



DALYKO APRAŠAS

Dalyko pavadinimas	Kodas
Duomenų struktūros ir algoritmai	

Dėstytojai	Padalinys
Koordinuojantis: Domas Meilūnas Kitas:	Matematikos ir informatikos fakultetas Vilniaus universitetas

Studijų pakopa	Dalyko tipas
Pirmoji	Privalomasis

Igyvendinimo forma	Vykdyto laikotarpis	Vykdyto kalbos
Auditorinė	2 semestras	Lietuvių / anglų

Reikalavimai studijuojančiajam
Išankstiniai reikalavimai: 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>Bendrosios kompetencijos:</p> <ul style="list-style-type: none"> žinias taikyti praktikoje (BK1). <p>Dalykinės kompetencijos:</p> <ul style="list-style-type: none"> taikyti programų projektavimo bendruosius metodus, formuluoti ir analizuoti programinės įrangos reikalavimus (DK1), analizuoti uždavinio algoritmo procesą pagal algoritmų bendrąsias savybes (DK2), kurti koncepcinius bei fizinius duomenų modelius pagal informacijos valdymo bei duomenų modeliavimo principus (DK9). 		
Dalyko studijų siekiniai	Studijų metodai	Vertinimo metodai
Pritaikyti duomenų struktūrų ir algoritmų teoriją bei praktiką, algoritmų konstravimo bei projektavimo žymėjimus bei principus įvairiose profesinėse veiklose.	Demonstravimai, diskusijos, susijusių programavimo aspektų analizė, praktinės užduotys, 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	Seminariai	Pratybos	Laboratoriniai darbai	E. mokymas(is)	Visas kontaktinis darbas	Savarankiškas darbas	Užduotys
1. Įvadas į dalyką, von Neumano principai, duomenų dėstymas atmintyje, abstraktūs duomenų tipai, duomenų struktūros, baziniai duomenų tipai.	1			2			3	4	Teorijos studijavimas. Praktinių užduočių sprendimas.
2. Abstraktūs duomenų tipai. Sąrašas (angl. list), eilutė (angl. queue), dėklas (angl. stack), jų programavimas masyvo ir rodyklės pagrindu. "Žiedinis" masyvas.	1			2			3	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
3. Vidinis ir išorinis 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.
4. Prioritetinė eilutė, „heap“ duomenų struktūra. Prioritetinės eilutės programavimas heap struktūros pagrindu. ADT dinaminė aibė.	1			2			3	4	Teorijos studijavimas. Praktinių užduočių sprendimas.
5. Dėstymo lentelės (angl. hash table) ir dėstymo algoritmai, maišos funkcijos. Išplėstinis dėstymas.	2			2			4	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
6. Binarinė paieška ir įterpimas. Hierarchinės struktūros, dvejetainiai paieškos medžiai, medžių dėstymas atmintyje. Duomenų paieška, įterpimas, trynimas, medžio mazgų apėjimas.	2			2			4	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
7. I programavimo projektas.				4			4	12	Praktinių užduočių sprendimas.
8. AVL-medžiai, 2-3-4 medžiai, raudoni-juodi medžiai.	2			4			6	8	Teorijos studijavimas. Praktinių užduočių sprendimas.
9. B-medžiai, Huffman'o algoritmas.	2			2			4	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
10. Skaitmeniniai medžiai, jų algoritmai, skaitmeninė paieška.	2			2			4	6	Teorijos studijavimas. Praktinių užduočių sprendimas.
11. Paieška tekstuose.	1			2			3	4	Teorijos studijavimas. Praktinių užduočių sprendimas.
12. II programavimo projektas.				4			4	12	Praktinių užduočių sprendimas.
Iš viso	15			30			45	80	

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% (1 Projektas – 15%, 2 projektas – 35%)	7-8 savaitės (1 projektas), 15-16 savaitės (2 projektas)	Studentas turi programiškai realizuoti paskirtas užduotis. Pagal užduoties specifiką turi parinkti duomenų saugojimui geriausiai tinkamas duomenų struktūras, realizuoti paskirtas duomenų struktūras, įgyvendinti rūšiavimo, paieškos, duomenų suliejimo ir kitus algoritmus. Studentas turi pristatyti ir apginti kompiuterines programas, paruošti pradinis duomenis programos testavimui, paaiškinti, kaip veikia įgyvendinti algoritmai, bei kaip jie realizuoti, atsakyti į klausimus, atlikti dėstytojo nurodytus pakeitimus. Studentas turi pademonstruoti pagrindinių algoritmų supratimą. Vertinamas atsakymų ir sprendimų korektiškumas bei optimalumas. Kiekvienas projektas vertinamas atskirai (galimi daliniai vertinimai, jei studentas apsigina tik dalį užduoties).
Egzaminas (raštu)	50%	birželio mėn.	Egzamine (raštu) pateikiami 10-15 klausimų, iš visų dalyko temų. Egzaminas vertinamas iki 5 balų.
			Galutinis dalyko vertinimas gaunamas sumuojant visus balus (praktinių užduočių ir egzamino) apvalinant juos iki sveiko skaitmens.
Eksterno tvarka			Dalyko studijuoti eksternu neleidžiama.

Autorius	Leidimo metai	Pavadinimas	Periodinio leidinio Nr. ar leidinio tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
Privalomoji literatūra				
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	Duomenų struktūros ir efektyvūs algoritmai		Vilnius: TEV
Papildoma literatūra				
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
Data Structures and Algorithms	

Lecturers	Department where the course unit is delivered
Coordinator: Domas Meilūnas Other lecturers:	Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
First	Individual

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

Prerequisites
Prerequisites: 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>Generic competences to be developed</p> <ul style="list-style-type: none"> Ability to apply knowledge in practical situations (<i>BK1</i>) <p>Subject-specific competences to be developed</p> <ul style="list-style-type: none"> Ability to apply general methods of the program design, state and analyse software requirements (<i>DK1</i>) Ability to analyse the algorithmic process of the task based on the general properties of the algorithm (<i>DK2</i>) Ability to build conceptual and physical data models based on information management and data modelling principles (<i>DK9</i>) 		
Learning outcomes of the course unit: students will be able to		
Ability to use the theory and practice of data structures and algorithms, notations and principles of algorithm construction and design in various professional activities	Demonstrations, discussions, analysis of related programming topics, practical exercises, reading of professional materials	Programming project, examination in written (test 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 the topics	Contact hours							Individual work: time and assignments	
	Lectures	Consultations	Seminars	Practice classes	Laboratory work	E. learning	Contact hours	Individual work	Assignments
1. Introduction to subject, von Neumann principles, data allocation in memory, abstract data types, data structures, basic data types.	1			2			3	4	Theory study. Practical tasks.
2. Abstract data types. List, queue, stack, programming of stack and queue based on array and pointer structures.	1			2			3	6	Theory study. Practical tasks.
3. Internal and external 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.
4. Priority queue, heap. Programming of priority queue based on heap structure. Dynamic sets ADT.	1			2			3	4	Theory study. Practical tasks.
5. Hash tables, hash algorithms, hash functions. Extendable hashing.	2			2			4	6	Theory study. Practical tasks.
6. Binary search and insertion. Hierarchical structures, 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 trees, 2-3-4 trees, red-black trees.	2			4			6	8	Theory study. Practical tasks.
9. B-trees, Huffman algorithm.	2			2			4	6	Theory study. Practical tasks.
10. Radix trees, radix algorithms, radix search.	2			2			4	6	Theory study. Practical tasks.
11. Text search.	1			2			3	4	Theory study. Practical tasks.
12. II programming project.				4			4	12	Practical tasks.
Total	15			30			45	80	

Settlement	Total
Defense of practical tasks	4
Examination	2

Assessment strategy	Weight %	Deadline	Assessment criteria
Programming project (2 projects)	50% (project 1 – 20% project 2 – 30%)	7-8 weeks (1 st project), 15-16 weeks (2 nd project)	Students should implement assigned tasks, choose the data structure most appropriate for data storage, implement various data structures, program sorting, search, merge, and other algorithms. Students should present and defend computer programs, prepare data suitable for project testing, explain the workflow and implementation of algorithms, answer the questions, and make additional changes according to the teacher's requests. Students should demonstrate an understanding of basic algorithms. The correctness and efficiency of the project design are evaluated. Each of the two projects is assessed separately (partial evaluations are possible if the student defends only part of the assigned task).
Examination (in written)	50%	June	The exam (in written form) consists of 10-15 questions, which cover all topics of the course. The exam is scored up to 5 points.
			The final evaluation of the course is the sum of all points obtained from both programming projects and the final exam. They are rounded to integer numbers, if necessary.
External order			The subject cannot be studied externally.

Author	Publis hing year	Title	Number or volume	Publisher or URL
Required reading				
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
Additional reading				
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,