

UČNI NAČRT PREDMETA / COURSE SYLLABUS (leto / year 2017/18)						
<b>Predmet:</b>		Algoritmi in podatkovne strukture 2				
<b>Course title:</b>		Algorithms and data structures 2				
<b>Študijski program in stopnja</b> Study programme and level		<b>Študijska smer</b> Study field		<b>Letnik</b> Academic year	<b>Semester</b> Semester	
Interdisciplinarni univerzitetni študijski program Računalništvo in matematika		ni smeri		2	drugi	
Interdisciplinary first cycle academic study programme Computer Science and Mathematics		none		2	second	
<b>Vrsta predmeta / Course type</b>				obvezni / compulsory		
<b>Univerzitetna koda predmeta / University course code:</b>				63280		
<b>Predavanja</b> Lectures	<b>Seminar</b> Seminar	<b>Vaje</b> Tutorial	<b>Klinične vaje</b> work	<b>Druge oblike študija</b>	<b>Samost. delo</b> Individ. work	<b>ECTS</b>
45		30			105	6
<b>Nosilec predmeta / Lecturer:</b>				prof. dr. Borut Robič		
<b>Jeziki / Languages:</b>		<b>Predavanja / Lectures:</b>		slovenski / Slovene		
		<b>Vaje / Tutorial:</b>		slovenski / Slovene		
<b>Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:</b>				<b>Prerequisites:</b>		
Vpis v letnik študija.				Enrolment in the programme.		
<b>Vsebina:</b>				<b>Content (Syllabus outline):</b>		

Predavanja:

Uvod: splošno o metodah razvoja algoritmov, o analizi algoritmov, o računski zahtevnosti algoritmov in problemov

Deli in vladaj: opis metode, primeri problemov in algoritmov (glejte primere v točki 12 spodaj)

Požrešna metoda: opis metode, primeri

Postopno izboljševanje: opis, primeri

Dinamično programiranje: opis, primeri

Sestopanje: opis metode, primeri

Razveji in omeji: opis metode, primeri

Linearno programiranje: opis metode, simpleksni algoritem, primeri

Izbrane višje podatkovne strukture

NP-težki računski problemi: spodnja meja časovne zahtevnosti, intuitivno o razredih P, NP in NP-težkih problemih

Metode reševanja NP-težkih problemov: heuristični algoritmi, aproksimacijski algoritmi, verjetnostni algoritmi, parametrizirani algoritmi, eksaktni eksponentni algoritmi, primeri

Primeri problemov in algoritmov: napredno urejanje &, Heapsort, Quicksort, problem izbiranja &, linearni algoritmi, matrično množenje &, Strassenov alg., diskretna Fourierova transformacija &, FFT alg., iskanje v nizih &, Knuth-Morris-Prattov algoritem, osnovni in zahtevnejši problemi in algoritmi na grafih (iskanje v grafu, topološko urejanje, maksimalni pretok &, Ford-Fulkersonov alg., najkrajše poti &, Bellman-Fordov ter Floyd-

Lectures:

Intro: about methods of algorithm design, analysis of algorithms, and computational complexity of algorithms and problems

Divide-and-Conquer: description of the method, examples of problems and algorithms (see examples 12 below)

Greedy method: description, examples

Iterative improvement: descr., examples

Dynamic programming: descr., examples

Backtracking: description, examples

Branch&Bound: description, examples

Linear programming: descr., Simplex algorithm, examples

Selected advanced data structures

NP-hard computational problems: lower bounds on time complexity, informally about P, NP and NP-hard problems,

Methods of solving NP-hard problems: heuristic algorithms, approximation algorithms, randomized algorithms, parameterized algorithms, exact exponential algorithms, examples

Example problems and algorithms: advanced sorting &, Heapsort, Quicksort, selection problem &, linear algorithms, matrix multiplication &, Strassen alg., Discrete Fourier Transformation &, FFT alg, string matching &, Knuth-Morris-Pratt, elementary and other graph problems and algorithms (searching a graph, topological sort, maximum flow &, Ford-Fulkerson alg., shortest paths &, algorithms of Bellman-Ford, and Floyd-Warshall), selected

<p>Warshallov alg.) , izbrani problemi iz računske geometrije.</p> <p>Vaje: Na vajah bodo študentje utrjevali snov, podano na predavanjih. Snov bodo uporabili za reševanje praktičnih problemov, pri čemer bo poudarek na samostojnem delu ob pomoči asistentov. Implementirali bodo več manjših programov (kot domače naloge) in obsežnejše programe (kot seminarske naloge), ki jih bodo zagovarjali na vajah.</p> <p>Domače in seminarske naloge:</p> <p>Namen domačih in seminarskih nalog je dati študentom priložnost za reševanje raznih računskih problemov s samostojnim razvojem algoritmov in njihovim programiranjem (in jih spodbuditi k sprotnemu študiju).</p>	<p>problems from computational geometry.</p> <p>Tutorial: Students will use the topics given during the lectures to independently solve practical problems (with the assistance of the TAs if needed). They will implement several smaller programs (home works) as well as larger programs (seminars), and present them at the tutorial.</p> <p>Home works and seminars:</p> <p>These are necessary for a student to independently practice the design and implementation of algorithms .</p>
--	--

#### Temeljni literatura in viri / Readings:

B. Robič: Algoritmi (to appear, instead of 2. below)

B. Vilfan: Osnovni algoritmi, Založba FE in FRI, 2002

Dodatna literatura:

T. Cormen et al. Introduction to Algorithms, McGraw-Hill, 3rd ed., 2009

B. Robič: Aproksimacijski algoritmi, Založba FE in FRI, 2. izdaja, 2009

#### Cilji in kompetence:

Cilj predmeta je pridobiti poglobljeno znanje s področij načrtovanja algoritmov, analize algoritmov, uporabe podatkovnih struktur, izbranih problemov in algoritmov ter ob vsem tem utrjevati in poglobljati znanje programiranja.

#### Objectives and competences:

To gain deeper knowledge of algorithm design methods, analysis of algorithms, use of data structures , selected problems and algorithms, and at the same time, to improve and deepen programming skills.

#### Predvideni študijski rezultati:

#### Intended learning outcomes:

Znanje in razumevanje:

Razumevanje različnih pristopov k programiranju in primernost raznih pristopov za reševanje raznih problemov,

Pregled principov in mehanizmov raznih vrst programskih jezikov,

Razumevanje načinov za opisovanje sintakse in pomena programskih jezikov ter formalno dokazovanje pravilnosti programov.

Uporaba:

Razvoj spretnosti simboličnega programiranja, programiranja v logiki in programiranja z omejitvami.

Refleksija:

Sposobnost razmišljanja o alternativnih formulacijah problemov ter pristopov k njihovem reševanju,

Kako različni modeli računanja, paradigme programiranja in vrste jezikov spodbujajo alternativne pristope k računalniškemu reševanju problemov.

Prenosljive spretnosti - niso vezane le na en

predmet:

Razširjene spretnosti snovanja programov.

Knowledge and understanding:

The ability to independently design algorithms and data structures for solving particular computational problems, the ability to independently analyze computational complexity of algorithms (and sometimes problems as well), the ability to independently develop and implement computer programs.

Application: use of the principles and methods in algorithm design and implementation

Reflection: understanding of the basic principles of algorithm design and their role in efficient solving of computational problems

Transferable skills: there are many and useful in other subjects. For example, the ability to plan, design, and implement algorithmic solutions to various problems (regardless of the programming language used)

**Metode poučevanja in učenja:**

Predavanja, domače naloge, seminarski način dela pri vajah. Poudarek je na sprotnem študiju in samostojnem delu pri vajah, seminarskih in domačih nalogah.

**Learning and teaching methods:**

Lectures, tutorial, home works, seminars.

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment:
Način: pisni izpit, ustno izpraševanje, naloge, projekt		Type (examination, oral, coursework, project):
Sprotno preverjanje:		Continuing (homework)
domače naloge, projektno delo	50%	Final (written and oral exam)
Končno preverjanje: pisni in ustni izpit	50%	Grading: 6-10 pass, 1-5 fail.
Ocene: 6-10 pozitivno, 1-5 negativno		
(v skladu s Statutom UL)		

**Reference nosilca / Lecturer's references:**

ČIBEJ, Uroš, SLIVNIK, Boštjan, ROBIČ, Borut. The complexity of static data replication in data grids. *Parallel Computing*, ISSN 0167-8191. [Print ed.], 2005, vol. 31, no. 8/9, str. [900]-912, ilustr. [COBISS.SI-ID 4995412]

SULISTIO, Anthony, ČIBEJ, Uroš, VENUGOPAL, Srikumar, ROBIČ, Borut, BUYYA, Rajkumar. A toolkit for modelling and simulating data Grids : an extension to GridSim. *Concurrency and computation*, ISSN 1532-0626. [Print ed.], Sep. 2008, vol. 20, no. 13, str. 1591-1609, ilustr. , doi: . [COBISS.SI-ID 6533716]

TROBEC, Roman, ŠTERK, Marjan, ROBIČ, Borut. Computational complexity and parallelization of the meshless local Petrov-Galerkin methods. *Computers & Structures*, ISSN 0045-7949. [Print ed.], 2009, vol. 87, no. 1/2, str. 81-90. [COBISS.SI-ID 21895463]

MIHELIČ, Jurij, ROBIČ, Borut. Flexible-attribute problems. *Computational optimization and applications*, ISSN 0926-6003. [Print ed.], 2010, vol. 47, no. 3, str. 553-566, ilustr. [COBISS.SI-ID 7087700]

MIHELIČ, Jurij, MAHJOUR, Amine, RAPINE, Christophe, ROBIČ, Borut. Two-stage flexible-choice problems under uncertainty. *European journal of operational research*, ISSN 0377-2217. [Print ed.], Mar. 2010, vol. 201, no. 2, str. 399-403, ilustr. [COBISS.SI-ID 7087444]