



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 bioinformatyki		11.3.0344	
Nazwa jednostki prowadzącej przedmiot			
Instytut Fizyki Teoretycznej i Astrofizyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Fizyka	forma	stacjonarne
		moduł	Podstawowa
		specjalnościowy	Podstawowa
		specjalizacja	Podstawowa
Nazwisko osoby prowadzącej (osób prowadzących)			
prof. dr hab. Danuta Makowiec			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		6 Przedmiot w wymiarze 30h wykładu i 45h ćwiczeń w laboratorium komputerowym + praca własna	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Wykład: 30 godz., Ćw. laboratoryjne: 45 godz.			
Cykl dydaktyczny			
2016/2017 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"> - praca własna - przygotowanie się do zaliczenia - wykład - ćwiczenia laboratoryjne w pracowni komputerowej 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Zaliczenie (zal) 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - kolokwium - wykonanie pracy zaliczeniowej - wykonanie określonej pracy praktycznej 	
		Podstawowe kryteria oceny	
		Ocena końcowa z zajęć praktycznych powstaje jako średnia z ocen cząstkowych uzyskanych z kolejnych indywidualnych projektów. Projekty dotyczą kolejno różnych metod sortowania danych, a także wykorzystania abstrakcyjnych struktur danych do przechowywania danych. Zaliczenie wykładu: na podstawie obecności albo kolokwium.	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	Wykonanie ćwiczeń	Kolokwium	mtd. dydakt 3	mtd. dydakt 4	mtd. dydakt 5	mtd. dydakt 6	mtd. dydakt 7	mtd. dydakt 8
Wiedza								
K_W03	+	+						
K_W04	+	+						
K_W08	+	+						
Umiejętności								
K_U01	+	+						
K_U05	+	+						
K_U06	+	+						
K_U07	+	+						
K_U08	+	+						

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

zaliczenie z przedmiotu *Programowanie*

B. Wymagania wstępne

praktyczna znajomość języka C/C++

Cele kształcenia

- Zrozumienie kluczowych algorytmów stosowanych w bioinformatyce, ich możliwości i ograniczeń.
- Poznanie podstawowych strategii algorytmicznych służących projektowaniu nowych algorytmów.
- Umiejętność zbadania poprawności algorytmu i jego wydajności czasowej.
- Umiejętność implementowania projektów efektywnych w C++ oraz korzystania z bibliotek Pythona.

Treści programowe

Wykład + ćwiczenia audytoryjne

Wprowadzenie:

1. Podstawy teorii obliczeń: notacja asymptotyczna, złożoność czasowa;
2. Przegląd podstawowych technik algorytmicznych: rekurencja, algorytmy grafowe, programowanie dynamiczne, algorytmy zachłanne.

Problemy bioinformatyczne i zastosowane techniki algorytmiczne:

1. Mapowanie restrykcje DNA: przykład zastosowania algorytmu wyczerpującego;
2. Wyszukiwanie motywów regulacyjnych i konstrukcja łańcucha mediany: przykłady zastosowania algorytmów wyczerpujących - technika przeszukiwania z ograniczeniami;
3. Przetasowania w genomie: przykłady korzystania z algorytmów zachłannych, ich ograniczenia;
4. Dopasowanie sekwencji DNA: ocena podobieństwa pomiędzy sekwencjami DNA metodami programowania dynamicznego (graf edycji, najdłuższy wspólny podciąg);
5. Sekwencjonowanie DNA, macierze DNA: algorytmy grafowe typu cykl Eulera i cykl Hamiltona w problemie konstrukcji najkrótszego superłańcucha;
6. Kombinatoryczne dopasowanie wzorców: technika tablic haszujących, drzewa przedrostkowe w wyszukiwaniu powtarzających się wzorców;
7. Klasyfikacja danych i ewolucja: algorytm klastrowania hierarchicznego, klastrowanie metodą k-średnich.

Laboratorium

Realizowane projekty:

- A. Przeszukiwanie zbioru tekstowego. Operacja wejścia- wyjścia w C++, programowanie wieloplikowe, dobre praktyki programistyczne.
- B. Tekst jako tablica dwuwymiarowa: operacje porządkowania kolumnami i w wierszu, przy wykorzystaniu bibliotek STL w C++.
- C. Podstawowe algorytmy sortowania: iteracyjne (InsertionSort, SelectionSort) i rekurencyjny (MergeSort). Utrwalenie pojęcia zmiennej referencyjnej w C++, wykorzystanie makr.
- D. Abstrakcyjne typy danych: kopiec i kolejka priorytetowa. Projektowanie klasy w C++.
- E. Ocena złożoności obliczeniowej algorytmu – złożoność implementacji algorytmów sortujących InsertionSort, SelectionSort, MergeSort, HeapSort. Zagadnienia przeciążania funkcji, tworzenia szablonów w C++.
- F. Wprowadzenie do Pythona: podstawowe typy danych, operacja wejścia-wyjścia, wyszukiwanie informacji.
- G. Zastosowanie implementacji algorytmów InsertionSort, SelectionSort, MergeSort, HeapSort w Pythonie do zmiennych tekstowych, ocena wydajności.
- H. Problem częściowego trawienia: algorytmy kombinatoryczne w Pythonie.
- I. Abstrakcyjne typy danych: drzewo przeszukiwań binarnych. Konstrukcja klasy w Python.
- J. Dopasowanie sekwencji DNA, korzystanie z biblioteki biopython.org.

Wykaz literatury Neil C. Jones , Pavel A. Pevzner, An Introduction to Bioinformatics Algorithms, MIT Press 2004 T. H. Cormen, C. E. Leiserson, Wprowadzenie do algorytmów, WNT 1998.	
Efekty kształcenia (obszarowe i kierunkowe) K_W03 ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie programowania, algorytmów i złożoności, języków i paradygmatów programowania, baz danych, inżynierii oprogramowania K_W04 zna podstawowe konstrukcje programistyczne oraz pojęcia składni i semantyki języków programowania; zna podstawowe metody projektowania, analizowania i programowania algorytmów; zna podstawowe struktury danych i wykonywane na nich operacje K_W08 ma wiedzę zakresie podstawowych technik i narzędzi badawczych stosowanych w naukach ścisłych i przyrodniczych K_U01 potrafi zastosować wiedzę matematyczną i informatyczną do formułowania, analizowania i rozwiązywania prostych zadań związanych z bioinformatyką K_U05 potrafi projektować wykorzystując podstawowe techniki algorytmiczne i struktury danych, analizować, pisać uruchamiać i testować programy w wybranym środowisku programistycznym K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy; wykorzystuje podstawowe techniki algorytmiczne i struktur danych K_U07 posługuje się przyjętymi formatami reprezentacji różnego rodzaju danych stosownie do sytuacji, pamiętając o ich ograniczeniach K_U08 potrafi stosować podstawowe pakiety oprogramowania użytkowego do prezentacji wyników i analizy danych	Wiedza Student zna: podstawowe zasady konstrukcji wydajnych algorytmów podstawowe zagadnienia bioinformatyki, zna podstawowe techniki rozwiązywania ich
	Umiejętności Student potrafi: zaprojektować i zaimplementować w języku C/C++ efektywny algorytm rozwiązujący prosty problem kombinatoryczny umie zastosować algorytm do danych rzeczywistych i ocenić uzyskane wyniki wyjaśnić i uzasadnić użycie stosowanych technik programistycznych
	Kompetencje społeczne (postawy)
Kontakt fizdm@univ.gd.pl	