

**FIȘA DISCIPLINEI****1. Date despre program**

1.1 Instituția de învățământ superior	Universitatea “Alexandru Ioan Cuza” din Iași
1.2 Facultatea	Facultatea de Matematică
1.3 Departamentul	Matematică
1.4 Domeniul de studii	Matematică
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Matematică informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Algoritmi și complexitate						
2.2 Titularul activităților de curs	Conf. dr. Ana-Maria Moșneagu Lect. dr. Andreea Florea						
2.3 Titularul activităților de seminar	Conf. dr. Ana-Maria Moșneagu Lect. dr. Andreea Florea						
2.4 An de studiu	K	2.5 Semestru	2	2.6 Tip de evaluare	EvP	2.7 Regimul disciplinei	OB

* OB – Obligatoriu / OP – Opțional

3. Timpul total estimat (ore pe semestru și activități didactice)

3.1 Număr de ore pe săptămână	4	din care: 3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și altele					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					35
Tutoriat					
Examinări					4
Alte activități.....					
3.7 Total ore studiu individual					65
3.8 Total ore pe semestru					125
3.9 Număr de credite					5

4. Precondiții (dacă este cazul)

4.1 De curriculum	Fundamentele programării
4.2 De competențe	Capacitatea de concepere a unor algoritmi simpli și implementarea lor corespunzătoare într-un limbaj de programare.

5. Condiții (dacă este cazul)

5.1 De desfășurare a cursului	Amfiteatru, calculator, videoproiector
5.2 De desfășurare a seminarului/laboratorului	Rețea de calculatoare cu mediul de programare Python instalat

6. Competențe specifice acumulate

Competențe profesionale	C1. Operarea cu noțiuni și metode matematice (0.5 credite) C2. Prelucrarea matematică a datelor, analiza și interpretarea unor fenomene și procese C3. Elaborarea și analiza unor algoritmi pentru rezolvarea problemelor (2.5 credite) C4. Conceperea modelelor matematice pentru descrierea unor fenomene C5. Demonstrarea rezultatelor matematice folosind diferite concepte și raționamente matematice (0.5 credite)
Competențe transversale	CT1. Aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unor atitudini responsabile față de domeniul științific și didactic, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională (0.5 credite) CT2. Desfășurarea eficientă și eficace a activităților organizate în echipă (0.5 credite) CT3. Utilizarea eficientă a surselor informaționale și a resurselor de comunicare și formare profesională asistată, atât în limba română, cât și într-o limbă de circulație internațională (0.5 credite)

7. Obiectivele disciplinei (din grila competențelor specifice acumulate)

7.1. Obiectivul general	Dezvoltarea gândirii algoritmice, înțelegerea, însușirea și utilizarea principalelor tehnici de proiectare și analiză a algoritmilor
7.2. Obiectivele specifice	La finalizarea cu succes a acestei discipline, studenții vor fi capabili să: <ul style="list-style-type: none"> ▪ Descrie principalele tehnici de proiectare a algoritmilor și a modalităților de analiză a acestora ▪ Utilizeze metode de programare adecvate pentru conceperea de algoritmi eficienți ▪ Determine complexitatea algoritmilor ▪ Analizeze și să clasifice algoritmii din punct de vedere al complexității acestora ▪ Explice avantajele și/sau dezavantajele folosirii unei anumite paradigme de proiectare pentru rezolvarea unei probleme

8. Conținut

8.1	Curs	Metode de predare	Observații
1.	Algoritmi: definiții, proprietăți, clasificări. Descrierea algoritmilor	Prelegerea participativă, expunerea, dialogul, exemplificarea	2 ore
2.	Analiza corectitudinii algoritmilor	Prelegerea participativă, , expunerea, dialogul, exemplificarea	2 ore
3.	Timp de execuție al unui algoritm. Analiza complexității în cazurile extreme și în cazul mediu	Prelegerea participativă, expunerea, dialogul, exemplificarea	2 ore
4.	Ordin de creștere a timpului de execuție. Notații asimptotice. Clase de complexitate	Prelegerea participativă, expunerea, dialogul, exemplificarea	2 ore
5.	Algoritmi elementari de sortare	Prelegerea participativă, expunerea, dialogul, exemplificarea	2 ore
6.	Analiza algoritmilor recursivi	Prelegerea participativă, expunerea, dialogul, exemplificarea	4 ore
7.	Tehnica reducerii Tehnica divizării	Prelegerea participativă, expunerea, dialogul, exemplificarea	4 ore
8.	Tehnica backtracking	Prelegerea participativă, expunerea, dialogul, exemplificarea	2 ore
9.	Tehnica Greedy	Prelegerea participativă, expunerea, dialogul, exemplificarea	2 ore

10.	Programare dinamică	Prelegerea participativă, expunerea, dialogul, exemplificarea	2 ore
11.	Algoritmi aleatori. Clase de probleme. Algoritmi de aproximare și euristici	Prelegerea participativă, expunerea, dialogul, exemplificarea	4 ore

Bibliografie

Referințe principale:

R. Andonie, I. Gârbacea, Algoritmi Fundamentali. O Perspectivă C++, Ed. Libris, Cluj-Napoca, 1995
G. Brassard, P. Bratley, Fundamental of Algorithmics, Prentice-Hall, 1996
T.H. Cormen, C.E. Leiserson, R.L. Rivest. Introduction to Algorithms (3rd ed.), MIT Press, 2009
T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introducere în Algoritmi, Computer Libris Agora, Cluj-Napoca, 2000 (traducere)
C.A. Giumale, Introducere în Analiza Algoritmilor. Teorie și aplicație, Ed. Polirom, 2004
D. Knuth. Arta Programării Calculatoarelor (vol. 1, 2, 3), Ed. Teora, 1999-2001 (traducere)
A. Levitin, Introduction to the design and analysis of algorithm (3rd ed.), Pearson Ed., 2012
D. Lucanu, M. Craus, Proiectarea Algoritmilor, Ed. Polirom, 2008
S. Skiena, The Algorithm Design Manual (2nd ed.), Springer-Verlag, Londra, 2008
D. Zaharie, Introducere în Proiectarea și Analiza Algoritmilor, Ed. Eubee, 2008

Referințe suplimentare:

A.M. Moșneagu, Algoritmi și complexitate, note de curs online

8.2	Seminar / Laborator	Metode de predare	Observații
1.	Introducere în limbajul de programare Python: tipuri de date, operatori, funcții de intrare/ieșire, structuri de control, funcții predefinite și funcții definite de utilizator	Exercițiul, lucrul individual, conversația	4 ore
2.	Analiza corectitudinii algoritmilor	Exercițiul, lucrul individual, conversația	2 ore
3.	Timp de execuție a unui algoritm. Analiza complexității în cazurile extreme și în cazul mediu. Ordin de creștere a timpului de execuție. Notății asimptotice. Clase de complexitate	Exercițiul, lucrul individual, conversația	2 ore
4.	Algoritmi elementari de sortare	Exercițiul, lucrul individual, conversația	2 ore
5.	Analiza algoritmilor recursivi Evaluare	Exercițiul, lucrul individual, conversația	4 ore
6.	Tehnica reducerii Tehnica divizării	Exercițiul, lucrul individual, conversația	4 ore
7.	Tehnica backtracking	Exercițiul, lucrul individual, conversația	2 ore
8.	Tehnica Greedy	Exercițiul, lucrul individual, conversația	2 ore
9.	Programare dinamică	Exercițiul, lucrul individual, conversația	2 ore
10.	Algoritmi aleatori. Algoritmi aproximativi și euristici Evaluare	Exercițiul, lucrul individual, conversația	4 ore

Bibliografie

T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introducere în Algoritmi, Computer Libris Agora, Cluj-Napoca, 2000 (traducere)
C.A. Giumale, Introducere în Analiza Algoritmilor. Teorie și aplicație, Ed. Polirom, 2004
C. Ivașc, A. Rusu, M. Prună. Tehnici de programare. Aplicații, Editura „Alexandru Myller”, Iași, 2006
A. Levitin, Introduction to the design and analysis of algorithm (3rd ed.), Pearson Ed., 2012
D. Zaharie, Introducere în Proiectarea și Analiza Algoritmilor, Ed. Eubee, 2008
M.T. Goodrich, R. Tamassia, M.H. Goldwasser, Data Structures & Algorithms in Python, Wiley, 2013

9. Coroborarea conținutului disciplinei cu așteptările reprezentanților comunității, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

Programa cursului este în concordanță cu tematica concursului de titularizare în învățământul preuniversitar și cu cerințele formulate de reprezentanții companiilor IT.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere în nota finală (%)
10.4 Curs	Cunoașterea și utilizarea corectă a noțiunilor fundamentale	Evaluare pe parcurs (verificare periodică: teste, verificări orale/ practice/ teme)	50%
10.5 Seminar/ Laborator	Aplicarea corectă a rezultatelor teoretice, identificarea metodelor eficiente pentru rezolvarea unor exerciții și probleme	Evaluare pe parcurs (verificare periodică: teste, verificări orale/ practice/ teme)	50%

10.6 Standard minim de performanță

1. Identificarea și selectarea tehnicilor de programare adecvate pentru conceperea de algoritmi eficienți în scopul rezolvării unor exerciții concrete
2. Determinarea complexității algoritmilor
3. Analizarea și clasificarea algoritmilor din punct de vedere al complexității acestora
4. Elaborarea și rularea de programe descrise în Python care să rezolve problemele propuse spre soluționare

Criterii: Nota finală (N) ≥ 5 , fiecare notă parțială (NP) ≥ 4 ;
Prezența la activitățile de laborator $\geq 85\%$.

Regula de stabilire a notei finale:

$$N = 0.8T + 0.2L, \text{ unde}$$

T = media aritmetică a notelor parțiale (NP) obținute la două teste susținute pe parcursul semestrului;

L = media aritmetică a notelor obținute la verificări orale/verificări practice/teme în timpul laboratoarelor.

Data completării
26.09.2022

Titular de curs
Conf. dr. Ana-Maria Moșneagu

Titular de seminar
Conf. dr. Ana-Maria Moșneagu

Lect. dr. Andreea Florea

Lect. dr. Andreea Florea

Data avizării în departament

Director de departament
Prof. dr. Ioan Bucătaru