

**SYLABUS****DOTYCZY CYKLU KSZTAŁCENIA 2024-2028***(skrajne daty)*

Rok akademicki 2024/2025

**1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE**

|                                                       |                                                                                             |
|-------------------------------------------------------|---------------------------------------------------------------------------------------------|
| Nazwa przedmiotu                                      | algorytmy i struktury danych                                                                |
| Kod przedmiotu*                                       |                                                                                             |
| Nazwa jednostki prowadzącej kierunek                  | Instytut Informatyki, Kolegium Nauk Przyrodniczych                                          |
| Nazwa jednostki realizującej przedmiot                | Instytut Informatyki, Kolegium Nauk Przyrodniczych                                          |
| Kierunek studiów                                      | Informatyka i ekonometria                                                                   |
| Poziom studiów                                        | studia pierwszego stopnia                                                                   |
| Profil                                                | praktyczny                                                                                  |
| Forma studiów                                         | Studia stacjonarne                                                                          |
| Rok i semestr/y studiów                               | rok I i II, sem. II i III                                                                   |
| Rodzaj przedmiotu                                     | przedmiot kierunkowy                                                                        |
| Język wykładowy                                       | polski                                                                                      |
| Koordinator                                           | dr hab. Jan Bazan, prof. UR, dr hab. Urszula Bentkowska, prof. UR                           |
| Imię i nazwisko osoby prowadzącej / osób prowadzących | dr hab. Jan Bazan, prof. UR, dr hab. Urszula Bentkowska, prof. UR, mgr inż. Adrian Cwiąkała |

\* -opcjonalnie, zgodnie z ustaleniami w Jednostce

**1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS**

| Semestr (nr) | Wykł. | Ćw. | Konw. | Lab. | Sem. | ZP | Prakt. | Inne (jakie?) | Liczba pkt. ECTS |
|--------------|-------|-----|-------|------|------|----|--------|---------------|------------------|
| 2            | 30    | 15  |       |      |      |    |        |               | 3                |
| 3            | 30    |     |       | 15   |      |    |        |               | 3                |

**1.2. Sposób realizacji zajęć**☒ zajęcia w formie tradycyjnej☐ zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość**1.3 Forma zaliczenia przedmiotu (z toku)** (egzamin, zaliczenie z oceną, zaliczenie bez oceny)

ZALICZENIE Z OCENĄ PO SEM. 2 ORAZ EGZAMIN PO SEM. 3

**2. WYMAGANIA WSTĘPNE**

Znajomość zagadnień realizowanych na przedmiotach: elementy logiki i teorii mnogości, analiza matematyczna, algebra liniowa z geometrią, podstawy programowania, programowanie obiektowe (wymaganie do części drugiej przedmiotu)

### 3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

#### 3.1 Cele przedmiotu

|                |                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------|
| C <sub>1</sub> | Nauczenie studentów metod konstrukcji algorytmów i metod analizy ich złożoności obliczeniowej.            |
| C <sub>2</sub> | Nauczenie studentów podstawowych struktur danych oraz metod ich implementacji i wykorzystania w praktyce. |
| C <sub>3</sub> | Zapoznanie studentów z przykładową biblioteką standardowych struktur danych.                              |

#### 3.2 Efekty uczenia się dla przedmiotu/ modułu

| EK ( efekt uczenia się) | Treść efektu kształcenia zdefiniowanego dla przedmiotu (modułu) - <b>wiedza minimalna i minimalne umiejętności</b> do zaliczenia                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Odniesienie do efektów kierunkowych (KEK) |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| EK_o1                   | Student zna notacje asymptotyczne, metody wykorzystywania ich do wyznaczania złożoności obliczeniowej algorytmów oraz techniki obliczeniowe pozwalające poprawnie wyznaczać złożoność obliczeniową (czasową i pamięciową) dla algorytmów iteracyjnych. Jednak niekoniecznie zna w wystarczającym stopniu techniki obliczeniowe pozwalające na wyznaczanie złożoności obliczeniowej algorytmów rekurencyjnych. Zna podstawowe klasy złożoności obliczeniowej algorytmów, ale być może nie potrafi poprawnie ocenić ich praktycznego znaczenia do rozwiązywania rzeczywistych problemów algorytmicznych.                                                                                                                           | K_Wo1, K_Wo2, K_Ko2                       |
| EK_o2                   | Student zna abstrakcyjne struktury danych, metody ich implementacji w przynajmniej jednym języku programowania oraz gotowe implementacje w dedykowanej bibliotece standardowej, w tym stosy, kolejki, listy, drzewa, grafy, słowniki, haszowanie, drzewa przeszukiwań binarnych. W szczególności student zna budowę tych struktur oraz operacje jakie mogą być wykonywane na tych strukturach. Jednakże możliwe jest, iż posiadana wiedza o efektywności tych operacji w kontekście złożoności obliczeniowej nie pozwala mu na w pełni poprawne porównanie tych struktur pod względem ich efektywności obliczeniowej oraz na dobieranie struktur danych do ustalonych wymagań dotyczących efektywności obliczeniowej algorytmów. | K_Wo1, K_Wo2, K_Ko2                       |
| EK_o3                   | Student zna zasady formułowania i algorytmizacji zadań oraz notację zapisu algorytmów w pseudojęzyku i w wybranym języku programowania, a także podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej                                                                                                                                                                                                                                                                                                                                                                                                                              | K_Wo1, K_Wo2                              |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
|       | siły, metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami. Zna także podstawowe algorytmy wyszukiwania i sortowania. Być może jednak nie zawsze potrafi poprawnie porównać te metody i algorytmy pod względem efektywności czasowej i dokładności otrzymanego rozwiązania. Jednak, dla trudniejszych problemów może zdarzyć się, że niepoprawnie dobierze metody i algorytmy do wymagań oczekiwanej efektywności rozwiązania.                                                                                                                                                                                                                                 |                     |
| EK_o4 | Student zna metodę dowodzenia semantycznej poprawności algorytmów, w tym poszczególne jej kroki (dowód częściowej poprawności, dowód własności określoności i dowód własności stopu) oraz rozumie te kroki. Nie musi jednak znać wszystkich metod dowodzenia tych kroków oraz uzasadnienia praktycznego dla zasadności każdego z tych kroków.                                                                                                                                                                                                                                                                                                                                                                                     | K_Wo1, K_Wo2, K_Ko2 |
| EK_o5 | Student umie poprawnie śledzić algorytm bez rekurencji zapisany w wybranym języku programowania lub w tzw. pseudojęzyku.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | K_Uo1, K_Uo2        |
| EK_o6 | Student potrafi zastosować abstrakcyjne typy danych do rozwiązywania problemów z użyciem języka programowania, przy czym zawsze poprawnie operuje na strukturach danych przechowujących tylko wartości typów konkretnych.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | K_Uo1, K_Uo2        |
| EK_o7 | Student potrafi poprawnie wyznaczać złożoność obliczeniową algorytmów (czasową i pamięciową) przy wykorzystaniu notacji asymptotycznych dla algorytmów iteracyjnych. Jednak być może nie zawsze poprawnie wyznacza złożoności obliczeniowe algorytmów rekurencyjnych                                                                                                                                                                                                                                                                                                                                                                                                                                                              | K_Uo1, K_Uo2        |
| EK_o8 | Student potrafi poprawnie wykorzystać podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej siły, metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami oraz algorytmy sortowania, wyszukiwania i przeszukiwania grafów. Zaprojektowane i zaimplementowane przez studenta algorytmy zawsze posiadają własność stopu. Jednak dopuszcza się, by skonstruowany przez studenta algorytm miał dwa błędy umożliwiające uznanie go za poprawny z punktu widzenia określoności operacji stosowanych w algorytmie lub poprawności uzyskanych na wyjściu algorytmu wyników. | K_Uo1, K_Uo2        |
| EK_o9 | Student weryfikuje semantyczną poprawność algorytmów w takim zakresie, że potrafi poprawnie zdefiniować warunki alfa i beta częściowej poprawności algorytmu oraz poprawnie formułuje warunek stopu i warunek określoności algorytmu. Nie zawsze jednak musi potrafić poprawnie udowodnić częściową poprawność oraz warunki stopu i określoności algorytmu.                                                                                                                                                                                                                                                                                                                                                                       | K_Uo1, K_Uo2        |

### 3.3 Treści programowe

#### A. Problematyka wykładu

|                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------|
| Treści merytoryczne                                                                                                |
| <i>Część pierwsza (semestr 2)</i>                                                                                  |
| 1. Ogólne wprowadzenie do przedmiotu. Pseudojęzyk. Śledzenie algorytmów bez rekurencji.                            |
| 2. Złożoność obliczeniowa algorytmów bez rekurencji.                                                               |
| 3. Wybrane metody rozwiązywania równań rekurencyjnych.                                                             |
| 4. Algorytmy z rekurencją i ich śledzenie.                                                                         |
| 5. Złożoność obliczeniowa algorytmów z rekurencją.                                                                 |
| 6. Zarys semantycznej poprawności algorytmów i jej praktyczny aspekt (asercje, testy jednostkowe, dzienniki itd.). |
| <i>Część druga (semestr 3)</i>                                                                                     |
| 7. Wprowadzenie do metod konstruowania algorytmów.                                                                 |
| 8. Algorytmy przeszukiwania wyczerpującego                                                                         |
| 9. Metoda dziel i zwyciężaj oraz programowanie dynamiczne                                                          |
| 10. Algorytmy aproksymacyjne.                                                                                      |
| 11. Abstrakcyjne struktury danych (lista, zbiór, drzewo, graf, słownik).                                           |
| 12. Konkretnie struktury danych (tablica dynamiczna, lista powiązana, drzewo binarne, tablica mieszająca).         |
| 13. Metody implementacji abstrakcyjnych struktur danych.                                                           |
| 14. Podstawowe algorytmy wyszukiwania i sortowania.                                                                |
| 15. Implementacja grafów i wybrane algorytmy grafowe.                                                              |
| 16. Trudność problemów.                                                                                            |

#### B. Problematyka ćwiczeń audytoryjnych, konwersatoryjnych, laboratoryjnych, zajęć praktycznych

|                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Treści merytoryczne                                                                                                                                     |
| <i>Część pierwsza (semestr 2)</i>                                                                                                                       |
| 1. Zadania na konstruowanie i śledzenie algorytmów bez rekurencji.                                                                                      |
| 2. Zadania na wyznaczanie złożoności obliczeniowej algorytmów bez rekurencji.                                                                           |
| 3. Zadania na rozwiązywanie równań rekurencyjnych.                                                                                                      |
| 4. Zadania na konstruowanie i śledzenie algorytmów rekurencyjnych.                                                                                      |
| 5. Zadania na wyznaczanie złożoności obliczeniowej algorytmów rekurencyjnych.                                                                           |
| 6. Zadania na dowodzenie semantycznej poprawności algorytmów.                                                                                           |
| <i>Część druga (semestr 3)</i>                                                                                                                          |
| 7. Zadanie na konstrukcję algorytmów metodami przeszukiwania wyczerpującego.                                                                            |
| 8. Zadanie na konstrukcję algorytmów metodami przeszukiwania z nawrotami.                                                                               |
| 9. Zadanie na konstrukcję algorytmów metodami dziel i zwyciężaj oraz na programowanie dynamiczne.                                                       |
| 10. Zadania na konstrukcję algorytmów zachłannych                                                                                                       |
| 11. Zadania na konstrukcję algorytmów metodami stochastycznymi.                                                                                         |
| 12. Zadania na implementację abstrakcyjnych struktur danych, także z użyciem standardowych struktur danych.                                             |
| 13. Zadania na wyszukiwanie i sortowanie, także z użyciem standardowych implementacji algorytmów dostępnych w bibliotece standardowych struktur danych. |

### 3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną

Ćwiczenia: rozwiązywanie zadań "tablicowych"(semestr 2); rozwiązywanie zadań, w tym programistycznych z użyciem komputera (semestr 3)

## 4. METODY I KRYTERIA OCENY

### 4.1 Sposoby weryfikacji efektów uczenia się

| Symbol efektu | Metody oceny efektów kształcenia<br>(np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć) | Forma zajęć dydaktycznych<br>(w, ćw, ...) |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| EK_01         | Kolokwium pisemne, egzamin pisemny                                                                                                      | W, ĆW                                     |
| EK_02         | Kolokwium pisemne, egzamin pisemny                                                                                                      | W, ĆW                                     |
| EK_03         | Kolokwium pisemne, egzamin pisemny                                                                                                      | W, LAB                                    |
| EK_04         | Kolokwium pisemne, egzamin pisemny                                                                                                      | W, ĆW                                     |
| EK_05         | Kolokwium pisemne                                                                                                                       | W, ĆW                                     |
| EK_06         | Kolokwium przy komputerze                                                                                                               | W, LAB                                    |
| EK_07         | Kolokwium pisemne                                                                                                                       | W, ĆW                                     |
| EK_08         | Kolokwium przy komputerze                                                                                                               | W, LAB                                    |
| EK_09         | Kolokwium pisemne                                                                                                                       | W, ĆW                                     |

### 4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

| Efekt | Ocena | Kryteria otrzymania oceny                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EK_01 | dst   | Student zna notacje asymptotyczne, metody wykorzystywania ich do wyznaczania złożoności obliczeniowej algorytmów oraz techniki obliczeniowe pozwalające poprawnie wyznaczać złożoność obliczeniową (czasową i pamięciową) tylko dla algorytmów iteracyjnych. Nie zna lub zna w niewystarczającym stopniu techniki obliczeniowe pozwalające na wyznaczanie złożoności dla algorytmów rekurencyjnych. Zna podstawowe klasy złożoności obliczeniowej algorytmów, ale nie zawsze potrafi je porównać z punktu widzenia złożoności obliczeniowej oraz poprawnie ocenić ich praktyczne znaczenie do rozwiązywania rzeczywistych problemów algorytmicznych. |
|       | db    | Student zna notacje asymptotyczne, metody wykorzystywania ich do wyznaczania złożoności obliczeniowej algorytmów, ale zna niezbędne techniki obliczeniowe pozwalające poprawnie wyznaczać złożoność obliczeniową (czasową i pamięciową) tylko dla algorytmów iteracyjnych oraz dla algorytmów z rekurencją prostą. Nie zna lub zna w niewystarczającym stopniu techniki obliczeniowe pozwalające na wyznaczanie złożoności dla algorytmów z rekurencją rozgałęzioną. Zna podstawowe klasy złożoności obliczeniowej algorytmów oraz potrafi je porównać z punktu widzenia złożoności obliczeniowej, ale nie zawsze potrafi poprawnie ocenić           |

|       |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |     | ich praktyczne znaczenie do rozwiązywania rzeczywistych problemów algorytmicznych.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|       | bdb | Student zna notacje asymptotyczne, metody wykorzystywania ich do wyznaczania złożoności obliczeniowej algorytmów, w tym niezbędne techniki obliczeniowe pozwalające poprawnie wyznaczać złożoność obliczeniową (czasową i pamięciową) zarówno dla algorytmów iteracyjnych, jak i dla algorytmów z rekurencją (w tym prostą i rozgałęzioną). Zna podstawowe klasy złożoności obliczeniowej algorytmów, potrafi je porównać z punktu widzenia złożoności obliczeniowej oraz potrafi ocenić ich praktyczne znaczenie do rozwiązywania rzeczywistych problemów algorytmicznych.                                                                                                                                                                                  |
| EK_02 | dst | Student zna abstrakcyjne struktury danych, metody ich implementacji w przynajmniej jednym języku programowania oraz gotowe implementacje w dedykowanej bibliotece standardowej, w tym stosy, kolejki, listy, drzewa, grafy, słowniki, haszowanie, drzewa przeszukiwań binarnych. W szczególności student zna budowę tych struktur oraz operacje jakie mogą być wykonywane na tych strukturach. Jednakże posiadana wiedza o efektywności tych operacji w konkretnych implementacjach tych struktur w kontekście złożoności obliczeniowej, nie pozwala mu na w pełni poprawne porównanie tych struktur pod względem ich efektywności obliczeniowej oraz na dobieranie struktur danych do ustalonych wymagań dotyczących efektywności obliczeniowej algorytmów. |
|       | db  | Student zna abstrakcyjne struktury danych, metody ich implementacji w przynajmniej jednym języku programowania oraz gotowe implementacje w dedykowanej bibliotece standardowej, w tym stosy, kolejki, listy, drzewa, grafy, słowniki, haszowanie, drzewa przeszukiwań binarnych. W szczególności student zna budowę tych struktur oraz operacje jakie mogą być wykonywane na tych strukturach. Ponadto, ma wiedzę o efektywności tych operacji w kontekście złożoności obliczeniowej, co daje mu możliwość porównania tych struktur pod względem ich efektywności obliczeniowej. Nie potrafi jednak w pełni poprawnie dobierać struktur danych do ustalonych wymagań dotyczących efektywności obliczeniowej rozwiązania danego problemu algorytmicznego.     |
|       | bdb | Student zna abstrakcyjne struktury danych, metody ich implementacji w przynajmniej jednym języku programowania oraz gotowe implementacje w dedykowanej bibliotece standardowej, w tym stosy, kolejki, listy, drzewa, grafy, słowniki, haszowanie, drzewa przeszukiwań binarnych. W szczególności student zna budowę tych struktur oraz operacje jakie mogą być wykonywane na tych strukturach. Ponadto, ma wiedzę o efektywności tych operacji w kontekście złożoności obliczeniowej, co daje mu możliwość porównania tych struktur pod względem ich efektywności obliczeniowej oraz dobierania struktur danych do ustalonych wymagań dotyczących efektywności obliczeniowej.                                                                                |
| EK_03 | dst | Student zna zasady formułowania i algorytmizacji zadań oraz notację zapisu algorytmów w pseudojęzyku i w wybranym języku programowania, a także podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej siły,                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

|       |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |     | metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami. Zna także podstawowe algorytmy wyszukiwania i sortowania. Jednak nie zawsze potrafi poprawnie porównać te metody algorytmy pod względem efektywności czasowej i dokładności otrzymanego rozwiązania. Nie zawsze potrafi także dobrze dobierać metody i algorytmy do wymagań oczekiwanej efektywności rozwiązania danego problemu.                                                                                                                                                                                                                                                                                  |
|       | db  | Student zna zasady formułowania i algorytmizacji zadań oraz notację zapisu algorytmów w pseudojęzyku i w wybranym języku programowania, a także podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej siły, metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami. Zna także podstawowe algorytmy wyszukiwania i sortowania. Potrafi porównać te metody i algorytmy pod względem efektywności czasowej i dokładności otrzymanego rozwiązania. Nie zawsze potrafi jednak dobrze dobierać metody i algorytmy do wymagań oczekiwanej efektywności rozwiązania danego problemu. |
|       | bdb | Student zna zasady formułowania i algorytmizacji zadań oraz notację zapisu algorytmów w pseudojęzyku i w wybranym języku programowania, a także podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej siły, metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami. Zna także podstawowe algorytmy wyszukiwania i sortowania. Potrafi porównać te metody i algorytmy pod względem efektywności czasowej i dokładności otrzymanego rozwiązania, a także poprawnie dobiera metody do wymagań oczekiwanej efektywności rozwiązania danego problemu.                             |
| EK_o4 | dst | Student zna metodę dowodzenia semantycznej poprawności algorytmów, w tym poszczególne jej kroki (dowód częściowej poprawności, dowód własności określoności i dowód własności stopu) oraz rozumie te kroki. Nie zna jednak metod dowodzenia wszystkich tych kroków oraz nie zna uzasadnienia praktycznego dla zasadności każdego z tych kroków.                                                                                                                                                                                                                                                                                                                                                                                             |
|       | db  | Student zna metodę dowodzenia semantycznej poprawności algorytmów, w tym poszczególne jej kroki (dowód częściowej poprawności, dowód własności określoności i dowód własności stopu) oraz rozumie te kroki oraz zna metody dowodzenia tych kroków. Jednak nie zna uzasadnienia praktycznego dla zasadności każdego z tych kroków.                                                                                                                                                                                                                                                                                                                                                                                                           |
|       | bdb | Student zna metodę dowodzenia semantycznej poprawności algorytmów, w tym poszczególne jej kroki (dowód częściowej poprawności, dowód własności określoności i dowód własności stopu), rozumie te kroki, zna metody dowodzenia tych kroków oraz potrafi uzasadnić praktycznie zasadność każdego z tych kroków.                                                                                                                                                                                                                                                                                                                                                                                                                               |
| EK_o5 | dst | Student umie śledzić algorytm iteracyjny bez rekurencji zapisany w wybranym języku programowania lub w tzw. pseudojęzyku.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

|       |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | db  | Student umie śledzić algorytm iteracyjny bez rekurencji rozgałęzionej zapisany w wybranym języku programowania lub w tzw. pseud języku.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|       | bdb | Student umie śledzić algorytm iteracyjny oraz rekurencyjny (w tym z rekurencją rozgałęzioną) zapisany w wybranym języku programowania lub w tzw. pseud języku.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| EK_o6 | dst | Student potrafi zastosować abstrakcyjne typy danych do rozwiązywania problemów z użyciem języka programowania, przy czym operuje na strukturach danych przechowujących tylko wartości typów prostych (np. int, byte, float, double, char, boolean itd.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|       | db  | Student potrafi zastosować abstrakcyjne typy danych do rozwiązywania problemów z użyciem języka programowania, przy czym operuje na strukturach danych przechowujących tylko wartości typów prostych i obiektowych-zdefiniowanych (np. String, Integer, Double, itd.).                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|       | bdb | Student potrafi zastosować abstrakcyjne typy danych do rozwiązywania problemów z użyciem języka programowania, przy czym operuje na strukturach danych przechowujących wartości dowolnych typów prostych i obiektowych, w tym typów zdefiniowanych przez studenta (np. Osoba, Punkt, itd.).                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| EK_o7 | dst | Student potrafi poprawnie wyznaczać złożoność obliczeniową algorytmów (czasową i pamięciową) przy wykorzystaniu notacji asymptotycznych dla algorytmów iteracyjnych. Nie potrafi poprawnie wyznaczać złożoności obliczeniowej algorytmów dla algorytmów rekurencyjnych.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|       | db  | Student potrafi poprawnie wyznaczać złożoność obliczeniową algorytmów (czasową i pamięciową) przy wykorzystaniu notacji asymptotycznych dla algorytmów iteracyjnych oraz dla algorytmów rekurencyjnych bez rekurencji rozgałęzionej. Nie potrafi poprawnie wyznaczać złożoności obliczeniowej algorytmów dla algorytmów z rekurencją rozgałęzioną.                                                                                                                                                                                                                                                                                                                                                                                                 |
|       | bdb | Student potrafi poprawnie wyznaczać złożoność obliczeniową algorytmów (czasową i pamięciową) przy wykorzystaniu notacji asymptotycznych dla algorytmów iteracyjnych oraz dla algorytmów rekurencyjnych (w tym z rekurencją rozgałęzioną).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| EK_o8 | dst | Student potrafi poprawnie wykorzystać podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej siły, metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami oraz algorytmy sortowania, wyszukiwania i przeszukiwania grafów. Zaprojektowane i zaimplementowane przez studenta algorytmy zawsze posiadają własność stopu. Jednak algorytm skonstruowany przez studenta podczas weryfikacji efektu kształcenia ma z błędy umożliwiające uznanie go za poprawny z punktu widzenia określoności operacji stosowanych w algorytmie lub poprawności uzyskanych na wyjściu algorytmu wyników. |



|       |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | db  | Student potrafi poprawnie wykorzystać podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej siły, metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami oraz algorytmy sortowania, wyszukiwania i przeszukiwania grafów. Zaprojektowane i zaimplementowane przez studenta algorytmy zawsze posiadają własność stopu. Jednak algorytm skonstruowany przez studenta podczas weryfikacji efektu kształcenia ma jeden błąd umożliwiający uznanie go za poprawny z punktu widzenia określoności operacji stosowanych w algorytmie lub poprawności uzyskanych na wyjściu algorytmu wyników. |
|       | bdb | Student potrafi poprawnie wykorzystać podstawowe techniki i metody projektowania i implementowania algorytmów, w tym metodę dynamicznego przydziału pamięci, rekurencję, metodę brutalnej siły, metodę dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, metodę Monte Carlo, przeszukiwanie z nawrotami oraz algorytmy sortowania, wyszukiwania i przeszukiwania grafów. Zaprojektowane i zaimplementowane przez studenta algorytmy zawsze posiadają własność stopu. Ponadto, algorytm skonstruowany przez studenta podczas weryfikacji efektu kształcenia jest poprawny z punktu widzenia określoności operacji stosowanych w algorytmie oraz poprawności uzyskanych na wyjściu algorytmu wyników.                                   |
| EK_09 | dst | Student weryfikuje semantyczną poprawność algorytmów w takim zakresie, że potrafi poprawnie zdefiniować warunki alfa i beta częściowej poprawności algorytmu oraz poprawnie formułuje warunek stopu i warunek określoności algorytmu. Nie potrafi jednak udowodnić formalnie częściowej poprawności oraz warunku stopu i określoności algorytmu.                                                                                                                                                                                                                                                                                                                                                                                                      |
|       | db  | Student weryfikuje semantyczną poprawność algorytmów w takim zakresie, że potrafi poprawnie zdefiniować warunki alfa i beta częściowej poprawności algorytmu oraz poprawnie formułuje warunek stopu i warunek określoności algorytmu. Potrafi także uzasadnić warunek stopu i określoności algorytmu. Nie potrafi jednak udowodnić formalnie częściowej poprawności algorytmu.                                                                                                                                                                                                                                                                                                                                                                        |
|       | bdb | Student w pełni potrafi przeprowadzić weryfikację semantycznej poprawności algorytmu, tzn. potrafi poprawnie zdefiniować warunki alfa i beta częściowej poprawności algorytmu oraz warunek stopu i warunek określoności algorytmu. Potrafi także uzasadnić warunek stopu i określoności algorytmu, oraz udowodnić formalnie częściową poprawność algorytmu.                                                                                                                                                                                                                                                                                                                                                                                           |

#### **Zasady uzyskania oceny końcowej:**

Zaliczenie ćwiczeń i laboratorium następuje na podstawie zaliczenia wszystkich efektów weryfikowanych przez planowane w danym okresie metody weryfikacji. Przy tym zakłada się, że każda metoda weryfikacji dostarcza osobne oceny dla każdego z weryfikowanych przez nią efektów kształcenia. Jeśli dany efekt jest weryfikowany przez więcej niż jedną metodę, to ocena weryfikująca

osiągnięcie tego efektu jest obliczana jako średnia arytmetyczna ocen uzyskanych w poszczególnych metodach weryfikowania tego efektu.

Student otrzymuje zaliczenia ocenę **niedostateczny**, gdy metody weryfikacji wykażą, iż co najmniej jeden z efektów nie został osiągnięty (średnia ocena dla tego efektu jest niższa niż 3.0);

Student otrzymuje ocenę **dostateczny**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.0, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 3.75;

Student otrzymuje ocenę **dobry**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.75, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 4.75;

Student otrzymuje ocenę **bardzo dobry**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 4.75;

Zaliczenie przedmiotu następuje na podstawie oceny uzyskanej na egzaminie

Student otrzymuje ocenę **niedostateczny**, gdy nie zaliczył ćwiczeń lub egzamin wykaże, iż co najmniej jeden z efektów nie został osiągnięty (średnia ocena dla tego efektu jest niższa niż 3.0);

Student otrzymuje ocenę **dostateczny**, gdy posiada zaliczenie z ćwiczeń, a przeciętnie każdy z efektów weryfikowanych na egzaminie zostanie osiągnięty na poziomie co najmniej 3.0, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 3.75;

Student otrzymuje ocenę **dobry**, gdy posiada zaliczenie z ćwiczeń oraz przeciętna ocena z zaliczenia każdego efektu spośród weryfikowanych na egzaminie wyniesie co najmniej 3.75, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 4.75;

Student otrzymuje ocenę **bardzo dobry**, gdy posiada zaliczenie z ćwiczeń oraz przeciętna ocena z zaliczenia każdego efektu spośród weryfikowanych na egzaminie wyniesie co najmniej 4.75.

## 5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

| Forma aktywności                                                                                          | Średnia liczba godzin na zrealizowanie aktywności |
|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| Godziny z harmonogramu studiów                                                                            | 90                                                |
| Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)                             | 4                                                 |
| Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.) | 56                                                |
| SUMA GODZIN                                                                                               | 150                                               |

**SUMARYCZNA LICZBA PUNKTÓW ECTS**

6

\* Należy uwzględnić, że 1 pkt ECTS odpowiada 25-30 godzin całkowitego nakładu pracy studenta.

**6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU**

|                                  |   |
|----------------------------------|---|
| wymiar godzinowy                 | - |
| zasady i formy odbywania praktyk | - |

**7. LITERATURA**

## Literatura podstawowa:

1. Aho A.V., Hopcroft J.E., Ullman J.D.: *Algorytmy i struktury danych*, Helion (2003).
2. Banachowski L., Diks K., Rytter W.: *Algorytmy i struktury danych*, WNT (2006).
3. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C.: *Wprowadzenie do algorytmów*, WNT (2004).
4. Lafore R.: *Java – algorytmy i struktury danych*, Helion (2003).

## Literatura uzupełniająca:

1. Banachowski L.: Kreczmar A.: *Elementy analizy algorytmów*, WNT (1989).
2. Bolc L., Cytowski J.: *Metody przeszukiwania heurystycznego*, tom I, PWN (1989).
3. Bolc L., Cytowski J.: *Metody przeszukiwania heurystycznego*, tom II, PWN (1991).
4. Wróblewski P.: *Algorytmy, struktury danych i techniki programowania*, Helion (2003).
5. Lipski W.: *Kombinatoryka dla programistów*, WNT (2004)

Akceptacja Kierownika Jednostki lub osoby upoważnionej