



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Algorytmy i struktury danych [S1Cybez1>AiSD]

Przedmiot

Kierunek studiów

Cyberbezpieczeństwo

Rok/Semestr

1/2

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratorium

0

Inne

0

Ćwiczenia

0

Projekty/seminaria

24

Liczba punktów ECTS

4,00

Koordynatorzy

prof. dr hab. inż. Marta Szachniuk

marta.szachniuk@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający niniejszy przedmiot powinien posiadać podstawową wiedzę z zakresu implementacji prostych algorytmów w językach programowania wysokiego poziomu (Python, C, C++). 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 wiedzy oraz rozumieć konieczność poszerzania swoich kompetencji w zakresie algorytmiki i programowania. W zakresie kompetencji społecznych student powinien wykazywać się uczciwością, odpowiedzialnością, wytrwałością, ciekawością poznawczą, kreatywnością, kulturą osobistą, szacunkiem dla innych ludzi.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy z zakresu złożoności obliczeniowej dotyczącej analizy złożoności problemów kombinatorycznych i algorytmów, 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 liczbowych z różną złożonością obliczeniową, programowania zachłannego, wyczerpującego i dynamicznego, przeszukiwania z nawracaniem, podstawowych algorytmów grafowych takich jak przeszukiwanie grafu, sortowanie topologiczne, znajdowanie cyklu Eulera i Hamiltona, podstaw dotyczących algorytmów probabilistycznych. 3. Przekazanie studentom podstawowej wiedzy o strukturach danych, obejmującej sposób działania na tablicach, listach, kolejkach, stosach, drzewach (w tym drzewach BST i kopcach) i grafach, analizę złożoności pamięciowej oraz złożoności obliczeniowej podstawowych operacji realizowanych na tych strukturach danych. 4. Rozwijanie u studentów umiejętności implementacji poznanych algorytmów oraz struktur danych. 5. Rozwijanie u studentów umiejętności doboru odpowiedniego algorytmu i struktury danych do rozwiązywanego problemu oraz ocenę złożoności obliczeniowej i pamięciowej ich implementacji. 6. Rozwijanie u studentów umiejętności testowania zaimplementowanych algorytmów oraz oceny ich poprawności, odporności na błędy, skalowalności i efektywności.

Przedmiotowe efekty uczenia się

Wiedza:

Ma rozszerzoną i pogłębioną wiedzę z badań operacyjnych przydatną do formułowania i rozwiązywania złożonych zadań informatycznych.

Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie wybranych algorytmów kombinatorycznych.

Ma szczegółową wiedzę dotyczącą algorytmiki, struktur danych oraz analizy złożoności obliczeniowej i pamięciowej algorytmów.

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

Umiejętności:

Potrafi planować i przeprowadzać eksperymenty, w tym pomiary czasu działania algorytmów, interpretować uzyskane wyniki i wyciągać wnioski o poprawności doboru i złożoności algorytmów.

Potrafi wykorzystać do formułowania i rozwiązywania zadań informatycznych metody analityczne i eksperymentalne w celu dobrania odpowiednich algorytmów i struktur danych. Potrafi ocenić złożoność obliczeniową algorytmów i problemów.

Ma umiejętność opracowywania 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:

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

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

- premiowanie aktywności studentów na wykładach

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

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

- ocenę zaimplementowanych algorytmów z wykorzystaniem różnorodnych struktur danych.

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

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

- ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym śródkresowym oraz końcowym:
- w formie kilku zadań zamkniętych, które polegają na wpisaniu w wolne miejsca wyniku obliczeń i analiz sprawdzających umiejętności studentów w zakresie rozwiązywania problemów algorytmicznych,
- do zaliczenia wykładu wymagane jest 50% punktów uzyskanych sumarycznie na egzaminie śródkresowym i końcowym.

Aktywność podczas zajęć premiowana jest przyznawaniem punktów, które uwzględniane są w czasie wystawiania oceny podsumowującej pracę studenta w semestrze.

Zależność oceny od liczby punktów definiuje Regulamin Studiów. Dodatkowo zasady zaliczania przedmiotu i dokładne progi zaliczeniowe zostaną przekazane studentom na początku semestru z wykorzystaniem uczelnianych systemów elektronicznych oraz na pierwszych zajęciach (w każdej formie zajęć).

Treści programowe

Wykłady z przedmiotu obejmują podstawową tematykę z zakresu algorytmiki:

- podstawowe terminy: problem i algorytm, dane i operacje na danych, instancja
- poprawność algorytmów, weryfikacja
- podział problemów na decyzyjne i optymalizacyjne
- deterministyczna i niedeterministyczna maszyna Turinga oraz maszyna RAM, jako przykłady abstrakcyjnego modelu komputera służącego do wykonywania algorytmów.
- 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
- złożoność obliczeniowa problemów oraz złożoność czasowa i pamięciowa algorytmów wraz ze sposobami jej wyznaczania oraz zapisywania w notacji $O()$
- metody konstruowania algorytmów takie jak metoda zstępująca, dziel i rządź oraz przeszukiwanie z nawrotami
- porównanie metody zachłannej oraz programowania dynamicznego wraz z omówieniem złożoności pseudowielomianowej na przykładzie problemu plecakowego
- komputerowe reprezentacje grafów uwzględniając macierz i listę incydencji, listę następników oraz macierz grafu

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. 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. Omówiona zostaje złożoność obliczeniowa i pamięciowa algorytmów wraz ze sposobami jej wyznaczania oraz zapisywania w notacji $O()$. Na przykładzie różnych algorytmów sortowania przedstawiony jest problem złożoności w najgorszym i najlepszym przypadku oraz złożoność średnia. W trakcie wykładu szczegółowo prezentowane są ogólne metody konstruowania algorytmów takie jak metoda zstępująca, dziel i rządź oraz algorytm z nawrotami. Porównywane są podejścia zachłanne, wyczerpujące oraz metoda programowania dynamicznego na przykładzie decyzyjnego problemu plecakowego. Omawiane są różne sposoby komputerowej reprezentacji grafów (macierz i lista incydencji, lista następników, macierz sąsiedztwa, lista krawędzi i macierz grafu). Przedstawione są wybrane problemy grafowe (wyszukiwanie ścieżki i cyklu Eulera, wyszukiwanie ścieżki i cyklu Hamiltona, sortowanie topologiczne grafu) wraz z przykładowymi metodami ich rozwiązywania. Oddzielnie omawiane są struktury drzewiaste (m.in. drzewa binarne, binarne drzewa przeszukiwań, kopce) wraz z podstawowymi operacjami na drzewach oraz zastosowaniami tych struktur do rozwiązywania problemów praktycznych. Dodatkowo podczas wykładu wprowadzane są podstawowe informacje na temat algorytmów probabilistycznych i funkcji haszujących.

Zajęcia projektowe uzupełniające wykład kładą duży nacisk na implementację oraz praktyczne zastosowania algorytmów i struktur danych przedstawianych na wykładzie. W ramach tych zajęć wyodrębniono pięć tematów. Każdy z nich wymaga realizacji projektu, na który składa się implementacja kilku algorytmów rozwiązujących zadany problem kombinatoryczny, walidacja poprawności algorytmów, implementacja skryptów do wykonania testów obliczeniowych, przeprowadzenie testów oraz opracowanie raportu z testowania oprogramowania. Pierwszy temat dotyczy sortowania ciągów

liczbowych. W ramach związanego z nim projektu należy zaimplementować kilka algorytmów sortowania różniących się złożonością obliczeniową (np. sortowanie bąbelkowe, sortowanie przez wybór i przez wstawianie, sortowanie szybkie, sortowanie z malejącymi przyrostami, sortowanie stogowe, sortowanie przez scalanie, sortowanie kubełkowe, sortowanie przez zliczanie). Drugim tematem są złożone struktury danych (lista jedno- i dwukierunkowa, drzewo binarne, drzewo BST, drzewo zrównoważone, kopiec), a w ramach projektu implementowane są przeróżne proste operacje na tych strukturach (przeszukiwanie, dodawanie elementów, usuwanie elementów). Przy trzecim temacie, którym są reprezentacje grafu i problemy grafowe, zadanie projektowe polega na implementacji algorytmów przeszukiwania grafu wszerz i wzdłuż oraz algorytmów sortowania topologicznego działających na różnych reprezentacjach maszynowych grafu skierowanego. Czwarty temat dotyczy algorytmów z powracaniem a w ramach projektu implementowane są algorytmy rozwiązujące problem wyszukiwania ścieżki lub cyklu Eulera i Hamiltona. W ramach piątego tematu przerabiany jest problem plecakowy, a związany z tym tematem projekt polega na implementacji algorytmu zachłannego, wyczerpującego oraz dynamicznego rozwiązujących binarną wersję tego problemu. Projekty realizowane są w zespołach dwuosobowych.

Metody dydaktyczne

1. Wykład: prezentacja ilustrowana przykładami.
2. Projekt: samodzielne programowanie wybranych algorytmów, uczenie przez doświadczenie, rozwiązywanie problemów praktycznych, weryfikacja poprawności kodu, przygotowanie i wykonywanie eksperymentów obliczeniowych, raportowanie i dokumentacja, prezentacja i obrona projektu, praca zespołowa, dyskusja.

Literatura

Podstawowa:

1. Elementy analizy algorytmów, L. Banachowski, A. Kreczmar, WNT, W-wa, 1982
 2. Algorytmy + struktury danych = programy, N. Wirth, WNT, W-wa, 2004
 3. Złożoność obliczeniowa problemów kombinatorycznych, J. Błażewicz, WNT, W-wa, 1988
 4. Wprowadzenie do algorytmów, T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, PWN, W -wa, 2012
- Uzupełniająca
1. Algorytmika praktyczna nie tylko dla mistrzów, P. Stańczyk, PWN, 2009

Uzupełniająca:

-

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	109	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	54	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	55	2,00