



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Algorytmy i struktury danych NS		11.3.1530	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	niestacjonarne (zaoczne)
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Paweł Pączkowski			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		8 Przedmiot w wymiarze 30h wykładu i 30h ćw. aud. + praca własna studenta	
Wykład, Ćw. audytoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. audytoryjne: 30 godz., Wykład: 30 godz.			
Termin realizacji przedmiotu			
2022/2023 zimowy			
Status przedmiotu		Język wykładowy	
obowiązkowy		polski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
<ul style="list-style-type: none"> - Rozwiązywanie zadań - Wykład z prezentacją multimedialną - pisanie programów 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - egzamin pisemny z pytaniami (zadaniami) otwartymi - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - kolokwium 	
		Podstawowe kryteria oceny	
		Ocena z ćwiczeń - na podstawie wyników kolokwium i oceny napisanego programu Ocena z wykładu - wynik egzaminu pisemnego	
Sposób weryfikacji założonych efektów uczenia się			

zakładany efekt kształcenia	egzamin	kolokwium	program	sprawdzian	referat	raport	aktywność w dyskusji	obserwacja postawy studenta
Wiedza								
K_W03	X	X	X					
P_W1	X	X	X					
P_W2	X	X						
P_W3	X	X						
Umiejętności								
K_U02		X						
K_U04								X
K_U06			X					
P_U1		X						
P_U2		X						
P_U3			X					
Kompetencje								
K_K01								X
K_K03							X	X
P_K1							X	X

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Matematyka Dyskretna, Języki Programowania

B. Wymagania wstępne

Umiejętność programowania, znajomość aparatu matematycznego na poziomie wykładu Matematyka Dyskretna

Cele kształcenia

Zapoznanie studentów z klasycznymi algorytmami i strukturami danych używanymi do efektywnego rozwiązania typowych zadań programistycznych, uzasadnienie poprawności poznawanych algorytmów i przeprowadzenie analizy złożoności czasowej tych algorytmów

Treści programowe

- Pojęcia wstępne: poprawność semantyczna, złożoność czasowa pesymistyczna i oczekiwana, notacja asymptotyczna złożoności
- Sortowanie przez porównania. Algorytmy o złożoności kwadratowej, o złożoności liniowo-logarytmicznej (heapsort), o średniej złożoności liniowo-logarytmicznej (quicksort). Twierdzenia o ograniczeniach dolnych złożoności czasowej pesymistycznej i oczekiwanej.
- Sortowanie w czasie liniowym.
- Podstawowe struktury danych: listy, stosy, kolejki, kolejki priorytetowe. Implementacje przy użyciu tablic i struktur dwojganiowych.
- Struktury danych dla operacji słownikowych (wstaw, usuń, szukaj): tablice z haszowaniem, drzewa poszukiwań binarnych, drzewa zrównoważone, B-drzewa.
- Metody konstruowania efektywnych algorytmów: metoda "dziel i zwyciężaj", programowanie dynamiczne (najdłuższy wspólny podciąg), algorytmy zachłanne (kody Huffmana).
- Przykłady algorytmów grafowych (najkrótsze ścieżki w grafie).

Wykaz literatury

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein -- Wprowadzenie do algorytmów, Wydawnictwo Naukowe PWN 2012.
- L. Banachowski, K. Diks, W. Rytter -- Algorytmy i struktury danych, WNT 2011.

Kierunkowe efekty uczenia się

K_W03: ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i struktur danych, języków formalnych, teorii automatów i złożoności obliczeniowej
K_U02: potrafi projektować i analizować algorytmy pod kątem ich poprawności i złożoności obliczeniowej,

Wiedza

Student:

- ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie klasycznych algorytmów sortowania oraz realizujących operacje słownikowe (wstaw, usuń, szukaj) i ich złożoności czasowej
- zna przykłady algorytmów reprezentujących podstawowe metody tworzenia efektywnych algorytmów: dziel i zwyciężaj, programowanie dynamiczne,

<p>wykorzystując odpowiednie techniki algorytmiczne i struktury danych</p> <p>K_U04: potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</p> <p>K_U06: potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców</p> <p>K_K01: zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego uczenia się</p> <p>K_K03: potrafi i jest gotów formułować opinie na temat podstawowych zagadnień informatycznych</p>	<p>strategia zachłanna</p> <ul style="list-style-type: none"> • ma podstawową wiedzę na temat analizowania poprawności i złożoności algorytmów • zna klasyczne struktury danych: stosy, listy, kolejki, kopce, drzewa, tablice z haszowaniem <p>P_W1: zna klasyczne struktury danych (listy, stosy, drzewa, tablice z haszowaniem, drzewa zrównoważone) i operacje na nich (K_W03)</p> <p>P_W2: zna wybrane algorytmy sortowania (K_W03)</p> <p>P_W3: zna fakty dotyczące złożoności czasowej algorytmów sortowania, szukania, wstawiania, usuwania (K_W03)</p>
	<p>Umiejętności</p> <p>Student:</p> <ul style="list-style-type: none"> • potrafi wyjaśnić, posługując się przykładem, działanie wybranych klasycznych algorytmów • potrafi podać definicje wybranych klasycznych, powszechnie używanych struktur danych i zilustrować je przykładem (stosy, kolejki, kopce, drzewa, tablice z haszowaniem) • potrafi podać przykłady algorytmów o różnej czasowej złożoności obliczeniowej i potrafi ocenić złożoność czasową prostego algorytmu <p>Efekty przedmiotowe</p> <p>P_U1: potrafi zilustrować na przykładzie działanie algorytmów sortowania oraz operujących na różnych rodzajach drzew (K_U02, K_U04)</p> <p>P_U2: potrafi podać definicje klasycznych struktur danych (K_U02)</p> <p>P_U3: potrafi zaprogramować wybrane proste algorytmy przedstawione w postaci pseudokodu (K_U06)</p>
	<p>Kompetencje społeczne (postawy)</p> <p>Student potrafi operować pojęciami w zakresie algorytmów i struktur danych w sposób umożliwiający mu porozumienie się z innymi informatykami w tym zakresie</p> <p>Efekty przedmiotowe</p> <p>P_K1: potrafi formułować wypowiedzi na temat algorytmów i struktur danych i rozumie konieczność dalszego kształcenia się (K_K01, K_K03)</p>
<p>Kontakt</p> <p>Pawel.Paczkowski@inf.ug.edu.pl</p>	