



Algorytmy dla problemów trudnych obliczeniowo

Karta opisu przedmiotu

Informacje podstawowe

Kierunek studiów Informatyka (kierunek wspólny - WI)	Cykl dydaktyczny 2022/2023	
Specjalność -	Kod przedmiotu WIINF5.II20.02814.22	
Jednostka organizacyjna Wydział Informatyki	Języki wykładowe polski	
Poziom kształcenia Studia inżynierskie I stopnia	Obligatoryjność Obowiązkowy	
Forma studiów Stacjonarne	Blok zajęciowy Przedmioty ogólne	
Profil studiów Ogólnoakademicki	Przedmiot powiązany z badaniami naukowymi Tak	
Koordynator przedmiotu	Piotr Faliszewski	
Prowadzący zajęcia	Piotr Faliszewski, Marcin Kurdziel, Marek Gajęcki, Bartosz Kusek, Andrzej Kaczmarczyk	
Okres Semestr 6	Forma zaliczenia Zaliczenie	Liczba punktów ECTS 3
	Forma prowadzenia i godziny zajęć Wykład: 14 Ćwiczenia laboratoryjne: 14 Ćwiczenia projektowe: 14	

Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy - Student zna i rozumie:			
W1	Student zna i rozumie pojęcia aproksymacji i algorytmów losowych.	INF1A_W02	Aktywność na zajęciach

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
W2	Student zna i rozumie podstawowe pojęcia parametrycznej teorii złożoności obliczeniowej.	INF1A_W02	Aktywność na zajęciach
Umiejętności - Student potrafi:			
U1	Student potrafi stworzyć program komputerowy rozwiązujący problem dużej złożoności obliczeniowej.	INF1A_U05, INF1A_U07	Zaliczenie laboratorium
Kompetencji społecznych - Student jest gotów do:			
K1	Student potrafi zaplanować wdrożenie programu komputerowego w sytuacji niepewności co do optymalnego rozwiązania.	INF1A_K04	Zaliczenie laboratorium

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

W ramach przedmiotu studenci poznają metody rozwiązywania problemów NP-trudnych (algorytmy dokładne, algorytmy klasy FPT, algorytmy aproksymacyjne).

Nakład pracy studenta

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Wykład	14
Ćwiczenia laboratoryjne	14
Ćwiczenia projektowe	14
Samodzielne studiowanie tematyki zajęć	32
Egzamin lub kolokwium zaliczeniowe	2
Dodatkowe godziny kontaktowe	5
Łączny nakład pracy studenta	Liczba godzin 81
Liczba godzin kontaktowych	Liczba godzin 42

* godzina (lekcyjna) oznacza 45 minut

Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
1.	<p>1. Implementacja algorytmów dokładnych (np. dla problemu Vertex Cover; 2 godz.)</p> <p>2. Implementacja algorytmów aproksymacyjnych (np. dla problemu Vertex Cover; 2 godz.)</p> <p>3. Implementacja redukcji do SAT (np. dla Vertex Cover lub dla Graph-k-Coloring) oraz wykorzystanie solwera SAT (2 godz.)</p> <p>4. Implementacja redukcji do ILP (np. dla Vertex Cover lub dla Graph-k-Coloring) oraz wykorzystanie solwera ILP (2 godz.)</p> <p>5. Implementacja algorytmu FPT parametryzowanego szerokością drzewową (2 godz.)</p> <p>6. Zaproponowanie metody rozwiązania oraz implementacja programu rozwiązującego zadany problem obliczeniowy (4 godz.)</p> <p>Oceny wyznaczane są na podstawie aktywności na laboratorium oraz skuteczności i efektywności programu stworzonego w punkcie 6. powyżej.</p>	U1, K1	Ćwiczenia laboratoryjne, Ćwiczenia projektowe
2.	<p>1. Wykładnicze algorytmy dla problemów NP-zupełnych (4 godz.) Metody pełnego przeglądu przestrzeni rozwiązań. Optymalizacja metod pełnego przeglądu. Algorytmy randomizowane. Przykłady algorytmów dla problemu spełnialności formuł logicznych.</p> <p>2. Algorytmy aproksymacyjne dla problemów NP-zupełnych (4 godz.) Pojęcie algorytmu aproksymacyjnego. Przykłady algorytmów o stałym współczynniku aproksymacji oraz o współczynniku aproksymacji zależnym od rozmiaru problemu. Pojęcie w pełni wielomianowego schematu aproksymacji.</p> <p>3. Twierdzenie o probabilistycznie weryfikowanych dowodach (1 godz.) Twierdzenie PCP. Równoważność probabilistycznie weryfikowanych dowodów i klasy NP. Znaczenie twierdzenia PCP dla wyznaczania granic aproksymacji problemów NP-zupełnych.</p> <p>4. Parametryczna teoria złożoności obliczeniowej (4 godz.) Klasy parametrycznej złożoności obliczeniowej (FPT, hierarchia $W1, W2, \dots$). Metody konstrukcji algorytmów FPT (przebieg przestrzeni rozwiązań, programowanie całkowitoliczbowe, kodowanie kolorami, kernelizacja). Przykłady problemów $W1$ i $W2$ trudnych.</p> <p>5. Algorytmy randomizowane (1 godz.) Pojęcie algorytmu randomizowanego. Podstawy matematyczne. Przykłady algorytmów.</p>	W1, W2	Wykład

Informacje rozszerzone

Metody i techniki kształcenia :

Wykłady odbywają się w formie zdalnej, Mini wykład, Wykłady odbywają się w formie zdalnej

Rodzaj zajęć	Metody zaliczenia	Warunki zaliczenia przedmiotu
Wykład	Aktywność na zajęciach	
Ćwiczenia laboratoryjne	Zaliczenie laboratorium	
Ćwiczenia projektowe	Zaliczenie laboratorium	

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

Ćwiczenia laboratoryjne zaliczane są na podstawie udziału w zajęciach oraz samodzielnej implementacji programu rozwiązującego zadany problem. Zaliczenia poprawkowe: Realizacja programu rozwiązującego zadany problem.

Sposób obliczania oceny końcowej

Ocena końcowa, zaliczenia ćwiczeń laboratoryjnych w pierwszym terminie jest oceną z ćwiczeń laboratoryjnych, ale nie niższą niż 3.5. W przypadku zaliczenia ćwiczeń laboratoryjnych w terminie późniejszym niż pierwszy, ocena końcowa nie może być wyższa niż 3.0.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

Wyrównywanie zaległości spowodowanych nieobecnością na zajęciach: W przypadku nieobecności na zajęciach studenci mogą uczestniczyć w zajęciach dla innej grupy. Jeśli jest to niemożliwe, studenci są zobowiązani opanować materiał ćwiczeń we własnym zakresie.

Wymagania wstępne i dodatkowe

Ogólna znajomość matematyki z naciskiem na matematykę dyskretną. Podstawowa wiedza dotycząca: logiki, teorii grafów, metod konstruowania algorytmów. Znajomość materiału przedmiotu Teoria Obliczeń i Złożoności Obliczeniowej.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Wykład: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego. Ćwiczenia laboratoryjne: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych. Ćwiczenia projektowe: Studenci wykonują prace praktyczne mające na celu uzyskanie kompetencji zakładanych przez sylabus. Ocenie podlega sposób wykonania projektu oraz efekt końcowy.

Literatura

Obowiązkowa

1. Literatura podstawowa:
1. Sipser M.: Wprowadzenie do teorii obliczeń. WNT 2009
2. Vazirani V.: Algorytmy aproksymacyjne, WNT 2016
4. Literatura uzupełniająca:
3. Papadimitriou C.H.: Złożoność obliczeniowa. Helion 2012
4. Bovet D.P, Crescenzi P.: Introduction to the theory of complexity. Prentice Hall, 1994
5. Wegener I.: Complexity Theory, Springer, 2005
6. Michalewicz Z., Fogel D.: Jak to rozwiązać czyli nowoczesna heurystyka. WNT 2006
7. Motwani R., Raghavam P.: Randomized Algorithms, Cambridge University Press 1995

Badania i publikacje

Publikacje

1. 1. Piotr Skowron, Piotr Faliszewski, Arkadii M. Slinko: Achieving fully proportional representation: Approximability results. *Artif. Intell.* 222: 67-103 (2015)
2. 2. Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra: The complexity of manipulative attacks in nearly single-peaked electorates. *Artif. Intell.* 207: 69-99 (2014)
3. 3. Robert Bredereck, Jiehua Chen, Piotr Faliszewski, André Nichterlein, Rolf Niedermeier: Prices Matter for the Parameterized Complexity of Shift Bribery. *AAAI 2014*: 1398-1404

Kierunkowe efekty uczenia się

Kod	Treść
INF1A_K04	Potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania oraz adekwatnie zaplanować pracę.
INF1A_U05	Potrafi wykorzystać poznane metody i modele do tworzenia programów o charakterze użytkowym, a także potrafi adekwatnie wykorzystać znane algorytmy i struktury danych w budowie systemu komputerowego.
INF1A_U07	Potrafi ocenić, dobrać i stosować właściwe metody i narzędzia stosowane przy realizacji zadań związanych z budową systemu komputerowego, potrafi ocenić przydatność i korzystać z dostępnych bibliotek oraz komponentów oprogramowania.
INF1A_W02	Ma szczegółową wiedzę w zakresie podstaw algorytmiki, struktur danych oraz złożoności obliczeniowej, a także w zakresie podstaw teoretycznych budowy wybranych narzędzi i systemów informatycznych.