul înfășurătorii convexe a unei mulţimi de puncte.	3		
14. Algoritmi pentru șiruri de caractere 14.1. Potrivirea șirurilor de caractere și algoritmul naiv de potrivire. 14.2. Algoritmul Rabin-Karp. 14.3. Algoritmul Boyer-Moore.	3		

Bibliografie minimală recomandată

1. Donald E. Knuth. 2022. The Art of Computer Programming, vol. 1-3, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. I – III Algoritmi Fundamentali, Ed. Teora)
2. Cristian A. Giumale. 2004. Introducere în analiza algoritmilor. Ed. Polirom
3. Radu-Daniel Vatavu. 2013. Ghid practic pentru analiza si proiectarea algoritmilor. Aplicații în C#. MatrixRom
4. Ştefan-Gheorghe Pentiuc, Radu-Daniel Vatavu. 2009. Algoritmi şi metode de programare în Java. Ed. Universităţii Ştefan cel Mare din Suceava
5. Radu-Daniel Vatavu, Lisa Anthony, Jacob O. Wobbrock. 2018. $Q: A super-quick, articulation-invariant stroke-gesture recognizer for low-resource devices. In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18). ACM, New York, NY, USA

Aplicații (Seminar/laborator/lucrări practice/proiect)	Nr. ore	Metode de predare	Observaţii
1. Analiza complexității algoritmilor (1)	2	Lucrări practice, experimentul, conversația, studiul de caz	
2. Analiza complexității algoritmilor (2)	2		
3. Metoda Divide et Impera: Introducere	2		
4. Metoda Divide et Impera: Algoritmul sortării prin interclasare. Algoritmul sortării rapide	2		
5. Test practic și evaluare	2	Probă practică	Rezolvarea unei probleme algoritmice prin implementarea corectă într-un limbaj de programare
6. Metoda programării dinamice: Introducere. Subsecvența crescătoare de lungime maximă	2	Lucrări practice, experimentul, conversația, studiul de caz	
7. Metoda programării dinamice: Subsecvența comună de lungime maximă	2		
8. Metoda programării dinamice: Parcurgerea labirintului	2		
9. Test practic și evaluare	2	Probă practică	Rezolvarea unei probleme algoritmice prin implementarea corectă într-un limbaj de programare
10. Grafuri: Introducere	2	Lucrări practice, experimentul, conversația, studiul de caz	
11. Grafuri: Drumuri de cost minim	2		
12. Grafuri: Arbori parțiali de cost minim	2		
13. Test practic și evaluare	2	Probă practică	Rezolvarea unei probleme algoritmice prin implementarea corectă într-un limbaj de programare
14. Rezolvarea de probleme algoritmice specifice	2	Lucrări practice, test practic, experimentul	

Bibliografie minimală recomandată

1. Donald E. Knuth. 2022. The Art of Computer Programming, vol. 1-3, Addison-Wesley Professional (sau traducerea în limba română: Arta programării calculatoarelor, vol. I – III Algoritmi Fundamentali, Ed. Teora)
2. Cristian A. Giumale. 2004. Introducere în analiza algoritmilor. Ed. Polirom
3. Radu-Daniel Vatavu. 2013. Ghid practic pentru analiza si proiectarea algoritmilor. Aplicații în C#. MatrixRom
4. Ştefan-Gheorghe Pentiuc, Radu-Daniel Vatavu. 2009. Algoritmi şi metode de programare în Java. Ed. Universităţii Ştefan cel Mare din Suceava
5. Radu-Daniel Vatavu, Lisa Anthony, Jacob O. Wobbrock. 2018. $Q: A super-quick, articulation-invariant stroke-gesture recognizer for low-resource devices. In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18). ACM, New York, NY, USA

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Expunerea clară a conceptelor teoretice și corectitudinea operării cu noțiuni specifice proiectării algoritmilor ca fundamente științifice, inginerești și ale informaticii. Rezolvarea corectă a problemelor de natură algoritmică pentru prelucrarea datelor în diverse reprezentări și pentru diverse aplicații. Analiza corectă a complexității teoretice a algoritmilor în vederea îmbunătățirii performanțelor sistemelor software.	Probă mixtă, alcătuită din: - test docimologic pe calculator și - test practic pe calculator (rezolvarea unei probleme de natură algoritmică) cu verificarea orală a abordării folosite pentru rezolvarea testului practic.	50%
Seminar	-	-	-
Laborator/ Lucrări practice	Corectitudinea rezolvării de probleme de natură algoritmică și a implementării acestora folosind un limbaj de programare și o platformă de dezvoltare a	Probe practice pe calculator, observația sistematică, verificarea temelor.	50%

	aplicațiilor software. Gradul de îndeplinire a activităților din lucrările practice de laborator și realizarea de teme suplimentare acestora. Participarea la diverse activități practice în legătură cu obiectivele specifice ale disciplinei.		
Proiect	-	-	-

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
22.09.2025	Prof. univ. dr. ing. Radu-Daniel VATAVU	Şef lucrări dr. ing. Alexandru-Ionuț ȘIEAN

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
25.09.2025	Şef lucrări dr.ing. Corneliu BUZDUGA

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
25.09.2025	Conf. univ. dr. ing. Eugen COCA

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
26.09.2025	Prof. dr. ing. Laurentiu-Dan MILICI