

# Graphs and networks in computer science - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Graphs and networks in computer science
Kod przedmiotu	11.9-WE-INFD-GaNiCS-Er
Wydział	Wydział Nauk Inżynieryjno-Technicznych
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus drugiego stopnia
Semestr rozpoczęcia	semestr zimowy 2022/2023

Informacje o przedmiocie	
Semestr	1
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
Sylabus opracował	<ul style="list-style-type: none"><li>dr hab. inż. Piotr Borowiecki, prof. UZ</li><li>dr inż. Grzegorz Łabiak</li></ul>

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	-	-	Zaliczenie na ocenę
Laboratorium	30	2	-	-	Zaliczenie na ocenę

## Cel przedmiotu

- 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.

## Wymagania wstępne

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

## Zakres tematyczny

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 algoirhtms (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.

## Metody kształcenia

**Lectures:** conventional lectures and discussions

**Laboratories:** computer laboratory exercises

## Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student can represent the appropoate tasks tn terms of the graph theory and apply the graph algorithms to solve them		<ul style="list-style-type: none"><li>diskusja</li><li>egzamin - ustny, opisowy, testowy i inne</li><li>sprawdzian</li></ul>	<ul style="list-style-type: none"><li>Wykład</li><li>Laboratorium</li></ul>

Opis efektu	Symbol efektywności	Metody weryfikacji	Forma zajęć
Student has skills in the area of graph algorithms and is able to implement such algorithms in one of the universal programming languages		<ul style="list-style-type: none"> <li>• obserwacje i ocena umiejętności praktycznych studenta</li> <li>• wykonanie sprawozdań laboratoryjnych</li> </ul>	<ul style="list-style-type: none"> <li>• Laboratorium</li> </ul>
Student has skills in the area of graph theory which can be applied to formulation and solving the tasks of system modelling and optimization		<ul style="list-style-type: none"> <li>• dyskusja</li> <li>• egzamin - ustny, opisowy, testowy i inne</li> <li>• kolokwium</li> </ul>	<ul style="list-style-type: none"> <li>• Wykład</li> </ul>
Student is able to represent a system (if appropriate) in a graph form, and the parallel processes as a Petri net		<ul style="list-style-type: none"> <li>• sprawdzian</li> <li>• wykonanie sprawozdań laboratoryjnych</li> </ul>	<ul style="list-style-type: none"> <li>• Laboratorium</li> </ul>

## Warunki zaliczenia

**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%

## Literatura podstawowa

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

## Literatura uzupełniająca

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.

## Uwagi

Zmodyfikowane przez dr hab. inż. Piotr Borowiecki, prof. UZ (ostatnia modyfikacja: 21-04-2022 01:11)

Wygenerowano automatycznie z systemu SylabUZ