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 / 1st Cycle	Computer and Information Science	2nd year	3rd

Vrsta predmeta/Course type	obvezni/obligatory

Univerzitetna koda predmeta/University course code	1_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:**
• Vpis v drugi letnik študijskega programa. • Študent mora pred izpitom pripraviti in predstaviti seminarsko nalogo.	• 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):**
• *Uvod:* osnovni principi reševanja problemov, pojem algoritma, zapis algoritma (psevdokoda, diagrami poteka, programski jezik), sled algoritma, osnovni podatkovni tipi, abstraktni podatkovni tipi. • *Analiza algoritmov:* časovna in prostorska zahtevnost algoritmov, velikostni red časovne zahtevnosti, modeliranje	• *Introduction:* basic principles of solving problems, the concept of algorithm, algorithm decryption, algorithm trace, pseudo code, basic data types, abstract data types. • *Analysis of algorithms:* time and space complexity of algorithms, order of magnitude of time complexity, modelling of the actual time of implementation, differences among complexity classes.

dejanskega časa izvajanja, razlike med razredi zahtevnosti.
- *Rekurzija:* 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.
- *Strategije reševanja problemov:* deli in vladaj (iskanje maksimuma, rekurzivni sortirni algoritmi, hanojski stolpi), dinamično programiranje (problem nahrbtnika, iskanje najkrajše poti, problem trgovskega potnika).
- *Podatkovne strukture:* kaj so podatkovne strukture, izbor strukture za določen problem, osnovne strukture (npr. polje) s primeri, pregled podatkovnih struktur.
- *Seznami in preslikave:* 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.
- *Drevo:* o drevesih, definicija abstraktnega podatkovnega tipa drevo, vrste obhodov drevesa, implementacije dreves, binarna drevesa.
- *Slovar:* 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.
- *Prioritetna vrsta:* definicija abstraktnega podatkovnega tipa vrsta, implementacija z delno urejenim levo poravnanim drevesom (kopico).
- *Graf:* 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

- *Recursion:* 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.
- *Strategies of solving problems:* divide and conquer (find the maximum, recursive sorting algorithms, towers of Hanoi), dynamic programming (knapsack problem, shortest path problem, traveling salesman problem).
- *Data structures:* what are data structures, selecting a structure for a particular problem, basic structures (e.g. array) with examples, an overview of data structures.
- *Lists and mappings:* 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.
- *Tree:* about trees, definition of an abstract data type tree, tree traversal types, tree implementations, binary trees.
- *Dictionary:* 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.
- *Priority queue:* definition of an abstract data type queue, implementation with a partially sorted left-aligned tree (heap).
- *Graph:* 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.

primerov požrešnih algoritmov in dinamičnega programiranja.	• *Sorting algorithms*: review of the basic methods, insertion sort, selection sort, bubble sort, shellsort, quicksort, heapsort, mergesort, radix sort. Analysis of time complexity of sorting algorithms.
• *Sortirni algoritmi*: 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.	

Temeljna literatura in viri/Readings:

Temeljna literatura/Basic literature
- Kononenko, I. in sod. (2008). Programiranje in algoritmi. Ljubljana: Založba FE in FRI.
- Cormen, T. H. et al. (2009). Introduction to Algorithms. Cambridge (Massachusetts), London: The MIT Press.

Priporočljiva literatura/Recommended literature
- Sedgewick, R. (2003). Algorithms in Java, Parts 1-3. Addison Wesley.

Cilji in kompetence:	**Objectives and competences:**
Učna enota prispeva predvsem k razvoju naslednjih splošnih in specifičnih kompetenc:	*The learning unit mainly contributes to the development of the following general and specific competences:*
• usposobljenost za poglobljeno razumevanje računalništva in informatike,	• competence for in-depth understanding of computer science and informatics,
• poznavanje in razumevanje procesov v tehniško-tehnološkem ter poslovnem okolju in sposobnost za njihovo analizo, sintezo in predvidevanje rešitev ter njihovih posledic,	• 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,
• sposobnost definiranja, razumevanja in ustvarjalnega reševanja strokovnih izzivov na področjih računalništva in informatike,	• the ability to define, understand and creatively solve professional challenges in the fields of computer science and informatics,
• pridobiti temeljno in aplikativno usposobljenost na področju računalništva in informatike, ki obsega osnovna teoretska in praktična znanja, bistvene za sodobno računalništvo in informatiko,	• 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,
• zmožnost opisati dano situacijo s pravilno uporabo matematičnih in računalniških simbolov ter zapisov,	• the ability to describe the given situation with a proper use of mathematical and computer symbols and records,
• usposobljenost za analizo in razvoj strojne in programske opreme,	• being qualified for the analysis and development of hardware and software,
• poznavanje zmožnosti in omejitev informacijskih tehnologij,	• knowing the capabilities and limitations of information technologies,
• praktično znanje in veščine pri razvoju programske in strojne opreme ter informacijskih tehnologij, ki so potrebne za uspešno delo na strokovnem področju računalništva in informatike	

(programiranje, računalniška arhitektura, omrežja itd.).	• 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:
Študent/študentka:	*Students:*
• pozna različne principe reševanja problemov,	• know various principles of problems solving,
• razume pomen abstraktne predstavitve problema,	• understand the importance of an abstract presentation of a problem,
• razume pojem algoritma in pozna tehnike za njegovo predstavitev,	• understand the concept of an algorithm and know the presentation techniques,
• pozna in razume analizo algoritmov,	• know and understand the analysis of algorithms,
• pozna in razume različne strategije in metode načrtovanja algoritmov,	• know and understand different strategies and methods for algorithm design,
• pozna in razume različne abstraktne podatkovne tipe in učinkovitost operacij pri različnih implementacijah,	• know and understand different abstract data types and effectiveness of operations in different implementations,
• pozna in razume delovanje različnih sortirnih algoritmov, pozna njihovo časovno kompleksnost in jih zna uporabiti pri reševanju različnih problemov,	• know and understand the operation of various sorting algorithms, know their time complexity and know how to use them to solve various problems,
• uporablja pridobljeno znanje načrtovanja algoritmov pri reševanju različnih problemov,	• use the acquired knowledge of algorithms design in solving various problems,
• uporablja teorijo časovne zahtevnosti algoritmov za analizo praktičnih problemov,	• use the theory of time complexity of algorithms for the analysis of practical problems,
• demonstrira in reflektira uporabnost različnih podatkovnih struktur in algoritmov na praktičnih problemih,	• demonstrates and reflects the usability of various data structures and algorithms on practical problems,
• spremlja in uporablja spletne vire ter knjižnice algoritmov in podatkovnih struktur,	• monitor and use web resources and libraries of algorithms and data structures,
• aktivno uporablja sodobna razvojna okolja,	• actively use modern development environments,
• pisno in ustno poroča o analizi in reševanju praktičnih problemov.	• report in written and spoken form on the analysis and solving of practical problems.

Metode poučevanja in učenja:	Learning and teaching methods:
• *predavanja* z aktivno udeležbo študentov (razlaga, diskusija, vprašanja, primeri, reševanje problemov), • *laboratorijske vaje:* v povezavi s predmetom (reševanje praktičnih problemov, uporaba programskih orodij), • *seminarska naloga,* • *samostojni študij.*	• *lectures* with active participation of students (explanation, discussion, questions, examples, problem solving), • *laboratory work:* in connection with the course (solving practical problems, the use of programming tools), • *seminar paper,* • *independent study.*

Načini ocenjevanja:	Delež (v %) Weight (in %)	Assessment:
Načini: • izpit • izdelava, predstavitev in zagovor seminarske naloge Ocenjevalna lestvica: ECTS.	60 % 40 %	Types: • exam • preparation, presentation and defence of the seminar paper Grading scheme: ECTS.