



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior/	Universitatea Națională de Știință și Tehnologie POLITEHNICA din București
1.2 Facultatea	Științe Aplicate
1.3 Departamentul	Facultatea de Științe Aplicate
1.4 Domeniul de studii universitare	Științe ingineresti aplicate
1.5 Programul de studii universitare	Matematică și Informatică Aplicată în Inginerie
1.6 Ciclul de studii universitare	Licență
1.7 Limba de predare	Română
1.8 Locația geografică de desfășurare a studiilor	București

2. Date despre disciplină

2.1 Denumirea disciplinei Course title (ro) (en)	Teoria algoritmilor (Theory of Algorithms)						
2.2 Titularul/ii activităților de curs	Lect.dr. Iuliana Munteanu						
2.3 Titularul/ii activităților de seminar/ laborator/proiect	Lect.dr. Iuliana Munteanu Ing.drd. Nicolae Tarba						
2.4 Anul de studiu	3	2.5 Semestrul	II	2.6. Tipul de evaluare	E	2.7 Statutul disciplinei	Ob
2.8 Categoria formativă	DD		2.9 Codul disciplinei				

3. Timpul total (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3seminar/laborator/proiect	2
3.4 Total ore din planul de învățământ	56	Din care: 3.5 curs	28	3.6seminar/laborator/proiect	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate/Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					15
Tutorat					2
Examinări					2
Alte activități (dacă există):					
3.7 Total ore studiu individual	19				
3.8 Total ore pe semestru	75				
3.9 Numărul de credite	3				



4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Parcursul disciplinelor: Programarea calculatoarelor și limbaje de programare 1, Programarea calculatoarelor și limbaje de programare 2, Programare orientată pe obiecte.
4.2 de rezultate ale învățării	Acumularea de cunoștințe privind procesul de dezvoltare și alcătuire a unui algoritm, pornind de la specificația unei probleme, verificarea corectitudinii și analiza detaliată a complexității algoritmului.

5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)/

5.1 de desfășurare a cursului	Cursul se va desfășura într-o sală dotată cu videoproiector și computer.
5.2 de desfășurare a seminarului/laboratorului/proiectului	Laboratorul se va desfășura într-o sală cu calculatoare pe care să fie instalat un compilator de C++ (de exemplu, CodeBlocks)

6. Obiectiv general: Însușirea procesului de dezvoltare a unui algoritm, pornind de la specificația unei probleme, verificarea corectitudinii și analiza detaliată a complexității algoritmului, cât și analiza problemei din punct de vedere computațional în scopul determinării dificultății procesului de rezolvare. Studiul principiilor de bază utilizate în alcătuirea algoritmilor ca etapă esențială în dezvoltarea eficientă a aplicațiilor software. Criterii de proiectare eficientă a programelor. Studii de caz și metode de evaluare a performanțelor algoritmilor.

Disciplina răspunde cerințelor actuale de dezvoltare și evoluție pe plan național și internațional al învățământului tehnic superior în domeniul Inginerie fizică/ Matematică și Informatică Aplicată în Inginerie. Programa disciplinei este integrată în programele de studii asociate domeniului de Inginerie fizică/ Matematică și Informatică Aplicată în Inginerie fiind corelată cu programe de studii similare din universitățile europene ce aplică sistemul Bologna.

În contextul actual, domeniile de activitate vizate sunt: industrial, economico-financiar (cu precădere în domeniul bancar), educație și cercetare – dezvoltare.

7. Rezultatele învățării: Crearea abilităților de a aplica cunoștințele generale privind algoritmizarea și modelarea unor probleme practice. Posibilitatea de a evalua eficiența implementării unei aplicații software folosind criteriile de performanță analizate la curs. Dezvoltarea calitatilor necesare unui bun programator: abstractizare, analiză, și sinteză.

Cunoștințe	- Identifică și descrie concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică. - Explică și interpretează rezultate teoretice și experimentale din matematică, fizică, chimie, economie, desen tehnic și informatică. - Crearea abilităților de a aplica cunoștințele generale privind structurarea algoritmică a programelor. - Se asigură studenților competențe adecvate cu necesitățile calificărilor actuale, o pregătire științifică și tehnică corespunzătoare nivelului de licență, care să le permită inserția rapidă pe piața muncii după absolvire, dar și posibilitatea continuării studiilor prin programe de masterat și doctorat.
------------	--



Abilități	- Operează cu concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică. - Proiectează experimente și sisteme ingineresti funcționale de complexitate mică/medie specifice. - Dezvoltarea calitatilor necesare unui programator: abstractizare, analiză, și sinteză: combinare (abstracție algebrică), echivalență (abstracție topologică/ sinteză), izolare (descompunere/ analiză), evidențiere (extragere din context), evidențiere a unei stări (abstracție temporală), idealizare (abstracție de ordine).
Responsabilitate și autonomie	- Aplică valorile eticii și deontologiei profesiei de inginer. - Practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor. - Comunică eficient despre activitățile de inginerie cu o gamă largă de public. - Este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate. - Selectează și analizează surse bibliografice. - Produce software și îl adaptează continuu la noile tehnologii și cerințe de piață. - Demonstrează autonomie în învățare - Își asumă în mod responsabil sarcinile profesionale și respectă normele de etică și deontologie profesională. - Lucrează eficient ca membru în echipă sau lider al acesteia. - Programul de studii este încadrat în politica și strategia Universității Politehnica din București, atât din punct de vedere al conținutului și structurii, cât și din punct de vedere al aptitudinii și deschiderii internaționale oferite studenților. - Parte din programa aferentă cursului a fost elaborată în urma discuțiilor avute între cadrele didactice de la Facultatea de Științe Aplicate și diverși potențiali angajatori ai absolvenților.

8. Metode de predare

Pornindu-se de analiza caracteristicilor de învățare ale studenților și de la nevoile lor specifice, procesul de predare va explora metode de predare atât expositive (prelegerea, expunerea), cât și conservative-interactive, bazate pe modele de învățare prin descoperire facilitate de explorarea directă și indirectă a realității (experimentul, demonstrația, modelarea), dar și pe metode bazate pe acțiune, precum exercițiul, activitățile practice și rezolvarea de probleme.

În activitatea de predare vor fi utilizate prelegeri, în baza unor prezentări Power Point sau diferite filmulețe care vor fi puse la dispoziția studenților. Fiecare curs va debuta cu recapitularea capitolelor deja parcurse, cu accent asupra noțiunilor parcurse la ultimul curs.

Prezentările utilizează imagini și scheme, astfel încât informațiile prezentate să fie ușor de înțeles și asimilat.

Prezentare la tabla și prezentare cu ajutorul calculatorului.

Se va exersa abilitatea de lucru în echipă pentru rezolvarea diferitelor sarcini de învățare.

9. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore



I	Preliminarii despre algoritmi - istoric, descoperirea algoritmilor, teoria rezolvarii problemelor; - definitia formală a unui algoritm; - reprezentarea algoritmilor: pseudocod, structuri iterative, recursivitate.	2
II	Analiza eficienței algoritmilor - notatii asimptotice; - complexitatea și corectitudinea algoritmilor; - tehnici de analiza a algoritmilor; - analiza algoritmilor recursivi.	4
III	Algoritmi backtracking: - descriere și aplicabilitate - implementare iterativă și recursivă.	4
IV	Algoritmi Greedy - tehnica Greedy; - interclasarea sirurilor ordonate; - implementarea arborilor de interclasare; - coduri Huffman.	4
V	Algoritmi Divide et Impera - Tehnica Divide et Impera; - Cautare binară, Quicksort, Mergesort.	2
VI	Algoritmi de programare dinamică - Principiile fundamentale ale programării dinamice; - Exemple de algoritmi ce folosesc tehnica programării dinamice; - Programarea dinamică comparată cu programarea Greedy.	4
VII	Algoritmi probabilisti	4
VIII	Algoritmi genetici	4
	Total:	28

Bibliografie:

1. Cursul în format electronic din Moodle.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, *Introducere în Algoritmi*, Agora, 2000.
3. Donald E. Knuth, *Tratat de programarea calculatoarelor. Algoritmi fundamentali*, Editura Tehnica, 1985.
4. Andonie R., Garbacea I., *Algoritmi fundamentali. O perspectiva C++*, Editura Libris, Cluj-Napoca, 1995.
5. Calude C., *Teoria algoritmilor*, Ed. Universitatii Bucuresti, 1987.



Nr. crt.	Conținutul	Nr. ore
1.	Exercitii si implementare aplicații backtracking si recursivitate	8
2.	Exercitii si implementare aplicații divide et impera	4
3.	Exercitii si implementare aplicații metoda Greedy si programare dinamica	6
4.	Exercitii si implementare aplicații algoritmi probabilisti	4
5.	Exercitii si implementare aplicații algoritmi genetici	6
	Total:	28

Bibliografie:

1. Cursul si laboratorul în format electronic din Moodle.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, *Introducere în Algoritmi*, Agora, 2000.
3. Donald E. Knuth, *Tratat de programarea calculatoarelor. Algoritmi fundamentali*, Editura Tehnica, 1985.
4. Andonie R., Garbacea I., *Algoritmi fundamentali. O perspectiva C++*, Editura Libris, Cluj-Napoca, 1995.
5. Calude C., *Teoria algoritmilor*, Ed. Universitatii Bucuresti, 1987.
6. Iulian Năstac, *Programarea calculatoarelor in limbajul C – Elemente fundamentale*, Editura Printech, București, 2006.
7. Dan Galațchi, Sorin Zoican, Roxana Zoican, *Limbajul C. Structuri de date și algoritmi*, Editura Politehnica Press, 2004.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Invatare curs si creativitate	Lucrare scrisa	50%
10.5 Seminar/laborator/proiect	Exercitii la seminar / Programe la laborator		25% / 25%
10.6 Condiții de promovare			
Obținerea a 50% din punctajul total.			

Data completării

23.09.2025

Titular de curs

Lect.dr. Iuliana MUNTEANU

Titular(ii) de aplicații

Lect.dr. Iuliana MUNTEANU

Ing.drd. Nicolae TARBA

Data avizării în
departament

Director de departament

Lect.dr. Romeo BERCIA



Universitatea Națională de Știință și Tehnologie
POLITEHNICA București
Facultatea de Științe Aplicate



Data aprobării în
Consiliul Facultății

Decan,
Conf. dr. Alina Claudia PETRESCU-NIȚĂ

26.09.2025