

## FIȘA DISCIPLINEI

## Analiza algoritmilor, anul universitar 2017-2018

## 1. Date despre program

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Electronică, Comunicații și Calculatoare
1.3	Departamentul	Electronică, Calculatoare și Inginerie Electrică
1.4	Domeniul de studii	Calculatoare și tehnologia informației
1.5	Ciclul de studii	Licență
1.6	Programul de studii / Calificarea	Calculatoare / Inginer calculatoare

## 2. Date despre disciplină

2.1 Denumirea disciplinei												Analiza algoritmilor			
2.2		Titularul activităților de curs						Conf. univ. dr. Costel BĂLCĂU							
2.3		Titularul activităților de laborator						Lector univ. dr. Adrian ȚURCANU							
2.4		Anul de studii	II	2.5		Semestrul	2	2.6		Tipul de evaluare	Examen	2.7		Regimul disciplinei	O

## 3. Timpul total estimat

3.1	Număr de ore pe săptămână	3	3.2	din care curs	2	3.3	laborator	1
3.4	Total ore din planul de inv.	42	3.5	din care curs	28	3.6	laborator	14
Distribuția fondului de timp								ore
Studiul după manual, suport de curs, bibliografie și notițe								10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								7
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								7
Tutoriat								2
Examinări								4
Alte activități .....								
3.7	Total ore studiu individual	30						
3.8	Total ore pe semestru	72						
3.9	Număr de credite	3						

## 4. Precondiții (acolo unde este cazul)

4.1	De curriculum	
4.2	De competențe	

## 5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Sală de curs cu o capacitate de minim 100 locuri dotată cu videoproiector și ecran de proiecție, 2 table.
5.2	De desfășurare a laboratorului	Sală de laborator de informatică, capacitate maximă 18 studenți/laborator.

## 6. Competențe specifice acumulate

Competențe profesionale	<b>C3 Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor (3 p.c.)</b> <b>C3.1</b> Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice (1.5 p.c.) <b>C3.2</b> Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor (1.5 p.c.)
Competențe transversale	-

## 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- Dezvoltarea gândirii algoritmice și însușirea tehnicilor de programare avansată.</li> <li>- Însușirea principalelor concepte din teoria algoritmilor: analiza eficienței, notații asimptotice, metode de elaborare a algoritmilor, structuri de date, tehnici de sortare și căutare, probleme NP-dificile și NP-complete.</li> </ul>
7.2 Obiectivele specifice	<i>Obiective cognitive:</i> <ul style="list-style-type: none"> <li>- Dobândirea de cunoștințe suport pentru evaluarea complexității algoritmilor.</li> <li>- Dobândirea de cunoștințe pentru demonstrarea corectitudinii și analiza eficienței algoritmilor.</li> <li>- Cunoașterea unor clase de probleme NP-complete și NP-dificile.</li> </ul> <i>Obiective procedurale:</i> <ul style="list-style-type: none"> <li>- Formarea abilităților de proiectare și implementare de algoritmi pentru diverse clase de</li> </ul>

	<p>probleme, inclusiv pentru probleme NP.</p> <ul style="list-style-type: none"> <li>- Formarea deprinderilor de investigare a problemelor din perspectiva diverselor metode posibile de rezolvare, în vederea obținerii unor soluții eficiente.</li> </ul> <p><i>Obiective atitudinale:</i></p> <ul style="list-style-type: none"> <li>- Rigurozitate în proiectarea algoritmilor și în programare.</li> </ul>
--	---

## 8. Conținuturi

8.1. Curs		Nr. ore	Metode de predare	Observații Resurse folosite
1.	<b>Elemente de complexitatea algoritmilor.</b> Fundamente matematice necesare analizei algoritmilor. Notatii asimptotice. Definiții, proprietăți, operații. Exemple.	2	Prelegerea Explicația Descrierea și exemplificarea Demonstrația Problematizarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar
2.	<b>Elemente de complexitatea algoritmilor.</b> Tehnici de analiză a complexității algoritmilor. Exemple.	2		
3.	<b>Elemente de complexitatea algoritmilor.</b> Algoritmi optimi. Algoritmi asimptotic-optimi. Exemple.	2		
4.	<b>Metoda Divide et Impera. Recursivitate.</b> Analiza algoritmilor recursivi. Metode de rezolvare a recurențelor. Exemple.	2		
5.	<b>Metoda Divide et Impera. Recursivitate.</b> Analiza complexității. Teorema Master. Exemple.	2		
6.	<b>Algoritmi de căutare și analiza complexității lor.</b> Căutare secvențială, căutare binară. Arbori binari stricți. Arbori de căutare.	2		
7.	<b>Algoritmi de căutare și analiza complexității lor.</b> Complexitatea algoritmilor de căutare. Optimalitatea algoritmului de căutare binară.	2		
8.	<b>Algoritmi de sortare prin comparații de chei și analiza comparativă a complexității lor.</b> Arbori de sortare. Complexitatea algoritmilor de sortare.	2		
9.	<b>Algoritmi de sortare prin comparații de chei și analiza comparativă a complexității lor.</b> Sortarea prin numărare. Sortarea prin selecție. Metoda bulelor (bubble sort). Sortarea prin inserție directă. Sortarea prin inserție binară. Complexitatea acestor algoritmi.	2		
10.	<b>Algoritmi de sortare prin comparații de chei și analiza comparativă a complexității lor.</b> Sortarea prin interclasare (mergesort). Sortare rapidă (quicksort). Optimalitatea acestor algoritmi.	2		
11.	<b>Algoritmi liniari în calculul matriceal.</b> Noțiunea de algoritm liniar. Exemple. Rangul pe linii, rangul pe coloane, rangul pe linii și coloane ale matricelor. Optimalitatea schemei lui Horner. Alte exemple de algoritmi optimi.	2		
12.	<b>Clasificarea problemelor în funcție de complexitate.</b> Algoritmi nedeterminiști. NP-completitudine. Probleme NP-dificile, probleme NP-complete. Problema validării (satisfaciabilității, SAT). Teorema Cook-Levin.	2		
13.	<b>Clasificarea problemelor în funcție de complexitate.</b> Exemple de probleme NP-dificile și de probleme NP-complete din teoria grafurilor: problema cliii maxime, problema stabilei maxime, problema suportului minim, problema arborelui Steiner de cost minim, problema ciclului hamiltonian, problema comisului voiajor. Compararea diferitelor soluții de rezolvare (algoritmi euristici, programare dinamică, Backtracking, Branch and Bound)	2		
14.	<b>Clasificarea problemelor în funcție de complexitate.</b> Alte exemple de probleme NP-dificile și de probleme NP-complete: problema submulțimii de sumă dată, problema rucsacului (Knapsack) – varianta discretă., Compararea diferitelor soluții de rezolvare. Calculabilitate în sens Turing. Ipoteza Church-Turing.	2		
<b>Bibliografie:</b> 1. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Data Structures and Algorithms, Addison-Wesley, 2009. 2. Gh. Barbu, M. Boloșteanu, I. Văduva, Bazele informaticii, Ed. Tehnică, București, 1997. 3. Gh. Barbu, V. Păun, Programarea în limbajul C/C++, Ed. Matrix Rom, București, 2011. 4. Gh. Barbu, V. Paun, Calculatoare personale și programare în C/C++, Ed. Did. și Ped., București, 2005. 5. C. Bălcău, Combinatorică și teoria grafurilor, Ed. Univ. din Pitești, Pitești, 2007. 6. C. Bălcău, Analiza algoritmilor – note de curs (format electronic). 7. O. Băscă, L. Livovschi, Algoritmi euristici, Ed. Univ. din București, București, 2003. 8. E. Ciurea, I. Ciupală, Algoritmi. Introducere în algoritmica fluxurilor în rețele, Ed. Matrix Rom, București, 2006.				

9. T.H. Cormen, Algorithms Unlocked, MIT Press, 2013.
10. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009.
11. H. Georgescu, Tehnici de programare, Ed. Univ. din București, București, 2005.
12. C.A. Giumale, Introducere în analiza algoritmilor. Teorie și aplicații, Ed. Polirom, Iași, 2004.
13. F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, Springer, 2007.
14. D. Jungnickel, Graphs, Networks and Algorithms, Springer, 2013.
15. D.E. Knuth, The Art Of Computer Programming. Vol. 4A: Combinatorial Algorithms, Addison-Wesley, 2011.
16. B. Korte, J. Vygen, Combinatorial Optimization. Theory and Algorithms, Springer, 2012.
17. L. Livovschi, H. Georgescu, Sinteza și analiza algoritmilor, Ed. Șt. și Encl., București, 1986.
18. D. Logofătu, Algoritmi fundamentali în C++: Aplicații, Ed. Polirom, Iași, 2007.
19. D. Lucanu, M. Craus, Proiectarea algoritmilor, Ed. Polirom, Iași, 2008.
20. R. Stephens, Essential Algorithms: A Practical Approach to Computer Algorithms, Wiley, 2013.
21. I. Tomescu, Data structures, Ed. Univ. București, București, 1997 (2004).

8.2. Aplicații – Laborator		Nr. ore	Metode de predare	Observații Resurse folosite
1.	<b>Fundamente ale programării în limbajul Java.</b> Structura lexicală a limbajului Java. Tipuri de date și variabile. Operatori, expresii, instrucțiuni. Vectori și tablouri multidimensionale. Obiecte și clase. Câmpuri și metode. Constructori. Definiții statice într-o clasă. Supraîncărcarea și suprascrierea metodelor. Probleme suplimentare.	2	Exercițiul Explicația Studiul de caz Dezbateri Lucrul în grup Teme Individuale	Calculator, Videoproiector, Suport documentar
2.	<b>Fundamente ale programării în limbajul Java.</b> Modificatori de vizibilitate. Pachete. Utilizarea unor clase predefinite: clase înfășurătoare, Math, String, Scanner, System, Arrays etc. <b>Algoritmi iterativi. Analiza complexității.</b> Implementări în limbajele C++ și Java. Probleme suplimentare.	2		
3.	<b>Algoritmi recursivi. Metoda Divide et Impera.</b> Implementări în limbajele C++ și Java. Probleme suplimentare.	2		
4.	<b>Implementarea algoritmilor de căutare și analiza comparativă a timpilor efectivi de execuție.</b> Căutare secvențială, căutare binară. Arbori binari de căutare. Implementări în limbajele C++ și Java. Probleme suplimentare.	2		
5.	<b>Implementarea algoritmilor de sortare și analiza comparativă a timpilor efectivi de execuție.</b> Sortarea prin numărare. Sortarea prin selecție. Metoda bulelor (bubble sort). Sortarea prin inserție directă. Sortarea prin inserție binară. Sortarea prin interclasare (mergesort). Sortare rapidă (quicksort). Implementări în limbajele C++ și Java. Probleme suplimentare.	2		
6.	<b>Algoritmi liniari în calculul matriceal.</b> Implementări în limbajele C++ și Java. Probleme suplimentare.	2		
7.	<b>Rezolvarea de probleme NP-dificile și NP-complete.</b> Implementări în limbajele C++ și Java. Probleme suplimentare.	2		
<b>Bibliografie:</b> 1. Gh. Barbu, V. Păun, Programarea în limbajul C/C++, Ed. Matrix Rom, București, 2011. 2. Gh. Barbu, V. Paun, Calculatoare personale și programare în C/C++, Ed. Did. și Ped., București, 2005. 3. C. Bălcău, Combinatorică și teoria grafurilor, Ed. Univ. din Pitești, Pitești, 2007. 4. E. Cerchez, M. Șerban, Programarea în limbajul C/C++ pentru liceu. Vol. 2: Metode și tehnici de programare, Ed. Polirom, Iași, 2005. 5. E. Ciurea, L. Ciupală, Algoritmi. Introducere în algoritmica fluxurilor în rețele, Ed. Matrix Rom, București, 2006. 6. T.H. Cormen, Algorithms Unlocked, MIT Press, 2013. 7. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009. 8. N. Dale, C. Weems, Programming and problem solving with JAVA, Jones and Bartlett Publ., 2008. 9. A. Ene, C. Știrbu, Programarea calculatoarelor și limbaje de programare: Teorie și aplicații, Ed. Univ. din Pitești, Pitești, 2005. 10. A. Ene, C. Știrbu, 100 de probleme rezolvate în JAVA, Ed. Univ. din Pitești, Pitești, 2007. 11. H. Georgescu, Tehnici de programare, Ed. Univ. din București, București, 2005. 12. C.A. Giumale, Introducere în analiza algoritmilor. Teorie și aplicații, Ed. Polirom, Iași, 2004. 13. F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, Springer, 2007. 14. D. Jungnickel, Graphs, Networks and Algorithms, Springer, 2013. 15. D. Logofătu, Algoritmi fundamentali în C++: Aplicații, Ed. Polirom, Iași, 2007. 16. D. Lucanu, M. Craus, Proiectarea algoritmilor, Ed. Polirom, Iași, 2008. 17. N. Popescu, Data structures and algorithms using Java, Ed. Politehnica Press, Bucuresti, 2008.				

18. R. Stephens, Essential Algorithms: A Practical Approach to Computer Algorithms, Wiley, 2013.
19. Ș. Tanasă, C. Olaru, Ș. Andrei, Java de la 0 la expert, Polirom, Iași, 2007.
20. I. Tomescu, Data structures, Ed. Univ. București, București, 1997 (2004).
21. A. Țurcanu, Analiza algoritmilor – suport pentru laborator (format electronic).

**9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor din domeniul aferent programului**

Conținuturile sunt corelate cu cele ale disciplinelor similare din universități de prestigiu din țară și din străinătate (precum MIT) și ajustate în urma discuțiilor cu reprezentanți ai angajatorilor locali din domeniul IT (precum RoWeb, Lisa, Prodnf, Kepler, Osf, Endava).

**10. Evaluare**

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Evaluare finală	Probă scrisă (teorie și implementări)	50%
10.5 Laborator	Activitate (implementarea problemelor propuse) Tema de casă (probleme suplimentare)	Verificare soluții, probă practică Verificare temă	30% 20%
10.6 Standard minim de performanță	* Note de minim 5 la activitatea de laborator, la tema de casă și la evaluarea finală (rezolvarea în proporție de 50% a cerințelor); nota finală minim 5. * Set de cunoștințe minimale pentru promovarea examenului final: - Cunoașterea principalelor noțiuni privind metodele de analiză a algoritmilor; - Cunoașterea principalelor modalități de aplicare adecvată a acestor metode în rezolvarea și implementarea eficientă a problemelor propuse.		

Data completării  
25 septembrie 2017

Titular de curs,  
Conf. univ. dr. Costel BĂLCĂU

Titular de laborator,  
Lector univ. dr. Adrian ȚURCANU

Data aprobării în Consiliul departamentului, 25 septembrie 2017  
Director de departament,  
(prestator)  
Conf. univ. dr. Doru CONSTANTIN

Director de departament,  
(beneficiar)  
Prof.univ.dr. Gheorghe ȘERBAN