



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Algorytmy i struktury danych [S1SI1E>AiSD]

### Przedmiot

Kierunek studiów

Sztuczna inteligencja/Artificial Intelligence

Rok/Semestr

1/2

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

angielski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

30

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

### Liczba punktów ECTS

6,00

### Koordynatorzy

dr hab. inż. Grzegorz Pawlak

grzegorz.pawlak@put.poznan.pl

### Wykładowcy

### Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu implementacji prostych programów w języku C/C++ (lub innym podobnym języku ustalonym z prowadzącym laboratoria). Powinien posiadać umiejętność rozwiązywania podstawowych problemów programistycznych oraz testowania i poprawiania błędów w zaimplementowanych przez siebie programach. Dodatkowo student powinien posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji. W zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

## Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy o złożoności obliczeniowej w zakresie jej analizy, działania deterministycznej i niedeterministycznej maszyny Turinga, maszyny RAM, klasyfikacji problemów i algorytmów oraz klas złożoności P i NP. 2. Przekazanie studentom podstawowej wiedzy z algorytmiki w zakresie sortowania ciągów danych z różną złożonością obliczeniową, programowania zachłannego i dynamicznego, przeszukiwania z powracaniem oraz podstawowych algorytmów grafowych takich jak BFS, DFS, znajdowanie cyklu Eulera i Hamiltona. 3. Przekazanie studentom podstawowej wiedzy o strukturach danych obejmujące sposób działania drzew, drzew BST, kopców i grafów oraz analizę ich złożoności. Programowania algorytmów i metod rozwiązywania problemów poszukiwania minimalnego drzewa rozpinającego w grafie nieskierowanym oraz znajdowania składowych spójnych i dwu-spójnych w dowolnym grafie oraz algorytmów sortowania topologicznego. 4. Przekazanie studentom wiedzy o złożoności problemu plecakowego oraz metodach rozwiązywania tego problemu: zachłannych, przybliżonych i przeglądu wyczerpującego. Implementacja i analiza złożoności zastosowanych algorytmów ze znajomością programowania dynamicznego. 5. Rozwijanie u studentów podstawowych umiejętności dowodzenia NP-zupełności problemów kombinatorycznych 6. Ocena złożoności obliczeniowej dla problemów optymalizacyjnych i odpowiadających im problemów decyzyjnych. 7. Podstawowa charakterystyka metody podziału i ograniczeń z przykładami rozwiązywania dla klasycznego problemu komiwojażera 8. Rozwijanie u studentów umiejętności implementacji programistycznej poznanych algorytmów oraz struktur danych. 9. Rozwijanie u studentów umiejętności doboru odpowiedniego algorytmu i struktury danych do rozwiązywanego problemu oraz ocenę złożoności obliczeniowej i pamięciowej ich implementacji. 10. Rozwijanie u studentów umiejętności testowania zaimplementowanych algorytmów oraz ich oceny.

## Przedmiotowe efekty uczenia się

### Wiedza:

Ma rozszerzoną i pogłębianą wiedzę z matematyki dyskretnej i podstaw teorii grafów przydatną do formułowania i rozwiązywania złożonych zadań informatycznych dotyczących analizy i formalnych dowodów poprawności oraz złożoności obliczeniowej algorytmów.

Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności obliczeniowej problemów kombinatorycznych. Ma szczegółową wiedzę nt. algorytmiki, struktur danych oraz analizy złożoności obliczeniowej i pamięciowej.

Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu analizy złożoności obliczeniowej algorytmów i problemów.

### Umiejętności:

Potrafi planować i przeprowadzać eksperymenty, w tym pomiary czasu działania algorytmów, interpretować uzyskane wyniki i wyciągać wnioski o poprawności doboru i złożoności algorytmów. Ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z podstawowych języków programowania wysokiego poziomu.

### Kompetencje społeczne:

Potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania poprzez rozstrzygnięcie dylematu, czy implementacja bardziej wydajnych algorytmów warta jest zwiększonego nakładu pracy na ich implementację.

## Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

### Ocena formująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę nieobowiązkowych zadań rozdawanych studentom w czasie wykładów: jednego dotyczącego implementacji i analizy struktury grafu, drugiego dotyczącego analizy złożoności obliczeniowej,
- premiowanie aktywności studentów na wykładach;

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę sprawozdań z wynikami projektów, których celem jest implementacja i analiza algorytmów i struktur danych,
- ocenę projektów programistycznych z implementacją algorytmów przy wykorzystaniu różnorodnych struktur danych,
- ocenę rozwiązań zadań demonstrujących sposób działania algorytmów prezentowanych przez

studentów na tablicy.

- poprzez weryfikacje kodów źródłowych programów i ocenę eksperymentów obliczeniowych wykonanych przez studentów

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę sprawozdań z wynikami projektów, których celem jest implementacja i analiza algorytmów i struktur danych,

- ocenę projektów programistycznych z implementacją algorytmów przy wykorzystaniu różnorodnych struktur danych,

- bieżącą ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych poprzez ewentualne kolokwia sprawdzające wiedzę teoretyczną (maks. 2 w semestrze),

- w zakresie laboratorium niezbędne jest uzyskanie pozytywnej oceny z każdego z wykonywanych

- ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym o charakterze problemowym:

a. w formie 5 zadań, które polegają na przeprowadzeniu obliczeń i analizie zastosowania odpowiednich algorytmów do rozwiązywania przykładowych problemów wraz z analizą ich efektywności i złożoności obliczeniowej i/lub pamięciowej sprawdzających umiejętności studentów w zakresie rozwiązywania problemów algorytmicznych,

b. zadania punktowane są w skali 0-5 punktów, ze skokiem co 0,25 punktu; do zaliczenia egzaminu wymagane jest przynajmniej 52% punktów.

c. Aktywność podczas zajęć może być premiowana przyznawaniem punktów, które uwzględniane są w czasie wystawiania oceny podsumowującej pracę w semestrze.

## Treści programowe

Wykłady z przedmiotu rozpoczynają się od wyjaśnienia podstawowych terminów z zakresu algorytmiki, takich jak problem i algorytm, dane i operacje na danych, instancja, pojęcie typu. Poruszona zostaje tematyka poprawności algorytmów, jej definiowanie oraz weryfikacja. Przedstawiony zostaje podział problemów na decyzyjne i optymalizacyjne, wraz z charakterystyką tych klas i przykładami należących do nich problemów. Przed przystąpieniem do omawiania implementacji algorytmów we współczesnych językach programowania omawiana jest deterministyczna i niedeterministyczna maszyna Turinga oraz maszyna RAM, jako przykłady abstrakcyjnego modelu komputera służącego do wykonywania algorytmów. W oparciu o ten materiał wyjaśniona zostaje idea i definicja klas problemów decyzyjnych P oraz NP, wraz z podklasami problemów NP-zupełnych i silnie NP-zupełnych oraz przedstawione sposoby dowodzenia przynależności problemów do tych klas. Omówiona zostaje złożoność obliczeniowa problemów oraz złożoność czasowa i pamięciowa algorytmów wraz ze sposobami jej wyznaczania oraz zapisywania w notacji  $O()$ . Poruszony zostaje problem złożoności w najgorszym i najlepszym przypadku oraz złożoność średnia. W trakcie wykładu szczegółowo prezentowane są ogólne metody konstruowania algorytmów sortowania, dzieli i rządzi oraz przeszukiwanie z powrotami. Przedstawiane jest również porównanie metody zachłannej oraz programowania dynamicznego wraz z omówieniem złożoności pseudowielomianowej. W tym celu wykorzystywana jest szczegółowa analiza problemu plecakowego. W czasie wykładu prezentowane są również możliwe sposoby komputerowej reprezentacji grafów uwzględniając macierz sąsiedztwa, macierz incydencji, listę krawędzi i listę incydencji wraz ze szczegółową analizą ich złożoności czasowej i pamięciowej w zależności od liczby wierzchołków i krawędzi w grafie oraz liczby wykonywanych operacji.

Zajęcia laboratoryjne kładą duży nacisk na zastosowanie w praktyce algorytmów i struktur danych prezentowanych na wykładzie poprzez realizację projektów oraz rozwiązywanie zadań na tablicy. Zajęcia podzielone są na kilka grup tematycznych, z których każda zakończona jest realizacją projektu implementującego omawiane algorytmy. W pierwszej grupie tematycznej prezentowane są algorytmy sortowania począwszy od najprostszych, działających ze złożonością kwadratową, takich jak sortowanie bąbelkowe, przez wybór i wstawianie, przez szybsze sortowanie Quick sort, Heap sort przez scalanie oraz Shella, po sortowania w czasie liniowym przez zliczanie. Dla każdego algorytmu analizowana jest jego zademonstrowana zostaje również koncepcja rekurencji. Kolejna grupa tematyczna obejmuje złożone struktury danych, takie jak lista jedno i dwu kierunkowa, drzewa, w tym drzewa BST oraz kopce. Dla każdej struktury przedstawiony jest algorytm dodawania i usuwania z nich elementów oraz możliwe sposoby ich przeszukiwania. Analizowana jest również ich złożoność oraz problemy, w których należy je wykorzystać. Trzecia grupa tematyczna to algorytmy grafowe obejmująca algorytmy dla grafów skierowanych i nieskierowanych, takie jak BFS, DFS, sortowanie topologiczne, drzewa rozpinające oraz wyszukiwanie cyklu Eulera i Hamiltona prezentujące również algorytmy z powracaniem. W trakcie omawiania grafów szczegółowo poruszana jest tematyka implementacji reprezentacji grafów przedstawionych na wykładzie. Ostatnia grupa tematyczna obejmuje implementację algorytmu

zachłannego - przybliżonego oraz dynamicznego dla problemu plecakowego a także algorytmu przeglądu wyczerpującego ich porównanie oraz analizę. Ponadto wprowadzono podstawy metody podziału i ograniczeń w raz z przykładami rozwiązań dla problemu komiwojażera.

## Tematyka zajęć

Wykłady z przedmiotu rozpoczynają się od wyjaśnienia podstawowych terminów z zakresu algorytmiki, takich jak problem i algorytm, dane i operacje na danych, instancja, pojęcie typu. Poruszona zostaje tematyka poprawności algorytmów, jej definiowanie oraz weryfikacja. Przedstawiony zostaje podział problemów na decyzyjne i optymalizacyjne, wraz z charakterystyką tych klas i przykładami należących do nich problemów. Przed przystąpieniem do omawiania implementacji algorytmów we współczesnych językach programowania omawiana jest deterministyczna i nondeterministyczna maszyna Turinga oraz maszyna RAM, jako przykłady abstrakcyjnego modelu komputera służącego do wykonywania algorytmów. W oparciu o ten materiał wyjaśniona zostaje idea i definicja klas problemów decyzyjnych P oraz NP, wraz z podklasami problemów NP-zupełnych i silnie NP-zupełnych oraz przedstawione sposoby dowodzenia przynależności problemów do tych klas. Omówiona zostaje złożoność obliczeniowa problemów oraz złożoność czasowa i pamięciowa algorytmów wraz ze sposobami jej wyznaczania oraz zapisywania w notacji  $O()$ . Poruszony zostaje problem złożoności w najgorszym i najlepszym przypadku oraz złożoność średnia. W trakcie wykładu szczegółowo prezentowane są ogólne metody konstruowania algorytmów sortowania, dziel i rządź oraz przeszukiwanie z powrotami. Przedstawiane jest również porównanie metody zachłannej oraz programowania dynamicznego wraz z omówieniem złożoności pseudowielomianowej. W tym celu wykorzystywana jest szczegółowa analiza problemu plecakowego. W czasie wykładu prezentowane są również możliwe sposoby komputerowej reprezentacji grafów uwzględniając macierz sąsiedztwa, macierz incydencji, listę krawędzi i listę incydencji wraz ze szczegółową analizą ich złożoności czasowej i pamięciowej w zależności od liczby wierzchołków i krawędzi w grafie oraz liczby wykonywanych operacji. Zajęcia laboratoryjne kładą duży nacisk na zastosowanie w praktyce algorytmów i struktur danych prezentowanych na wykładzie poprzez realizację projektów oraz rozwiązywanie zadań na tablicy. Zajęcia podzielone są na kilka grup tematycznych, z których każda zakończona jest realizacją projektu implementującego omawiane algorytmy. W pierwszej grupie tematycznej prezentowane są algorytmy sortowania począwszy od najprostszyc, działających ze złożonością kwadratową, takich jak sortowanie bąbelkowe, przez wybór i wstawianie, przez szybsze sortowanie Quick sort, Heap sort przez scalanie oraz Shella, po sortowania w czasie liniowym przez zliczanie. Dla każdego algorytmu analizowana jest jego zademonstrowana zostaje również koncepcja rekurencji. Kolejna grupa tematyczna obejmuje złożone struktury danych, takie jak lista jedno i dwu kierunkowa, drzewa, w tym drzewa BST oraz kopce. Dla każdej struktury przedstawiony jest algorytm dodawania i usuwania z nich elementów oraz możliwe sposoby ich przeszukiwania. Analizowana jest również ich złożoność oraz problemy, w których należy je wykorzystać. Trzecia grupa tematyczna to algorytmy grafowe obejmująca algorytmy dla grafów skierowanych i nieskierowanych, takie jak BFS, DFS, sortowanie topologiczne, drzewa rozpinające oraz wyszukiwanie cyklu Eulera i Hamiltona prezentujące również algorytmy z powracaniem. W trakcie omawiania grafów szczegółowo poruszana jest tematyka implementacji reprezentacji grafów przedstawionych na wykładzie. Ostatnia grupa tematyczna obejmuje implementację algorytmu zachłannego - przybliżonego oraz dynamicznego dla problemu plecakowego a także algorytmu przeglądu wyczerpującego ich porównanie oraz analizę. Ponadto wprowadzono podstawy metody podziału i ograniczeń w raz z przykładami rozwiązań dla problemu komiwojażera.

## Metody dydaktyczne

Wykład: prezentacja ilustrowana przykładami podawanymi na tablicy.

Ćwiczenia laboratoryjne: prezentacja ilustrowana przykładami podanymi na tablicy oraz wykonywanie zadań podanych przez nauczyciela - ćwiczenia praktyczne.

## Literatura

Podstawowa:

1. Introduction to Algorithms, T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, The MIT Press ([https://edutechlearners.com/download/Introduction\\_to\\_algorithms-3rd%20Edition.pdf](https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf))
2. Algorithms + Data Structures = Programs, N. Wirth, 2004 (<http://www.ethoberon.ethz.ch/WirthPubl/AD.pdf>)
3. Computers and Intractability. A Guide to the theory of NP-hardness, M.R. Garey, D.S Johnson, Freeman, 1979

Uzupełniająca:

1. Algorithms, Robert Sedgewick, Kevin Wayne, Wesley, Edition IV, 2011

### Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	150	6,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	88	3,50