



KARTA OPISU MODUŁU KSZTAŁCENIA		
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 2/3
Specjalność Systemy informatyczne	Przedmiot oferowany w języku: polskim	Kurs (obligatoryjny/obieralny) obligatoryjny
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty / seminaria: -		Liczba punktów 5
Stopień studiów: I	Forma studiów (stacjonarna/niestacjonarna) stacjonarne	Obszar(y) kształcenia nauki techniczne
		Podział ECTS (liczba i %) 5 100%
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) kierunkowy		
Jednostka prowadząca przedmiot: Instytut Informatyki		
Osoba odpowiedzialna za przedmiot / wykładowca: Lista osób prowadzących zajęcia: dr hab. inż. Małgorzata Sterna, prof. PWSZ dr hab. inż. Małgorzata Sterna, prof. PWSZ e-mail: malgorzata.sterna@cs.put.poznan.pl e-mail: malgorzata.sterna@cs.put.poznan.pl tel. 61 424 2942 tel. 61 424 2942 Instytut Informatyki Instytut Informatyki ul. Ks. S. Wyszyńskiego 36, 62-200 Gniezno ul. Ks. S. Wyszyńskiego 36, 62-200 Gniezno		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Wiedza z zakresu matematyki dyskretnej
2	Umiejętności:	Umiejętność programowania w co najmniej jednym języku programowania
3	Kompetencje społeczne	Świadomość konieczności poszerzania swoich kompetencji
Cel przedmiotu: Zapoznanie studentów z teoretycznymi i praktycznymi problemami związanymi z algorytmami i strukturami danych, w szczególności wykształcenie umiejętności korzystania z różnych technik algorytmicznych i umiejętności analizy efektywności algorytmów, a także prezentacja podstawowych zagadnień z zakresu teorii złożoności obliczeniowej.		
Efekty kształcenia		
Wiedza. W wyniku przeprowadzonych zajęć student powinien/ będzie w stanie:		Odniesienie do Kierunkowych Efektów Kształcenia
01	Objąć bazy techniki algorytmiczne i podstawowe abstrakcyjne struktury danych. Scharakteryzować klasy złożoności problemów kombinatorycznych. Wy tłumaczyć pojęcie złożoności obliczeniowej algorytmu i wymienić rodzaje algorytmów.	K_W07++
02	Dobierać algorytmy do specyfiki problemu, w szczególności dobierać algorytmy sortowania do charakterystyki danych wejściowych, identyfikować podstawowe problemy z zakresu teorii grafów i wybierać algorytmy je rozwiązujące, wybierać metody dokładne i heurystyczne w zależności od klasy złożoności problemu.	K_W11+++
03	Formułować algorytmy, w szczególności z użyciem schematów blokowych i współczesnych języków programowania.	K_W06+++

Umiejętności. W wyniku przeprowadzonych zajęć student będzie potrafił:		Odniesienie do Kierunkowych Efektów Kształcenia
01	Zaimplementować algorytmy w wybranym języku programowania, przeprowadzić proste eksperymenty obliczeniowe, porównywać algorytmy z różnych punktów widzenia oraz opracowywać i analizować wyniki działania algorytmów.	K_U03+++ K_U17++
02	Wykorzystywać wiedzę z zakresu teorii złożoności obliczeniowej w procesie rozwiązywania problemu, w szczególności odróżniać i właściwie stosować algorytmy wielomianowe/wykładnicze, dokładne/przybliżone, szacować efektywność algorytmów.	K_U21++
03	Wyszukiwać dostępne w literaturze algorytmy i struktury oraz posłużyć się nimi w procesie rozwiązywania postawionego problemu.	K_U01+++ K_U06++
Kompetencje społeczne. W wyniku przeprowadzonych zajęć student zdobędzie następujące kompetencje:		Odniesienie do Kierunkowych Efektów Kształcenia
01	Dbłość o dobór właściwych metod algorytmicznych w rozwiązywaniu problemów praktycznych, gwarantujących efektywność pamięciową i czasową.	K_K04++
02	Świadomość nieustannego rozwoju technik algorytmicznych powodującego konieczność śledzenia literatury i dokształcania.	K_K01++

Sposoby sprawdzenia efektów kształcenia	
<u>Wykład</u>	
<ul style="list-style-type: none"> • sprawdzian pisemny weryfikujący umiejętność posługiwania się algorytmami i strukturami prezentowanymi podczas zajęć oraz znajomość podstawowych definicji, • ocenianie ciągle na każdych zajęciach (premiowanie aktywności i jakości percepcji). 	
<u>Laboratoria:</u>	
<ul style="list-style-type: none"> • ocenianie ciągle, na każdych zajęciach – w szczególności ocena przyrostu umiejętności posługiwania się poznanymi algorytmami i strukturami danych, • ocena poprawności realizacji ćwiczeń z wykorzystaniem dostarczonego lub samodzielnie wykonanego oprogramowania, • ocena sprawozdań z realizacji ćwiczeń, • ocena poprawności doboru algorytmów i struktur danych do specyfiki rozwiązywanego problemu, • ocena poprawności sposobu przeprowadzania eksperymentów obliczeniowych, • ocena umiejętności przeprowadzania analizy i wyciągania wniosków na podstawie wyników eksperymentów obliczeniowych, • ocena umiejętności konfrontowania wiedzy teoretycznej z wynikami eksperymentów, w szczególności uwzględniania wpływu czynników technologicznych, • ocena staranności sprawozdań i zrozumiałości formułowanego tekstu. 	

Treści programowe		
<p>Abstrakcyjne struktury danych i ich implementacje: tablice z haszowaniem, listy, drzewa, drzewa poszukiwań binarnych, stosy, kolejki, kolejki priorytetowe, słowniki. Algorytm selekcji, wyszukiwania liniowego i binarnego. Algorytmy sortowania. Grafy i ich reprezentacje maszynowe. Podstawowe algorytmy grafowe: przeglądanie grafu „w głąb”, „wszerz”, sortowanie topologiczne, wyznaczanie najkrótszych ścieżek w grafie, poszukiwanie minimalnego drzewa rozpinającego, wyznaczanie cyklu Eulera i Hamiltona. Przykłady algorytmów typu dziel i rządź, algorytmów zachłannych. Algorytm przeszukiwania z nawrotami, programowanie dynamiczne i heurystyki na przykładzie problemu plecakowego. Podstawy teorii złożoności obliczeniowej: problem nierozstrzygalny, przeszukiwania, decyzyjny, optymalizacyjny, algorytm, podstawowe modele obliczeń, funkcja złożoności obliczeniowej, klasy problemów NP, P, NP-zupełnych, silnie NP-zupełnych, ocena efektywności czasowej algorytmów i jakości rozwiązań.</p>		
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. Aho A.V., Hopcroft J.E., Ulman J.D., Algorytmy i struktury danych, Helion, Gliwice, 2003. 2. Dasgupta S., Papadimitriou Ch., Vazirani U., Algorytmy, PWN, Warszawa, 2010. 3. Lipski W., Kombinatoryka dla programistów, WNT, Warszawa, 2004. 4. Wirth N., Algorytmy + struktury danych = programy, WNT, Warszawa 2002. 		
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. Błażewicz J., Złożoność obliczeniowa problemów kombinatorycznych, WNT, Warszawa, 1988. 2. Cormen T.H., Leiserson Ch.E., Rivest R.L., Wprowadzenie do algorytmów, WNT, Warszawa, 1997. 3. Papadimitriou Ch., Złożoność obliczeniowa, WNT, Warszawa 2002. 4. Stephens R., Algorytmy i struktury danych, Helion, Gliwice, 2000. 5. Wróblewski P., Algorytmy, struktury danych i techniki programowania, Helion, Gliwice, 2010. 		
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	125 ¹⁾	5
Zajęcia wymagające indywidualnego kontaktu z nauczycielem	75 ²⁾	3
Zajęcia o charakterze praktycznym	50 ³⁾	2

1 pkt ECTS ≈ 25-30 h pracy studenta – do określenia poszczególnych składowych przyjęto:

- 1) łączne obciążenie studenta
- 2) - zajęcia dydaktyczne {w+c+L+p} + konsultacje + egzamin; dla stacjonarnych liczba godzin > 50 % godzin z poz1.
- 3) Zajęcia laboratoryjne+przygotowanie do tych zajęć+opracowanie sprawozdań+zajęcia projektowe+przygotowanie do zajęć projektowych+konsultacje w sprawie projektów+realizacja projektu.