



## DALYKO APRAŠAS

Dalyko pavadinimas	Kodas
Duomenų struktūros ir algoritmai	

Dėstytojai	Padalinys
<b>Koordinuojantis:</b> dr. Margarita Beniušė <b>Kitas:</b>	Matematikos ir informatikos fakultetas Vilniaus universitetas

Studijų pakopa	Dalyko tipas
Pirmoji	Individualiųjų studijų dalykas

Įgyvendinimo forma	Vykdymo laikotarpis	Vykdymo kalbos
Auditorinė	2 semestras	Lietuvių / anglų

Reikalavimai studijuojančiajam
<b>Išankstiniai reikalavimai:</b> programavimo, IT ir matematikos pagrindai.

Dalyko apimtis kreditais	Visas studento darbo krūvis	Kontaktinio darbo valandos	Savarankiško darbo valandos
5	125	45	80

Dalyko tikslas: studijų programos ugdomos kompetencijos		
<p><b>Bendrosios kompetencijos:</b></p> <ul style="list-style-type: none"> <li>• žinias taikyti praktikoje (<i>BK1</i>).</li> </ul> <p><b>Dalykinės kompetencijos:</b></p> <ul style="list-style-type: none"> <li>• taikyti programų projektavimo bendruosius metodus, formuluoti ir analizuoti programinės įrangos reikalavimus (<i>DK1</i>),</li> <li>• analizuoti uždavinio algoritmo procesą pagal algoritmų bendrąsias savybes (<i>DK2</i>),</li> <li>• kurti koncepcinius bei fizinius duomenų modelius pagal informacijos valdymo bei duomenų modeliavimo principus (<i>DK9</i>).</li> </ul>		
Dalyko studijų siekiniai	Studijų metodai	Vertinimo metodai
Pritaikyti duomenų struktūrų ir algoritmų teoriją bei praktiką, algoritmų konstravimo bei projektavimo principus įvairiose profesinėse veiklose.	Teorinės ir praktinės paskaitos, pavyzdžių analizavimas, praktinės užduotys, diskusijos, susijusių programavimo aspektų analizė, dalykinės literatūros studijavimas	Programavimo projektai, egzaminas raštu (atvirojo ir uždarojo tipo klausimai)
Parinkti ir naudoti/įgyvendinti duomenų struktūras bei algoritmus įvairiuose taikymuose, apdorojant duomenų aibes pagal suformuluotus reikalavimus.		
Analizuoti bendrąsias pateikto algoritmo savybes ir įvertinti jo sudėtingumą asimptotiškai.		
Atlikti modifikacijas (eigos, tvarkos, duomenų specifikos ir kt. pakeitimus) įgyvendintame algoritme dėl pasikeitusių užduoties reikalavimų.		

Temos	Kontaktinio darbo valandos							Savarankiškų studijų laikas ir užduotys	
	Paskaitos	Konsultacijos	Seminarai	Pratybos	Laboratoriniai darbai	E. mokymas(is)	Visas kontaktinis darbas	Savarankiškas darbas	Užduotys
1. Įvadas į dalyką, duomenų dėstymas atmintyje, abstraktūs duomenų tipai, duomenų struktūros, baziniai duomenų tipai. Didžiojo O žymėjimas, algoritmo sudėtingumas.	1			2			3	4	Teorijos studijavimas. Praktinių užduočių sprendimas.
2. Vidinis duomenų rūšiavimas. Rūšiavimo algoritmai (bubble sort, selection sort, insertion sort, shell sort, quick sort, merge sort). Iteravimas, rekursija.	1			2			3	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
3. Abstraktūs duomenų tipai, tiesinės duomenų struktūros. Sąrašas (angl. list), eilutė (angl. queue), dėklas (angl. stack), aibė (set), jų programavimas masyvo ir rodyklės pagrindu. ADT dinaminė aibė.	1			2			3	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
4. Prioritetinė eilutė. Heap duomenų struktūra. Heap rūšiavimas. Binarinė paieška ir įterpimas.	1			2			3	4	Teorijos studijavimas. Praktinių užduočių sprendimas.
5. Dėstymo lentelės (angl. hash table) ir dėstymo algoritmai, maišos funkcijos. Išplėstinis dėstymas. Bucket rūšiavimas.	2			2			4	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
6. Hierarchinės struktūros. Dvejetainiai medžiai. Dvejetainiai paieškos medžiai. Medžių dėstymas atmintyje. Duomenų paieška, įterpimas, trynimasis, medžio apėjimas.	2			2			4	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
7. I programavimo projektas.				4			4	12	Praktinių užduočių sprendimas.
8. AVL-medžiai, spalv medžiai, 2-3-4 medžiai, raudoni-juodi medžiai.	2			4			6	8	Teorijos studijavimas. Praktinių užduočių sprendimas.
9. B-medžiai, B+ medžiai. Huffman'o algoritmas.	1			1			2	2	Teorijos studijavimas. Praktinių užduočių sprendimas.
10. Radix medžiai. Radix rūšiavimas. Skaitmeniniai paieškos medžiai. Patricia medžiai.	2			2			4	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
11. Grafai. Paieška į gylį, plotį. Trumpiausio kelio paieška.	1			1			2	4	Teorijos studijavimas. Praktinių užduočių sprendimas.
12. Paieška tekstuose.	1			2			3	4	Teorijos studijavimas. Praktinių užduočių sprendimas.

13. II programavimo projektas.			4		4	12	Praktinių užduočių sprendimas.
<b>Iš viso</b>	<b>15</b>		<b>30</b>		<b>45</b>	<b>80</b>	

Atsiskaitymai ir konsultacijos	Iš viso valandų
Praktinių užduočių gynimas	4
Egzaminas	2

Vertinimo forma	Svoris proc.	Atsiskaitymo laikas	Vertinimo kriterijai
Programavimo projektai (2 projektai)	50% (kiekvienas projektas po 25%)	7-8 savaitės (1 projektas), 15-16 savaitės (2 projektas)	Studentas turi programiškai realizuoti paskirtas užduotis. Pagal užduoties specifika turi parinkti duomenų saugojimui geriausiai tinkamas duomenų struktūras, realizuoti paskirtas duomenų struktūras, įgyvendinti rūšiavimo, paieškos, duomenų suliejimo, sukirtimo, grupavimo ir kitus algoritmus. Studentas turi pristatyti praktinę užduotį, pademonstruoti jos veikimą, paruošti pradinis duomenis programos testavimui, žodžiu paaiškinti, kaip veikia įgyvendinti algoritmai, bei kaip jie realizuoti, atsakyti į klausimus, atlikti dėstytojo nurodytus pakeitimus. Studentas turi pademonstruoti pagrindinių algoritmų supratimą. Vertinamas sugebėjimas paaiškinti, kaip yra suprogramuota užduotis, sugebėjimas argumentuoti, kodėl užduočiai parinktos tam tikros duomenų struktūros ir algoritmai, vertinami atsakymai į pateiktus klausimus, sprendimų korektiškumas bei optimalumas. Kiekvienas projektas vertinamas atskirai iki 2,5 balo.
Egzaminas (raštu)	50%	sesijos metu	Egzamine (raštu) pateikiami 15-20 klausimų iš visų dėstytų temų. Egzaminas vertinamas iki 5 balų.
			Galutinis dalyko vertinimas gaunamas sumuojant praktinių užduočių ir egzamino balus, apvalinant juos iki sveiko skaitmens. Dalyko galutiniam vertinimui privaloma, kad būtų atsiskaitytos visos praktinės užduotys ir laikytas teorijos egzaminas.

Autorius	Leidimo metai	Pavadinimas	Periodinio leidinio Nr. ar leidinio tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
<b>Privalomoji literatūra</b>				
T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein	2022	Introduction to Algorithms		MIT Press
M. T. Goodrich, R. Tamassia, M. H. Goldwasser	2014	Data Structures and Algorithms in Java		Wiley
T. H. Cormen C. E. Leiserson R. L. Rivest C. Stein	2009	Introduction to Algorithms		MIT Press
Simon Harris, James Ross	2006	Beginning Algorithms		Wiley
Robert Sedgwick	2000	Algorithms in C. Part 1-4		Addison-Wesley
Algimantas Juozapavičius	2007	Duomenų struktūros ir efektyvūs algoritmai		Vilnius: TEV
<b>Papildoma literatūra</b>				
Raimondas Čiegis	2007	Duomenų struktūros, algoritmai ir jų analizė.		Vilnius, VGTU
Kazys Baniulis, Bronius Tamulynas	2005	Duomenų struktūros		Kaunas, KTU

James Keogh, Ken Davidson	2004	Data Structures Demystified		McGraw-Hill/Osborne, ISBN 0-07-225359-2, USA
Nerijus Aukštakalnis ir kt.	2004	Praktinis vadovas duomenų struktūrų kurso studijoms		Kaunas, KTU,



## COURSE UNIT DESCRIPTION

Course unit title	Course unit code
<b>Data Structures and Algorithms</b>	

Lecturers	Department where the course unit is delivered
<b>Coordinator:</b> dr. Margarita Beniušė <b>Other lecturers:</b>	Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
First	Individualized studies

Mode of delivery	Semester or period when the course unit is delivered	Languages of instruction
Face-to-face	2nd semester	Lithuanian / English

Prerequisites
<b>Prerequisites:</b> basics of programming, IT, and mathematics.

Number of ECTS credits allocated	Student's workload	Contact hours	Individual work
5	125	45	80

Purpose of the course unit: programme competences to be developed		
<p><i>Generic competences to be developed</i></p> <ul style="list-style-type: none"> <li>• Ability to apply knowledge in practical situations (<i>BK1</i>)</li> </ul> <p><i>Subject-specific competences to be developed</i></p> <ul style="list-style-type: none"> <li>• Ability to apply general methods of the program design, state and analyse software requirements (<i>DK1</i>)</li> <li>• Ability to analyse the algorithmic process of the task based on the general properties of the algorithm (<i>DK2</i>)</li> <li>• Ability to build conceptual and physical data models based on information management and data modelling principles (<i>DK9</i>)</li> </ul>		
Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Ability to use the theory and practice of data structures and algorithms, notations and principles of algorithm construction and design in various professional activities	Theoretical and practical lectures, analysis of examples, practical exercises, discussions, analysis of related programming topics, reading of professional materials	Programming projects, examination in written (test with closed and open questions)
Ability to choose and use/implement data structures and algorithms in order to process data sets in various applications based on the formulated requirements		
Ability to analyze the general properties of the given algorithm and evaluate its complexity asymptotically.		
Ability to make modifications (changes in flow, order, data specifics, etc.) in the implemented algorithm due to the changes in requirements of the task		

Course content: breakdown of	Contact hours	Individual work: time and
------------------------------	---------------	---------------------------

the topics								assignments	
	Lectures	Consultations	Seminars	Practice classes	Laboratory work	E. learning	Contact hours	Individual work	Assignments
1. Introduction to subject, data allocation in memory, abstract data types, data structures, basic data types. Big-O notation, algorithm complexity.	1			2			3	4	Theory study. Practical tasks.
2. Internal sorting. Sorting algorithms (bubble sort, selection sort, insertion sort, shell sort, quick sort, merge sort). Iteration, recursion.	1			2			3	6	Theory study. Practical tasks.
3. Abstract data types. Linear data structures. List, queue, stack, set. Programming based on array and reference structures. Dynamic set ADT.	1			2			3	6	Theory study. Practical tasks.
4. Priority queue. Heap. Heap sort. Binary search and insertion.	1			2			3	4	Theory study. Practical tasks.
5. Hash tables, hash algorithms, hash functions. Extendable hashing. Bucket sort.	2			2			4	6	Theory study. Practical tasks.
6. Hierarchical structures. Binary tree. Binary search trees. Tree allocation in memory. Data search, insertion, deletion. Tree traversing.	2			2			4	6	Theory study. Practical tasks.
7. I programming project.				4			4	12	Practical tasks.
8. AVL tree, splay tree, 2-3-4 tree, red-black tree.	2			4			6	8	Theory study. Practical tasks.
9. B tree, B+ tree, Huffman algorithm.	1			1			2	2	Theory study. Practical tasks.
10. Radix tree, radix sort. Digital search tree. Patricia tree.	2			2			4	6	Theory study. Practical tasks.
11. Graph. Depth first search, breadth first search. Shortest path.	1			1			2	4	
12. Text search.	1			2			3	4	Theory study. Practical tasks.
13. II programming project.				4			4	12	Practical tasks.
<b>Total</b>	<b>15</b>			<b>30</b>			<b>45</b>	<b>80</b>	

Settlement	Total
Defense of practical tasks	4
Examination	2

Assessment strategy	Weight %	Deadline	Assessment criteria
Programming project (2 projects)	50% (25% each project)	7-8 weeks (1 <sup>st</sup> project), 15-16 weeks (2 <sup>nd</sup> project)	Student should implement assigned tasks, choose the data structure most appropriate for data storage, implement various data structures, sorting, search, merge, intersection, grouping and other algorithms. Student should present practical task program, demonstrate it's functionality, prepare data suitable for project testing, explain the workflow and implementation of algorithms, answer the questions and make additional changes according to the teacher requests. Student should demonstrate understanding of basic algorithms. The final mark depends on student's ability to explain how the project is programmed, ability to argue the choices of data structures and algorithms, the correctness and efficiency of project design, the ability to answer the questions. Each of two projects is assessed separately up to 2.5 point.
Examination (in written)	50%	session	Exam (in written form) consists of 15-20 questions, which cover all topics of the course. Exam is scored up to 5 points.
			The final evaluation of course is the sum of all points obtained from programming projects and final exam. They are rounded to integer numbers, if necessary. The presentations of all practical projects and exam are mandatory.

Author	Publishing year	Title	Number or volume	Publisher or URL
<b>Reading</b>				
T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein	2002	Introduction to Algorithms		MIT Press
M. T. Goodrich, R. Tamassia, M. H. Goldwasser	2014	Data Structures and Algorithms in Java		Wiley
T. H. Cormen C. E. Leiserson R. L. Rivest C. Stein	2009	Introduction to Algorithms		The MIT Press
Simon Harris, James Ross	2006	Beginning Algorithms		Wiley Publishing
Robert Sedgwick	2000	Algorithms in C. Part 1-4		Addison-Wesley, 2000
Algimantas Juozapavičius	2007	Data Structures and Efficient Algorithms		Vilnius: TEV
<b>Additional reading</b>				
Raimondas Čiegis	2007	Data Structures, Algorithms and Analysis		Vilnius, VGTU
Kazys Baniulis, Bronius Tamulynas	2005	Data Structures		Kaunas, KTU
James Keogh, Ken Davidson	2004	Data Structures Demystified		McGraw-Hill/Osborne, ISBN 0-07-225359-2, USA
Nerijus Aukštakalnis, et others	2004	Practical Guide for Data Structures		Kaunas, KTU,