

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	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus pierwszego stopnia
Semestr rozpoczęcia	semestr zimowy 2019/2020

Informacje o przedmiocie	
Semestr	2
Liczba punktów ECTS do zdobycia	5
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

- familiarize students with the basic theoretical terms of computer science, like algorithm, formal languages, context-free grammar, deterministic and non-deterministic finite automata,
- learning basic skills in assigning of complexity class of algorithms, proof of algorithms halting problem and partial correctness
- familiarize students with Church-Turing thesis, easy and hard problems as well as NP, NP-complex and NP-hard problems,
- familiarize students with basic terms of parallel algorithms theory.

Wymagania wstępne

Algorithms and data structures, Computational logic, Foundations of discrete systems, Mathematical analysis.

Zakres tematyczny

Introduction: algorithm and its properties, asymptotic notation.

Algorithmic correctness: correct algorithm, partial correctness, semantic correctness, halting problem.

Foundations of automata theory and languages: finite automata and regular expressions, context-free grammars, automata with stack and context-free languages.

Simple algorithmic models:. Church-Turing thesis, Turing machine and its variants. Random access machine. Counting programs.

Algorithmic complexity: Time and space complexity of algorithms, pessimistic and average complexity. Top and bottom limit of complexity, natural complexity. Open and close algorithmic problems, algorithmic wound.

Algorithmic problems classification: easy and hard problems, logarithmic, polynomial, NP, NP-complex and expotential problems. Decidability and undecidability.

Parallel and probabilistic algorithms. Produkt complexity, nets. Thesis about parallel calculations. Nick class. RNC algoriithms..

Metody kształcenia

Wykład: wykład konwencjonalny/tradycyjny.

Ćwiczenia: ćwiczenia rachunkowe.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Potrafi oszacować złożoność czasową zadanego algorytmu.		sprawdzian z progami punktowymi	Wykład

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Jest kreatywny w rozwiązywaniu problemów uwzględniając dostępne techniki i środki.		• obserwacja i ocena aktywności na zajęciach • sprawdzian z progami punktowymi	• Ćwiczenia
Jest w stanie przeprowadzić analizę poprawności algorytmu sekwencyjnego zapisanego w postaci blokowej i w języku programowania strukturalnego.		• obserwacja i ocena aktywności na zajęciach • sprawdzian z progami punktowymi	• Ćwiczenia
Student zna pojęcie złożoności obliczeniowej dotyczące algorytmów równoległych.		• sprawdzian z progami punktowymi	• Wykład
Potrafi definiować i charakteryzować klasy złożoności problemów algorytmicznych. Zna podstawowe pojęcia teorii poprawności algorytmicznej, automatów, gramatyk i języków.		• sprawdzian z progami punktowymi	• Wykład

Warunki zaliczenia

Lecture - to pass written exam.

Exercises - to pass two written tests

Literatura podstawowa

1. Cormen T. H., Leiserson C. E., Rivest R. L.: Introduction to algorithms, MIT Press, Boston, 1994
2. Aho A. V., Hopcroft J. E., Ullman J.D.: Algorithms and Data Structures, Addison-Wesley Longman Publishing Co., Inc. Boston 1983
3. Dasgupta S., Papadimitriou Ch., Vazirani U.: Algorithms, McGraw-Hill, New York 2008
4. Harel D.: Algorithmics, the Spirit of Computing, Addison-Wesley Publ., Reading, 1987
5. Hopcroft J. E., Ullmann J.D.: Introduction to Automata Theory, Languages, and Computation, Addison-Wesley Publ., Reading, 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: 30-10-2019 13:53)

Wygenerowano automatycznie z systemu SylabUZ