

KARTA KURSU

Nazwa	Algorytmy i Struktury Danych
Nazwa w j. ang.	Algorithms and Data Structures

Koordynator	dr Roman Czapla	Zespół dydaktyczny
		dr inż. Magdalena Andrzejewska dr Roman Czapla dr Leszek Głowacki dr Zdobysław Świerczyński
Punktacja ECTS*	6	

Opis kursu (cele kształcenia)

Celem przedmiotu jest zapoznanie studentów z fundamentalnymi i zaawansowanymi algorytmami oraz strukturami danych. Uczestnicy zdobędą umiejętności analizy złożoności algorytmów, implementacji oraz praktycznego zastosowania różnych struktur danych do rozwiązywania problemów informatycznych.

Wykłady są prowadzone z wykorzystaniem pseudokodu, który służy do przedstawiania ogólnych algorytmów i metod. Natomiast na ćwiczeniach studenci realizują praktyczne implementacje algorytmów w wybranym języku programowania, w zależności od preferencji i wymagań kursu.

Warunki wstępne

Wiedza	Student zna podstawowe pojęcia matematyki dyskretnej.
Umiejętności	Student posiada podstawowe umiejętność programowania w wybranym języku proceduralnym.
Kursy	<u>Wymagane zaliczenie kursu:</u> Teoretyczne Podstawy Informatyki, Programowanie

Efekty uczenia się

	Efekt uczenia się	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student:	
	W01: rozumie pojęcie złożoności obliczeniowej i notacji asymptotycznej oraz zna metodę analizy algorytmów.	K_W02 K_W03 K_W04
	W02: zna klasyfikację i rodzaje algorytmów sortowania oraz algorytmów wyszukiwania – zna algorytmy sortowania przez porównywanie oraz zna algorytmy sortowania o złożoności liniowej.	K_W03 K_W04
	W03: rozumie podstawowe i złożone struktury danych.	K_W03 K_W04
	W04: zna podstawowe definicje dot. grafów oraz algorytmy grafowe i ich zastosowania.	K_W03 K_W04
	W05: zna podstawowe algorytmy tekstowe i wie do czego są wykorzystywane.	K_W03 K_W04
	W06: rozumie pojęcie problemów trudnych obliczeniowo.	K_W03 K_W04

	Efekt uczenia się	Odniesienie do efektów kierunkowych
Umiejętności	Po zakończeniu kursu student:	
	U01: potrafi analizować złożoność algorytmu – potrafi ocenić złożoność obliczeniową algorytmów, stosując notację asymptotyczną i analizować wydajność algorytmów w kontekście czasowym oraz pamięciowym.	K_U02 K_U05
	U02: potrafi implementować algorytmy sortowania i wyszukiwania (w tym wyznaczania statystyk pozycyjnych)	K_U02 K_U05
	U03: potrafi implementować algorytmy rekurencyjne oraz oparte na metodzie „dziel i zwyciężaj”.	K_U02 K_U05
	U04: potrafi zaimplementować i stosować struktury danych w różnych problemach algorytmicznych.	K_U02 K_U05
	U05: potrafi stosować algorytmy grafowe w praktyce.	K_U02 K_U05
	U06: potrafi stosować techniki programowania dynamicznego.	K_U02 K_U05
	U07: potrafi rozwiązywać problemy z wykorzystaniem metod z nawrotami i algorytmów zachłannych.	K_U02 K_U05
	U08: potrafi zaimplementować wybrane algorytmy tekstowe.	K_U02 K_U05

	Efekt uczenia się	Odniesienie do efektów kierunkowych
Kompetencje społeczne	Po zakończeniu kursu student:	
	K01: potrafi korzystać z różnych źródeł informacji (zasobów sieci Internet) do poszerzania własnej wiedzy i zdobywania nowych umiejętności.	K_K02
	K02: potrafi przekazać wiedzę informatyczną w sposób zrozumiały dla innych.	K_K01

Studia stacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	30					45					

Studia niestacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	20					30					

Opis metod prowadzenia zajęć

Podczas zajęć laboratoryjnych studenci rozwiązują zadania dot. algorytmiki, implementują wybrane algorytm i struktury oraz prezentują i omawiają rozwiązania zadań.

Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01					x			x				x	
W02					x			x				x	
W03					x			x				x	
W04					x			x				x	
W05					x			x				x	
W06					x			x				x	
U01					x			x				x	
U02					x			x				x	
U02					x			x				x	
U03					x			x				x	
U04					x			x				x	
U05					x			x				x	
U06					x			x				x	
U07					x			x				x	
U08					x			x				x	
K01								x					
K02								x					

Kryteria oceny	<p>Osiągnięcie efektów kształcenia podanych powyżej uprawnia studentów do uzyskania oceny nie wyższej niż dostateczna.</p> <p>Ocenę dobrą lub bardzo dobrą może uzyskać student, który wykaże się dobrą lub bardzo dobrą znajomością teoretycznych aspektów algorytmów i struktur danych oraz umiejętnością ich implementacji w wybranym języku programowania.</p>
----------------	--

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

<ol style="list-style-type: none"> Podstawy algorytmiki <ul style="list-style-type: none"> podstawowe definicje; rekurencja i elementy matematyki dyskretnej; praktyczne przykłady algorytmów rekurencyjnych; złożoność obliczeniowa, notacja asymptotyczna; praktyczna i teoretyczna analiza złożoności obliczeniowej algorytmów; Algorytm wyszukiwania i algorytmy sortowania przez porównywanie: <ul style="list-style-type: none"> proste algorytmy sortowania: bąbelkowe, przez wybieranie, przez wstawianie; metoda dziel i zwyciężaj oraz sortowanie przez scalanie, sortowanie szybkie, sortowanie przez kopcowanie i kopiec binarny; porównanie złożoności czasowej i wydajności poszczególnych metod sortowania. algorytm wyboru k-tego co do wielkości elementu; wyszukiwanie binarne, wyszukiwanie interpolacyjne i statystyki pozycyjne. Algorytmy sortowania o złożoności liniowej: <ul style="list-style-type: none"> sortowanie przez zliczanie,; sortowanie kubełkowe;

- sortowanie pozycyjne.
- 4. Podstawowe i złożone struktury danych:
 - tablica, listy, drzewa ukorzenione;
 - tablica z haszowaniem;
 - drzewa wyszukiwania binarnego, drzewa czerwono-czarne i AVL.
 - inne zrównoważone drzewa, np. B-drzewa i splay trees;
 - abstrakcyjne struktury danych i ich implementacja - stos, kolejka, tablica asocjacyjna.
 - drzewa przedziałowe – rodzaje, implementacja, zastosowania.
- 5. Grafy – rodzaje i reprezentacja:
 - omówienie typów reprezentacji grafów;
 - kluczowe algorytmy grafowe: BFS, DFS;
 - przykłady zastosowań: najkrótsze ścieżki bez wag, sortowanie topologiczne, silnie spójne składowe, mosty.
- 6. Najkrótsze ścieżki oraz minimalne drzewa rozpinające:
 - algorytmy Dijkstry, Bellmana-Forda;
 - algorytm Floyda-Warshalla;
 - algorytmy Kruskala i Prima.
- 7. Metoda z nawrotami i algorytmy zachłanne
 - omówienie metody nawrotów.
 - wybrane problemy: problem n hetmanów, kody Huffmana, problem wyboru zajęć, problem pokrycia zbioru, problem plecakowy (wersja zachłanna), problem wydawania reszty (wersja zachłanna).
- 8. Programowanie dynamiczne:
 - podstawy techniki programowania dynamicznego i omówienie wybranych problemów, takich jak ciąg Fibonacciego, najdłuższy rosnący podciąg, problem plecakowy (wersja dynamiczna), zbiór wierzchołków niezależnych w drzewie, problem wydawania reszty (wersja dynamiczna).
- 9. Wprowadzenie do algorytmów tekstowych
 - problem wyszukiwania wzorca w tekście (algorytm naiwny);
 - algorytm Rabina-Karpa;
 - algorytm Knutha-Morrisa-Pratta (KMP);
 - inne techniki przetwarzania tekstu;
 - drzewa i tablice sufiksów.
- 10. Problemy trudne obliczeniowo
 - omówienie klas P i NP;
 - problemy NP-zupełne i NP-trudne wraz z przykładami;
 - algorytmy aproksymacyjne.

Wykaz literatury podstawowej

Wskazane przez prowadzącego rozdziały:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Wprowadzenie do algorytmów*, Wydawnictwo Naukowe PWN, Warszawa 2023;
2. J. Kubica, *Struktury danych z przymrużeniem oka. Zabawna przygoda z przykładami pachnącymi kawą*, Helion, Gliwice 2024;
3. G. Heineman, *Nauka algorytmów. Poradnik pisania lepszego kodu*, Helion, Gliwice 2022.
4. P. Wróblewski, *Algorytmy, struktury danych i techniki programowania. Wydanie VI*, Helion, Gliwice 2019.
5. L. Banachowski, W. Rytter, K. M. Diks, *Algorytmy i struktury danych*, Wydawnictwo Naukowe PWN, Warszawa 2018;
6. S. Dasgupta, C. Papadimitriou, U. Vazirani, *Algorytmy*, Wydawnictwo Naukowe PWN, Warszawa 2010;
7. P. Kotowski, *Algorytmy + Struktury Danych = Abstrakcyjne Typy Danych*, Wydawnictwo BTC, Warszawa 2006.
8. D. Knuth, *Sztuka programowania, tom 1 i 3*, WNT, 2002

Wykaz literatury uzupełniającej

1. A. Bhargava, *Algorytmy. Ilustrowany przewodnik*, Helion, Gliwice 2017.
2. T. H. Cormen, *Algorytmy bez tajemnic*, Helion, Gliwice 2013.
3. K. Pieńkosz, J. Wojciechowski, *Grafy i sieci*, Wydawnictwo Naukowe PWN, Warszawa 2013;

4. Z.J. Czech, S. Deorowicz, P. Fabian, *Algorytmy i Struktury Danych. Wybrane zagadnienia*, Wydawnictwo Politechniki Śląskiej, Gliwice 2010;
5. A. Debudaj-Grabysz, S. Deorowicz, J. Widuch, *Algorytmy i Struktury Danych. Wybór zaawansowanych metod*, Wydawnictwo Politechniki Śląskiej, Gliwice 2012;
6. Drozdek A., *C++. Algorytmy i struktury danych*. Helion, Gliwice 2004.

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) - **studia stacjonarne**

Liczba godzin w kontakcie z prowadzącymi	Wykład	30
	Konwersatorium (ćwiczenia, laboratorium itd.)	45
	Pozostałe godziny kontaktu studenta z prowadzącym	5
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	30
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	0
	Przygotowanie do egzaminu	40
Ogółem bilans czasu pracy		150
Liczba punktów ECTS w zależności od przyjętego przelicznika		6

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) - **studia niestacjonarne**

Liczba godzin w kontakcie z prowadzącymi	Wykład	20
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	5
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	45
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	0
	Przygotowanie do egzaminu	50
Ogółem bilans czasu pracy		150
Liczba punktów ECTS w zależności od przyjętego przelicznika		6