

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Rețele de Comunicații și Sisteme Distribuite / Master
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	7.3

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Algoritmi si Calculabilitate</b>				
2.2 Titularii de curs	Prof. dr. ing. Rodica Potolea - <a href="mailto:Rodica.Potolea@cs.utcluj.ro">Rodica.Potolea@cs.utcluj.ro</a>				
2.3 Titularul / Titularii activităților de seminar / laborator / proiect	Prof. dr. ing. Rodica Potolea - <a href="mailto:Rodica.Potolea@cs.utcluj.ro">Rodica.Potolea@cs.utcluj.ro</a>				
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6 Tipul de evaluare ( E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

### 3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	1	Laborator	-	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	14	Laborator	-	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										40
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										-
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										16
(d) Tutoriat										-
(e) Examinări										2
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))							58			
3.5 Total ore pe semestru (3.2+3.4)							100			
3.6 Numărul de credite							4			

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Structuri de Date, Tehnici de Programare, Algoritmi,
4.2 de competențe	Evaluare eficienta algoritmi, Cunosterea algoritmilor fundamentali pe structurile de date de baza

### 5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator
5.2. de desfășurare a seminarului / laboratorului / proiectului	Calculatoare

### 6. Competențele specifice acumulate

6.1 Competențe profesionale	<p><b>C1</b> - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> <li>• <b>C1.1</b> - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software</li> <li>• <b>C1.2</b> - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate</li> <li>• <b>C1.3</b> - Utilizarea unor modele și metode specifice pentru identificarea de</li> </ul>
-----------------------------	---

	<p>componente si solutii software viabile în condiții de specificare parțială</p> <ul style="list-style-type: none"> <li>• <b>C1.4</b> - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor si modelelor de dezvoltare software, precum si a sistemelor software complexe.</li> <li>• <b>C1.5</b> - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate in toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare si integrare, validare).</li> </ul> <p><b>C2</b> - Elaborarea de tehnici, metode, si metodologii avansate in domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> <li>• <b>C2.1</b> - Identificarea și descrierea structurii și a modului de funcționare a sistemelor de software complexe și a aplicațiilor dezvoltate pe baza acestora</li> <li>• <b>C2.2</b> - Exploatarea cunoștințelor de specialitate în vederea identificării și înțelegerii metodologiilor și tehnicilor de realizare a componentelor hardware și software</li> <li>• <b>C2.3</b> - Construirea unor componente software originale ale sistemelor avansate de programe, folosind algoritimi, tehnici, metode de proiectare, metodologii, protocoale, limbaje de programare, structuri de date, tehnologii si medii de programare complexe, raportate in literatura de specialitate.</li> <li>• <b>C2.4</b> - Utilizarea de metode, criterii si metrici de evaluare si selectie a metodologiilor de realizarea a sistemelor software, a caracteristicilor lor funcționale și non-funcționale</li> <li>• <b>C2.5</b> - Elaborarea de proiecte software originale, implementarea, testarea si validarea acestora pe baza combinării inovative a celor raportate în literatura de specialitate</li> </ul>
6.2 Competențe transversale	CT3 - Exersarea deprinderii de autoeducare continuă și demonstrarea de abilități critice, inovatoare și de cercetare

### 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Analiza obiectiva a unei probleme si identificarea complexitatii sale; identificarea de solutii potentiale si alegerea celei potrivite contextului dat.
7.2 Obiectivele specifice	Calcul de complexitate; diferentierea claselor de complexitate; cunosterea tipurilor de probleme care nu au solutii „usoare”.

### 8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Complexitate	2	Scenariul <b>onsite</b> : Expunere la tablă, prezentare cu videoproiectorul, discuții Scenariul <b>online</b> : Prezentare si discutii sincrone Teams, materiale de curs, teme platforma Moodle, Skype	
Timp polinomial, P si NP	2		
NP completitudine	2		
Reductibilitate	2		
P=?NP	2		
Demonstrare NP-comp	2		
Probleme NP-comp	2		
Model Computational	2		
Masina Turing	2		
Reducere NP-comp1 (Circuit-SAT, SAT, 3-FNC-SAT)	2		
Reducere NP-comp2 (Clica, Acoperire varfuri, Suma)	2		
Reducere NP-comp3 (Ciclu Hamiltonian, Comis Voiajor)	2		
Aplicatii NP-comp si aproximari 1	2		
Aplicatii NP-comp si aproximari 2	2		
Bibliografie			
1. Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. <a href="#">Introduction to Algorithms</a> . 2nd ed. Cambridge, MA: MIT Press, 2001. ISBN: 9780262032933			

<p>2. Papadimitriou, C. H. <a href="#">Computational Complexity</a>. 1st ed. Boston: Addison Wesley Publishing Company, 1994. ISBN: 0201530821</p> <p>3. Arora, Sanjeev and Barak, Boaz <i>Complexity Theory: A Modern Approach</i>, Princeton University, available on the web, <a href="http://www.cs.princeton.edu/theory/index.php/Compbook/Draft">http://www.cs.princeton.edu/theory/index.php/Compbook/Draft</a></p> <p>4. Sipser, Michael. <i>Introduction to the Theory of Computation</i>. 2nd ed. Boston, MA: Course Technology, 2005. ISBN: 9780534950972</p>			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
P vs NPC	1	Expuneri la tablă, explicații suplimentare, discuții	
Demonstrari apartenenta NPC	1		
Algoritmi de reducere1	1		
Algoritmi de reducere2	1		
Aproximari1	1		
Aproximari2	1		
Probleme NPC	1		
<p><b>Bibliografie</b></p> <p>1. Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. <a href="#">Introduction to Algorithms</a>. 2nd ed. Cambridge, MA: MIT Press, 2001. ISBN: 9780262032933</p> <p>2. Papadimitriou, C. H. <a href="#">Computational Complexity</a>. 1st ed. Boston: Addison Wesley Publishing Company, 1994. ISBN: 0201530821</p> <p>3. Arora, Sanjeev and Barak, Boaz <i>Complexity Theory: A Modern Approach</i>, Princeton University, available on the web, <a href="http://www.cs.princeton.edu/theory/index.php/Compbook/Draft">http://www.cs.princeton.edu/theory/index.php/Compbook/Draft</a></p> <p>4. Sipser, Michael. <i>Introduction to the Theory of Computation</i>. 2nd ed. Boston, MA: Course Technology, 2005. ISBN: 9780534950972</p>			

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

Identificarea problemelor “dificile” și rezolvarea lor aproximativă; cunoașterea de soluții exacte ale problemelor “usoare”.
--

**10. Evaluare**

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Examinare	Examen scris	50%
Seminar	Prezentare/eseu/rezolvare probleme complexe	Prezentare (P)	50%
Laborator	-	-	-
Proiect	-	-	-
Standard minim de performanță: Min 5 (50%E+50%P>=5)			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
26.06.2023	Curs	Prof. dr. ing. Rodica Potolea	
	Aplicații	Prof. dr. ing. Rodica Potolea	

<b>Data avizării în Consiliul Departamentului Calculatoare</b>	Director Departament, Prof. dr. ing. Rodica Potolea
<b>Data aprobării în Consiliul Facultății de Automatică și Calculatoare</b>	Decan, Prof. dr. ing. Liviu Miclea