Graphs and networks in computer science - course description

General information

Course name	Graphs and networks in computer science
Course ID	11.9-WE-INFD-GaNiCS-Er
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Computer Science
Education profile	academic
Level of studies	Second-cycle Erasmus programme
Beginning semester	winter term 2022/2023

Course information

Semester	1
ECTS credits to win	5
Course type	obligatory
Teaching language	english
Author of syllabus	• dr hab. inż. Piotr Borowiecki, prof. UZ • dr inż. Grzegorz Łabiak

Classes forms

The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	-	-	Credit with grade
Laboratory	30	2	-	-	Credit with grade

Aim of the course

- gaining basic skills and competences in the field of algorithmic graph theory.
- acquiring the ability to use graphs for modeling and solving real life problems.

Prerequisites

Basics of programming, Algorithms and data structures, Theoretical foundations of computer science.

Scope

Basic concepts of graph theory. Overview of application areas. Examples of important graph classes.

Selected graph frameworks (graph representations). Generating random graphs. Graph isomorphism. Graph and network datasets.

Graph searching algorihtms (breadth-first and depth-first searches, backtracking). Computing strongly connected components, topological sorting.

Minimum spanning trees (the algorithms of Prim and Kruskal).

Shortest paths in graphs (Dijkstra's, Bellman-Ford and Floyd-Warshall algorithms).

Algorithms for Euler tour/path and chinese postman problems.

Graph coloring - selected models, variants and algorithms for vertex and edge colorings.

Hamiltonian cycle/path and traveling salesperson problems (algorithms and applications).

Flow networks - determinig maximum flow (Ford–Fulkerson algorithm).

Graph problems in the context of Petri net theory - modeling parallel systems.

Teaching methods

Lectures: conventional lectures and discussions
Laboratories: computer laboratory exercises

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Student is able to repersent a system (if appropriate) in a graph form, and the parallel processes as a Petri net		• a quiz • carrying out laboratory reports	• Laboratory
Student has skills in the area of graph theory which can be applied to formulation and solving the tasks of system modelling and optimization		• a discussion • an evaluation test • an exam - oral, descriptive, test and other	• Lecture

Outcome description	Outcomesymbols	Methods of verification	The class form
Student has skills in the area of graph algorithms and is able to implement such algorithms in one of the universal programming languages		• an observation and evaluation of the student's practical skills • carrying out laboratory reports	• Laboratory
Student can represent the approproate tasks tn terms of the graph theory and apply the graph algorithms to solve them		• a discussion • a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory

Assignment conditions

Lectures – the passing condition is to obtain a positive grade from the final test.

Laboratories – the passing condition is to obtain a positive grade from all laboratory assignments.

Course - it is necessary to pass both lectures and laboratories.

Calculation of the final grade: lecture 50% + laboratory 50%

Recommended reading

1. Robin Wilson: Introduction to graph theory. Pearson Education Limited, 1996 (or other editions).
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms. MIT Press and McGraw-Hill, 1990 (or other editions).
3. Maciej M. Sysło, N. Deo, Janusz S. Kowalik: Discrete optimization Algorithms, Prentice-Hall, 1983.
4. Marek Kubale (Ed.), Graph Colorings. Contemporary Mathematics 352, American Mathematical Society, 2004

Further reading

1. Narsing Deo: Graph Theory with Application to Engineering and Computer Science, Prentice-Hall, Englewood Cliffs, N.J., 1974
2. Reinhard Diestel: Graph theory. Electronic edition, Springer Verlag New York, 2000.
3. Wolfgang Reisig: A Primer in Petri Net Design. Springer-Verlag, 1992.

Notes

Modified by dr hab. inż. Piotr Borowiecki, prof. UZ (last modification: 21-04-2022 01:11)