



DALYKO APRAŠAS

Dalyko pavadinimas	Kodas
Grafų algoritmai	ITGTA

Dėstytojas	Padalinys
Koordinuojantis: dr. Rokas Astrauskas	Informatikos institutas Matematikos ir informatikos fakultetas Vilniaus universitetas

Studijų pakopa	Dalyko tipas
Pirmoji	Pasirenkamasis

Igyvendinimo forma	Vykdymo laikotarpis	Vykdymo kalba
Auditorinė	6 semestras	Lietuvių

Reikalavimai studijuojančiajam
Išankstiniai reikalavimai: programavimo pagrindai, bazinės duomenų struktūrų žinios.

Dalyko apimtis kreditais	Visas studento darbo krūvis	Kontaktinio darbo valandos	Savarankiško darbo valandos
5	130	64	66

Dalyko tikslas: studijų programos ugdomos kompetencijos		
Bendrosios kompetencijos: - Nuolatinis mokymasis (BK2), - Gebės savarankiškai įsisavinti naujas žinias, metodus ir įrankius bei taikyti juos praktikoje (BK2.3). Dalykinės kompetencijos: - Konceptualių pagrindų žinios ir gebėjimai (DK4). - Gebės taikyti matematikos pagrindų, mokslo, inžinerijos, kompiuterių mokslo teorines žinias ir algoritminius principus programų sistemų kūrime (DK4.2)/ - Gebės abstrakčiai mąstyti, naudoti formalius aprašymo metodus, įrodinėti jų teisingumą, formalizuoti ir specifiuoti realaus pasaulio problemas (DK4.3).		
Dalyko studijų siekiniai	Studijų metodai	Vertinimo metodai
Naudoti egzistuojančius teorinius modelius, tinkamą terminologiją, rekomenduojamus programavimo, modeliavimo principus ir įrankius įvairiose taikomosiose srityse ar kasdieninėje veikloje.	Įtraukiamosios paskaitos, užduočių sprendimas pratybų metu ir savarankiškai, įvairių šaltinių studijavimas, pratybų užduočių įgyvendinimas ir atsiskaitymas, konsultacijos.	Klausimai žodžiu ir raštu, pratybų užduočių įgyvendinimo ir atsiskaitymo vertinimas.
Apibrėžti taikomosios srities problemą ir taikyti egzistuojančius sprendimus.		
Pateikti algoritmą įvairiais būdais (pseudo-kodu, schema ar kt.) suformuluotam uždaviniui ar programai ir realizuoti kitų pateiktus algoritmus, formuluoti uždavinį įvairiuose abstrakcijos lygmenyse.		
Įvertinti pateikto algoritmo sudėtingumą, remiantis algoritmų vertinimo metodais, rekomenduoti efektyvų algoritmą užduočiai spręsti.		
Realizuoti, modifikuoti ir taikyti pagrindinius praktikoje naudojamus grafų teorijos algoritmus.		

Temos	Kontaktinio darbo valandos							Savarankiškų studijų laikas ir užduotys	
	P a s k a i t o s	K o n s u l t a c i j o s	S e m i n a r a i	P r a t y b o s	L a b o r a t o r i n i a i d a r b a i	P r a k t i k a	V i s a s k o n t a k t i n i s d a r b a s	S a v a r a n k i š k a s d a r b a s	Užduotys
1. Grafai ir jų reprezentacija duomenų struktūromis.	4				4		8	8	Pirmosios pratybų užduoties temos. Namų darbų užduotys, literatūros studijavimas, pratybų užduoties įgyvendinimas
2. Paieška grafuose: paieškos į gylį ir paieškos į plotį algoritmai. Taikymai: topologinis rūšiavimas, trumpiausio kelio radimas bei grafo jungumo patikrinimas.	6				6		12	12	
3. Trumpiausio kelio tarp dviejų viršūnių radimas: Dijkstra algoritmas. Heap duomenų struktūra, pirmumo eilutės.	4				4		8	8	Antrosios pratybų užduoties temos. Namų darbų užduotys, literatūros studijavimas, pratybų užduoties įgyvendinimas
4. Trumpiausias kelias kryptiniame becikliame grafe (DAGe). Taikymai: užduočių tvarkaraštis.	2				2		4	4	
5. Trumpiausio kelias grafe su neigiamais svoriais: Bellman – Ford algoritmas. Taikymai: valiutų arbitražas.	2				2		4	4	
6. Minimalaus dengiančiojo medžio radimas: Prim algoritmas, Kruskal algoritmas. Taikymai.	2				2		4	4	
7. Eulerio ciklo paieška, kiniškasis paštininko uždavinys: Algoritmai, taikymai.	4				4		8	8	Trečiosios pratybų užduoties temos. Namų darbų užduotys, literatūros studijavimas, pratybų užduoties įgyvendinimas
8. Hamiltono ciklo paieška ir keliaujančio pirklio uždavinys: algoritmai ir taikymai. Euristikos.	4				4		8	8	
9. Kiti NP-C grafų uždaviniai: grafų spalvinimas, viršūnių padengimas, klikos.	2				2		4	4	
10. Maksimalaus srauto apskaičiavimas duotame srauto tinkle: Ford–Fulkerson algoritmas.	2				2		4	6	Literatūros studijavimas
Iš viso	32				32		64	66	

Vertinimo strategija	Svoris proc.	Atsiskaitymo laikas	Vertinimo kriterijai
Praktinės programavimo užduotys	60	Po užduotį kovo, balandžio, gegužės mėnesiais	Per semestrą kiekvienas studentas privalo įgyvendinti tris pratybų užduotis, pateikdamas ir apgindamas sukurtas kompiuterines realizacijas. Reikalaujama, kad studento parašyta kompiuterinė realizacija ne tik duotų teisingus rezultatus, tačiau studentas sugebėtų paaiškinti jos veikimo principus, atsakyti į susijusius su užduotimi klausimus (apie sąvokas, algoritmus). Vertinimo kriterijai: algoritmo veikimo teisingumas, efektyvus programavimas, teisingos ir išsamios darbo išvados, atsakymų į klausimus teisingumas, nuoseklumas ir aiškumas.
Egzaminas raštu	40	Birželis	Raštu pateikiami klausimai apie sąvokas, algoritmus, jų taikymo pavyzdžius ir įgyvendinimo aspektus. Vertinimo kriterijai: teisingi ir išsamūs atsakymai į klausimus, aiškus atsakymų pateikimas.
Eksterno tvarka			Reikalavimai: reikia turėti surinkus bent 50% iš programavimo užduočių, kad būtų leidžiama laikyti eksternu. Perlaikyti bus galima tik egzaminą, surinkti balai pernešami.

Autorius	Leidimo metai	Pavadinimas	Periodinio leidinio Nr. ar leidinio tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
Privalomoji literatūra				
T. Cormen, C. Leiserson, R. Rivest, C. Stein	2009	Introduction to Algorithms (3rd Edition)		The MIT Press
Jean-Claude Fournier	2009	Graph Theory and Applications		John Wiley & Sons
Papildoma literatūra				
J. Gross, J. Yellen	1999	Graph theory and its applications		Chapman & Hall/CRC
Robert Sedgewick	2007	Algorithms in C. Part 5: Graph Algorithms (3rd Edition)		Addison-Wesley
R. Sedgewick, K. Wayne	2011	Algorithms (4th Edition)		Addison-Wesley



COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Graph Algorithms	ITGTA

Lecturer	Department where the course unit is delivered
Coordinator: dr Rokas Astrauskas	Institute of Computer Science Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
First	Optional

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	6th semester	Lithuanian

Prerequisites
Student should have basic programming skills and knowledge of data structures.

Number of ECTS credits allocated	Student's workload	Contact hours	Individual work
5	130	64	66

Purpose of the course unit: programme competences to be developed		
Generic competences: - Life-long learning (GC2). - An ability independently to acquire new knowledge, methodologies, and tools and to apply them in practice (GC2.3). Specific competences: - Knowledge and skills of underlying conceptual basis (SC4). - An ability to apply mathematical foundations, knowledge of science and engineering, computer science theory, and algorithmic principles in software systems development (SC4.2). - An ability to reason at an abstract level, to use formal notation, to prove correctness, and to apply formalization and specification for real-world problems (SC4.3).		
Learning outcomes of the course unit	Teaching and learning methods	Assessment methods
Ability to use existing theoretical models, terminology, recommended programming, modelling principles and tools in applied sciences and everyday life.	Retractable lectures, solution of tasks during laboratory work and solo, analysis of information from different resources, implementation of laboratory work tasks, consultations.	Oral and written questions, assessment of implementation of laboratory work tasks.
Ability to define a problem in application area and apply the existing solutions.		
Ability to present the algorithm using various techniques (pseudo-code, schema, etc.) for the given task or program; ability to implement others' algorithms; ability to formulate the task in different levels of abstraction.		
Ability to evaluate the complexity of the algorithm and to recommend the effective algorithm for the task to solve.		
Ability to implement, modify and apply common graph algorithms, related to applications.		

Course content: breakdown of the topics	Individual work: time and assignments							Assignments
	L e c t u r e s	T u t o r i a l s	S e m i n a r s	L a b o r a t o r y w o r k	I n t e r n s h i p / w o r k p l a c e m e n t	C o n t a c t h o u r s	I n d i v i d u a l w o r k	
1. Introduction to graph theory. Representations of graphs by data structures.	4				4	8	8	Topics for the first laboratory work task. Homework assignments, literature analysis, implementation of the laboratory work task
2. Graph searching: depth-first search and breadth-first search algorithms. Applications: topological sorting, finding the shortest path and graph connectivity test.	6				6	12	12	
3. Finding the shortest path between two vertices: Dijkstra's algorithm. Heap data structure, priority queues.	4				4	8	8	Topics for the second laboratory work task. Homework assignments, literature analysis, implementation of the laboratory work task
4. Shortest path in directed acyclic graph. Applications: job scheduling.	2				2	4	4	
5. Shortest path in graph with negative weights. Applications: currency arbitrage.	2				2	4	4	
6. Finding the minimum spanning tree: Prim's algorithm, Kruskal's algorithm. Applications.	2				2	4	4	Topics for the third laboratory work task. Homework assignments, literature analysis, implementation of the laboratory work task
7. Search for Eulerian cycle and Chinese postman problem: algorithms and applications.	4				4	8	8	
8. Hamiltonian cycle and travelling salesman problem: algorithms and applications. Heuristics.	4				4	8	8	
9. Other NP-C graph problems: graph coloring, vertex cover, cliques.	2				2	4	4	
10. Computing the maximum flow in a flow network: Ford–Fulkerson algorithm.	2				2	4	6	Literature analysis
Total	32				32	64	66	

Assessment strategy	Weight %	Deadline	Assessment criteria
Laboratory work task	60	One task for each month of March, April and May	During the semester, each student is required to implement three laboratory work tasks, by providing and defending developed computer programs. Good understanding of concepts and algorithms is required, error-free implementation is not enough. Assessment criteria: correctness of algorithm implementation, effective programming, correct and complete conclusions, correct, consistent and clear answer to questions.
Exam	40	June	Questions (in written form) about concepts, algorithms, examples of their applications and implementation aspects are asked. Assessment criteria: correctness and completeness of answers, clear presentation of thoughts.

Rules of external			Requirements: at least 50% of points from laboratory tasks must be scored to be allowed to take the external. Only the exam can be retaken, the scores from tasks will carry over.	
Author	Publishing year	Title	Issue No or volume	Publishing house or Internet site
Required reading				
T. Cormen, C. Leiserson, R. Rivest, C. Stein	2009	Introduction to Algorithms (3rd Edition)		The MIT Press
Jean-Claude Fournier	2009	Graph Theory and Applications		John Wiley & Sons
Optional reading				
J. Gross, J. Yellen	1999	Graph theory and its applications		Chapman & Hall/CRC
Robert Sedgewick	2007	Algorithms in C. Part 5: Graph Algorithms (3rd Edition)		Addison-Wesley
R. Sedgewick, K. Wayne	2011	Algorithms (4th Edition)		Addison-Wesley