

	UČNI NAČRT PREDMETA/COURSE SYLLABUS
Predmet	Algoritmi in podatkovne strukture
Course title	Algorithms and Data Structures

Študijski program in stopnja Study programme and level	Študijska smer Study field	Letnik Academic year	Semester Semester
Poslovna informatika / 1. stopnja	Računalništvo in informatika	2. letnik	3.
Business Informatics / 1 st Cycle	Computer and Information Science	2 nd year	3 rd

Vrsta predmeta/Course type obvezni/obligatory

Univerzitetna koda predmeta/University course code I_RI_2_UN2

Predavanja Lectures	Seminar Seminar	Sem. vaje Tutorial	Lab. vaje Laboratory work	Teren. vaje Field work	Samost. delo Individ. work	ECTS
45			45		85	7

Nosilec predmeta/Lecturer: dr. Borut Čampelj, pred.

Jeziki/ Languages:	Predavanja/Lectures:	slovenski/Slovenian
	Vaje/Tutorial:	slovenski/Slovenian

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:

Prerequisites:

<ul style="list-style-type: none"> Vpis v drugi letnik študijskega programa. Študent mora pred izpitom pripraviti in predstaviti seminarsko nalogo. 	<ul style="list-style-type: none"> The prerequisite for inclusion is enrolment in the second year of study. Students have to successfully prepare and present a seminar paper before the examination.
---	---

Vsebina:

Content (Syllabus outline):

<ul style="list-style-type: none"> <i>Uvod:</i> osnovni principi reševanja problemov, pojem algoritma, zapis algoritma (pseudokoda, diagrami poteka, programski jezik), sled algoritma, osnovni podatkovni tipi, abstraktni podatkovni tipi. <i>Analiza algoritmov:</i> časovna in prostorska zahtevnost algoritmov, velikostni red časovne zahtevnosti, modeliranje 	<ul style="list-style-type: none"> <i>Introduction:</i> basic principles of solving problems, the concept of algorithm, algorithm decryption, algorithm trace, pseudo code, basic data types, abstract data types. <i>Analysis of algorithms:</i> time and space complexity of algorithms, order of magnitude of time complexity, modelling of the actual time of implementation, differences among complexity classes.
--	---

<p>dejanskega časa izvajanja, razlike med razredi zahtevnosti.</p> <ul style="list-style-type: none"> • <i>Rekurzija</i>: kaj je rekurzija, primeri rekurzivnih algoritmov (funkcija faktoriele, Fibonaccijevo zaporedje, Evklidov algoritem), repna rekurzija, spreminjanje repne rekurzije v iteracijo, prednosti in slabosti rekurzivnih algoritmov v primerjavi z iterativnimi. • <i>Strategije reševanja problemov</i>: deli in vladaj (iskanje maksimuma, rekurzivni sortirni algoritmi, hanojski stolpi), dinamično programiranje (problem nahrbtnika, iskanje najkrajše poti, problem trgovskega potnika). • <i>Podatkovne strukture</i>: kaj so podatkovne strukture, izbor strukture za določen problem, osnovne strukture (npr. polje) s primeri, pregled podatkovnih struktur. • <i>Seznami in preslikave</i>: definicije abstraktnih podatkovnih tipov z različnimi implementacijami (s poljem, s kazalci...): enosmerni seznam, dvosmerni seznam, množica, vrsta, sklad, zgoščena tabela. Primerjava časovne zahtevnosti operacij znotraj različnih implementacij. • <i>Drevo</i>: o drevesih, definicija abstraktnega podatkovnega tipa drevo, vrste obhodov drevesa, implementacije dreves, binarna drevesa. • <i>Slovar</i>: definicija abstraktnega podatkovnega tipa slovar in operacij nad slovarjem, implementacije slovarja z različnimi drevesnimi strukturami (binarna iskalna drevesa, AVL drevesa, 2-3 drevesa, B- drevesa). Prednosti in slabosti navedenih drevesnih struktur, primerjava in analiza učinkovitosti metod različnih implementacij. • <i>Prioritetna vrsta</i>: definicija abstraktnega podatkovnega tipa vrsta, implementacija z delno urejenim levo poravnanim drevesom (kopico). • <i>Graf</i>: definicija abstraktnih podatkovnih tipov usmerjeni graf (DI-graph) in neusmerjeni graf (graph), primeri grafov, analiza kritične poti, minimalno vpeto drevo in iskanje najkrajših poti kot 	<ul style="list-style-type: none"> • <i>Recursion</i>: what is recursion, examples of recursive algorithms (Factorial function, Fibonacci sequence, Euclid algorithm), tail recursion, the change of the tailed recursion into iteration, advantages and disadvantages of recursive algorithms comparing to iterative. • <i>Strategies of solving problems</i>: divide and conquer (find the maximum, recursive sorting algorithms, towers of Hanoi), dynamic programming (knapsack problem, shortest path problem, traveling salesman problem). • <i>Data structures</i>: what are data structures, selecting a structure for a particular problem, basic structures (e.g. array) with examples, an overview of data structures. • <i>Lists and mappings</i>: definition of abstract data types with different implementations (arrays, pointers...): one-way list and two-way list, set, queue, stack, hash table. Comparison of time complexity of operations in different implementations. • <i>Tree</i>: about trees, definition of an abstract data type tree, tree traversal types, tree implementations, binary trees. • <i>Dictionary</i>: definition of an abstract data type glossary in its operations, dictionary implementations with different tree structures (binary search trees, AVL trees, 2-3 trees, B-trees). Advantages and disadvantages of the above tree structures, comparison and analysis of the effectiveness of different methods implementations. • <i>Priority queue</i>: definition of an abstract data type queue, implementation with a partially sorted left-aligned tree (heap). • <i>Graph</i>: definition of an abstract data types directed graph (DI-graph) and undirected graph (graph), graph examples, critical path analysis, minimum spanning tree and search for the shortest path as examples of greedy algorithms and dynamic programming.
--	--

<p>primerov požrešnih algoritmov in dinamičnega programiranja.</p> <ul style="list-style-type: none"> • <i>Sortirni algoritmi</i>: pregled osnovnih metod, urejanje z vstavljanjem, urejanje z izbiranjem, urejanje z mehurčki, Shellovo urejanje, hitro urejanje, urejanje s kopico, urejanje z zlivanjem in radix. Analiza časovne zahtevnosti. 	<ul style="list-style-type: none"> • <i>Sorting algorithms</i>: review of the basic methods, insertion sort, selection sort, bubble sort, shellsort, quicksort, heapsort, mergesort, radix sort. Analysis of time complexity of sorting algorithms.
--	--

Temeljna literatura in viri/Readings:

Temeljna literatura/Basic literature

- Cormen, T. H. et al. (2009). Introduction to Algorithms. 3. Izdaja. Cambridge (Massachusetts), London, England: The MIT Press.
- Fajfar, I. (2020). Algoritmi in podatkovne strukture. Ljubljana: Fakulteta za elektrotehniko.
- Kononenko, I. in sod. (2008). Programiranje in algoritmi. Ljubljana: Založba FE in FRI.
- Kononnenko, I. in Robnik Šikonja, M. (2004). Osnove algoritmov in podatkovnih struktur II. Ljubljana: Založba FE in FRI.
- Sedgewick, R. (2003). Algorithms in Java, Parts 1-3. Addison Wesley.

Cilji in kompetence:

Učna enota prispeva predvsem k razvoju naslednjih splošnih in specifičnih kompetenc:

- usposobljenost za poglobljeno razumevanje računalništva in informatike,
- poznavanje in razumevanje procesov v tehniško-tehnološkem ter poslovnem okolju in sposobnost za njihovo analizo, sintezo in predvidevanje rešitev ter njihovih posledic,
- sposobnost definiranja, razumevanja in ustvarjalnega reševanja strokovnih izzivov na področjih računalništva in informatike,
- pridobiti temeljno in aplikativno usposobljenost na področju računalništva in informatike, ki obsega osnovna teoretska in praktična znanja, bistvene za sodobno računalništvo in informatiko,
- zmožnost opisati dano situacijo s pravilno uporabo matematičnih in računalniških simbolov ter zapisov,
- usposobljenost za analizo in razvoj strojne in programske opreme,
- poznavanje zmožnosti in omejitev informacijskih tehnologij,
- praktično znanje in veščine pri razvoju programske in strojne opreme ter informacijskih tehnologij, ki so potrebne

Objectives and competences:

The learning unit mainly contributes to the development of the following general and specific competences:

- competence for in-depth understanding of computer science and informatics,
- knowledge and understanding of processes in the technical-technological and business environment, as well as the ability for their analysis, synthesis and prediction of the solutions and their consequences,
- the ability to define, understand and creatively solve professional challenges in the fields of computer science and informatics,
- to acquire basic and applicative qualification in the field of computer science and informatics, which encompasses basic theoretical and practical knowledge essential for modern computer science and informatics,
- the ability to describe the given situation with a proper use of mathematical and computer symbols and records,

<p>za uspešno delo na strokovnem področju računalništva in informatike (programiranje, računalniška arhitektura, omrežja itd.).</p>	<ul style="list-style-type: none"> • being qualified for the analysis and development of hardware and software, • knowing the capabilities and limitations of information technologies, • practical knowledge and skills in the development of software and hardware and information technologies necessary for successful work in the field of computer science and informatics (programming, computer architecture, networks, etc.).
---	---

Predvideni študijski rezultati:

Intended learning outcomes:

<p>Študent/študentka:</p> <ul style="list-style-type: none"> • pozna različne principe reševanja problemov, • razume pomen abstraktne predstavitve problema, • razume pojem algoritma in pozna tehnike za njegovo predstavitev, • pozna in razume analizo algoritmov, • pozna in razume različne strategije in metode načrtovanja algoritmov, • pozna in razume različne abstraktne podatkovne tipe in učinkovitost operacij pri različnih implementacijah, • pozna in razume delovanje različnih sortirnih algoritmov, pozna njihovo časovno kompleksnost in jih zna uporabiti pri reševanju različnih problemov, • uporablja pridobljeno znanje načrtovanja algoritmov pri reševanju različnih problemov, • uporablja teorijo časovne zahtevnosti algoritmov za analizo praktičnih problemov, • demonstrira in reflektira uporabnost različnih podatkovnih struktur in algoritmov na praktičnih problemih, • spremlja in uporablja spletne vire ter knjižnice algoritmov in podatkovnih struktur, • aktivno uporablja sodobna razvojna okolja, • pisno in ustno poroča o analizi in reševanju praktičnih problemov. 	<p>Students:</p> <ul style="list-style-type: none"> • know various principles of problems solving, • understand the importance of an abstract presentation of a problem, • understand the concept of an algorithm and know the presentation techniques, • know and understand the analysis of algorithms, • know and understand different strategies and methods for algorithm design, • know and understand different abstract data types and effectiveness of operations in different implementations, • know and understand the operation of various sorting algorithms, know their time complexity and know how to use them to solve various problems, • use the acquired knowledge of algorithms design in solving various problems, • use the theory of time complexity of algorithms for the analysis of practical problems, • demonstrates and reflects the usability of various data structures and algorithms on practical problems, • monitor and use web resources and libraries of algorithms and data structures, • actively use modern development environments,
---	--

	<ul style="list-style-type: none"> report in written and spoken form on the analysis and solving of practical problems.
--	--

Metode poučevanja in učenja:

Learning and teaching methods:

<ul style="list-style-type: none"> <i>predavanja</i> z aktivno udeležbo študentov (razlaga, diskusija, vprašanja, primeri, reševanje problemov), <i>laboratorijske vaje</i>: v povezavi s predmetom (reševanje praktičnih problemov, uporaba programskih orodij), <i>seminarska naloga</i>, <i>samostojni študij</i>. 	<ul style="list-style-type: none"> <i>lectures</i> with active participation of students (explanation, discussion, questions, examples, problem solving), <i>laboratory work</i>: in connection with the course (solving practical problems, the use of programming tools), <i>seminar paper</i>, <i>independent study</i>.
---	---

Delež (v %)

Načini ocenjevanja:

Weight (in %)

Assessment:

<p>Načini:</p> <ul style="list-style-type: none"> izpit izdelava, predstavitev in zagovor seminarske naloge <p>Ocenjevalna lestvica: ECTS.</p>	<p>60 %</p> <p>40 %</p>	<p>Types:</p> <ul style="list-style-type: none"> exam preparation, presentation and defence of the seminar paper <p>Grading scheme: ECTS.</p>
--	-------------------------	---