

WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

KARTA OPISU PRZEDMIOTU

Wydział	Informatyki		
Kierunek	Informatyka		
Specjalność			
Semestr	I	Program studiów,	2019/2020
Stopień studiów	II	dla którego obowiązuje	
		syllabus	

Nazwa przedmiotu	Zaawansowane algorytmy i struktury danych			
Kod przedmiotu	ZAiSD			
Łączna liczba godzin	54	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	wykład + laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	7 (4+3)			

Prowadzący zajęcia	
Forma prowadzonych zajęć	Wykład
Wymiar zajęć	27 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Prowadzący zajęcia	
Forma prowadzonych zajęć	Laboratorium
Wymiar zajęć	27 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Wymagania wstępne	Umiejętność programowania w języku C++, znajomość podstawowych algorytmów i struktur danych.
Założenia i cele przedmiotu	Przegląd zaawansowanych technik algorytmicznych i analiza ich złożoności obliczeniowej – tradycyjna oraz zamortyzowana i oczekiwana. Przegląd zaawansowanych struktur danych i ich wykorzystanie w rozwiązywaniu problemów obliczeniowych. Wprowadzenie do różnych technik projektowania algorytmów.
Metody dydaktyczne	<ol style="list-style-type: none"> 1. Wykład – w formie tradycyjnej wspomagany prezentacją multimedialną. 2. Laboratorium – w trakcie zajęć studenci analizują i rozwiązują zadania obliczeniowe, programują wybrane algorytmy i struktury danych.

WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)		Odniesienie do efektów dla kierunku	Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji
WIEDZA – absolwent zna i rozumie:	W01. Zaawansowane algorytmy i struktury danych. W02. Różnice pomiędzy różnymi algorytmami.	K_W01 K_W06 K_W07 K_W09	P7S_WG
UMIEJĘTNOŚCI – absolwent potrafi:	U01. Pozyskiwać potrzebne informacje z literatury. U02. Wyciągać wnioski z przykładów rozwiązań prezentowanych na wykładzie oraz w fachowej literaturze. U03. Dyskutować na tematy poruszane na wykładzie. U04. Weryfikować i oceniać wyniki eksperymentów. U05. Rozwiązywać zadania z zakresu zaawansowanych problemów algorytmicznych. U06. Definiować problemy i podawać właściwy sposób ich rozwiązania.	K_U01 K_U02 K_U06 K_U08 K_U10 K_U11 K_U13	P7S_UW P7S_KO P7S_UK P7S_KK P7S_UO
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	K01. Podnoszenia umiejętności i pogłębiania wiedzy.	K_K03	P7S_UU

Treści programowe		
Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – wykład		
1	Efektywność algorytmów, szacowanie złożoności obliczeniowej: liczby Fibonacciego.	1
2	Analiza złożoności obliczeniowej prostych algorytmów iteracyjnych: sortowanie bąbelkowe, sortowanie przez wstawianie, sortowanie przez wybór.	1
3	Analiza złożoności obliczeniowej prostych algorytmów rekurencyjnych: wyszukiwanie binarne, szybkie potęgowanie, algorytm Euklidesa.	1
4	Analiza amortyzowana złożoności obliczeniowej: stos z operacją multi-pop (kolejka z operacją multi-dequeue), tablice dynamiczne.	2
5	Analiza probabilistyczna złożoności obliczeniowej: wybór k-tego elementu (algorytm Hoare’a), sortowanie przez podział (sortowanie szybkie).	1
6	Rozwiązywanie problemów obliczeniowych za pomocą techniki dziel i zwyciężaj: mnożenie długich liczb, mnożenie macierzy.	2
7	Rozwiązywanie problemów obliczeniowych za pomocą programowania dynamicznego: najdłuższy wspólny podciąg, najdłuższy podciąg rosnący.	2
8	Rozwiązywanie problemów obliczeniowych za pomocą strategii zachłannej: problemy	2

WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

	typu ASP i MPP, przydział zadań do procesora.	
9	Efektywna realizacja słownika na drzewie BST: drzewa AVL, drzewa czerwono-czarne.	2
10	Efektywna realizacja słownika na nośniku zewnętrznym (na dysku): B-drzewa, B+drzewa.	1
11	Efektywna realizacja kolejki priorytetowej: kopiec binarny umieszczony w tablicy.	1
12	Efektywna realizacja złączalnych kolejek priorytetowych: kopce dwumianowe, kopce lewicowe.	2
13	Efektywna realizacja zbiorów rozłącznych: reprezentacja listowa ze zbalansowanym łączeniem oraz reprezentacja drzewiasta ze zbalansowanym łączeniem i kompresją ścieżek.	3
14	Grafy: przeglądanie grafów za pomocą DFS i BFS, grafy eulerowskie, grafy hamiltonowskie, digrafy, grafy ważone. Podstawowe algorytmy grafowe: algorytm Kruskala (najkrótsze drzewo rozpinające), algorytm Dijkstry (najkrótsze ścieżki pomiędzy wierzchołkami).	3
15	Klasy problemów P i NP. Problemy NP-zupełne. Dowodzenie trudności problemów obliczeniowych.	3
Forma zajęć – laboratorium		
1	Implementacja różnych algorytmów obliczających liczby Fibonacciego.	1
2	Implementacja i analiza złożoności podstawowych algorytmów sortujących.	1
3	Implementacja i analiza złożoności algorytmu szybkiego potęgowania.	1
4	Implementacja i analiza złożoności algorytmu Euklidesa.	1
5	Implementacja i analiza złożoności sortowania szybkiego i algorytmu Hoare’a.	2
6	Implementacja algorytmu mnożenia długich liczb techniką dziel i zwyciężaj.	3
7	Implementacja algorytmu LCS przy pomocy programowania dynamicznego.	2
8	Implementacja algorytmu LIS przy pomocy programowania dynamicznego.	2
9	Implementacja algorytmu wyboru zajęć z wykorzystaniem strategii zachłannej.	2
10	Implementacja słownika w postaci zrównoważonego drzewa BST.	3
11	Implementacja kolejki priorytetowej w postaci kopca binarnego.	1
12	Implementacja złączalnej kolejki priorytetowej w postaci kopca lewicowego.	1
13	Implementacja zbiorów rozłącznych w postaci drzewiastej.	1
14	Implementacja grafu w postaci macierzy sąsiedztwa – badanie spójności grafu.	3
15	Implementacja grafu w postaci macierzy list sąsiadów – najkrótsza ścieżka. Zaliczenie.	3

WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

Forma i warunki zaliczenia przedmiotu	Egzamin pisemny z wykładu. Wykonanie zadań w ramach laboratorium.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Egzamin pisemny	W01-W02
	Ocena zadań wykonanych w ramach laboratorium	U01-U06, K01

Literatura podstawowa	<ol style="list-style-type: none"> 1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, <i>Wprowadzenie do algorytmów</i>, WNT, Warszawa 2012. 2. L. Banachowski, K. Diks, W. Rytter, <i>Algorytmy i struktury danych</i>, Helion, Warszawa 2011.
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. R. Sedgewick, <i>Algorytmy w C++</i>, RM, Warszawa 1999. 2. R. Sedgewick, <i>Algorytmy w C++. Grafy</i>, RM, Warszawa 2003. 3. S. Dasgupta, C. Papadimitriou, U. Vazirani, <i>Algorytmy</i>, PWN, Warszawa 2012. 4. P. Wróblewski, <i>Algorytmy, struktury danych i techniki programowania</i>, Helion, Gliwice 2010.

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	54
Przygotowanie się do zajęć	44
Studiowanie literatury	34
Udział w konsultacjach	10
Przygotowanie projektu / eseju / prezentacji itp.	-
Przygotowanie się do egzaminu / zaliczenia	33
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	175
Liczba punktów ECTS	7