

## FIȘA DISCIPLINEI

### Proiectarea 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

#### 2. Date despre disciplină

2. Date despre disciplina											
2.1	Denumirea disciplinei					Proiectarea 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	I	2.6	Tipul de evaluare	E	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 învățământ	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								18
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								14
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								14
Tutorat								2
Examinări								6
Alte activități .....								-
3.7	Total ore studiu individual	54						
3.8	Total ore pe semestru	96						
3.9	Număr de credite	4						

#### 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>C2 Proiectarea componentelor hardware, software și de comunicații (2 p.c.):</b> <b>C2.3</b> Construirea unor componente hardware, software și de comunicații folosind metode de proiectare, limbaje, algoritmi, structuri de date, protocoale și tehnologii (1 p.c.). <b>C2.5</b> Implementarea componentelor sistemelor hardware, software și de comunicație (1 p.c.). <b>C3 Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor (2 p.c.):</b> <b>C3.1</b> Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice (1 p.c.). <b>C3.5</b> Dezvoltarea și implementarea de soluții informatice pentru probleme concrete (1 p.c.).
Competențe transversale	-

#### 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- Însușirea principalelor concepte din teoria proiectării algoritmilor: metode de elaborare a algoritmilor, introducerea în evaluarea complexității algoritmilor.</li> <li>- Dezvoltarea gândirii algoritmice și însușirea unor tehnici de programare avansată.</li> </ul>
7.2 Obiectivele specifice	<i>Obiective cognitive:</i> <ul style="list-style-type: none"> <li>- Cunoașterea metodelor de bază în elaborarea algoritmilor, a unor modalități de implementare eficientă și de aplicare adecvată la situații concrete a acestor metode.</li> <li>- Însușirea unor algoritmi fundamentali din teoria grafurilor.</li> </ul>

	<ul style="list-style-type: none"> <li>- Dobândirea de cunoștințe pentru demonstrarea corectitudinii și analiza eficienței algoritmilor.</li> </ul> <p><i>Obiective procedurale:</i></p> <ul style="list-style-type: none"> <li>- Formarea abilităților de proiectare și implementare de algoritmi pentru diverse clase de probleme.</li> <li>- Formarea deprinderi pentru rezolvarea de probleme modelate prin grafuri.</li> <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 și teoria grafurilor</b> Notății asimptotice. Exemple. Definiții generale privind grafurile. Clase particulare de grafuri. Exemple.	2	Prelegerea Explicația Descrierea și exemplificarea Demonstrația Problematizarea Conversația euristică Studiul de caz	Calculator, Videoproiector, Suport documentar
2	<b>Elemente de complexitatea algoritmilor și teoria grafurilor</b> Algoritmi pentru parcurgerile DF și BF ale grafurilor oarecare. Aplicații: determinarea componentelor conexe și a componentelor tare-conexe. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
3	<b>Proiectarea algoritmilor bazați pe metoda Greedy</b> Inegalitatea rearanjamentelor. Aplicații. Memorarea optimă a textelor pe benzi. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
4	<b>Proiectarea algoritmilor bazați pe metoda Greedy</b> Problema planificării spectacolelor. Problema rucsacului. Varianta continuă. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
5	<b>Proiectarea algoritmilor bazați pe metoda Greedy</b> Codificarea Huffman. Algoritmi pentru determinarea distanțelor și drumurilor minime și a arborilor parțiali de cost minim în grafuri ponderate. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
6	<b>Proiectarea algoritmilor bazați pe metoda Greedy</b> Algoritmi Ford-Fulkerson și Edmonds-Karp pentru determinarea fluxurilor maxime în rețele. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
7	<b>Proiectarea algoritmilor bazați pe metoda Backtracking</b> Scheme generale, iterative și recursive. Problema colorării grafurilor (hărților). Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
8	<b>Proiectarea algoritmilor bazați pe metoda Backtracking</b> Probleme pe tabla de șah: problema damelor, problema nebunilor, problema parcurgerii cu un cal. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
9	<b>Proiectarea algoritmilor bazați pe metoda Backtracking</b> Generarea de obiecte combinatoriale și de matrice/tablouri cu anumite proprietăți. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
10	<b>Proiectarea algoritmilor bazați pe metoda Divide et Impera</b> O problemă de acoperire. Problema celor mai apropiate două puncte. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
11	<b>Proiectarea algoritmilor bazați pe metoda programării dinamice</b> Descrierea metodei. Algoritmi generali. Subșir crescător de lungime maximă. Distanța Levenshtein. Subșir comun de lungime maximă.	2		

	Drumuri de sumă minimă/maximă în tablouri de numere. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.			
12	<b>Proiectarea algoritmilor bazați pe metoda programării dinamice</b> Algoritmul Roy-Warshall pentru determinarea matricei drumurilor unui graf. Algoritmul Roy-Floyd pentru determinarea matricei drumurilor minime a unui graf ponderat. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
13	<b>Metoda Branch and Bound</b> Descrierea metodei. Algoritm general. Jocul Perspico. Problema circuitului hamiltonian de cost minim. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.	2		
14	<b>Alegerea și combinarea metodelor. Alte tipuri de algoritmi</b> Analiza comparativă a metodelor anterioare. Exemple. Algoritmi euristici. Algoritmi probabilisti. Algoritmi pentru jocuri. Jocul NIM. Jocul SUDOKU.	2		
Bibliografie: 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, Proiectarea algoritmilor – note de curs (format electronic). 7. O. Băscă, L. Livovschi, Algoritmi euristici, Ed. Univ. din București, București, 2003. 8. E. Ciurea, L. 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).				
<b>8.2. Aplicații – Laborator</b>		Nr. ore	Metode de predare	Observații Resurse folosite
1	<b>Elemente de complexitatea algoritmilor și teoria grafurilor. Implementări în C++/Java</b> Notății asimptotice. Exemple. Definiții generale privind grafurile. Clase particulare de grafuri. Exemple. Algoritmi pentru parcurgerile DF și BF ale grafurilor oarecare. Aplicații: determinarea componentelor conexe și a componentelor tare-conexe. Probleme suplimentare.	2		
2	<b>Proiectarea algoritmilor bazați pe metoda Greedy. Implementări în C++/Java</b> Inegalitatea rearanjamentelor. Aplicații. Memorarea optimă a textelor pe benzi. Problema planificării spectacolelor. Problema rucsacului. Varianta continuă. Probleme suplimentare.	2	Exercițiul Explicația Studiul de caz Dezbateră Lucrul în grup Teme individuale	Calculator, Videoprojector, Suport documentar
3	<b>Proiectarea algoritmilor bazați pe metoda Greedy. Implementări în C++/Java</b> Codificarea Huffman. Algoritmi pentru determinarea distanțelor și drumurilor minime și a arborilor parțiali de cost minim în grafuri ponderate. Algoritmi Ford-Fulkerson și Edmonds-Karp pentru determinarea fluxurilor maxime în rețele. Probleme suplimentare.	2		

4	<b>Proiectarea algoritmilor bazați pe metoda Backtracking. Implementări în C++/Java</b> Problema colorării grafurilor (hărților). Probleme pe tabla de șah: problema damelor, problema nebunilor, problema parcurgerii cu un cal. Probleme suplimentare.	2		
5	<b>Proiectarea algoritmilor bazați pe metodele Backtracking și Divide et Impera. Implementări în C++/Java</b> Generarea de obiecte combinatoriale și de matrice/tablouri cu anumite proprietăți. O problemă de acoperire. Problema celor mai apropiate două puncte. Probleme suplimentare.	2		
6	<b>Proiectarea algoritmilor bazați pe metoda programării dinamice. Implementări în C++/Java</b> Subșir crescător de lungime maximă. Distanța Levenshtein. Subșir comun de lungime maximă. Drumuri de sumă minimă/maximă în tablouri de numere. Algoritmul Roy-Warshall pentru determinarea matricei drumurilor unui graf. Algoritmul Roy-Floyd pentru determinarea matricei drumurilor minime a unui graf ponderat. Probleme suplimentare.	2		
7	<b>Metoda Branch and Bound. Alegerea și combinarea metodelor. Alte tipuri de algoritmi. Implementări în C++/Java</b> Jocul Perspico. Problema circuitului hamiltonian de cost minim. Algoritmi euristici. Algoritmi probabiliști. Algoritmi pentru jocuri. Jocul NIM. Jocul SUDOKU. Probleme suplimentare.	2		
<p><b>Bibliografie:</b></p> <ol style="list-style-type: none"> <li>1. Gh. Barbu, V. Păun, Programarea în limbajul C/C++, Ed. Matrix Rom, București, 2011.</li> <li>2. Gh. Barbu, V. Paun, Calculatoare personale și programare în C/C++, Ed. Did. și Ped., București, 2005.</li> <li>3. C. Bălcău, Combinatorică și teoria grafurilor, Ed. Univ. din Pitești, Pitești, 2007.</li> <li>4. E. Cerchez, M. Șerban, Programarea în limbajul C/C++ pentru liceu. Vol. 2: Metode și tehnici de programare, Ed. Polirom, Iași, 2005.</li> <li>5. E. Ciurea, L. Ciupală, Algoritmi. Introducere în algoritmica fluxurilor în rețele, Ed. Matrix Rom, București, 2006.</li> <li>6. T.H. Cormen, Algorithms Unlocked, MIT Press, 2013.</li> <li>7. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009.</li> <li>8. N. Dale, C. Weems, Programming and problem solving with JAVA, Jones and Bartlett Publ., 2008.</li> <li>9. A. Ene, C. Știrbu, Programarea calculatoarelor și limbaje de programare: Teorie și aplicații, Ed. Univ. din Pitești, Pitești, 2005.</li> <li>10. A. Ene, C. Știrbu, 100 de probleme rezolvate în JAVA, Ed. Univ. din Pitești, Pitești, 2007.</li> <li>11. H. Georgescu, Tehnici de programare, Ed. Univ. din București, București, 2005.</li> <li>12. C.A. Giumale, Introducere în analiza algoritmilor. Teorie și aplicații, Ed. Polirom, Iași, 2004.</li> <li>13. F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, Springer, 2007.</li> <li>14. D. Jungnickel, Graphs, Networks and Algorithms, Springer, 2013.</li> <li>15. D. Logofătu, Algoritmi fundamentali în C++: Aplicații, Ed. Polirom, Iași, 2007.</li> <li>16. D. Lucanu, M. Craus, Proiectarea algoritmilor, Ed. Polirom, Iași, 2008.</li> <li>17. N. Popescu, Data structures and algorithms using Java, Ed. Politehnica Press, București, 2008.</li> <li>18. R. Stephens, Essential Algorithms: A Practical Approach to Computer Algorithms, Wiley, 2013.</li> <li>19. Ș. Tanasă, C. Olaru, Ș. Andrei, Java de la 0 la expert, Polirom, Iași, 2007.</li> <li>20. I. Tomescu, Data structures, Ed. Univ. București, București, 1997 (2004).</li> <li>21. A. Turcanu, Proiectarea algoritmilor – suport pentru laborator (format electronic).</li> </ol>				

**9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori 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, Prodin, 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	<ul style="list-style-type: none"> <li>* 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.</li> <li>* Set de cunoștințe minimale pentru promovarea examenului final: <ul style="list-style-type: none"> <li>- Cunoașterea principalelor noțiuni privind metodele de elaborare a algoritmilor;</li> <li>- Cunoașterea principalelor modalități de aplicare adecvată și de implementare eficientă a acestor metode în rezolvarea problemelor propuse.</li> </ul> </li> </ul>
------------------------------------	--

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