

<b>Nazwa zajęć:</b> Algorytmy i złożoność		Algorithms and complexity	
<b>Kierunek:</b> Informatyka			<b>Obowiązuje od roku ak.</b> 2019/2020
<b>Poziom:</b> I st. inżynierski	<b>Profil:</b> praktyczny	<b>Grupa zajęć:</b> Kierunkowe	
<b>Semestr:</b> II	<b>Forma zaliczenia:</b> Z - zaliczenie na ocenę	<b>Punkty ECTS:</b> 2	<b>Zajęcia do wyboru:</b> Nie
			<b>Język zajęć:</b> polski

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych:

<b>Wykład</b> 15 / 8	<b>Ćwiczenia</b> 15 / 8	<b>Suma godzin:</b> 30 / 16
-------------------------	----------------------------	--------------------------------

**Specjalność:**

**Nazwiska osób odpowiedzialnych za zajęcia:**

prof. dr hab. Andrzej Giryń, dr Andrzej Pankowski

**Opis zajęć:**

Omówienie podstawowych struktur algorytmicznych. Implementacjach algorytmów. Poznanie zasad budowania i analizy algorytmów

**Cele dydaktyczne:**

Zapoznanie studentów z podstawowymi zagadnieniami algorytmiki. Metody projektowania algorytmów oraz ich weryfikacji i oceny.

Specyfikacje algorytmów oraz ich klasy złożoności.

Wdrożenie umiejętności opracowywania analizowania oraz implementacji algorytmów.

Uzyskanie kompetencji w określaniu sposobów postępowania w trakcie wykonywania analizy i konstrukcji prostych algorytmów obliczeniowych.

**Metody dydaktyczne:**

MP1 wykład informacyjny  
MP2 objaśnienie  
MC1 ćwiczenie praktyczne  
MC2 projekt  
MS1 wykład problemowy  
MS2 metoda symulacyjna

**Metody oceniania:**

MO1 praca projektowa  
MO2 sprawozdanie z ćwiczeń

**Wykład**

W1 Wprowadzenie do teorii algorytmów.

W2 Podstawowe struktury algorytmiczne. Specyfikacja algorytmów.

W3 Przegląd metod i technik projektowania algorytmów.

W4 Elementy analizy algorytmów.

W5 Algorytmy iteracyjne i rekurencyjne.

W6 Strategia "dziel i zwyciężaj".

W7 Elementarne algorytmy sortowania.

W8 Zaawansowane algorytmy sortowania.

**Ćwiczenia**

C1 Wprowadzenie do algorytmów, budowanie sieci działań.

C2 Schemat blokowy, budowanie algorytmów.

C3 Implementacja wybranych schematów blokowych.

C4 Analiza schematów blokowych.

C5 Budowanie złożonych schematów blokowych (iteracyjnych, rekurencyjnych).

C6 Strategia "dziel i zwyciężaj".

C7 Algorytmy sortujące: przez wybór, bąbelkowe, przez wstawianie.

C8 Algorytmy sortujące: QuickSort, MergeSort.

**Literatura podstawowa**

1 Cormen T., Lieserson Ch., Rivest R.: Wprowadzenie do algorytmów, WNT, 2000.

2 Harris S.: Algorytmy od podstaw. Wiley 2006, Helion 2006.

3 Sedgewick R.: Algorytmy w C++, RM, 1999.

4 Kubale M.: Łagodne wprowadzenie do analizy algorytmów. Wyd. Pol. Gdańskiej, 1999.

5 Wróblewski P.: Algorytmy. Struktury danych i techniki programowania. Helion 2010.

**Literatura uzupełniająca**

1 Banachowski L., Kreczmar A., Rytter W., Analiza algorytmów i struktur danych, WNT, 1987.

2 Knuth D.: Sztuka programowania, T1, T3. Addison Wesley 1997, WNT 2002.

3 Lipski W.: Kombinatoryka dla programistów. WNT 2004.

**Źródła dodatkowe**

1 Materiały na platformie.

**Warunki zaliczenia**

Wykonanie zadań praktycznych poprzez realizację schematów blokowych, algorytmów omawianych na wykładzie i ćwiczeniach. Wykonanie sprawozdania i analizy wybranych algorytmów. Implementacja wybranych algorytmów.

**Przykłady pytań zaliczeniowych**

Wymienić podstawowe cechy algorytmu i opisać je

Przedstawić graficznie przykład wybranego algorytmu obliczeniowego

**Obciążenie pracą studenta**

Studia stacjonarne/niestacjonarne

Forma pracy studenta	Wykład		Ćwiczenia		Suma	
Zajęcia z bezpośrednim udziałem nauczyciela	15 g	8 g	15 g	8 g	30 g	16 g
Zapoznanie się z literaturą przedmiotu	1 g	2 g			1 g	2 g
Przygotowanie się do zajęć	1 g	2 g	2 g	4 g	3 g	6 g
Przygotowanie się do kolokwium						
Realizacja zadanych ćwiczeń i zadań	2 g		10 g	20 g	12 g	20 g

Przygotowanie sprawozdania z ćwiczeń			4 g	6 g			4 g	6 g
Przygotowanie projektu / pracy								
Przygotowanie się i udział w egzaminie								
	19 g	12 g	31 g	38 g			50 g	50 g

<b>Efekty uczenia się</b>	<b>KEK</b>	<b>Treści kształcenia</b>	<b>Metody dydaktyczne</b>	<b>M. oceniania</b>
Student zna podstawowe metody budowania algorytmów i sposoby ich przedstawiania.	K_W01	W1, W2, W3 C1, C2, C3	MP1, MP2, MS1, MC1	MO1, MO2
Student objaśnia algorytmy i ich implementację.	K_W05	W5, W6, W7, W8 C3, C5, C6, C7, C8	MP1, Mp2, MS 2	MO1, MO2
Student potrafi rozwiązać postawione zadanie algorytmiczne, implementując (wybrany) algorytm.	K_U05	W3, W5 C3, C5	MC1, MC2	MO1, MO2
Student potrafi dokonać analizy złożoności obliczeniowej oraz poprawności sematycznej wybranych algorytmów.	K_U06	W2, W4 C3, C4, C5	MC1, MC2, MS1, MS2	MO1, MO2
Student rozumie potrzebę ciągłego uzupełniania wiedzy.	K_K01	W1 C1	MP1, MS1	MO1, MO2