

ALGORYTMY I STRUKTURY DANYCH

Kod modułu: ASD

Rodzaj przedmiotu: kierunkowy; obowiązkowy

Wydział: Informatyki

Kierunek: Informatyka

Poziom studiów: pierwszego stopnia – VI poziom PRK

Profil studiów: praktyczny

Forma studiów: stacjonarna/niestacjonarna

Rok: 1

Semestr: 2

Formy zajęć i liczba godzin:

Forma stacjonarna

wyklady – 45

laboratorium – 35

Forma niestacjonarna

wyklady – 20

laboratorium – 20

Zajęcia prowadzone są w języku polskim.

Liczba punktów ECTS: 5

Osoby prowadzące:

wykład:

laboratorium:

1. Założenia i cele przedmiotu:

Celem przedmiotu jest przekazanie studentom wiedzy na temat podstaw algorytmiki: począwszy od pojęcia i cech algorytmu, poprzez metody weryfikacji poprawności i złożoności algorytmów do przeglądu algorytmów rozwiązujących różnorodne problemy – od najprostszych, do NP-trudnych. Towarzyszy temu omówienie (przypomnienie) struktur danych, w szczególności tablic, rekordów czy wskaźników, za pomocą których zapisuje się przetwarzane dane. Studenci powinni zapoznać się z szeroką gamą istniejących algorytmów zapisywanych w różnych notacjach (opis słowny, schematy blokowe, pseudokody), aby potrafić je zastosować w praktyce (np. na przedmiotach związanych z programowaniem), a także aby algorytmy te modyfikować lub na ich podstawie tworzyć nowe algorytmy rozwiązujące nowe zadania.

2. Określenie przedmiotów wprowadzających wraz z wymaganiami wstępnymi:

Podstawy programowania, Języki programowania – student powinien znać podstawowe struktury danych stosowane w językach programowania oraz podstawowy zestaw instrukcji do zapisu algorytmów.

3. Opis form zajęć

a) Wykłady

• **Treści programowe**

1. Pojęcie algorytmu.
2. Cechy i rodzaje algorytmów.
3. Metody zapisu algorytmu: opis słowny, schematy blokowe, pseudokod.
4. Pojęcie typu danych.
5. Proste typy danych – liczbowe, znakowe, logiczne.
6. Złożone typy danych – tablice, rekordy, zbiory, pliki. Typy wskaźnikowe
7. Reprezentacja fizyczna struktur danych.
8. Weryfikacja poprawności algorytmów. Pojęcie niezmiennika, metoda Floyda.
9. Złożoność obliczeniowa algorytmów – pamięciowa i czasowa. Złożoność czasowa średnia, pesymistyczna.
10. O-notacja dla złożoności algorytmów. Podział algorytmów ze względu na złożoność. Problemy NP-zupełne, nierozstrzygalność.
11. Przykłady szacowania złożoności czasowej.
12. Sortowanie: przez proste wstawianie, przez proste wybieranie, sortowanie bąbelkowe.
13. Sortowanie drzewiaste i stogowe (kopcowanie).
14. Metoda dziel i zwyciężaj - sortowanie szybkie.
15. Wyszukiwanie liniowe, blokowe i binarne. Rola wartownika w wyszukiwaniu.
16. Haszowanie. Metody przewyżczania kolizji. Doskonała, minimalna funkcja haszująca.
17. Wyszukiwanie wzorca w tekście, słowniki.
18. Abstrakcyjne struktury danych – listy, stosy, kolejki, kolejki priorytetowe.
19. Realizacja fizyczna list, stosów i kolejek (wskaźniki, tablice).
20. Grafy – definicja, cechy, rodzaje.
21. Reprezentacje grafów skierowanych i nieskierowanych (wskaźniki, tablice, uporządkowanie lewolistowe).
22. Podstawowe operacje na grafach, przeszukiwanie grafu w głąb, wszerz. Przeszukiwanie z nawrotami.
23. Pojęcie i definicje drzew. Cechy drzew – wysokość, moment, rząd itd.
24. Drzewa binarne, reprezentacja tablicowa.
25. Zastosowanie drzew (np. drzewo rozbioru gramatycznego wyrażeń arytmetycznych, drzewa BST w sortowaniu i wyszukiwaniu, drzewa jako reprezentacja wyrażeń arytmetycznych – odczyt postaci prefiksowej, infiksowej i postfiksowej).
26. Generowanie kodu Huffmana algorytmem zachłannym z wykorzystaniem drzewa.
27. Znajdowanie minimalnego drzewa rozpinającego algorytmami Prima i Kruskala.
28. Znajdowanie cyklu Eulera w grafach.
29. Znajdowanie najkrótszej ścieżki w grafach (porównanie algorytmu zachłannego z programowaniem dynamicznym).
30. Znajdowanie cyklu Hamiltona w grafach.
31. Analiza problemu komiwojażera, zastosowanie heurystyki.
32. Zalety i wady rekurencji na przykładzie algorytmów (porównanie algorytmów rekurencyjnych i iteracyjnych):
 - silnia,
 - liczby Fibonacciego
 - wieże Hanoi.
33. Algorytmy tworzenia trójkąta Sierpińskiego i innych fraktali.

• **Metody dydaktyczne:**

Wykład prowadzony metodą tradycyjną z wykorzystaniem rzutnika multimedialnego prezentacją algorytmów oraz programów.

• **Forma i warunki zaliczenia:**

Warunkiem zaliczenia całości przedmiotu jest zdanie egzaminu w formie zadaniowej i testowej. Część testowa powinna uwzględniać przede wszystkim część teoretyczną.

• **Literatura podstawowa:**

1. Wilson R.J.: Wprowadzenie do teorii grafów, PWN, Warszawa 2002.
2. Wróblewski P.: Algorytmy, struktury danych i techniki programowania dla programistów Java. Gliwice: HELION, cop. 2019
3. Jamro M.: Struktury danych i algorytmy w języku C#. Projektowanie efektywnych aplikacji. Gliwice: HELION, cop. 2019
4. Cutajar J.: *Struktury danych i algorytmy w języku Java. Przewodnik dla początkujących.* Gliwice: HELION, cop. 2019
5. Bhargava A.Y.: *Algorytmy. Ilustrowany przewodnik.* Gliwice: Helion, cop. 2017.

• **Literatura uzupełniająca:**

1. Knuth D.E.: Sztuka programowania, WNT, Warszawa 2003
2. Aho A.V., Hopcroft J.E., Ullman J.D.: Projektowanie i analiza algorytmów komputerowych. PWN, Warszawa 1983.
3. Buczek B.: Algorytmy. Ćwiczenia. Helion, 2008.
4. Krok E., Stempnakowski Z.: Podstawy algorytmów. Schematy blokowe. Difin, 2008.
5. Wirth N.: Algorytmy + struktury danych = programy. WNT, Warszawa 2004.

b) Ćwiczenia audytoryjne

• **Treści programowe (tematyka zajęć):**

1. Konstrukcja algorytmów i ich zapis za pomocą metod: opis słowny, schematy blokowe, pseudokod.
2. Wyznaczanie złożoności obliczeniowej algorytmów – średniej i pesymistycznej.
3. Analiza metod sortowania: przez proste wstawianie, przez proste wybieranie, bąbelkowego, drzewiastego, stogowego i szybkiego.
4. Analiza metod wyszukiwania: liniowego, binarnego.
5. Haszowanie. Metody przewyżczania kolizji. Wyszukiwanie wzorca w tekście, słowniki.
6. Realizacja fizyczna list, stosów i kolejek (wskaźniki, tablice).
7. Operacje na listach w reprezentacji ze wskaźnikami osadzonymi i w reprezentacji tablicowej (*Front, Push, Pop, Rear, Inject, Eject*).
8. Grafy – przykłady, reprezentacja grafów skierowanych i nieskierowanych (wskaźniki, tablice, uporządkowanie lewolistowe).
9. Wykonywanie podstawowych operacji na grafach, przeszukiwanie grafu w głąb, wszerz.
10. Drzewa: cechy i metody zapisu, reprezentacja tablicowa, wskaźnikowa, zapis lewolistowy.
11. Przykładowe zastosowania drzew (np. drzewo rozbioru gramatycznego wyrażeń arytmetycznych, drzewa BST w sortowaniu i wyszukiwaniu).
12. Generowanie kodu Huffmana algorytmem zachłannym z wykorzystaniem drzewa.
13. Znajdowanie minimalnego drzewa rozpinającego algorytmami Prima i Kruskala.
14. Znajdowanie cyklu Eulera w grafach.
15. Znajdowanie najkrótszej ścieżki w grafach (porównanie algorytmu zachłannego z programowaniem dynamicznym).
16. Znajdowanie cyklu Hamiltona w grafach.
17. Rekurencyjne i iteracyjne realizacje algorytmów:
 - silnia,
 - liczby Fibonacciego,
 - wieże Hanoi.
18. Algorytmy tworzenia trójkąta Sierpińskiego i innych fraktali.

• **Metody dydaktyczne:**

Na ćwiczeniach studenci powinni analizować, modyfikować i rozszerzać podane algorytmy oraz budować ich nowe wersje służące do rozwiązywania nowych problemów. W miarę możliwości i umiejętności programowania, niektóre algorytmy mogą być zaprogramowane na komputer.

• **Forma i warunki zaliczenia:**

Warunkiem zaliczenia całości przedmiotu jest zdanie egzaminu w formie zadaniowej i testowej. Część testowa powinna uwzględniać przede wszystkim część teoretyczną.

• **Literatura podstawowa:**

6. Wilson R.J.: Wprowadzenie do teorii grafów, PWN, Warszawa 2002.
7. Wróblewski P.: Algorytmy: struktury danych i techniki programowania. Helion, Gliwice 2003.

• **Literatura uzupełniająca:**

6. Knuth D.E.: Sztuka programowania, WNT, Warszawa 2003
7. Aho A.V., Hopcroft J.E., Ullman J.D.: Projektowanie i analiza algorytmów komputerowych. PWN, Warszawa 1983.
8. Buczek B.: Algorytmy. Ćwiczenia. Helion, 2008.
9. Krok E., Stempnackowski Z.: Podstawy algorytmów. Schematy blokowe. Difin, 2008.
10. Wirth N.: Algorytmy + struktury danych = programy. WNT, Warszawa 2004.

4. Opis sposobu wyznaczania punktów ECTS

a. forma stacjonarna

Forma zajęć	Formy aktywności studenta	Średnia ilość godzin na zrealizowanie aktywności
Wykład	kontakt z nauczycielem	45
	studiowanie literatury oraz zasobów internetowych	5
	przygotowanie prezentacji	5
	przygotowanie do egzaminu	5
Laboratorium	Kontakt z nauczycielem	35
	projektowanie i tworzenie algorytmów	20
	implementacja algorytmów	10
	przygotowanie do zajęć	5
Konsultacje	Kontakt z nauczycielem	3
Zal./Egzamin	Kontakt z nauczycielem	3
Całkowita ilość godzin aktywności studenta		125
Liczba punktów ECTS dla modułu		5

b. forma niestacjonarna

Forma zajęć	Formy aktywności studenta	Średnia ilość godzin na zrealizowanie aktywności
Wykład	kontakt z nauczycielem	20
	studiowanie literatury oraz zasobów internetowych	15
	przygotowanie prezentacji	10
	przygotowanie do egzaminu	15
Laboratorium	Kontakt z nauczycielem	20
	projektowanie i tworzenie algorytmów	20
	implementacja algorytmów	10
	przygotowanie do zajęć	15
Konsultacje	Kontakt z nauczycielem	3

Zal./Egzamin	Kontakt z nauczycielem	3
Całkowita ilość godzin aktywności studenta		125
Liczba punktów ECTS dla modułu		5

5. Wskaźniki sumaryczne

a. forma stacjonarna

- a) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich
 - Liczba godzin kontaktowych – 86
 - Liczba punktów ECTS – 3,4
- b) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach o charakterze praktycznym.
 - Liczba godzin kontaktowych – 35
 - Liczba punktów ECTS – 2,6

b. forma niestacjonarna

- a) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich
 - Liczba godzin kontaktowych – 46
 - Liczba punktów ECTS – 1,8
- b) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach o charakterze praktycznym.
 - Liczba godzin kontaktowych – 20
 - Liczba punktów ECTS – 2,6

6. Zakładane efekty uczenia się

Numer (Symbol)	Efekty uczenia się dla modułu	Odniesienie do efektów uczenia się dla kierunku
ASD_01	Zna pojęcie i metody zapisu algorytmu i struktur danych oraz ich implementację, pojęcie optymalizacji oraz rekurencji	K_W03 K_W12
ASD_02	Zna pojęcie, cechy, rodzaje i sposób implementacji grafów, w tym drzew	K_W03 K_W12
ASD_03	Rozumie działanie algorytmów sortowania, wyszukiwania, wybranych algorytmów grafowych i stosujących drzewa	K_W03
ASD_04	Potrafi ocenić poprawność algorytmu oraz jego złożoność obliczeniową	K_U01 K_U02 K_U12
ASD_05	Potrafi zastosować odpowiedni algorytm do rozwiązania postawionego problemu	K_U01 K_U02 K_U11
ASD_06	Potrafi skonstruować odpowiednie struktury danych dla podanego zadania oraz algorytmu	K_U01 K_U02 K_U11 K_U12
ASD_07	Rozumie konieczność podnoszenia swoich kompetencji zawodowych	K_K01

ASD_08	Potrafi pracować w zespole projektowo-programistycznym	K_U05 K_U09 K_K02
--------	--	-------------------------

7. Odniesienie efektów uczenia się do form zajęć i sposób oceny osiągnięcia przez studenta efektów uczenia się

Numer (Symbol)	Forma zajęć		Sposób sprawdzenia osiągnięcia efektu
	wykład	ćwiczenia	
ASD_01	x		sprawdzian egzamin
ASD_02	x		sprawdzian egzamin
ASD_03	x		sprawdzian egzamin
ASD_04	x	x	sprawdzian praca kontrolna egzamin
ASD_05		x	sprawdzian praca kontrolna
ASD_06		x	sprawdzian praca kontrolna
ASD_07	x	x	prezentacja egzamin
ASD_08		x	sprawdzian praca kontrolna

8. Kryteria uznania osiągnięcia przez studenta efektów uczenia się.

Numer (Symbol)	Efekt jest uznawany za osiągnięty gdy:
ASD_01	Na sprawdzianach oraz egzaminie wykaże się znajomością pojęcia i metod zapisu algorytmu i struktur danych oraz ich implementacji, pojęcia optymalizacji oraz rekurencji
ASD_02	Na sprawdzianach oraz egzaminie wykaże się znajomością pojęcia, cech, rodzajów i sposobów implementacji grafów, w tym drzew
ASD_03	Na sprawdzianach oraz egzaminie zademonstruje działanie algorytmów sortownia, wyszukiwania, wybranych algorytmów grafowych i stosujących drzewa
ASD_04	Na sprawdzianach oraz egzaminie oceni poprawność podanego algorytmu oraz jego złożoność obliczeniową
ASD_05	Na sprawdzianach oraz egzaminie zastosuje odpowiedni algorytm do rozwiązania postawionego problemu
ASD_06	Na sprawdzianach oraz egzaminie skonstruuje odpowiednie struktury danych dla podanego zadania oraz algorytmu
ASD_07	W prezentacji pokaże rozwinięcie pojęć przekazanych na zajęciach
ASD_08	W czasie realizacji sprawdzianu zespołowego i przy tworzeniu pracy kontrolnej potrafi pracować w zespole projektowo-programistycznym