

Graph Algorithms - course description

General information	
Course name	Graph Algorithms
Course ID	11.0-WK-IDP-AG-W-S14_pNadGenBY6PH
Faculty	Faculty of Mathematics, Computer Science and Econometrics
Field of study	Data Engineering
Education profile	academic
Level of studies	First-cycle studies leading to Engineer's degree
Beginning semester	winter term 2022/2023

Course information	
Semester	4
ECTS credits to win	6
Available in specialities	Design and Maintenance of Analytical Systems
Course type	optional
Teaching language	polish
Author of syllabus	- dr hab. Ewa Drgas-Burchardt, prof. UZ - dr Anna Fiedorowicz

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	-	-	Exam
Laboratory	30	2	-	-	Credit with grade

Aim of the course

Zapoznanie studentów z podstawowymi algorytmami grafowymi i sieciowymi, ze szczególnym uwzględnieniem możliwości wykorzystania tych algorytmów do rozwiązywania zagadnień praktycznych. Zajęcia na laboratorium będą polegały na analizie, badaniu poprawności i implementacji poznanych algorytmów oraz ich zastosowaniu w problemach praktycznych.

Prerequisites

Podstawy programowania. Wprowadzenie do teorii grafów.

Scope

Wykład:

1. Asymptotyka. Klasy złożoności obliczeniowej algorytmów.
2. Algorytmy przeszukiwania grafów i digrafów (BFS, DFS) i ich zastosowania.
3. Digrafy spójne i silnie spójne. Algorytm wyznaczania komponent silnie spójnych w digrafach.
4. Sortowanie topologiczne.
5. Zagadnienie najkrótszych ścieżek. Najkrótsze ścieżki w sieciach (sieci o nieujemnych wagach, bez cykli o ujemnej wadze). Algorytmy (Dijkstra, Bellman – Ford). Najkrótsze ścieżki pomiędzy wszystkimi parami wierzchołków. Algorytm Floyd-Warshall.
6. Minimalne drzewa rozpinające. Twierdzenia charakteryzujące. Algorytmy (Prim, Kruskal).
7. Drzewa rozpinające spełniające dodatkowe warunki (np. ograniczona średnica, ograniczony stopień).
8. Przepływy w sieciach.
9. Grafy Eulera i Hamiltona.
10. Problem komiwożacza (TSP), wybrane heurystyki.
11. Kolorowanie grafów. Wybrane algorytmy kolorowania.

Laboratorium:

1. Analiza działania i badanie poprawności wybranych algorytmów grafowych.
2. Rozwiązywanie podanych zagadnień (praktycznych) poprzez wybór (lub zaprojektowanie) odpowiedniego algorytmu i dobór odpowiedniej struktury danych.
3. Implementacja kilku wybranych algorytmów grafowych w języku programowania. Napisanie i przetestowanie odpowiednich programów oraz przygotowanie dokumentacji.

Teaching methods

Wykład konwersatoryjny. Ćwiczenia laboratoryjne w pracowni komputerowej.

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
---------------------	-----------------	-------------------------	----------------

Outcome description	Outcome symbols	Methods of verification	The class form
Student umie rozwiązać wybrane problemy grafowe metodami algorytmicznymi.	• K_U15	• egzamin kolokwium, bieżąca kontrola na zajęciach	• Lecture • Laboratory
Student umie przeanalizować zaprojektowane algorytmy pod kątem poprawności i złożoności obliczeniowej.	• K_U05	• kolokwium, bieżąca kontrola na zajęciach	• Laboratory
Student zna klasy złożoności algorytmów.	• K_W11	• egzamin	• Lecture
Student zna podstawowe algorytmy grafowe oraz sieciowe oraz rozumie ich zastosowanie do rozwiązywania zagadnień praktycznych.	• K_W02 • K_W04 • K_W10	• egzamin kolokwium, bieżąca kontrola na zajęciach	• Lecture • Laboratory
Student potrafi zaprojektować algorytmy rozwiązujące typowe problemy grafowe i sieciowe poprzez zaprojektowanie i dobór odpowiednich technik algorytmicznych i struktur danych.	• K_U04	• kolokwium, bieżąca kontrola na zajęciach	• Laboratory
Student potrafi zaimplementować wybrane algorytmy grafowe i sieciowe przy użyciu narzędzi programistycznych oraz przedstawić rozwiązanie w czytelnej formie.	• K_U06 • K_K02	• projekt, sprawozdanie	• Laboratory

Assignment conditions

Wykład: egzamin pisemny.

Laboratorium: na ocenę składają się wyniki osiągnięte na kolokwiach (z zadaniami o zróżnicowanym stopniu trudności) oraz ocena projektu polegającego na implementacji wybranego algorytmu (grafowego lub sieciowego) i przygotowaniu dokumentacji (w formie sprawozdania).

Na ocenę z przedmiotu składa się ocena z laboratorium (50%) oraz ocena z egzaminu pisemnego (50%). Warunkiem zaliczenia przedmiotu jest pozytywna ocena z laboratorium i egzaminu.

Recommended reading

1. W. Lipski, Kombinatoryka dla programistów, WNT, Warszawa, 2005.
2. K.A. Ross, Ch.R.B. Wright, Matematyka dyskretna, PWN, Warszawa, 1996.
3. D. Jungnickel, Graphs, Networks and Algorithms, Springer-Verlag, Berlin Heidelberg, 2008
4. E. M. Reingold, J. Deo, N. Nievergelt, *Algorytmy kombinatoryczne*, PWN, Warszawa, 1985.
5. M. Sysło, N. Deo, J. Kowalik, Algorytmy optymalizacji dyskretniej z programami w języku Pascal, PWN, Warszawa, 1993.
6. J. Błażewicz, Złożoność obliczeniowa problemów kombinatorycznych, WNT, Warszawa, 1988.

Further reading

1. M. Kubale (Redaktor): Optymalizacja dyskretna – modele i metody kolorowania grafów, WNT, 2002.
2. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Projektowanie i analiza algorytmów komputerowych, Helion, 2003.
3. B. Jankowski, Grafy – Algorytmy w Pascalu, MIKOM, 2003.
4. P. Wróblewski, Algorytmy, struktury danych i techniki programowania, Helion, 2003.

Notes

Modified by dr Alina Szelecka (last modification: 19-05-2022 21:46)

Generated automatically from SylabUZ computer system