

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ



ПТВЕРДЖУЮ

Декан

Говорушенко Т.О.

1 вересня 2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Структури даних і алгоритми

Назва

Статус дисципліни: вибіркова, дисципліна професійної підготовки

Факультет – Інформаційних технологій

Кафедра – Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Курс	Семестр	Загальне навантаження		Кількість годин						Форма семестрового контролю			
			Кредити ЄКТС	Години	Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС	Курсовий проект	Курсова робота	Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття						
ОД	3.5	не парний	8.0	240	102	34	34	34		138	-	-	+	
Разом			8	240	102	34	34	34		138			1	

Програма складена


Підпис

Мельниченко О.В.

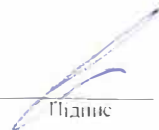
Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри

комп'ютерної інженерії та інформаційних систем

Протокол 2 від 30.08.2024 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем


Підпис

Засорнова І.О.

Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій №1 від 05.09.2024р.

Голова Вченої ради


Підпис

Говорушенко Т.О.

Ініціали, прізвище

Хмельницький 2024

Вступ

Мета викладання дисципліни. Дисципліна “Структури даних і алгоритми” є вибірковою для студентів спеціальності «Комп’ютерна інженерія» та забезпечує підготовку студентів з розроблення алгоритмів обробки даних та принципів побудови структур для збереження даних.

Метою курсу є оволодіння студентами основними принципами побудови структур для збереження даних і алгоритмами та методами обробки даних, які в подальшому можуть сприяти їх успішному застосуванню в професійній діяльності.

Предмет дисципліни. Принципи побудови структур для збереження даних; концепція абстрактних типів даних, основні структури, що реалізують абстрактні типи даних і їх представлення; методи та алгоритми обробки даних; оцінка складності алгоритмів.

Завдання дисципліни:

- навчити застосовувати сумісне збереження даних різних типів на прикладі компіляторів C++;
- розглянути основні можливості мов програмування C/C++ по організації складних структур даних;
- навчити використовувати базові алгоритми та структури даних для розв’язку прикладних задач;
- навчити застосовувати алгоритми обробки даних на всіх етапах життєвого циклу прикладної програмної системи, починаючи з аналізу вимог до програмної системи і її попереднього проектування, і закінчуючи її реалізацією, тестуванням і наступним супроводом.

В результаті вивчення курсу «Структури даних і алгоритми» студент повинен досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

знати:

- теоретичні положення, що лежать в основі функціонування алгоритмічного та програмного забезпечення; концепцію абстрактних типів даних та основні структури, що реалізують абстрактні типи даних; принципи побудови структур для збереження даних; базові алгоритми обробки даних; основи алгоритмізації та сучасної методології розробки алгоритмічного програмного забезпечення; методи оцінки складності алгоритмів;

вміти:

- здійснювати пошук, оброблення та аналіз інформації з різних джерел для розв’язання прикладних задач з використанням теорії алгоритмів та принципів побудови структур для збереження даних;
- застосовувати знання з теорії алгоритмів та основних принципів побудови структур для збереження даних при розв’язуванні задач аналізу та синтезу програмних систем;
- застосовувати знання з теорії алгоритмів для ідентифікації, формулювання і розв’язування прикладних задач, використовуючи методи побудови алгоритмів, алгоритми обробки даних, принципи побудови структур для збереження даних та абстрактні типи даних, що є найбільш придатними для досягнення поставлених цілей;
- поєднувати теорію і практику побудови структур для збереження даних та алгоритмів обробки даних, а також приймати оптимальні рішення при виробленні стратегії розв’язку прикладних задач з урахуванням виробничих інтересів;
- виконувати експериментальні дослідження з метою пошуку оптимальних алгоритмів розв’язку прикладних задач;
- системно мислити та застосовувати творчі здібності до формування нових ідей при побудові алгоритмів для розв’язування прикладних задач;
- оцінювати отримані результати та аргументовано захищати прийняті рішення при реалізації алгоритмів та виборі структур даних для розв’язку прикладних задач;

бути здатним:

- до абстрактного мислення, аналізу і синтезу при побудові алгоритмів для розв'язування прикладних задач;
- до навчання та оволодіння сучасними знаннями з метою вироблення стратегії побудови алгоритмів для розв'язку прикладних задач та застосовувати одержані знання у практичних ситуаціях;
- до розуміння теорії алгоритмів та принципів побудови структур для збереження даних;
- застосовувати знання з теорії алгоритмів, оцінки складності алгоритмів та принципи побудови структур для збереження даних у практичних ситуаціях;
- використовувати поняття теорії алгоритмів та концепції абстрактних типів даних, а також принципи побудови структур для збереження даних та оцінки складності алгоритмів для розроблення алгоритмічного та програмного забезпечення на сучасних мовах програмування;
- аргументувати вибір методів розв'язування алгоритмічних прикладних задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення;
- розв'язувати складні задачі під час розробки прикладного програмного забезпечення, що передбачає застосування понять теорії алгоритмів та принципи побудови структур для збереження даних і характеризується комплексністю та невизначеністю умов.

Програмні компетентності

Інтегральна компетентність. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов.

ЗК1. Здатність до абстрактного мислення, аналізу і синтезу

ЗК2. Здатність вчитися і оволодівати сучасними знаннями

ЗК3. Здатність застосовувати знання у практичних ситуаціях

ЗК7. Вміння виявляти, ставити та вирішувати проблеми

ЗК12. Здатність використовувати інформаційні та комунікаційні технології

ЗК13. Здатність розв'язувати поставлені задачі та приймати відповідні рішення

ЗК14. Здатність розробляти та управляти проектами, оцінювати та забезпечувати якість виконуваних робіт

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

ФК11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

ФК13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.

ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати та захищати прийняті рішення.

Спеціальні компетентності, визначені за освітньою програмою:

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН13. Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

Результати навчання, визначені за освітньою програмою:

ПРН23. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання програмно-технічних засобів комп'ютерних систем та мереж.

СТРУКТУРИ ДАНИХ І АЛГОРИТМИ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	Непарний
Кредити ЄКТС	8,0
Форми здобуття освіти	Очна денна

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен: знати теоретичні положення, що лежать в основі функціонування алгоритмічного та програмного забезпечення, концепцію абстрактних типів даних та основні структури, що реалізують абстрактні типи даних, принципи побудови структур для збереження даних, основи алгоритмізації та сучасної методології розробки алгоритмічного програмного забезпечення та методи оцінки складності алгоритмів; вміло застосовувати знання з теорії алгоритмів для ідентифікації, формулювання і розв'язування прикладних задач, використовуючи методи побудови алгоритмів, алгоритми обробки даних, принципи побудови структур для збереження даних та абстрактні типи даних, що є найбільш придатними для досягнення поставлених цілей; поєднувати теорію і практику побудови структур для збереження даних та алгоритмів обробки даних, реалізовувати абстрактні типи даних, дані з динамічною структурою, алгоритми для роботи з деревами, графами, внутрішнього та зовнішнього сортування, пошуку, управління пам'яттю, а також приймати оптимальні рішення при виробленні стратегії розв'язку прикладних задач; виконувати експериментальні дослідження з метою пошуку оптимальних шляхів розв'язку прикладних задач; використовувати поняття теорії алгоритмів та концепції абстрактних типів даних, а також принципи побудови структур для збереження даних та оцінки складності алгоритмів при розробленні алгоритмічного та програмного забезпечення на сучасних мовах програмування.

Зміст навчальної дисципліни. Основні поняття структур даних. Абстрактні типи даних. Статичні структури даних. Динамічні структури даних. Лінійні структури даних. Нелінійні структури даних. Асимптотичні оцінки складності алгоритмів. Класи складності. Параметри оцінки алгоритмів сортування. Класифікація алгоритмів сортування. Оцінка складності алгоритмів сортування. Найпростіші методи сортування. Пришвидшені методи сортування. Спеціальні методи сортування. Рекурсивні алгоритми. Принципи побудови рекурсивних алгоритмів. Етапи розроблення рекурсивних алгоритмів. Перехід від рекурсивної до ітеративної форми. Особливості роботи рекурсивних функцій. Доцільність застосування рекурсії в програмуванні. Аналіз трудомісткості рекурсивних алгоритмів. Методи пошуку. Хешування. Хеш-таблиці. Хеш-функції. Вимоги до хеш-функції. Методи вирішення проблеми колізій. Графи. Способи представлення графів. Методи обходу графів. Алгоритми на графах. Дерева. Способи представлення та обходу дерев. Методи розроблення алгоритмів. Метод грубої сили (повний перебір). Декомпозиція ("розділай і володарюй"). Зменшення розміру задачі ("зменшуй і володарюй"). Перетворення ("перетворюй і володарюй"). Жадібні алгоритми. Динамічне програмування. Пошук з поверненням. Локальний пошук.

Запланована навчальна діяльність: лекції - 34 год., лабораторні заняття - 34 год., практичні заняття - 34 год., самостійна робота - 138 год.; разом - 240 год.

Методи навчання: словесні, наочні, проблемно-пошукові (лекції); пояснювально-ілюстративні, практичні, частково-пошукові (практичні та лабораторні заняття), практичні, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми і методи оцінювання результатів навчання: захист лабораторних та практичних робіт, тестовий контроль, контрольна робота

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Шаховська Н.Б., Голошук Р.О. Алгоритми і структури даних: навч. посіб. Львів : Видавництво «Магнолія 2006», 2024. 215 с.
2. Бармак О.В., Манзюк Е.А., Радюк П.М. Теорія алгоритмів. Теоретичний курс та лабораторний практикум : навч. посіб. Хмельницький : ХНУ, 2023. 168 с.
3. Кублій, Л. І. Алгоритми та структури даних. Основи алгоритмізації [Електронний ресурс] : підручник для здобувачів ступеня бакалавра за спеціальністю 121 «Інженерія програмного забезпечення» / Л. І. Кублій ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2022. 528 с.
4. Mehlhorn, Kurt. Data structures and algorithms 2: graph algorithms and NP-completeness. Vol. 2. - Springer Science & Business Media, 2021. p. 542.
5. Sedgewick, Robert, and Kevin Wayne. Algorithms. Addison-wesley professional, 2021. p. 504.
6. Крєневич А.П. Алгоритми і структури даних. Підручник. Київ: ВПЦ "Київський Університет", 2021. 200 с.
7. Berztsiss, Alfs T. Data structures: theory and practice. Academic press, 2020. p. 682.

Викладач: док. філ., Мельниченко О.В.

1. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	Денна форма			
	Лекції	Лабораторні роботи	Практичні роботи	СРС
Тема 1. Базові концепції структур даних та алгоритмів.	12	4	4	16
Тема 2. Сортування та пошук.	12	16	14	64
Тема 3. Задачі на графах.	8	12	14	48
Тема 4. Класичні задачі.	2	-	2	10
Підсумкове заняття.	-	2	2	-
Разом за семестр:	34*	34*	34*	138

Примітка. * по чисельнику - 18 годин, по знаменнику - 16 годин (розрахунок здійснюється відповідно до розкладу занять)

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
1.	Тема 1. Базові концепції структур даних та алгоритмів. Лекція № 1. Основні поняття структур даних. Основні поняття структур даних. Абстрактні типи даних. Статичні структури даних. Динамічні структури даних. Лінійні структури даних. Нелінійні структури даних. Літ. [1, 2, 4, 7, 8]	4
2.	Тема 1. Базові концепції структур даних та алгоритмів. Лекція № 2. Складність алгоритмів. Асимптотичні оцінки складності. Класи складності. Клас P (задачі з поліноміальною складністю). Клас NP (поліноміальної перевірки). Клас NPC (NP - повні задачі). Інші класи складності. Літ. [1, 4, 6, 7, 8, 10]	4
3.	Тема 1. Базові концепції структур даних та алгоритмів. Лекція № 3. Рекурсивні алгоритми. Рекурсія. Поняття та основні визначення. Рекурентності. Принципи побудови рекурсивних алгоритмів. Етапи розроблення рекурсивних алгоритмів. Параметризація. База рекурсії. Декомпозиція. Умови завершення рекурсії. Перехід від рекурсивної до ітеративної форми. Особливості роботи рекурсивних функцій. Доцільність застосування рекурсії в програмуванні. Аналіз трудомісткості рекурсивних алгоритмів методом підрахунку вершин дерева рекурсії. Літ. [1, 2, 5]	4
4.	Тема 2. Сортування та пошук. Лекція № 4. Методи сортування. Параметри оцінки алгоритмів сортування. Класифікація алгоритмів сортування за сферою застосування. Стійке сортування. Нестійке сортування. Оцінка складності алгоритмів сортування. Найпростіші методи сортування. Пришвидшені методи сортування. Спеціальні методи сортування. Сортування обміном (бульбашкою). Сортування вибором. Сортування вставками. Сортування Шелла. Швидке сортування. Пірамідалне сортування. Сортування злиттям. Кишенькове сортування. Кишенькове сортування для повторюваних ключів. Порозрядне сортування. Прийоми при роботі з масивами. Літ. [1, 3, 8, 10]	4
5.	Тема 2. Сортування та пошук. Лекція № 5. Методи пошуку. Класифікація методів інформаційного пошуку. Лінійний пошук. Лінійний пошук з бар'єром. Пошук з перестановкою в початок. Пошук із транспозицією. Бінарний пошук. Рекурсивний та ітеративний підходи реалізації бінарного пошуку. Індексно - послідовний метод пошуку. Літ. [1, 3, 8, 10]	4
6.	Тема 2. Сортування та пошук. Лекція № 6. Хешування. Поняття хешування. Хеш-таблиці. Хеш-функції. Мультиплікативна хеш-функція для ключів з плаваючою точкою. Модульна хеш-функція для цілих ключів. Об'єднання мультиплікативного і модульного методів. Модульна хеш-функція для довгих ключів. Вирішення проблеми колізій за допомогою ланцюжків (відкрите хешування). Відкрита адресація (закрите хешування). Поширені способи переборів: лінійне зондування, квадратичне зондування, подвійне хешування. Якою повинна бути хороша хеш-функція. Ключі як натуральні числа. Ділення з остачею. Мультиплікативна схема хешування. Хешування рядків змінної довжини. Сфери застосування хешування. Літ. [1, 3, 8, 10]	4

7.	<p>Тема 3. Задачі на графах. Лекція № 7. Графи. Основні означення. Типи графів. Спеціальні види графів. Основні властивості графів. Способи представлення графів. Матриця інцидентності. Матриця суміжності. Списки суміжності. Перетворення матриці суміжності на список суміжності. Перерахування ребер. Перетворення перерахування ребер на список суміжності. Методи обходу графів. Пошук в глибину. Пошук в ширину. Пошук компонент зв'язності. Пошук циклів в неорієнтованому графі. Пошук циклів в орієнтованому графі. Перевірка графа на дводольність. Алгоритм Дейкстри для визначення найкоротших відстаней в зваженому графі. Алгоритм Флойда-Уоршелла для визначення найкоротших відстаней в зваженому графі. Літ. [1, 2, 4]</p>	4
8.	<p>Тема 3. Задачі на графах. Лекція № 8. Дерева. Впорядковані та неупорядковані дерева. Правила обходу дерев. Прямий, зворотний та симетричний обходи дерева. Абстрактний тип даних TREE. Подання дерев за допомогою масивів. Подання дерев з використанням списків синів. Двійкові дерева. Представлення двійкових дерев. Реалізація двійкових дерев за допомогою вказівників. Збалансоване дерево. Дерево бінарного пошуку. Префіксні дерева. Б-дерево. AVL-дерево. Червоно-чорне дерево. Розширюване дерево. Літ. [1, 2, 4]</p>	4
9.	<p>Тема 4. Класичні задачі. Лекція № 9. Методи розроблення алгоритмів. Метод грубої сили (повний перебір). Декомпозиція ("розділяй і володарюй"). Зменшення розміру задачі ("зменшуй і володарюй"). Перетворення ("перетворюй і володарюй"). Жадібні алгоритми. Динамічне програмування. Пошук з поверненням. Локальний пошук. Літ. [1, 3-5, 7, 10]</p>	2
Разом за семестр:		34

2.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
1.	<i>Лабораторна робота №1.</i> Алгоритми нестійкого сортування: швидке сортування, сортування Шелла.	4
2.	<i>Лабораторна робота №2.</i> Алгоритми стійкого сортування: сортування вставками, сортування злиттям, кишенькове сортування, порозрядне сортування.	4
3.	<i>Лабораторна робота №3.</i> Рекурсія.	4
4.	<i>Лабораторна робота №4.</i> Методи пошуку.	4
5.	<i>Лабораторна робота №5.</i> Хешування. Пошук з використанням хеш-функцій.	4
6.	<i>Лабораторна робота №6.</i> Неорієнтовані графи.	4
7.	<i>Лабораторна робота №7.</i> Орієнтовані графи.	4
8.	<i>Лабораторна робота №8.</i> Деревя.	4
9.	Підсумкове заняття.	2
<i>Разом за семестр:</i>		34*

Примітка. * по чисельнику - 18 годин, по знаменнику - 16 годин (розрахунок здійснюється відповідно до розкладу занять)

2.3 Зміст практичних занять

№ з/п	Тема практичного заняття	Кількість годин
1.	<i>Практична робота №1.</i> Сортування обміном (бульбашкою).	2
2.	<i>Практична робота №2.</i> Сортування вибором.	2
3.	<i>Практична робота №3.</i> Пірамідальне сортування.	2
4.	<i>Практична робота №4.</i> Сортування прямим злиттям.	2
5.	<i>Практична робота №5.</i> Зовнішнє сортування природним злиттям.	2
6.	<i>Практична робота №6.</i> Рекурсивні алгоритми.	2
7.	<i>Практична робота №7.</i> Використання Standard Template Library та алгоритмів стандартної бібліотеки C++ <algorithm> для розв'язування прикладних задач.	2
8.	<i>Практична робота №8.</i> Хешування. Хеш-таблиці.	2
9.	<i>Практична робота №9.</i> Хеш-таблиці. Невпорядковані асоціативні контейнери.	2
10.	<i>Практична робота №10.</i> Графи. Пошук в глибину.	2
11.	<i>Практична робота №11.</i> Графи. Обхід в ширину.	
12.	<i>Практична робота №12.</i> Топологічне сортування	2
13.	<i>Практична робота №13.</i> Пошук розділяючих вершин та мостів в графі.	
14.	<i>Практична робота №14.</i> Задачі на графах.	2
15.	<i>Практична робота №15.</i> Двійкове дерево пошуку. Способи обходу дерев.	
16.	<i>Практична робота №16.</i> Префіксні дерева.	2
17.	Підсумкове заняття.	2
Разом за семестр:		34

2.4 Зміст самостійної (індивідуальної) роботи

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту практичних та лабораторних робіт, тестування з теоретичного матеріалу, виконанні індивідуальних завдань тощо.

Номер тижня	Вид самостійної роботи	К-ть годин
1.	Опрацювання лекційного матеріалу. Підготовка до ЛР1.	8
2.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР1.	8
3.	Опрацювання лекційного матеріалу. Підготовка до ЛР2.	8
4.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР2.	8
5.	Опрацювання лекційного матеріалу. Підготовка до ЛР3.	8
6.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР3.	8
7.	Опрацювання лекційного матеріалу. Підготовка до ЛР4.	8
8.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР4.	8
9.	Опрацювання лекційного матеріалу. Підготовка до ЛР5.	8
10.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР5.	8
11.	Опрацювання лекційного матеріалу. Підготовка до ЛР6.	8
12.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР6.	8
13.	Опрацювання лекційного матеріалу. Підготовка до ЛР7.	8
14.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР7.	8
15.	Опрацювання лекційного матеріалу. Підготовка до ЛР8.	8
16.	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР8.	8
17.	Опрацювання лекційного матеріалу. Підготовка до підсумкового лабораторного заняття. Підготовка до тестування.	10
	<i>Разом за семестр:</i>	138

3. МЕТОДИ НАВЧАННЯ

Лекції проводяться, в основному, з використанням словесних, наочних, проблемно-пошукових методів; практичні та лабораторні заняття проводяться пояснювально-ілюстративними методами, практичними та частково-пошуковими методами; самостійна робота передбачає виконання індивідуальних завдань із залученням практичних, дослідницьких, частково-пошукових методів.

4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних, практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі заліку. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Процес оцінювання підготовленості студента можна розділити на етапи. Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань предметної області, вміє розширити їх. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) - особа має орієнтовне уявлення щодо понять, які вивчаються, здатна складати базові алгоритми для обробки даних та розробляти програми за цими алгоритмами.

Понятійно-аналітичний (ПА) - особа має чітке уявлення щодо навчального об'єкту, здатна перенести раніше засвоєні знання на типові ситуації.

Продуктивно-синтетичний (ПС) - особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєні знання на нетипові, нестандартні ситуації.

Кожний вид роботи з дисципліни оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (тестування), вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи - здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється під час проведення лабораторних занять; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

При оцінюванні знань студентів викладач керується такими критеріями.

Оцінка „зараховано” А виставляється студенту, який глибоко засвоїв методи розв'язування практичних задач та вміє їх раціонально застосувати. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „зараховано” В отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування у вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку „зараховано” С отримує студент за правильну відповідь з однією-двома суттєвими помилками.

Оцінки „зараховано” D заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як

правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок застосування методів розв'язування практичних задач, але припустився неточностей. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки „зараховано” Е заслуговує студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички застосування технологій розв'язування практичних задач.

Оцінка „незараховано” FX виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань.

Як правило, оцінка „незараховано” F виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

Кожний вид роботи оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота		Самостійна, індивідуальна робота		Семестровий контроль (залік)
Лабораторні роботи №:	Практичні роботи №:	Контрольна робота	Тестовий контроль Т 1-4	
1-8	1-16			
ВК: 0,7	0,1	0,1	0,1	0

Умовні позначення: Т - тема дисципліни; ВК - ваговий коефіцієнт.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1-11	12-14	15-18	19-20
Оцінка	2	3	4	5

Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в онлайн режимі в модульному середовищі MOODLE. Викладач виставляє результати тестування згідно журналу оцінок модульного середовища MOODLE.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у наступній таблиці.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС - буквене позначення оцінки, що відповідає набраній студентом кількості балів.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інституцій на шкала балів	Інституцій на оцінка	Критерії оцінювання	
			Зараховано	Незараховано
A	4,75-5,00	5	Зараховано	Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків.
B	4,25-4,74	4		Добре – повне знання навчального матеріалу з кількома незначними помилками.
C	3,75-4,24	4		Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками.
D	3,25-3,74	3		Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією.
E	3,00-3,24	3		Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00-2,99	2	Незараховано	Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2		Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни.

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ

1. Основні поняття структур даних.
2. Абстрактні типи даних.
3. Статичні структури даних.
4. Динамічні структури даних.
5. Лінійні структури даних.
6. Нелінійні структури даних.
7. Асимптотичні оцінки складності.
8. Класи складності.
9. Клас P (задачі з поліноміальною складністю).
10. Клас NP (поліноміальної перевірки).
11. Клас NPC (NP - повні задачі).
12. Параметри оцінки алгоритмів сортування.
13. Класифікація алгоритмів сортування за сферою застосування.
14. Стійке сортування. Нестійке сортування.
15. Оцінка складності алгоритмів сортування.
16. Найпростіші методи сортування.
17. Пришвидшені методи сортування.
18. Спеціальні методи сортування.
19. Сортування обміном (бульбашкою).
20. Сортування вибором.
21. Сортування вставками.
22. Сортування Шелла.
23. Швидке сортування.
24. Пірамідальне сортування.
25. Сортування злиттям.
26. Кишенькове сортування.
27. Кишенькове сортування для повторюваних ключів.
28. Порозрядне сортування.
29. Рекурсія. Поняття та основні визначення. Рекурентності.
30. Принципи побудови рекурсивних алгоритмів.

31. Етапи розроблення рекурсивних алгоритмів.
32. Умови завершення рекурсії.
33. Перехід від рекурсивної до ітеративної форми.
34. Особливості роботи рекурсивних функцій.
35. Доцільність застосування рекурсії в програмуванні.
36. Аналіз трудомісткості рекурсивних алгоритмів методом підрахунку вершин дерева рекурсії.
37. Класифікація методів інформаційного пошуку.
38. Лінійний пошук.
39. Лінійний пошук з бар'єром.
40. Пошук з перестановкою в початок.
41. Пошук із транспозицією.
42. Бінарний пошук.
43. Рекурсивний та ітеративний підходи реалізації бінарного пошуку.
44. Індексно - послідовний метод пошуку.
45. Поняття хешування.
46. Хеш-таблиці.
47. Хеш-функції.
48. Мультиплікативна хеш-функція для ключів з плаваючою точкою.
49. Модульна хеш-функція для цілих ключів.
50. Об'єднання мультиплікативного і модульного методів.
51. Модульна хеш-функція для довгих ключів.
52. Вирішення проблеми колізій за допомогою ланцюжків (відкрите хешування).
53. Відкрита адресація (закрите хешування).
54. Поширені способи переборів: лінійне зондування, квадратичне зондування, подвійне хешування.
55. Якою повинна бути хороша хеш-функція.
56. Ключі як натуральні числа.
57. Ділення з остачею.
58. Мультиплікативна схема хешування.
59. Хешування рядків змінної довжини.
60. Сфери застосування хешування.
61. Типи графів. Спеціальні види графів.
62. Основні властивості графів.
63. Матриця інцидентності.
64. Матриця суміжності.
65. Списки суміжності.
66. Перетворення матриці суміжності на список суміжності.
67. Перерахування ребер.
68. Перетворення перерахування ребер на список суміжності.
69. Методи обходу графів.
70. Пошук в глибину.
71. Пошук в ширину.
72. ошук компонент зв'язності.
73. Пошук циклів в неорієнтованому графі.
74. Пошук циклів в орієнтованому графі.
75. Перевірка графа на дводольність.
76. Топологічне сортування.
77. Алгоритм Дейкстри для визначення найкоротших відстаней в зваженому графі.
78. Алгоритм Флойда-Уоршелла для визначення найкоротших відстаней в зваженому графі.
79. Впорядковані та невпорядковані дерева.
80. Правила обходу дерев. Прямий, зворотний та симетричний обходи дерева.
81. Абстрактний тип даних TREE.

82. Подання дерев за допомогою масивів.
83. Подання дерев з використанням списків синів.
84. Двійкові дерева.
85. Представлення двійкових дерев.
86. Реалізація двійкових дерев за допомогою вказівників.
87. Збалансоване дерево.
88. Дерево бінарного пошуку.
89. Префіксні дерева.
90. Б-дерево.
91. AVL-дерево.
92. Червоно-чорне дерево.
93. Розширюване дерево.
94. Метод грубої сили (повний перебір).
95. Декомпозиція ("розділяй і володарюй").
96. Зменшення розміру задачі ("зменшуй і володарюй").
97. Перетворення ("перетворюй і володарюй").
98. Жадібні алгоритми.
99. Динамічне програмування.
100. Пошук з поверненням.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни «Структури даних і алгоритми» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані:

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Шаховська Н.Б., Голошук Р.О. Алгоритми і структури даних: навч. посіб. Львів : Видавництво «Магнолія 2006», 2024. 215 с.
2. Бармак О.В., Манзюк Е.А., Радюк П.М. Теорія алгоритмів. Теоретичний курс та лабораторний практикум : навч. посіб. Хмельницький : ХНУ, 2023. 168 с.
3. Кублій, Л. І. Алгоритми та структури даних. Основи алгоритмізації [Електронний ресурс] : підручник для здобувачів ступеня бакалавра за спеціальністю 121 «Інженерія програмного забезпечення» / Л. І. Кублій ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2022. 528 с.
4. Mehlhorn, Kurt. Data structures and algorithms 2: graph algorithms and NP-completeness. Vol. 2. - Springer Science & Business Media, 2021. p. 542.
5. Sedgewick, Robert, and Kevin Wayne. Algorithms. Addison-wesley professional, 2021. p. 504.
6. Кренивч А.П. Алгоритми і структури даних. Підручник. Київ: ВПЦ "Київський Університет", 2021. 200 с.
7. Berziss, Alfs T. Data structures: theory and practice. Academic press, 2020. p. 682.
8. Sanders, P., Mehlhorn, K., Dietzfelbinger, M., & Dementiev, R. Sequential and Parallel Algorithms and Data Structures. Springer, 2019. p. 302.
9. Storer, James Andrew. An introduction to data structures and algorithms. Springer Science & Business Media, 2019. p. 348.
10. Adamson, Iain T. Data structures and algorithms: a first course. -Springer Science & Business Media, 2019. p. 460.
11. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.

8. ІНФОРМАЦІЙНІ РЕСУРСИ

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Електронна бібліотека університету.