

FIȘA DISCIPLINEI
PROIECTAREA ȘI IMPLEMENTAREA ALGORITMILOR
Anul universitar 2020-2021

1. Date despre program

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Științe, Educație Fizică și Informatică
1.3	Departamentul	Matematică-Informatică
1.4	Domeniul de studii	Informatică
1.5	Ciclul de studii	Licență
1.6	Programul de studii / Calificarea	Informatică / Informatică

2. Date despre disciplină

2.1	Denumirea disciplinei	Proiectarea și implementarea algoritmilor									
2.2	Titularul activităților de curs	Conf. univ. dr. Bălcău Costel									
2.3	Titularul activităților de laborator	Asist. univ. dr. Diaconu Crina									
2.4	Anul de studii	2	2.5	Semestrul	1	2.6	Tipul de evaluare	Examen	2.7	Regimul disciplinei	O

3. Timpul total estimat

3.1	Număr de ore pe săptămână	4	3.2	din care curs	2	3.3	laborator	2
3.4	Total ore din planul de inv.	56	3.5	din care curs	28	3.6	laborator	28
Distribuția fondului de timp								Ore
Studiul după manual, suport de curs, bibliografie și notițe								58
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								14
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								14
Tutoriat								4
Examinări								4
Alte activități								
3.7	Total ore studiu individual			94				
3.8	Total ore pe semestru			150				
3.9	Număr de credite			6				

4. Precondiții (acolo unde este cazul)

4.1	De curriculum	
4.2	De competențe	Capacitate de analiză și sinteză

5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Sală dotată cu videoprojector
5.2	De desfășurare a seminarului/laboratorului	Laborator de informatică

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> Programarea în limbaje de nivel înalt. Dezvoltarea și întreținerea aplicațiilor informatice. Utilizarea instrumentelor informatice în context interdisciplinar. Utilizarea bazelor teoretice ale informaticii și a modelelor formale. Proiectarea și gestiunea bazelor de date.
Competențe transversale	<ul style="list-style-type: none"> Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatică de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională.

7. Obiectivele disciplinei

7.1	Obiectivul general al disciplinei	Disciplina are ca obiectiv general însușirea de către studenți a principalelor concepte din teoria proiectării și implementării algoritmilor, formarea de deprinderi pentru aplicarea acestora în rezolvarea de probleme și realizarea de aplicații, dezvoltarea gândirii algoritmice și însușirea unor tehnici de programare avansată.
7.2	Obiectivele specifice	Obiective cognitive: <ul style="list-style-type: none"> Cunoașterea metodelor de bază în elaborarea algoritmilor, a unor modalități de

	<p>implementare eficientă și de aplicare adecvată la situații concrete a acestor metode.</p> <ul style="list-style-type: none"> • Dobândirea de cunoștințe pentru demonstrarea corectitudinii și analiza eficienței algoritmilor. • Însușirea unor algoritmi eficienți de căutare și sortare <p>Obiective procedurale:</p> <ul style="list-style-type: none"> • Formarea abilităților de proiectare și implementare de algoritmi pentru diverse clase de probleme și aplicații. • Aplicarea unor algoritmi eficienți de căutare și sortare în rezolvarea de probleme și în realizarea de aplicații. • Formarea deprinderilor de investigare a problemelor din perspectiva diverselor metode posibile de rezolvare, în vederea obținerii unor soluții eficiente. • Dobândirea și dezvoltarea unui stil de programare profesionist. <p>Obiective atitudinale:</p> <ul style="list-style-type: none"> • Rigurozitate în proiectarea algoritmilor și în programare.
--	--

8. Conținuturi

8.1. Curs		Nr. ore	Metode de predare	Observații Resurse folosite
1	<p>1. Elemente de complexitatea algoritmilor</p> <p>1.1. Notății asimptotice. Exemple.</p> <p>1.2. Analiza eficienței algoritmilor. Exemple.</p>	2		
2	<p>2. Proiectarea algoritmilor bazați pe metoda Greedy</p> <p>2.1. Descrierea metodei. Algoritmi generali. Aplicabilitate.</p> <p>2.2. Inegalitatea rearanjamentelor.</p> <p>2.3. Memorarea optimală a textelor pe benzi.</p> <p>2.4. Problema rucsacului, varianta continuă.</p> <p>2.5. Problema planificării spectacolelor.</p> <p>2.6. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.</p>	4		
3	<p>3. Proiectarea algoritmilor bazați pe metoda Backtracking</p> <p>3.1. Descrierea metodei. Algoritmi generali, nerecursivi și recursivi. Aplicabilitate.</p> <p>3.2. Problema colorării grafurilor.</p> <p>3.3. Problema damelor pe tabla de șah.</p> <p>3.4. Problema nebunilor pe tabla de șah.</p> <p>3.5. Generarea obiectelor combinatoriale: produs cartezian, submulțimi, aranjamente cu repetiție, aranjamente, combinări, combinări cu repetiție, permutări, permutări cu repetiție, compuneri și partiții ale unui număr natural.</p> <p>3.6. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.</p>	6	Prelegerea Explicația Descrierea și exemplificarea Demonstrația Problematizarea	Calculator, Videoprojector, Suport documentar
4	<p>4. Proiectarea algoritmilor bazați pe metoda Divide et Impera</p> <p>4.1. Descrierea metodei. Algoritmi generali. Aplicabilitate.</p> <p>4.2. Problema turnurilor din Hanoi.</p> <p>4.3. Căutarea binară.</p> <p>4.4. Sortarea prin interclasare. Sortarea rapidă.</p> <p>4.5. O problemă de acoperire.</p> <p>4.6. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.</p>	6	Conversația euristică Studiul de caz	
5	<p>5. Proiectarea algoritmilor bazați pe metoda programării dinamice</p> <p>5.1. Descrierea metodei. Algoritmi generali. Aplicabilitate.</p> <p>5.2. Distanța Levenshtein.</p> <p>5.3. Subșir crescător de lungime maximă.</p> <p>5.4. Drumuri de sumă minimă/maximă în tablouri de numere.</p> <p>5.5. Demonstrarea corectitudinii și evaluarea complexității algoritmilor propuși.</p>	4		
6	<p>6. Alte metode de programare</p> <p>6.1. Metoda Branch and Bound. Aplicabilitate. Jocul Perspico. Problema circuitului hamiltonian de cost minim.</p> <p>6.2. Algoritmi euristici. Aplicabilitate. Exemple.</p> <p>6.3. Demonstrarea corectitudinii, evaluarea optimalității soluțiilor și a complexității algoritmilor propuși.</p>	6		
<p>Bibliografie:</p> <ol style="list-style-type: none"> 1. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Data Structures and Algorithms, Addison-Wesley, Massachusetts, 2009. 2. Gh. Barbu, V. Păun, Programarea în limbajul C/C++, Ed. Matrix Rom, București, 2011. 3. Gh. Barbu, I. Văduva, M. Boloșteanu, Bazele informaticii, Ed. Tehnică, București, 1997. 4. C. Bălcău, Combinatorică și teoria grafurilor, Ed. Univ. din Pitești, Pitești, 2007. 				

5. C. Bălcău, Proiectarea și implementarea algoritmilor – note de curs (format electronic).
6. O. Bâscă, L. Livovschi, Algoritmi euristici, Ed. Univ. din București, București, 2003.
7. E. Cerchez, M. Șerban, Programarea în limbajul C/C++ pentru liceu. Vol. 2: Metode și tehnici de programare, Ed. Polirom, Iași, 2005.
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, Cambridge, 2013.
10. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Cambridge, 2009.
11. C. Croitoru, Tehnici de bază în optimizarea combinatorie, Ed. Univ. Al. I. Cuza, Iași, 1992.
12. N. Dale, C. Weems, Programming and problem solving with JAVA, Jones and Bartlett Publ., 2008.
13. P. Deitel, H. Deitel, Java SE 8 for programmers, Deitel & Associates, Boston, 2014.
14. S. Even, Graph Algorithms, Cambridge University Press, 2012.
15. H. Georgescu, Tehnici de programare, Ed. Univ. din București, București, 2005.
16. C.A. Giumale, Introducere în analiza algoritmilor. Teorie și aplicații, Ed. Polirom, Iași, 2004.
17. M. Goodrich, R. Tamassia, Algorithm Design. Foundations, Analysis and Internet Examples, Wiley, New Delhi, 2011.
18. R. Harper, Practical foundations for programming languages, Cambridge University Press, Cambridge, 2013.
19. F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, Springer, 2007.
20. D. Jungnickel, Graphs, Networks and Algorithms, Springer, 2013.
21. D.E. Knuth, The Art Of Computer Programming. Vol. 4A: Combinatorial Algorithms, Addison-Wesley, Massachusetts, 2011.
22. B. Korte, J. Vygen, Combinatorial Optimization. Theory and Algorithms, Springer, 2012.
23. R. Lafore, Data Structures and Algorithms in Java, Sams Publishing, Indianapolis, 2002.
24. A. Levitin, Introduction to The Design and Analysis of Algorithms, Pearson, Boston, 2012.
25. L. Livovschi, H. Georgescu, Sinteza și analiza algoritmilor, Ed. Științifică și Enciclopedică, București, 1986.
26. D. Logofătu, Algoritmi fundamentali în Java: Aplicații, Ed. Polirom, Iași, 2007.
27. D. Lucanu, M. Craus, Proiectarea algoritmilor, Ed. Polirom, Iași, 2008.
28. S. Miller, Mathematics of Optimization: How to do Things Faster, AMS, Providence, 2017.
29. D.A. Popescu, Bazele programării - JAVA după C++, ebooks.infobits.ro, 2019.
30. D.R. Popescu, Combinatorică și teoria grafurilor, Soc. de Șt. Matem. din România, București, 2005.
31. N. Popescu, Data structures and algorithms using Java, Ed. Politehnica Press, București, 2008.
32. C.P. Popovici, H. Georgescu, L. State, Bazele informaticii. Vol. I, II, Ed. Univ. din București, București, 1990, 1991.
33. R. Sedgewick, P. Flajolet, An Introduction to the Analysis of Algorithms, Addison-Wesley, New Jersey, 2013.
34. R. Sedgewick, K. Wayne. Algorithms, Addison Wesley, Massachusetts, 2011.
35. R. Stephens, Essential Algorithms: A Practical Approach to Computer Algorithms, Wiley, Indianapolis, 2013.
36. Ș. Tanasă, C. Olaru, Ș. Andrei, Java de la 0 la expert, Polirom, Iași, 2007.
37. T. Toadere, Grafe. Teorie, algoritmi și aplicații, Ed. Albastră, Cluj-Napoca, 2002.
38. I. Tomescu, Data structures, Ed. Univ. București, București, 1997 (2004).
39. I. Tomescu, Probleme de combinatorică și teoria grafurilor, Ed. Did. și Ped., București, 1981.
40. M.A. Weiss, Data Structures and Algorithm Analysis in Java, Addison-Wesley, New Jersey, 2012.
41. Revista MATINF, Ed. Univ. din Pitești, Pitești, 2018-2020.

8.2. Aplicații – Laborator		Nr. ore	Metode de predare	Observații Resurse folosite
1	1. Fundamente ale programării în limbajul Java 1.1. Structura lexicală a limbajului Java. 1.2. Tipuri de date și variabile. 1.3. Operatori, expresii, instrucțiuni. 1.4. Vectori și tablouri multidimensionale. 1.5. Obiecte și clase. 1.6. Pachete. Utilizarea unor clase predefinite. 1.7. Probleme.	8	Exercițiul Explicația Studiul de caz Dezbaterea Lucrul în grup Teme individuale	Calculator, Videoproiector, Suport documentar
2	2. Metoda Greedy. Implementări în Java 2.1. Memorarea optimală a textelor pe benzi. 2.2. Problema rucsacului, varianta continuă. 2.3. Problema planificării spectacolelor. 2.4. Probleme suplimentare.	4		
3	3. Metoda Backtracking. Implementări în Java 3.1. Problema colorării grafurilor. 3.2. Problema damelor pe tabla de șah. 3.3. Problema nebunilor pe tabla de șah. 3.4. Generarea obiectelor combinatoriale: produs cartezian, submulțimi, aranjamente cu repetiție, aranjamente, combinări, combinări cu repetiție, permutări, permutări cu repetiție, compuneri și partiții ale unui număr natural. 3.5. Probleme suplimentare.	4		
4	4. Metoda Divide et Impera. Implementări în Java 4.1. Problema turnurilor din Hanoi. 4.2. Căutarea binară. 4.3. Problema interclasării. 4.4. Sortarea prin interclasare. Sortarea rapidă.	4		

	4.5. O problemă de acoperire. 4.6. Probleme suplimentare.			
5	5. Metoda programării dinamice. Implementări în Java 5.1. Distanța Levenshtein. 5.2. Subșir crescător de lungime maximă. 5.3. Drumuri de sumă minimă/maximă în tablouri de numere. 5.4. Probleme suplimentare.	4		
6	6. Alte metode de programare. Implementări în Java 6.1. Jocul Perspico. 6.2. Problema circuitului hamiltonian de cost minim. 6.3. Probleme suplimentare.	4		

Bibliografie:

1. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Data Structures and Algorithms, Addison-Wesley, Massachusetts, 2009.
2. Gh. Barbu, V. Păun, Programarea în limbajul C/C++, Ed. Matrix Rom, București, 2011.
3. Gh. Barbu, I. Văduva, M. Boloșteanu, Bazele informaticii, Ed. Tehnică, București, 1997.
4. C. Bălcău, Combinatorică și teoria grafurilor, Ed. Univ. din Pitești, Pitești, 2007.
5. O. Bâscă, L. Livovschi, Algoritmi euristici, Ed. Univ. din București, București, 2003.
6. E. Cerchez, M. Șerban, Programarea în limbajul C/C++ pentru liceu. Vol. 2: Metode și tehnici de programare, Ed. Polirom, Iași, 2005.
7. E. Ciurea, L. Ciupală, Algoritmi. Introducere în algoritmica fluxurilor în rețele, Ed. Matrix Rom, București, 2006.
8. T.H. Cormen, Algorithms Unlocked, MIT Press, Cambridge, 2013.
9. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Cambridge, 2009.
10. C. Croitoru, Tehnici de bază în optimizarea combinatorie, Ed. Univ. Al. I. Cuza, Iași, 1992.
11. N. Dale, C. Weems, Programming and problem solving with JAVA, Jones and Bartlett Publ., 2008.
12. P. Deitel, H. Deitel, Java SE 8 for programmers, Deitel & Associates, Boston, 2014.
13. S. Even, Graph Algorithms, Cambridge University Press, 2012.
14. H. Georgescu, Tehnici de programare, Ed. Univ. din București, București, 2005.
15. C.A. Giumale, Introducere în analiza algoritmilor. Teorie și aplicații, Ed. Polirom, Iași, 2004.
16. M. Goodrich, R. Tamassia, Algorithm Design. Foundations, Analysis and Internet Examples, Wiley, New Delhi, 2011.
17. R. Harper, Practical foundations for programming languages, Cambridge University Press, Cambridge, 2013.
18. F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs, Springer, 2007.
19. D. Jungnickel, Graphs, Networks and Algorithms, Springer, 2013.
20. D.E. Knuth, The Art Of Computer Programming. Vol. 4A: Combinatorial Algorithms, Addison-Wesley, Massachusetts, 2011.
21. B. Korte, J. Vygen, Combinatorial Optimization. Theory and Algorithms, Springer, 2012.
22. R. Lafore, Data Structures and Algorithms in Java, Sams Publishing, Indianapolis, 2002.
23. A. Levitin, Introduction to The Design and Analysis of Algorithms, Pearson, Boston, 2012.
24. L. Livovschi, H. Georgescu, Sinteza și analiza algoritmilor, Ed. Științifică și Enciclopedică, București, 1986.
25. D. Logofătu, Algoritmi fundamentali în Java: Aplicații, Ed. Polirom, Iași, 2007.
26. D. Lucanu, M. Craus, Proiectarea algoritmilor, Ed. Polirom, Iași, 2008.
27. S. Miller, Mathematics of Optimization: How to do Things Faster, AMS, Providence, 2017.
28. D.A. Popescu, Bazele programării - JAVA după C++, ebooks.infobits.ro, 2019.
29. D.R. Popescu, Combinatorică și teoria grafurilor, Soc. de Șt. Matem. din România, București, 2005.
30. N. Popescu, Data structures and algorithms using Java, Ed. Politehnica Press, București, 2008.
31. C.P. Popovici, H. Georgescu, L. State, Bazele informaticii. Vol. I, II, Ed. Univ. din București, București, 1990, 1991.
32. R. Sedgewick, P. Flajolet, An Introduction to the Analysis of Algorithms, Addison-Wesley, New Jersey, 2013.
33. R. Sedgewick, K. Wayne. Algorithms, Addison Wesley, Massachusetts, 2011.
34. R. Stephens, Essential Algorithms: A Practical Approach to Computer Algorithms, Wiley, Indianapolis, 2013.
35. Ș. Tanasă, C. Olaru, Ș. Andrei, Java de la 0 la expert, Polirom, Iași, 2007.
36. T. Toadere, Grafe. Teorie, algoritmi și aplicații, Ed. Albastră, Cluj-Napoca, 2002.
37. I. Tomescu, Data structures, Ed. Univ. București, București, 1997 (2004).
38. I. Tomescu, Probleme de combinatorică și teoria grafurilor, Ed. Did. și Ped., București, 1981.
39. C. Diaconu, Proiectarea și implementarea algoritmilor – îndrumar pentru laborator (format electronic).
40. M.A. Weiss, Data Structures and Algorithm Analysis in Java, Addison-Wesley, New Jersey, 2012.
41. Revista MATINF, Ed. Univ. din Pitești, Pitești, 2018-2020.

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

Competențele dobândite în cadrul disciplinei permit absolvenților să utilizeze eficient modelele și tehnicile specifice proiectării, analizei și implementării algoritmilor în rezolvarea cerințelor aferente practicii și cercetării în domeniul informaticii. 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, Proding, 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, algoritmi - cu implementare - și probleme)	50%
10.5 Laborator	Activitate (rezolvarea și implementarea problemelor propuse) Tema de casă	Verificare soluții, probe practice Verificare temă	30% 20%
10.6 Standard minim de performanță	* Promovarea probelor practice; 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 elaborare a algoritmilor; - Cunoașterea principalelor modalități de aplicare adecvată și de implementare eficientă a acestor metode în rezolvarea problemelor propuse.		

Data completării
20.09.2020

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

Titular de laborator,
Asist univ. dr. Crina DIACONU

Data avizării în Departament
23.09.2020

Director Departament (prestator)
Conf.univ.dr. Doru CONSTANTIN

Director Departament (beneficiar)
Conf.univ.dr. Doru CONSTANTIN