Чернівецький національний університет імені Юрія Федьковича
Факультет математики та інформатики
Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”
Декан факультету
математики та інформатики

_____ Ольга МАРТИНЮК

“ 28 ” серпня 2026 року

РОБОЧА ПРОГРАМА
навчальної дисципліни

Алгоритми і структури даних
(назва навчальної дисципліни)

Обов’язкова
(вказати: обов’язкова / вибіркова)

Освітньо-професійна програма Технології програмування та комп’ютерне моделювання
(назва програми)

Спеціальність F1 Прикладна математика
(вказати: код, назва)

Галузь знань F Інформаційні технології

Рівень вищої освіти перший (бакалаврський)
(вказати: перший бакалаврський/другий магістерський)

Факультет математики та інформатики
(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська
(вказати: на якій мові читається дисципліна)

Чернівці – 2026

Робоча програма навчальної дисципліни "*Алгоритми і структури даних"* **складена відповідно до** освітньо-професійної програми "Технології програмування та комп'ютерне моделювання" та навчального плану підготовки здобувачів вищої освіти за спеціальністю F1 Прикладна математика (галузь знань F Інформаційні технології). Дисципліна реалізує обов'язковий освітній компонент ОК18 "Алгоритми і структури даних" в обсязі 4 кредити ECTS (120 годин).

Розробник:

Краснокутська Інесса Володимирівна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Викладачі, що забезпечують читання даної навчальної дисципліни:

Краснокутська Інесса Володимирівна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Погоджено з гарантом ОП _____ **Василь МАЦЕНКО**
(підпис)

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій

Протокол № 2 від « 28 » серпня 2026 року

Завідувач кафедри _____ **Ярослав БІГУН**
(підпис)

Схвалено методичною радою факультету математики та інформатики

Протокол № 1 від « 28 » серпня 2026 року

Голова методичної ради _____ **Тоня ФРАТАВЧАН**
(підпис)

МЕТА ОСВІТНЬОГО КОМПОНЕНТА

Мета дисципліни:

• формування у студентів системного уявлення про основні алгоритми і структури даних, їх властивості, переваги та області застосування;

• оволодіння теоретичними засадами побудови та аналізу алгоритмів, оцінювання їх ефективності за часовими та просторовими характеристиками;

• набуття практичних навичок використання лінійних і нелінійних структур даних для розв'язування прикладних задач;

• розвиток алгоритмічного мислення та вміння обирати оптимальні структури даних і алгоритмічні підходи залежно від умов задачі.

Завдання дисципліни:

• вивчення базових структур даних, зокрема списків, списків з ітераторами, однозв'язних, двозв'язних та кільцевих списків;

• засвоєння принципів роботи та практичного застосування стеків і черг;

• вивчення деревоподібних структур даних і пірамід та основних операцій над ними;

• опанування алгоритмів сортування та пошуку, аналіз їх складності та ефективності;

• вивчення алгоритмічної стратегії «розділяй та володарюй» та її використання для побудови ефективних алгоритмів;

• формування навичок реалізації алгоритмів і структур даних мовами програмування з використанням сучасних підходів та інструментів.

Знання, уміння та навички, отримані в результаті вивчення дисципліни, є фундаментальними для підготовки фахівця у галузі інформаційних технологій, сприяють розвитку логічного та алгоритмічного мислення й становлять основу для подальшого вивчення складніших програмних та обчислювальних дисциплін.

ПРЕРЕКВІЗИТИ

Для успішного засвоєння дисципліни необхідні добрі знання та навички, що отримані під час вивчення ОК6 «Програмування» (класичне програмування мовами Python та C), ОК7 «Дискретна математика» та ОК24 «Математична логіка та теорія алгоритмів».

РЕЗУЛЬТАТИ НАВЧАННЯ

В результаті вивчення дисципліни студент має набути таких **компетентностей**:

знати:

• основні поняття теорії алгоритмів та структур даних;

• класифікацію та властивості алгоритмів, методи оцінювання їх часової та просторової складності;

• лінійні структури даних: списки, списки з ітераторами, однозв'язні, двозв'язні та кільцеві списки;
• принципи роботи стеків і черг та сфери їх застосування;
• деревоподібні структури даних та основні операції над ними;
• алгоритми сортування та пошуку, їх переваги, недоліки та області використання;
• алгоритмічну стратегію «розділяй та володарюй» та приклади її застосування;
• основні підходи до організації та зберігання даних у пам'яті комп'ютера.

вміти:
• аналізувати поставлені задачі та обирати адекватні алгоритми і структури даних для їх розв'язання;
• реалізовувати лінійні та нелінійні структури даних мовами програмування;
• застосовувати ітератори для обходу та модифікації структур даних;
• використовувати стеки та черги для розв'язування прикладних задач;
• будувати та опрацьовувати дерева, виконувати основні операції над ними;
• реалізовувати та порівнювати алгоритми сортування і пошуку;
• застосовувати підхід «розділяй та володарюй» при розробці ефективних алгоритмів;
• оцінювати ефективність алгоритмів і оптимізувати їх за часовими та просторовими характеристиками.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

ЗК01. Здатність учитися і оволодівати сучасними знаннями.

ЗК06. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК07. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК10. Навички у використанні інформаційних і комунікаційних технологій.

фахові компетентності:

ФК02. Здатність виконувати завдання, сформульовані у математичній формі.

ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.

ФК06. Здатність розв'язувати професійні задачі за допомогою комп'ютерної техніки, комп'ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків.

ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.

ФК09. Здатність до проведення математичного і комп'ютерного моделювання, аналізу та обробки даних, обчислювального експерименту, розв'язання формалізованих задач за допомогою спеціалізованих програмних засобів.

та отримуються наступні **програмні результати навчання**:

ПРН04. Виконувати математичний опис, аналіз та синтез дискретних об'єктів та систем, використовуючи поняття й методи дискретної математики та теорії алгоритмів.

ПРН09. Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.

ПРН10. Володіти методиками вибору раціональних методів та алгоритмів розв'язання математичних задач оптимізації, дослідження операцій, оптимального керування і прийняття рішень, аналізу даних.

ПРН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	3	4	120	30	-	-	30	60	-	іспит
Заочна	-	-	-	-	-	-	-	-	-	-	-

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі										
		л	п	лаб	інд	с.р.						
1	2	3	4	5	6	7						
Змістовий модуль 1. ***Динамічні структури даних та алгоритми їх обробки***												
Тема 1. Асимптотичний аналіз. Оцінка складності роботи алгоритму. Однозв'язні та двозв'язні списки	16	4		4		8						
Тема 2. Узагальнене програмування.	15	4		4		7						

Ітератори. Списки з ітераторами												
Тема 3. Стеки, черги та їх застосування. Обернена польська нотація	15	4		4		7						
Тема 4. Кільцеві списки. Кільцеві списки з ітераторами. Деки	14	3		3		8						
Разом за змістовим модулем 1	60	15		15		30						
Змістовий модуль 2. ***Складніші структури даних, алгоритми та методи розробки алгоритмів***												
Тема 5. Дерева. Бінарні дерева. Бінарні дерева пошуку. Піраміди. Черги з пріоритетами	20	6		4		10						
Тема 6. Алгоритми пошуку. Алгоритми сортування	20	4		6		10						
Тема 7. Метод декомпозиції (розділяй та володарюй)	20	5		5		10						
Разом за змістовим модулем 2	60	15		15		30						
Усього годин	120	30		30		60						

Тематика та питання лекційних занять

МОДУЛЬ 1. ДИНАМІЧНІ СТРУКТУРИ ДАНИХ ТА АЛГОРИТМИ ЇХ ОБРОБКИ	
Тема 1	Асимптотичний аналіз. Оцінка складності роботи алгоритму. Однозв'язні та двозв'язні списки 1. Поняття ефективності алгоритму: часова (time complexity) та просторова (space complexity) складність. 2. Асимптотичні нотації: O. Класи складності алгоритмів та їх порівняння: O(1), O(log n), O(n), O(n log n), O(n^2), O(2^n), O(n!). Аналіз найгіршого, середнього та найкращого випадків виконання алгоритму. 3. Поняття динамічної структури даних та порівняння зв'язних списків із масивами (виділення пам'яті, швидкість доступу та модифікації). 4. Структура вузла однозв'язного списку (дані та вказівник на наступний елемент). 5. Базові операції в однозв'язному списку: ініціалізація, додавання (на початок, у кінець, після заданого елемента), видалення, пошук та обхід.

	6. Особливості організації двозв'язного списку (наявність вказівників next та prev, використання вказівника на початок і кінець списку). 7. Алгоритми вставки та видалення у двозв'язному списку; переваги та недоліки порівняно з однозв'язним.
Тема 2	Узагальнене програмування. Ітератори. Списки з ітераторами 1. Концепція узагальненого програмування. 2. Поняття ітератора як абстракції для послідовного доступу до елементів структури даних без розкриття її внутрішньої реалізації. 3. Категорії ітераторів (однонаправлені, двонаправлені, довільного доступу) та базовий інтерфейс (розіменування, інкремент/декремент, порівняння). 4. Реалізація власного ітератора для зв'язного списку на основі структур даних. 5. Проблема інвалідації ітераторів під час модифікації списку.
Тема 3	Стеки, черги та їх застосування. Обернена польська нотація. 1. Абстрактний тип даних (АТД) Стек (принцип LIFO): базові операції (push, pop, isEmpty). 2. Варіанти реалізації стека: на базі статичного/динамічного масиву та на базі однозв'язного списку. 3. Практичне застосування стеків (стек викликів функцій, перевірка збалансованості дужок, скасування дій Undo). 4. Обернена польська нотація (ОПН / постфіксний запис): алгоритм переведення інфіксного виразу в постфіксний (алгоритм сортувальної станції Дейкстри) та обчислення значення виразу за допомогою стека. 5. АТД Черга (принцип FIFO): операції (enqueue, dequeue), реалізація на базі списку та кільцевого буфера (масиву).
Тема 4	Кільцеві списки. Кільцеві списки з ітераторами. Деки 1. Організація однозв'язних та двозв'язних кільцевих списків (замикання останнього елемента на перший). 2. Особливості обходу кільцевого списку та умови зупинки циклу; задачі, що ефективно вирішуються кільцевими списками (наприклад, задача Йосифа Флавія, циклічне планування процесів). 3. Побудова ітератора для кільцевого списку та уникнення нескінченних циклів. 4. Двостороння черга (Дек / Deque): визначення, основні операції додавання та видалення з обох кінців. 5. Варіанти реалізації дека (на базі двозв'язного списку та на базі кільцевого масиву).
МОДУЛЬ 2. СКЛАДНІШІ СТРУКТУРИ ДАНИХ, АЛГОРИТМИ ТА МЕТОДИ РОЗРОБКИ АЛГОРИТМІВ	
Тема 5	Дерева. Бінарні дерева. Бінарні дерева пошуку. Піраміди. Черги з пріоритетами 1. Основні терміни теорії дерев: корінь, вузол, лист, предок, нащадок, глибина, висота дерева, степінь вузла. Бінарні дерева та їх види (повне, ідеально збалансоване, вироджене); способи подання в пам'яті. 2. Алгоритми обходу бінарних дерев: у глибину (DFS: прямий / pre-order, симетричний / in-order, зворотний / post-order) та в ширину (BFS / по рівнях). 3. Бінарні дерева пошуку (BST): властивість впорядкованості, алгоритми пошуку, вставки та видалення вузла (три випадки видалення). Проблема виродження BST у список та огляд концепції збалансованих дерев (АВЛ-дерева, червоно-чорні дерева). 4. Поняття бінарної купи (піраміди): властивість форми (майже повне бінарне дерево) та властивість купи (min-heap та max-heap). Компактне представлення бінарної купи в одновимірному масиві (формули обчислення індексів батька та нащадків). 5. Базові операції підтримки властивості купи: просіювання вгору (siftUp) та просіювання вниз (siftDown). Побудова купи з невпорядкованого масиву за лінійний час O(n) та алгоритм пірамідального сортування (Heap Sort). АТД Черга з пріоритетами (Priority Queue): реалізація на основі бінарної купи та приклади застосування (планування задач, алгоритми на графах).

Тема 6	Алгоритми пошуку. Алгоритми сортування 1. Постановка задачі пошуку; поняття ключа та критеріїв ефективності. Лінійний (послідовний) пошук. Бінарний (двійковий) пошук у відсортованому масиві: ітеративна та рекурсивна реалізації. 2. Класифікація алгоритмів сортування: внутрішні та зовнішні, стійкі (stable) та нестійкі, сортування на місці (in-place). Прості квадратичні сортування ($O(n^2)$): сортування бульбашкою (Bubble Sort), вибором (Selection Sort) та вставками (Insertion Sort) — порівняння кількості порівнянь та обмінів. 3. Ефективні алгоритми сортування ($O(n \log n)$): швидке сортування (Quick Sort — вибір опорного елемента, розбиття Ломуто та Хоара) та сортування злиттям (Merge Sort). Пірамідальне сортування (Heap Sort). 4. Сортування без порівнянь за лінійний час ($O(n)$): сортування підрахунком (Counting Sort) та порозрядне сортування (Radix Sort).
Тема 7	Метод декомпозиції (розділяй та володарюй) 1. Сутність парадигми «розділяй та володарюй» (Divide and Conquer): три етапи (розбиття, рекурсивне розв'язання, об'єднання результатів). 2. Зв'язок методу з рекурсією та рекурентними рівняннями складності. 3. Класичні приклади застосування: бінарний пошук, сортування злиттям та швидке сортування. 4. Застосування декомпозиції до математичних та геометричних задач (алгоритм Карацуби для швидкого множення чисел, пошук пари найближчих точок на площині). 5. Оцінка складності циклів та рекурсивних алгоритмів (основна теорема рекурентних співвідношень / Master Theorem).

Тематика та завдання семінарських занять

Семінарські заняття з даної навчальної дисципліни навчальним планом не передбачені.

Тематика та завдання практичних занять

Практичні заняття з даної навчальної дисципліни навчальним планом не передбачені.

Тематика та завдання лабораторних занять

МОДУЛЬ 1. ДИНАМІЧНІ СТРУКТУРИ ДАНИХ ТА АЛГОРИТМИ ЇХ ОБРОБКИ	
ЛР 1	Однозв’язні та двозв’язні списки 1. Проєктування структури вузла та реалізація базових операцій вставки елементів (на початок, у кінець, за індексом). 2. Реалізація операцій видалення вузлів та пошуку елементів за значенням. 3. Порівняння ефективності виконання операцій вставки та видалення для однозв’язного і двозв’язного списків.
ЛР 2	Списки з ітераторами 1. Реалізація списків з використанням структур та ітераторів. 2. Розробка власного ітератора з підтримкою базових операцій навігації (перехід до наступного/попереднього елемента, розіменування, порівняння).

	3. Використання ітераторів для обходу, вставки та видалення елементів списку.
ЛР 3	Стеки, черги. Обернена польська нотація. 1. Реалізація абстрактного типу даних Стек (LIFO) на базі динамічних структур даних. 2. Реалізація абстрактного типу даних Черга (FIFO) на базі динамічних структур даних. 3. Обчислення значення виразу, записаного в оберненій польській нотації, із використанням стека.
ЛР 4	Кільцеві списки. Кільцеві списки з ітераторами 1. Організація замкненої структури однозв'язного або двозв'язного кільцевого списку та базових операцій модифікації. 2. Реалізація ітератора для кільцевого списку з контролем умови повного обходу без зациклення. 3. Розв'язання прикладної задачі циклічного перебору елементів (наприклад, задачі Йосифа Флавія).
МОДУЛЬ 2. СКЛАДНІШІ СТРУКТУРИ ДАНИХ, АЛГОРИТМИ ТА МЕТОДИ РОЗРОБКИ АЛГОРИТМІВ	
ЛР 5	Дерева. Бінарні дерева. Бінарні дерева пошуку 1. Побудова структури бінарного дерева пошуку (BST) та реалізація операцій додавання і пошуку елементів. 2. Реалізація алгоритмів обходу дерева в глибину (прямий, симетричний, зворотний) та в ширину. 3. Реалізація алгоритму видалення вузла з дерева пошуку (для випадків відсутності, одного або двох нащадків).
ЛР 6	Алгоритми сортування 1. Програмна реалізація простих квадратичних алгоритмів сортування (бульбашкою, вибором або вставками) та швидких алгоритмів (Quick Sort або Merge Sort). 2. Експериментальне вимірювання часу виконання та підрахунок кількості порівнянь і обмінів на різних наборах даних (випадкових, відсортованих, зворотно відсортованих). 3. Порівняльний аналіз отриманих практичних результатів із теоретичними асимптотичними оцінками складності.
ЛР 7	Метод декомпозиції (розділяй та володарюй) 1. Розробка рекурсивного алгоритму розв'язання задачі шляхом розбиття вхідних даних на незалежні підзадачі (наприклад, сортування злиттям, швидкий пошук k-ї порядкової статистики або пошук пари найближчих точок). 2. Реалізація етапу об'єднання (злиття) результатів розв'язання підзадач. 3. Оцінка глибини рекурсії та швидкодії розробленого алгоритму порівняно з ітеративним підходом.

Індивідуальні науково-дослідні завдання (ІНДЗ)

ІНДЗ з даної навчальної дисципліни не передбачені.

Завдання для самостійної роботи студентів

Назва теми та завдання самостійної роботи	Кіль-кість годин
Тема 1. Асимптотичний аналіз. Оцінка складності роботи алгоритму. Однозв'язні та двозв'язні списки 1. Амортизований аналіз складності алгоритмів та його практичне застосування. 2. Особливості організації циклічних та багаторівневих зв'язних списків. 3. Використання сторожових вузлів (sentinel nodes) для спрощення операцій зі зв'язними списками. 4. Застосування зв'язних списків для реалізації інших структур даних.	8
Тема 2. Узагальнене програмування. Ітератори. Списки з ітераторами 1. Константні та зворотні ітератори: призначення та особливості використання. 2. Реалізація ітераторів для різних типів структур даних на основі структур. 3. Використання ітераторів для виконання операцій пошуку, фільтрації та модифікації елементів списку.	7
Тема 3. Стеки, черги та їх застосування. Обернена польська нотація 1. Використання стеків для реалізації алгоритму пошуку в глибину (DFS). 2. Застосування черг для організації обробки подій та моделювання систем масового обслуговування. 3. Декілька стеків в одній області пам'яті: принципи організації та особливості використання. 4. Алгоритми обробки виразів із дужками та функціями (sin, cos, log) за допомогою стеків.	7
Тема 4. Кільцеві списки. Кільцеві списки з ітераторами. Деки 1. Застосування кільцевих списків у мультимедійних системах для організації циклічного відтворення даних. 2. Алгоритми розв'язання задачі Йосифа Флавія різними способами та порівняння їх ефективності. 3. Реалізація циклічного буфера для обробки потоків даних. 4. Монотонні дека та їх використання для пошуку мінімуму й максимуму у ковзному вікні.	8
Тема 5. Дерева. Бінарні дерева. Бінарні дерева пошуку. Піраміди. Черги з пріоритетами 1. AVL-дерева: фактор балансу, повороти та підтримка збалансованості. 2. Червоно-чорні дерева: основні властивості та принципи балансування. 3. B-дерева та B+ дерева: особливості організації та застосування в базах даних. 4. Біноміальні та Фібоначчієві купи: принципи організації та порівняння з бінарними купами. 5. Використання дерев для представлення ієрархічних даних та організації файлових систем.	10
Тема 6. Алгоритми пошуку. Алгоритми сортування 1. Інтерполяційний та експоненціальний пошук: принципи роботи й оцінка ефективності. 2. Алгоритми пошуку підрядка в тексті: Кнута – Морріса – Пратта та Бойєра –	10

Мура. 3. Зовнішнє сортування: принципи організації обробки великих обсягів даних. 4. Сортування Шелла: особливості алгоритму та порівняння з іншими методами сортування. 5. Паралельні алгоритми сортування та особливості їх застосування.	
Тема 7. Метод декомпозиції (розділяй та володарюй) 1. Розв'язання задач методом декомпозиції з використанням рекурсивних алгоритмів. 2. Метод швидкого перетворення Фур'є (FFT) та його застосування. 3. Алгоритм Штрассена для множення матриць. 4. Порівняння методів «розділяй та володарюй» і динамічного програмування. 5. Застосування методу декомпозиції для обробки великих масивів даних.	10
Разом	60

Самостійна робота студента полягає в опрацюванні лекційного матеріалу, більш детальному розгляді окремих питань курсу, підготовці до виконання та захисту лабораторних робіт.

Самостійна робота студента загалом складає 60 годин. Розподіл цих годин за видами робіт: опрацювання лекційного матеріалу, більш детальний розгляд окремих питань курсу – 30 годин; підготовка до виконання та захисту лабораторних робіт – 30 годин.

Оцінювання самостійної роботи студента є складовою частиною оцінювання захисту лабораторних робіт (30% оцінки кожної лабораторної роботи складає оцінювання самостійної роботи).

МЕТОДИ НАВЧАННЯ

Під час викладання навчальної дисципліни «Алгоритми і структури даних» застосовується комплексний підхід до вибору методів та освітніх технологій, спрямований на формування як ґрунтовного теоретичного фундаменту з алгоритмізації, так і стійких практичних навичок розробки, відлагодження та тестування програмного забезпечення мовою C++.

Освітній процес поєднує такі класичні та сучасні методи й освітні технології:

1. **За джерелами передачі та сприйняття інформації:**
   - *Словесні методи:* розповідь, лекція-пояснення, навчальний діалог та лекція-дискусія для розкриття теоретичних основ алгоритмів та структур даних.
   - *Наочні класичні та цифрові методи:* метод мультимедійних презентацій, демонстрація розбору та виконання програмного коду в режимі реального часу в інтегрованому середовищі розробки (IDE), візуалізація алгоритмів (трасування стеку викликів, схеми розподілу динамічної пам'яті).
   - *Практичні методи:* розв'язання алгоритмічних задач та виконання лабораторного практикуму на ПК з індивідуальним захистом робіт.
2. **За характером пізнавальної діяльності здобувачів освіти:**
   - *Пояснювально-ілюстративний та репродуктивний методи:* засвоєння принципів організації динамічних структур даних (однозв'язних, двозв'язних і кільцевих списків, стеків, черг), узагальненого програмування та ітераторів; відтворення класичних алгоритмів пошуку, сортування (бульбашкою, вибором, вставками), а також базових операцій на бінарних деревах пошуку та пірамідах за поданими зразками.

- *Частково-пошуковий та проблемний методи:* розв’язання алгоритмічних задач підвищеної складності, аналіз та порівняння альтернативних рішень (зокрема ітераційних та рекурсивних підходів), самостійний пошук та усунення помилок у програмному коді (дебагінг).
- *Дослідницький метод:* аналіз ефективності алгоритмів за часом виконання та використанням пам’яті, проектування складних структур даних, побудова алгоритмів.

3. За формою організації та практичної спрямованості (основний блок):

- *Практико-орієнтований лабораторний практикум:* проведення лабораторних занять у комп’ютерних класах із персональним виконанням та захистом лабораторних робіт.
- *Алгоритмічний тренінг та розв'язання задач:* систематична робота на лабораторних заняттях із подальшим аналізом і рев'ю коду (code review).
- *Інформаційно-комунікаційні технології та електронне навчання:* використання електронного навчального курсу в LMS Moodle ЧНУ для доступу до лекційних матеріалів та подання звітів із скріншотами виконання програм.
- *Самонавчання та самостійна робота:* опрацювання рекомендованої літератури, виконання завдань для самоконтролю за методичними вказівками викладача та підготовка до контрольних робіт.

СИСТЕМА КОНТРОЛЮ ТА ОЦІНЮВАННЯ

Методами та формами поточного контролю є:

- усні поточні опитування;
- захист виконаних лабораторних робіт;
- контрольні роботи.

Формою підсумкового контролю є усно-письмовий іспит.

Політика освітнього процесу

Політика щодо відвідування. Відвідування занять є обов’язковим. За наявності поважних причин (хвороба, індивідуальний графік навчання) навчання, за погодженням із викладачем, може відбуватись у змішаній (очно-дистанційній) формі. За необхідності, з метою з’ясування всіх незрозумілих під час самостійної роботи питань, потрібно відвідувати консультації викладача.

Політика щодо академічної доброчесності. Студент зобов’язаний самостійно, своєчасно та добросовісно виконувати усі отримані завдання, дотримуючись принципів академічної доброчесності. Під час проведення контрольних заходів заборонені використання сторонньої допомоги, навчальної літератури, конспектів, мобільних пристроїв, інших джерел інформації.

Політика щодо дедлайнів та перескладання. Якщо студент був відсутній на заняттях (з будь-якої причини), він повинен самостійно опрацювати пропущений матеріал та прозвітувати викладачу про виконання відповідних завдань у встановлені ним терміни під час консультацій. Якщо студент з поважної причини пропустив захист лабораторної роботи, він може з дозволу викладача захистити її під час консультації. Відсутність студента на захисті лабораторної роботи без поважної причини відповідає оцінці «0 балів», тобто призводить до заборгованості, яка повинна бути ліквідована студентом до початку підсумкового контролю з наступного модуля. Кінцевий термін ліквідації заборгованості з модульного контролю обмежується початком

заліково-екзаменаційної сесії. Лабораторні роботи, які захищаються невчасно без поважних причин, оцінюються на нижчу оцінку (від -10% до -50% від максимальної кількості балів – залежно від терміну затримки здачі). Перезахист лабораторних робіт не дозволяється. Складання (перескладання) іспиту відбувається за встановленим деканатом розкладом.

КРИТЕРІЇ ОЦІНЮВАННЯ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ

Оцінювання знань студентів здійснюється на основі результатів поточного, модульного та підсумкового контролю. Оцінювання здійснюється за програмним матеріалом навчальної дисципліни, засвоєння якого перевіряється пропонованими видами контролю.

Форми проведення поточного контролю та періодичність їх здійснення визначаються робочим планом викладача відповідно до програми дисципліни. Поточний контроль здійснюється під час проведення лабораторних занять і перевірки самостійної роботи студентів, а також під час читання лекцій. Модульний контроль здійснюється за результатами виконаних модульних контрольних робіт та перевірки лабораторних робіт. Завданнями поточного та модульного контролю є перевірка рівня розуміння та засвоєння лекційного матеріалу, набуття практичних навичок і досвіду виконання індивідуальних і комплексних задач.

Форми підсумкового семестрового контролю визначаються навчальним планом спеціальності F1 «Прикладна математика». Для дисципліни «Алгоритми і структури даних» встановлено семестровий іспит по завершенню вивчення курсу.

Завданням підсумкового контролю (іспиту) є перевірка розуміння студентом програмного матеріалу в цілому, здатності успішно розв’язувати поставлені практичні задачі та комплексно використовувати отримані знання.

Оцінювання знань здійснюється за 100-бальною шкалою. Результати роботи впродовж навчального семестру оцінюються в ході поточного та модульного контролю на інтервалі оцінок від 0 до 60 балів, а результати підсумкового контролю (іспиту) оцінюються максимум у 40 балів. Загальна підсумкова оцінка з навчальної дисципліни виставляється за загальною сумою балів поточного та модульного контролю і підсумкового контролю.

Нормативно-процесуальні умови оцінювання, допуску та апеляції

Оцінювання знань здобувачів вищої освіти здійснюється відповідно до «Положення про організацію освітнього процесу в Чернівецькому національному університеті імені Юрія Федьковича» та «Положення про контроль і систему оцінювання результатів навчання здобувачів вищої освіти у Чернівецькому національному університеті імені Юрія Федьковича».

Умови допуску до підсумкового семестрового контролю (іспиту):

- Обов'язкове виконання, подання та успішний захист усіх 7 лабораторних робіт, передбачених робочою програмою;
- Написання двох контрольних робіт;
- Набрання за результатами поточного оцінювання не менше 35 балів із 60 можливих.

Політика ліквідації заборгованостей та апеляції:

- Здобувачі освіти, які не були допущені до іспиту або отримали під час семестрового контролю незадовільну оцінку FX (35–49 балів), мають право на ліквідацію академічної

заборгованості за затвердженим графіком у визначені терміни до початку наступного навчального семестру.

- Перескладання іспиту дозволяється не більше двох разів: один раз — викладачу, вдруге – комісії, створеній деканом факультету. Перескладання семестрового контролю з метою підвищення позитивної оцінки не допускається.
- Здобувачі, які отримали оцінку F (1-34 бали), зобов'язані пройти повторний курс вивчення дисципліни.
- У разі незгоди з отриманою на семестровому іспиті оцінкою здобувач вищої освіти має право подати письмову апеляційну заяву на ім'я декана факультету (або ректора Університету) не пізніше 12:00 наступного робочого дня після оголошення результату відповідно до Положення про апеляцію на результати підсумкового семестрового контролю ЧНУ.

Розподіл балів, які отримують студенти

Поточне оцінювання та самостійна робота									Іспит	Сума
Змістовий модуль №1					Змістовий модуль №2					
ЛР 1	ЛР 2	ЛР 3	ЛР 4	КР	ЛР 5	ЛР 6	ЛР 7	КР	40	100
5	5	5	5	10	5	5	5	15		

Іспит (40 балів) складається з перевірки теоретичного матеріалу на 20 балів та комплексного практичного завдання на 20 балів. Білет складається з 3 питань: 2 теоретичних (оцінюються максимум по 10 балів кожне) і 1 практичного (оцінюється максимально в 20 балів). За суттєві помилки в викладенні теоретичного завдання знімається 5-9 балів. За несуттєві помилки в викладенні теоретичного завдання знімається 1-4 бали. Часткове викладення теоретичного завдання із грубими помилками оцінюється не вище 5 балів. За суттєві помилки в розв’язуванні практичного завдання знімається 10-18 балів. За несуттєві помилки в розв’язуванні практичного завдання знімається 2-8 балів. Часткове розв’язання практичного завдання із грубими помилками оцінюється не вище 8 балів.

Загальна підсумкова оцінка з навчальної дисципліни виставляється за загальною сумою балів, набраних студентом, згідно з наступною таблицею:

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	**Оцінка за шкалою ECTS**	
	Оцінка (бали)	**Пояснення за розширеною шкалою**
відмінно	A (90-100)	відмінно
добре	B (80-89)	дуже добре
	C (70-79)	добре
задовільно	D (60-69)	задовільно
	E (50-59)	достатньо
незадовільно з можливістю повторного складання	FX (35-49)	(незадовільно) з можливістю повторного складання

незадовільно з обов'язковим самостійним опрацюванням освітнього компонента до перескладання	F (1-34)	(незадовільно) з обов'язковим самостійним опрацюванням освітнього компонента до перескладання

ПЕРЕЛІК ПИТАНЬ ДЛЯ САМОКОНТРОЛЮ ТА ПІДСУМКОВОГО КОНТРОЛЮ НАВЧАЛЬНИХ ДОСЯГНЕНЬ СТУДЕНТІВ

1. Що таке алгоритм та які основні вимоги висуваються до алгоритмів?
2. Які існують способи оцінювання ефективності алгоритмів?
3. У чому полягає різниця між часовою та просторовою складністю алгоритму? Що означає асимптотичне позначення O?
4. Як визначити складність алгоритму залежно від кількості виконуваних операцій?
5. У чому полягає різниця між найкращим, середнім та найгіршим випадками виконання алгоритму?
6. Що таке амортизований аналіз складності алгоритмів?
7. Які переваги та недоліки динамічних структур даних порівняно зі статичними?
8. У чому полягають основні відмінності між масивами та зв'язними списками?
9. Які особливості організації однозв'язних списків?
10. Яку роль відіграють вказівники у зв'язних списках?
11. Які операції можна виконувати над однозв'язним списком?
12. Як реалізувати вставку та видалення елемента в однозв'язному списку?
13. Які особливості організації двозв'язного списку?
14. Які переваги та недоліки двозв'язних списків порівняно з однозв'язними?
15. Що таке ітератор та для чого він використовується? Які основні види ітераторів існують?
16. Які операції можна виконувати за допомогою ітераторів? Як організувати послідовний доступ до елементів зв'язного списку? Що таке інвалідація ітераторів та за яких умов вона виникає?
17. Що таке абстрактний тип даних та які його основні характеристики?
18. У чому полягає принцип LIFO та для якої структури даних він характерний?
19. Які основні операції виконуються над стеком?
20. Які існують способи реалізації стеків?
21. Які особливості використання стеків для перевірки правильності дужкових виразів?
22. Як стек використовується під час рекурсивного виклику функцій?
23. Що таке обернена польська нотація та які її переваги?
24. У чому полягає алгоритм сортувальної станції Дейкстри?
25. Як здійснюється обчислення арифметичних виразів у постфіксній формі?
26. У чому полягає принцип FIFO та для якої структури даних він використовується?
27. Які основні операції виконуються над чергою?
28. Які існують способи реалізації черг?
29. Що таке кільцевий буфер та де він застосовується?

30. Які особливості організації кільцевих зв’язних списків?
31. Чим відрізняються однозв’язні та двозв’язні кільцеві списки?
32. Які особливості обходу кільцевих списків? Як уникнути нескінченного циклу під час обходу кільцевого списку?
33. У чому полягає задача Йосифа Флавія та як її можна розв’язати за допомогою кільцевого списку?
34. Що таке дек та які операції можна виконувати над цією структурою? Чим дек відрізняється від звичайної черги? Які способи реалізації дека існують?
35. Що таке дерево та які основні поняття використовуються для його опису?
36. Які існують види бінарних дерев? Які способи представлення дерев у пам’яті комп’ютера?
37. Які існують алгоритми обходу бінарних дерев?
38. У чому полягає різниця між обходом дерева в глибину та в ширину?
39. Що таке бінарне дерево пошуку та які його основні властивості? Як виконуються операції пошуку, вставки та видалення вузлів у бінарному дереві пошуку?
40. Які випадки видалення вузлів існують у бінарному дереві пошуку? Що таке вироджене бінарне дерево та як воно впливає на ефективність пошуку?
41. Що таке AVL-дерево та для чого використовується балансування?
42. Які основні властивості червоно-чорних дерев?
43. Що таке B-дерева та B+ дерева і де вони застосовуються?
44. Що таке бінарна купа та які її основні властивості? У чому полягає різниця між мінімальною та максимальною купою?
45. Як здійснюється представлення бінарної купи за допомогою одновимірного масиву? У чому полягають операції просіювання вгору та просіювання вниз?
46. Як здійснюється побудова бінарної купи з невпорядкованого масиву? Що таке пірамідальне сортування та яка його часова складність?
47. Що таке черга з пріоритетами та які її основні операції?
48. Які переваги використання бінарної купи для реалізації черги з пріоритетами?
49. У чому полягає задача пошуку та які критерії визначають ефективність алгоритмів пошуку?
50. Як працює алгоритм лінійного пошуку?
51. У чому полягає принцип бінарного пошуку та які умови необхідні для його застосування?
52. Яка різниця між ітеративною та рекурсивною реалізацією бінарного пошуку?
53. Як класифікуються алгоритми сортування? Що таке стійкість алгоритму сортування?
54. У чому полягає різниця між сортуванням на місці та сортуванням із використанням додаткової пам’яті?
55. Як працює алгоритм сортування бульбашкою? У чому полягають особливості сортування вибором? Як працює алгоритм сортування вставками?
56. Які переваги та недоліки простих алгоритмів сортування?
57. Як працює алгоритм швидкого сортування? Яку роль відіграє опорний елемент у швидкому сортуванні?
58. У чому полягає різниця між схемами розбиття Ломуто та Хоара?
59. Як працює алгоритм сортування злиттям?
60. Які переваги та недоліки сортування злиттям порівняно зі швидким сортуванням?
61. У чому полягають особливості сортування Шелла?

62. Як працює алгоритм сортування підрахунком?
63. У чому полягає принцип порозрядного сортування?
64. Які обмеження мають алгоритми сортування без порівнянь?
65. Що таке зовнішнє сортування та в яких випадках воно використовується?
66. У чому полягає метод декомпозиції «розділяй та володарюй»? Які основні етапи реалізації алгоритмів методом декомпозиції?
67. Який зв'язок існує між рекурсією та методом «розділяй та володарюй»? Що таке рекурентне співвідношення та для чого воно використовується?
68. Як застосовується метод декомпозиції в алгоритмі бінарного пошуку? Які особливості використання декомпозиції в алгоритмах швидкого сортування та сортування злиттям?
69. У чому полягає алгоритм Карацуби та які його переваги?
70. Як застосовується метод декомпозиції для пошуку пари найближчих точок на площині? Які особливості алгоритму Штрассена для множення матриць?

ЗАРАХУВАННЯ РЕЗУЛЬТАТІВ НЕФОРМАЛЬНОЇ ОСВІТИ

Порядок та умови визнання й зарахування результатів навчання, здобутих шляхом неформальної та/або інформальної освіти, регламентуються чинним у Чернівецькому національному університеті імені Юрія Федьковича «Порядком визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти».

Здобувачі вищої освіти мають право на визнання результатів навчання, отриманих під час проходження онлайн-курсів (на платформах Coursera, edX, Prometheus, Udemy, Stepik, Pluralsight тощо), участі у воркшопах, хакатонах, літніх/зимових школах з програмування за наявності підтверджуючих документів (сертифікатів, свідоцтв).

Механізм та умови визнання результатів:

1. **Перезарахування окремих тем / лабораторних робіт / поточного контролю:**
   - У разі часткової відповідності змісту онлайн-курсу або сертифікованої програми тематиці окремих змістових модулів дисципліни «Алгоритми і структури даних», здобувач звертається в усній формі до викладача, подаючи підтверджуючі документи.
   - За умови підтвердження відповідності результатів навчання здобувачу зараховується максимальний бал з відповідної теми (лабораторної роботи) та звільняється від виконання відповідних завдань. Загальний обсяг балів, який може бути зарахований здобувачу за результатами неформальної/інформальної освіти з окремих тем, не може перевищувати 25% від загальної кількості балів з дисципліни (до 25 балів зі 100).
2. **Повне зарахування освітнього компонента (підсумкового контролю):**
   - У разі необхідності повного зарахування навчальної дисципліни здобувач подає письмову заяву на ім'я декана факультету, до якої додає освітню декларацію про попереднє навчання та сертифікати.
   - Розгляд документів, співставлення результатів навчання із програмою дисципліни та оцінювання здійснює Предметна комісія, призначена розпорядженням декана.
3. **Процедура підтвердження та оцінювання:**
   - У разі прийняття Предметною комісією рішення про проведення контрольного заходу (співбесіди, контрольного завдання), здобувачу надається до 10 робочих днів для підготовки за питаннями і завданнями, що виносяться на контроль.
   - Загальний обсяг визнаних результатів неформальної освіти за весь строк навчання за ОПП «Технології програмування та комп'ютерне моделювання» (галузь знань F

«Інформаційні технології») не може перевищувати 35% від загального обсягу освітньої програми.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Краснокутська І.В. Алгоритми та структури даних (лабораторний практикум). – Чернівці: Чернівецький національний ун-т, 2017. – 48 с.
2. Караванова Т. П. Теорія алгоритмів. Частина 1. Необчислювальні алгоритми : навч. посіб. Чернівці : Чернівецький нац. ун-т ім. Юрія Федьковича, 2022. – 268 с.
3. Караванова Т. П. Теорія алгоритмів. Частина 2. Обчислювальні алгоритми : навч. посіб. 2-ге вид., випр. Чернівці : Чернівецький нац. ун-т ім. Юрія Федьковича, 2024. – 287 с. ISBN 978-966-423-867-7.
4. Креневич А. П. Алгоритми і структури даних : підручник. Київ : ВПЦ «Київський університет», 2021. – 200 с.
5. Марченко О. І., Марченко О. О. Структури даних та алгоритми : підручник : у 2 ч. Київ : Просвіта, 2024. Ч. 1. – 268 с.
6. Мелешко Є. В., Якименко М. С., Поліщук Л. І. Алгоритми та структури даних : навч. посіб. для студ. техн. спец. денної та заочної форм навчання. Кропивницький : Лисенко В. Ф., 2019. – 156 с.
7. Трофименко О. Г., Прокоп Ю. В., Яновський О. Г. Структури даних : практикум. Одеса, 2021. – 113 с.
8. Кормен Т.Х., Лейзерсон Ч.Е., Рівест Р.Л., Стайн К. Вступ до алгоритмів. – К. : К. І. С., 2019. – 1288 с.

Додаткова (допоміжна)

1. Ткачук В. М. Алгоритми та структури даних : навч. посіб. Івано-Франківськ : Вид-во Прикарпат. нац. ун-ту ім. Василя Стефаника, 2016. – 286 с.
2. Глазунова Л. В. Алгоритми та структури даних : метод. вказівки до практ. та лаб. занять. Одеса : ОНАЗ ім. О. С. Попова, 2020. – 59 с.
3. Караванова Т. П. Теорія алгоритмів. Частина 3. Збірник завдань з рекомендаціями та прикладами : навч. посіб. Чернівці : Чернівецький нац. ун-т, 2025. – 223 с. ISBN 978-966-423-980-3.
4. Karumanchi N. Data Structures and Algorithms Made Easy – CareerMonk Publications, 2016. – 418 p.
5. Wengrow J. Common–Sense Guide to Data Structures and Algorithms. – Pragmatic Bookshelf, 2020. – 481 p.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Електронний курс в системі Moodle
   https://moodle.chnu.edu.ua/course/view.php?id=7301
2. Візуалізація алгоритмів та структур даних Національного університету Сінгапуру VisuAlgo

https://visualgo.net/en

3. Курс “Розробка та аналіз алгоритмів” на Prometheus
   https://prometheus.org.ua/prometheus-free/algorithms-design-analysis-part1/

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- «Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича»
  https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича»
  https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/
- «Політика використання штучного інтелекту в Чернівецькому національному університету імені Юрія Федьковича»
  https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha-zi-zminamy-zatverdzhenymy-vchenoiu-radoiu-chnu-imeni-yuriia-fedkovycha/

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

- допускається за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов’язаний зазначити факт і спосіб його застосування);
- не допускається використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- додаткові обмеження чи вимоги щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.