

Theoretical foundations of computer science - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Theoretical foundations of computer science
Kod przedmiotu	11.3-WE-INFP-TFCS-Er
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	WIEiA - oferta ERASMUS / Informatyka
Profil	-
Rodzaj studiów	Program Erasmus pierwszego stopnia
Semestr rozpoczęcia	semestr zimowy 2018/2019

Informacje o przedmiocie	
Semestr	3
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
Sylabus opracował	prof. dr hab. inż. Andrzej Obuchowicz

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	-	-	Egzamin
Ćwiczenia	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

- provide basic knowledge of concepts of algorithms, formal languages, deterministic and undeterministic finite automats, context-free grammars;
- give basic skills of algorithms complexity calculation, algorithms stop property and partialy-correctness proving
- provide basic knowledge of the Church-Turing thesis, concepts of P, EXP, NP, NPC and NP-hard problems,
- provide basic knowledge of concepts of the parallel algorithms theory.

Wymagania wstępne

Algorithms and data structures, foundation of discrete systems, mathematical analysis.

Zakres tematyczny

Principle informations: algorithm and its properties, asymptotic notations. Algorithm correctness.

Principles of automaton and languages theory: finite automats and regular terms, context-free gramars, pushdown automaton and context-free languages.

Primitive algorithmic models: Church-Turing thesis and its variants, Turing machine and its variants, random access memory machine, counter machine.

Algorithms effectiveness measures: space and time complexity, pessimistic and average complexity, bottom and upper limit of complexity, natural complexity, algorithmically open and closed problems.

Classification of algorithmic problems: logarithmic, polynomial, NP, NP-complex, NP-hard and exponential class; open problems connected with algorithmic problems classification; proving of NP complexity; undecidable problems.

Parallel and probabilistic algorithms, stable and expanding parallelism, product complexity. Nets: parallelism with stable connections. Thesis about parallel calculations, Nick class.

Probabilistic algorithms for chosen standard algorithmic problems. probabilistic complexity classes.

Metody kształcenia

Lecture, exercise classes.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Can analyze correctness of a sequential algorithm described by a block scheme or program written in a structural programming language.		• obserwacja i ocena aktywności na zajęciach • sprawdzian z pytaniami punktowymi	• Ćwiczenia

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Can estimate a time complexity of a given algorithm.		sprawdzian z progami punktowymi	Wykład
Is inventive in problems solving using accessible techniques and resources.		- obserwacja i ocena aktywności na zajęciach - sprawdzian z progami punktowymi	Ćwiczenia
Knowns concepts of parallel algorithms complexity.		sprawdzian z progami punktowymi	Wykład
Can define and characterize classes of algorithmic problems complexity. Knows the basic concepts of theories of algorithmic correctness, automaton, grammars, and languages.		sprawdzian z progami punktowymi	Wykład

Warunki zaliczenia

Lecture – the passing condition is to obtain positive marks from written or oral tests conducted at least once per semester.

Exercise classes – the passing condition is to obtain positive marks from all exercises and tests conducted during the semester.

Calculation of the final grade: lecture 50% + exercise classes 50%

Literatura podstawowa

1. Banachowski L., Kreczmar A.: Elementy analizy algorytmów, WNT, Warszawa, 1982
2. Bilski T., Chmiel K., Stokłosa J.: Zbiór zadań ze złożoności obliczeniowej algorytmów, Wydawnictwo Politechniki Poznańskiej, Poznań, 1992
3. Cormen T. H., Leiserson C. E., Rivest R. L.: Wprowadzenie do algorytmów, WNT, Warszawa, 1997
4. Harel D.: Rzecz o istocie informatyki, WNT, Warszawa, 2000
5. Hopcroft J. E., Ullmann J.D.: Wprowadzenie do teorii automatów, języków i obliczeń, PWN, Warszawa, 2003
6. Aho A.V., Ullmann J.D.: Wykłady z informatyki z przykładami w języku C, Helion, Gliwice, 2003

Literatura uzupełniająca

1. Ben Ari M.: Logika matematyczna w informatyce, WNT, Warszawa, 2005
2. Graham R.L., Knuth D.E., Patashnik O.: Matematyka konkretna, PWN, Warszawa, 2002
3. Papadimitriou C. H.: Złożoność obliczeniowa, Helion, Gliwice, 2012
4. Ross K. A., Wright C. R. B.: Matematyka dyskretna, PWN, Warszawa, 2000
5. Wróblewski P.: Algorytmy, struktury danych i języki programowania, Helion, Gliwice, 1997

Uwagi

Zmodyfikowane przez prof. dr hab. inż. Andrzej Obuchowicz (ostatnia modyfikacja: 27-03-2018 22:51)

Wygenerowano automatycznie z systemu SylabUZ