

Робоча програма дисципліни «Алгоритми та методи обчислень»

СПЕЦІАЛЬНІСТЬ 123 · КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

БАКАЛАВР

120

Годин

Загальний обсяг дисципліни

4

Кредити ЄКТС

Відповідно до стандарту

4

Модулі

Змістовних блоки курсу



Загальна інформація про дисципліну

Ідентифікація

Назва: Алгоритми та методи обчислень

Спеціальність: 123 Комп'ютерна інженерія

Ступінь: Бакалавр

Обсяг: 120 годин / 4 кредити ЄКТС

Модулів: 4 змістовних блоки

Призначення дисципліни

Дисципліна «Алгоритми та методи обчислень» формує у студентів системне розуміння математичних основ алгоритмізації, методів чисельного аналізу та обчислювальних технологій, необхідних для вирішення інженерних задач у сфері комп'ютерних систем та інформаційних технологій.

Дисципліна є обов'язковою для підготовки бакалаврів зі спеціальності «Комп'ютерна інженерія» та забезпечує формування фахових компетентностей, передбачених освітньо-професійною програмою, зокрема здатності обирати, обґрунтовувати та реалізовувати ефективні алгоритми розв'язання обчислювальних задач.

Загальні компетентності

Математичне мислення

Здатність до абстрактного та формального математичного мислення: аналіз задач, побудова математичних моделей, виявлення алгоритмічних закономірностей, формування обґрунтованих висновків на основі обчислювального експерименту.

Комунікація

Здатність до усної та письмової технічної комунікації: чітко описувати алгоритми, аргументовано обирати методи обчислень, оформлювати звіти, аналітичні записки та пояснювальні записки до програмного коду.

Самоорганізація

Уміння самостійно навчатися, планувати роботу та організовувати виконання обчислювальних та програмних завдань у визначені строки, зокрема під час підготовки індивідуальних проектів і розрахункових робіт.

Командна робота

Відповідальність, ініціативність та готовність працювати у команді під час групового розв'язання обчислювальних задач, обговорення підходів до реалізації алгоритмів та спільного аналізу результатів.

Студент повинен вміти: аналізувати умову задачі та формалізувати її математично; вибирати оптимальний метод обчислень; оцінювати точність та ефективність алгоритму; реалізовувати алгоритми мовами програмування; коректно інтерпретувати та представляти результати обчислень.

Фахові компетентності

Теорія алгоритмів

Розуміння математичних основ теорії алгоритмів: концепції складності, класи P та NP, машина Тюрінга, рекурсія та індукція. Знання ключових алгоритмічних парадигм та вміння застосовувати їх для аналізу і розв'язання задач.

Чисельні методи

Здатність застосовувати чисельні методи розв'язання систем рівнянь, інтерполяції, апроксимації, чисельного диференціювання та інтегрування; оцінювати похибки методів та забезпечувати стійкість обчислень.

Структури даних та алгоритми

Знання базових структур даних (масиви, списки, стеки, черги, дерева, графи), алгоритмів пошуку та сортування, методів динамічного програмування та жадібних алгоритмів; вміння обирати оптимальну структуру для конкретної задачі.

Програмна реалізація

Здатність реалізовувати алгоритми та методи обчислень мовами програмування C++ та Python; використовувати спеціалізовані бібліотеки чисельних обчислень; верифікувати коректність реалізації на тестових прикладах.

Результати навчання

01

Аналіз та оцінка алгоритмів

Аналізувати алгоритми з точки зору часової та просторової складності; застосовувати O -нотацію; порівнювати альтернативні алгоритмічні підходи та обґрунтовувати вибір оптимального рішення для конкретної задачі.

03

Проектування та реалізація

Проектувати та реалізовувати ефективні алгоритми і структури даних для вирішення інженерних задач; верифікувати коректність реалізації; документувати програмний код відповідно до стандартів.

02

Застосування чисельних методів

Застосовувати методи чисельного аналізу для розв'язання систем лінійних та нелінійних рівнянь, задач інтерполяції та апроксимації, чисельного інтегрування та диференціювання; оцінювати похибки обчислень.

04

Робота з науковими джерелами

Орієнтуватися в науковій та навчально-методичній базі з теорії алгоритмів і методів обчислень; використовувати наукові та інформаційні ресурси для підготовки обґрунтованих аналітичних та розрахункових матеріалів.

Математичні основи алгоритмізації та складність обчислень

1

Тема 1.1

Основи теорії алгоритмів. Поняття алгоритму, властивості, способи опису. Машина Тюрінга.

2

Тема 1.2

Аналіз ефективності алгоритмів. Часова та просторова складність. O-нотація та класи складності.

3

Тема 1.3

Рекурсія та індукція. Рекурентні співвідношення. Теорема Мастера. Рекурсивні алгоритми.

4

Тема 1.4

Похибки обчислень. Класифікація та джерела похибок. Стійкість та числова точність алгоритмів.

Основи теорії алгоритмів

Лекційний матеріал (2 год.)

Лекція 1: Поняття алгоритму та його властивості: дискретність, визначеність, результативність, масовість. Способи опису алгоритмів: словесний, графічний (блок-схеми), псевдокод, мовою програмування. Класифікація алгоритмів: лінійні, розгалужені, циклічні, рекурсивні.

Машина Тюрінга: формальна модель обчислень, поняття обчислюваності, проблема зупинки. Вступ до теорії складності: класи P, NP, NP-повні задачі. Алгоритмічно нерозв'язні задачі.

Ключові питання лекції

- Поняття алгоритму та його основні властивості
- Способи формального опису алгоритмів
- Машина Тюрінга як модель обчислень
- Поняття обчислюваності та алгоритмічної нерозв'язності
- Класи складності P та NP: визначення та значення

Практичне заняття (2 год.)

Завдання 1. Розробка алгоритмів для типових задач обробки даних: складання блок-схем, запис псевдокодом, порівняння різних підходів до опису одного алгоритму.

Завдання 2. Кейс: аналіз та класифікація алгоритмів за структурою управління. Визначення ознак лінійності, розгалуженості, циклічності на практичних прикладах.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання підручника: Марченко О. І., Марченко О. О. «Структури даних та алгоритми» — розділ «Основи алгоритмізації» (КПІ, 2024)
- Підготовка реферату: «Машина Тюрінга та її значення для теорії обчислень»
- Складання порівняльної таблиці способів опису алгоритмів



Очікуваний результат: студент вміє формально описувати алгоритми, класифікувати їх за структурою та пояснювати концепцію обчислюваності.

Аналіз ефективності алгоритмів

1

О-нотація

Асимптотичні позначення O , Ω , Θ . Оцінка «найкращого», «найгіршого» та «середнього» випадку.

2

Часова складність

$O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$. Підрахунок операцій. Порівняння алгоритмів.

3

Просторова складність

Аналіз споживання пам'яті. Компроміс «час–пам'ять». In-place алгоритми.

4

Амортизований аналіз

Методи агрегування, бухгалтерії, потенціалів. Практичні приклади амортизованої складності.

Лекційний матеріал (2 год.)

Поняття ефективності алгоритму. Асимптотична нотація: O , Ω , Θ . Правила підрахунку складності: послідовні, розгалужені, циклічні структури. Найкращий, найгірший і середній випадки. Клас складності алгоритмів. Просторова складність та компроміс «час–пам'ять». Основи амортизованого аналізу.

Практика (2 год.) та СРС (7 год.)

Практика: Аналіз часової та просторової складності конкретних алгоритмів; побудова таблиць порівняння; розв'язання задач на визначення асимптотичної складності за псевдокодом.

СРС: Опрацювання: Кублій Л. І. «Алгоритми та структури даних. Основи алгоритмізації» (КПІ, 2022); підготовка аналітичної записки «Порівняльний аналіз ефективності алгоритмів пошуку».

Рекурсія та похибки обчислень

Тема 1.3 — Рекурсія та індукція (Лекція 3, 2 год.)

Поняття рекурсії та принцип математичної індукції. Рекурентні співвідношення та методи їх розв'язання. Теорема Мастера для аналізу алгоритмів «розділяй і володарюй». Приклади рекурсивних алгоритмів: обчислення факторіалу, числа Фібоначчі, алгоритм Ханойської вежі. Усунення хвостової рекурсії. Порівняння рекурсивних та ітеративних підходів.

Практика (2 год.): Розробка рекурсивних алгоритмів для задач перебору та розбиття; аналіз трудомісткості рекурсивних викликів; трасування рекурсивних програм.

СРС (7 год.): Опрацювання: Грудзинський Ю. Є. «Алгоритми та структури даних» (КПІ, 2022) — розділ «Рекурсивні алгоритми»; підготовка реферату «Рекурсія та динамічне програмування: порівняльний аналіз».

Тема 1.4 — Похибки обчислень (Лекція 4, 2 год.)

Джерела та класифікація похибок: абсолютна, відносна, похибка методу, похибка округлення. Подання чисел у комп'ютері: формати з фіксованою та плаваючою комою, стандарт IEEE 754. Накопичення похибок у обчислювальних процесах. Обумовленість задачі та стійкість алгоритму. Методи оцінки та мінімізації похибок. Інтервальна арифметика.

Практика (2 год.): Розрахунок абсолютної та відносної похибок для конкретних обчислювальних задач; аналіз впливу округлення на точність результату; дослідження стійкості чисельних алгоритмів.

СРС (7 год.): Опрацювання: Кветний Р. Н. та ін. «Методи та алгоритми комп'ютерних обчислень. Теорія і практика» (ВНТУ, 2023) — розділ «Похибки обчислень»; підготовка розрахункової роботи з аналізу точності алгоритмів.

Структури даних та алгоритмічні стратегії

Тема 2.1

Лінійні структури даних: масиви, зв'язні списки, стеки, черги.
Алгоритми роботи зі списками.

Тема 2.2

Нелінійні структури: дерева, графи. Алгоритми обходу, пошуку в ширину та глибину.

Тема 2.3

Алгоритми сортування та пошуку. Порівняльний аналіз: бульбашка, вставка, злиття, швидке сортування.

Тема 2.4

Алгоритмічні стратегії: «розділяй і володарюй», жадібні алгоритми, динамічне програмування.

Лінійні структури даних

Лекційний матеріал (2 год.)

Поняття структур даних та їх класифікація. Масиви: одновимірні та багатовимірні, операції, складність. Зв'язні списки: односпрямовані, двоспрямовані, кільцеві. Операції: вставка, видалення, пошук. Стек: принцип LIFO, реалізація на масиві та списку, застосування (обробка виразів, рекурсія). Черга: принцип FIFO, кільцева черга, черга з пріоритетами, двостороння черга (deque).

Ключові питання

- Порівняння масивів та зв'язних списків за часовою та просторовою складністю
- Реалізація стеку та черги на основі масиву та списку
- Застосування стеку для обробки арифметичних виразів
- Черга з пріоритетами та сфери застосування
- Вибір структури даних залежно від задачі

Практичне заняття (2 год.)

Завдання 1. Реалізація зв'язного списку мовою C++ або Python: вставка, видалення, пошук елементу. Аналіз складності операцій.

Завдання 2. Кейс: реалізація стеку для перевірки правильності розставлення дужок у виразі. Тестування на граничних значеннях.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання: Марченко О. І., Марченко О. О. «Структури даних та алгоритми» (КПІ, 2024) — розділ «Лінійні структури даних»
- Підготовка розрахункової роботи: «Порівняльний аналіз реалізацій черги на масиві та зв'язному списку»
- Розв'язання задач на застосування стеку та черги в алгоритмах обходу



Очікуваний результат: студент вміє реалізовувати лінійні структури даних, обирати оптимальну структуру для задачі та аналізувати складність операцій.

Нелінійні структури даних: дерева та графи



Лекційний матеріал (2 год.)

Дерево як структура даних: термінологія, властивості. Бінарне дерево пошуку: операції та складність. Збалансовані дерева. Граф: визначення, види, способи подання. Алгоритми обходу: BFS та DFS. Алгоритм Дейкстри для пошуку найкоротшого шляху. Мінімальне кістякове дерево.

Практика (2 год.) та СРС (7 год.)

Практика: Реалізація бінарного дерева пошуку; програмування обходів дерева; реалізація алгоритму BFS/DFS для графа; розв'язання задачі пошуку найкоротшого шляху.

СРС: Опрацювання: Бульба С. С. та ін. «Алгоритми та структури даних» (ХПІ, 2024) — розділ «Дерева та графи»; підготовка звіту «Порівняльний аналіз алгоритмів пошуку найкоротшого шляху».

Алгоритми сортування та алгоритмічні стратегії

Тема 2.3 — Алгоритми сортування та пошуку (Лекція 7, 2 год.)

Алгоритми сортування $O(n^2)$: бульбашкою, вставкою, вибором — порівняльна характеристика. Ефективні алгоритми: злиттям $O(n \log n)$, швидке сортування (QuickSort) $O(n \log n)$. Сортування підрахунком, пірамідальне сортування. Нижня межа складності сортування порівняннями. Алгоритми пошуку: лінійний та бінарний пошук, хешування — принципи та складність.

Практика (2 год.): Реалізація та порівняльне тестування алгоритмів сортування; вимірювання часу виконання на наборах різного розміру; побудова графіків залежності часу від розміру вхідних даних.

СРС (7 год.): Опрацювання: Кублій Л. І. «Алгоритми та структури даних. Основи алгоритмізації» (КПІ, 2022); підготовка розрахункової роботи «Реалізація та порівняльний аналіз алгоритмів сортування».

Тема 2.4 — Алгоритмічні стратегії (Лекція 8, 2 год.)

«Розділяй і володарюй»: принцип, аналіз складності через теорему Мастера, приклади (злиттям, QuickSort, бінарний пошук). Жадібні алгоритми: принцип, умови застосовності, задача про рюкзак, алгоритми на графах. Динамічне програмування: принцип оптимальності Беллмана, мемоізація та табуляція, задачі: числа Фібоначчі, найдовша спільна підпоследовність, задача про рюкзак 0/1. Порівняння стратегій.

Практика (2 год.): Розв'язання задач динамічного програмування; розробка жадібних алгоритмів для задач планування; порівняння підходів «розділяй і володарюй» та динамічного програмування.

СРС (7 год.): Підготовка аналітичного звіту «Динамічне програмування vs жадібні алгоритми: вибір стратегії для практичних задач».

Чисельні методи розв'язання алгебраїчних задач

Тема 3.1

Чисельні методи розв'язання нелінійних рівнянь: метод бісекції, дотичних (Ньютона), хорд, ітерацій.

Тема 3.2

Системи лінійних алгебраїчних рівнянь. Прямі методи: Гауса, LU-розкладання. Ітераційні методи.

Тема 3.3

Інтерполяція та апроксимація. Поліноми Лагранжа та Ньютона. МНК. Сплайни.

Тема 3.4

Чисельне диференціювання та інтегрування. Формули прямокутників, трапецій, Сімпсона, квадратурні формули.

Чисельні методи розв'язання нелінійних рівнянь

Лекційний матеріал (2 год.)

Постановка задачі: знаходження кореня рівняння $f(x) = 0$. Локалізація кореня: відокремлення коренів графічним та аналітичним методами. Метод бісекції (половинного ділення): алгоритм, умова збіжності, оцінка похибки. Метод дотичних (Ньютона): алгоритм, квадратична збіжність, умови застосовності. Метод хорд та метод простої ітерації: умови збіжності, порівняльна характеристика.

Ключові питання

- Відокремлення коренів нелінійного рівняння
- Метод бісекції: алгоритм та оцінка кількості ітерацій
- Метод Ньютона: переваги, недоліки, умови збіжності
- Порівняльна характеристика ітераційних методів
- Критерії зупинки ітераційного процесу

Практичне заняття (2 год.)

Завдання 1. Реалізація методу бісекції та методу Ньютона для заданого рівняння: програмування, тестування, порівняння кількості ітерацій та точності.

Завдання 2. Кейс: вибір та обґрунтування методу розв'язання нелінійного рівняння залежно від характеристик функції. Аналіз збіжності.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання: Кветний Р. Н. та ін. «Методи та алгоритми комп'ютерних обчислень. Теорія і практика» (ВНТУ, 2023) — розділ «Нелінійні рівняння»
- Виконання індивідуального завдання: розв'язання нелінійного рівняння трьома методами, порівняння результатів
- Підготовка реферату: «Метод Ньютона та його модифікації в інженерних задачах»



Очікуваний результат: студент вміє застосовувати чисельні методи розв'язання нелінійних рівнянь, оцінювати їх збіжність та реалізовувати програмно.

Системи лінійних алгебраїчних рівнянь

Лекційний матеріал (2 год.)

Постановка задачі розв'язання СЛАР. Умова існування та єдиності розв'язку: визначник, ранг матриці. Прямі методи: метод Гауса з вибором ведучого елемента, LU-розкладання. Трикутні системи. Обчислення визначника та оберненої матриці. Ітераційні методи: Якобі, Гауса–Зейделя — умови збіжності. Обумовленість матриці та вплив на точність розв'язку. Оцінка похибок.

Практика (2 год.) та СРС (7 год.)

Практика: Реалізація методу Гауса з вибором ведучого елемента; розв'язання СЛАР методом Якобі; порівняння прямих та ітераційних методів за точністю та часом виконання; аналіз обумовленості матриці.

СРС: Опрацювання: Назаренко О. А., Третьяк О. І. «Методи обчислень у системі комп'ютерної математики Maple» (Астропринт, 2023); виконання індивідуального завдання на розв'язання СЛАР кількома методами з оцінкою точності.

❏ **Очікуваний результат:** студент вміє застосовувати прямі та ітераційні методи розв'язання СЛАР, оцінювати точність та обумовленість задачі.

Інтерполяція та апроксимація функцій



Інтерполяція (постановка задачі)

Задача інтерполяції: побудова функції, що проходить через задані вузлові точки. Існування та єдиність інтерполяційного полінома.



Поліноми Лагранжа та Ньютона

Інтерполяційний поліном Лагранжа. Поліном Ньютона з рівновіддаленими та довільними вузлами. Формули кінцевих різниць. Оцінка похибки інтерполяції.



МНК та сплайни

Метод найменших квадратів: лінійна та поліноміальна апроксимація. Кубічні сплайни: побудова, переваги над поліноміальною інтерполяцією.

Лекційний матеріал (2 год.)

Задача інтерполяції та апроксимації: відмінності та область застосування. Інтерполяційний поліном Лагранжа: побудова та оцінка похибки. Поліном Ньютона: ефективний алгоритм обчислення. Кубічні сплайни та їх переваги. МНК: лінійна та нелінійна апроксимація, матриця Гаусса.

Практика (2 год.) та СРС (7 год.)

Практика: Побудова інтерполяційного полінома Лагранжа для заданої таблиці значень; реалізація МНК для лінійної апроксимації; порівняння інтерполяції та апроксимації на конкретних даних.

СРС: Опрацювання: Кветний Р. Н. та ін. «Методи та алгоритми комп'ютерних обчислень» (ВНТУ, 2023) — розділ «Інтерполяція»; виконання індивідуальної розрахункової роботи.

Чисельне диференціювання та інтегрування

Лекційний матеріал (2 год.)

Чисельне диференціювання: кінцево-різницеві апроксимації похідних першого та другого порядків. Оцінка похибки. Рекомендації щодо вибору кроку. Чисельне інтегрування: прямокутники (ліво-, право-, середньоточковий варіанти), метод трапецій, метод Сімпсона ($1/3$ та $3/8$). Оцінка похибки квадратурних формул. Метод Гауса. Адаптивне інтегрування. Невласні інтеграли.

Ключові питання

- Кінцево-різницеві формули для наближеного обчислення похідних
- Порівняльна характеристика квадратурних формул
- Оцінка похибки методу трапецій та Сімпсона
- Метод Гауса та його переваги
- Адаптивне інтегрування: принцип та алгоритм

Практичне заняття (2 год.)

Завдання 1. Обчислення визначеного інтеграла методами прямокутників, трапецій та Сімпсона. Порівняння точності при різних кроках інтегрування.

Завдання 2. Чисельне диференціювання функції; оцінка похибки при різних значеннях кроку h ; аналіз ефекту скорочення значущих цифр.

Завдання 3. Практичне завдання: обчислення довжини дуги та площі фігури чисельними методами.

Самостійна робота (7 год.)

- Опрацювання: Кветний Р. Н. та ін. «Методи та алгоритми комп'ютерних обчислень» (ВНТУ, 2023) — розділ «Чисельне інтегрування»
- Виконання індивідуального завдання з чисельного інтегрування кількома методами
- Підготовка аналітичного звіту «Порівняльний аналіз квадратурних формул»



Очікуваний результат: студент вміє застосовувати квадратурні формули, оцінювати похибку чисельного інтегрування та реалізовувати алгоритми програмно.

Чисельні методи оптимізації та диференціальних рівнянь

Тема 4.1

Задачі оптимізації. Методи одновимірної оптимізації: золотий перетин, парабол. Градієнтні методи.

Тема 4.2

Чисельне розв'язання звичайних диференціальних рівнянь (ЗДР). Методи Ейлера та Рунге–Кутта.

Тема 4.3

Алгоритми на рядках та хешування. Пошук підрядка: наївний, Кнута–Морріса–Пратта, Боєра–Мура.

Тема 4.4

Рандомізовані алгоритми та алгоритми на великих даних. Методи Монте–Карло. Поточкові алгоритми.

Методи оптимізації

Лекційний матеріал (2 год.)

Постановка задачі оптимізації: мінімізація/максимізація функції. Одновимірна безумовна оптимізація: метод золотого перетину, метод парабол — алгоритм та порівняння збіжності. Багатовимірна безумовна оптимізація: метод градієнтного спуску, метод найшвидшого спуску, метод Ньютона. Умовна оптимізація: метод множників Лагранжа. Задача лінійного програмування: симплекс-метод (вступ).

Ключові питання

- Поняття унімодальної функції та умови застосовності методу золотого перетину
- Градієнтні методи: принцип, умови збіжності
- Метод Ньютона для оптимізації: переваги та обмеження
- Задача умовної оптимізації та метод множників Лагранжа
- Глобальна та локальна оптимізація: відмінності

Практичне заняття (2 год.)

Завдання 1. Реалізація методу золотого перетину для мінімізації унімодальної функції; аналіз кількості ітерацій та точності результату.

Завдання 2. Дискусія: «Вибір методу оптимізації залежно від характеристик задачі». Розгляд практичних прикладів з комп'ютерної інженерії.

Завдання 3. Розв'язання задачі умовної оптимізації методом множників Лагранжа.

Самостійна робота (7 год.)

- Опрацювання: Кветний Р. Н. та ін. «Методи та алгоритми комп'ютерних обчислень» (ВНТУ, 2023) — розділ «Задачі оптимізації»
- Підготовка реферату «Методи оптимізації в задачах машинного навчання»
- Виконання індивідуального завдання: мінімізація функції двох змінних кількома методами



Очікуваний результат: студент вміє застосовувати чисельні методи оптимізації, обирати метод залежно від властивостей задачі та реалізовувати алгоритми програмно.

Чисельне розв'язання диференціальних рівнянь

Лекційний матеріал (2 год.)

Задача Коші для ЗДР першого порядку. Метод Ейлера: явний та неявний, похибка та порядок точності. Метод Рунге–Кутта 2-го та 4-го порядків: алгоритм, коефіцієнти, оцінка похибки. Метод предиктор–коректор (Адамса). Стійкість чисельних методів для ЗДР. Вибір кроку інтегрування. Системи ЗДР та рівняння вищих порядків.

Практика (2 год.) та СРС (7 год.)

Практика: Розв'язання задачі Коші методом Ейлера та Рунге–Кутта 4-го порядку; порівняння точності методів на тестових задачах з відомим аналітичним розв'язком; аналіз залежності похибки від кроку.

СРС: Опрацювання: Кветний Р. Н. та ін. «Методи та алгоритми комп'ютерних обчислень» (ВНТУ, 2023) — розділ «ЗДР»; підготовка звіту «Методи Рунге–Кутта: порівняльний аналіз порядків точності»; виконання індивідуального завдання.



Очікуваний результат: студент вміє чисельно розв'язувати задачу Коші, порівнювати методи за точністю та стійкістю, реалізовувати алгоритми на мові програмування.

Алгоритми на рядках та хешування

Лекційний матеріал (2 год.)

Алгоритми пошуку підрядка: наївний алгоритм $O(nm)$. Алгоритм Кнута–Морріса–Пратта (КМП): побудова префікс-функції, лінійна складність $O(n+m)$. Алгоритм Боєра–Мура: евристики поганого символу та доброго суфікса. Хешування: поняття хеш-функції, колізії, методи розв'язання (ланцюжки, відкрита адресація). Хеш-таблиці: складність операцій. Поліноміальне хешування рядків (алгоритм Рабіна–Карпа).

Ключові питання

- Алгоритм КМП: побудова та застосування префікс-функції
- Порівняння алгоритмів пошуку підрядка за складністю
- Принцип хешування та методи вирішення колізій
- Хеш-таблиці: амортизована складність операцій
- Алгоритм Рабіна–Карпа: множинний пошук підрядків

Практичне заняття (2 год.)

Завдання 1. Реалізація алгоритму КМП для пошуку підрядка; тестування та порівняння з наївним алгоритмом за кількістю порівнянь.

Завдання 2. Реалізація хеш-таблиці з обробкою колізій методом ланцюжків; аналіз ефективності при різних коефіцієнтах завантаження.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання: Марченко О. І., Марченко О. О. «Структури даних та алгоритми» (КПІ, 2024) — розділ «Хешування та пошук»
- Підготовка реферату: «Алгоритми пошуку підрядка: порівняльний аналіз та застосування»
- Реалізація алгоритму Боєра–Мура та порівняння з КМП



Очікуваний результат: студент вміє реалізовувати ефективні алгоритми рядкового пошуку та структури хеш-таблиць, аналізувати їх складність.

Рандомізовані алгоритми та алгоритми на великих даних

Лекційний матеріал (2 год.)

Рандомізовані алгоритми: класифікація (Las Vegas, Monte Carlo), порівняння з детермінованими. Методи Монте–Карло: стохастичне моделювання, обчислення інтегралів, оцінка π . Рандомізоване швидке сортування. Алгоритми потокової обробки даних (streaming): принципи, задачі підрахунку, алгоритм Флажолє–Мартена. Зовнішнє сортування: проблема великих масивів, методи зовнішнього злиття. Основи паралельних обчислень: закони Амдала та Густафсона.

Ключові питання

- Рандомізовані алгоритми: переваги та обмеження
- Метод Монте–Карло: принцип та застосування
- Алгоритми потокової обробки: задачі та обмеження пам'яті
- Зовнішнє сортування: проблема та алгоритм злиття
- Закон Амдала та межі паралельного прискорення

Практичне заняття (2 год.)

Завдання 1. Реалізація методу Монте–Карло для обчислення площі фігури та значення π ; дослідження залежності точності від кількості випадкових точок.

Завдання 2. Дискусія: «Рандомізовані алгоритми в задачах комп'ютерної інженерії — де межа між Las Vegas та Monte Carlo?».

Завдання 3. Підготовка та презентація міні-проекту: «Застосування алгоритмів на великих даних у сучасних системах».

Самостійна робота (7 год.)

- Опрацювання: Новотарський М. А., Порєв В. М. «Алгоритми та методи обчислень. Лабораторний практикум» (КПІ, 2025)
- Підготовка аналітичного звіту «Паралельні алгоритми та закон Амдала: практичні наслідки для багатопроцесорних систем»
- Реалізація рандомізованого алгоритму для задачі з варіанту



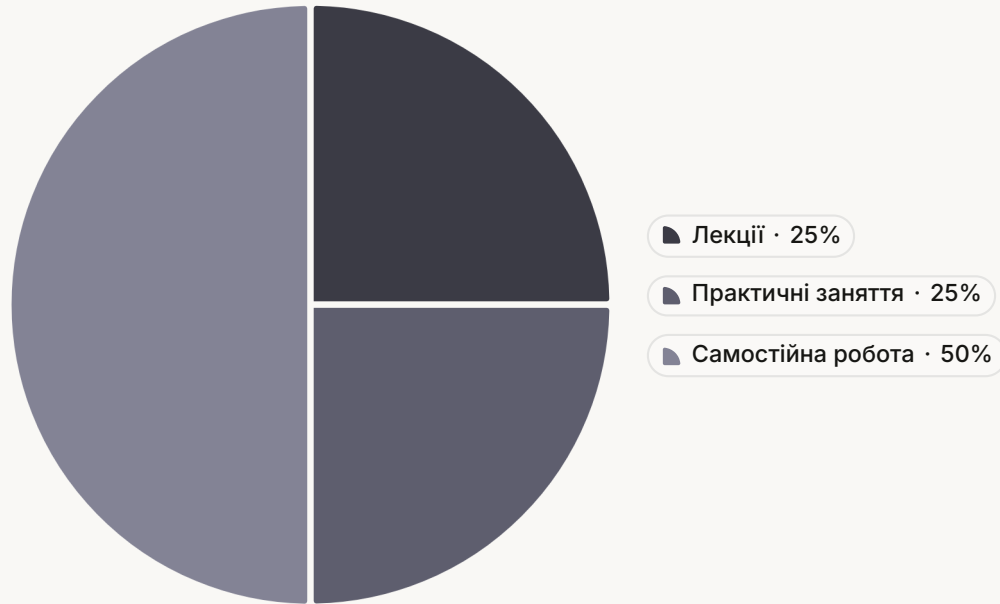
Очікуваний результат: студент вміє аналізувати рандомізовані алгоритми, застосовувати метод Монте–Карло та розуміти принципи роботи з великими масивами даних.

Розподіл навчального часу — 120 годин

Модуль	Тема	Лекції (год.)	Практичні (год.)	СРС (год.)
Модуль 1	Тема 1.1 — Основи теорії алгоритмів	2	2	7
	Тема 1.2 — Аналіз ефективності алгоритмів	2	2	7
	Тема 1.3 — Рекурсія та індукція	2	2	7
	Тема 1.4 — Похибки обчислень	2	2	7
Модуль 2	Тема 2.1 — Лінійні структури даних	2	2	7
	Тема 2.2 — Дерева та графи	2	2	7
	Тема 2.3 — Алгоритми сортування та пошуку	2	2	7
	Тема 2.4 — Алгоритмічні стратегії	2	2	7
Модуль 3	Тема 3.1 — Нелінійні рівняння	2	2	7
	Тема 3.2 — Системи лінійних рівнянь	2	2	7
	Тема 3.3 — Інтерполяція та апроксимація	2	2	7
	Тема 3.4 — Чисельне диф-ня та інтегрування	2	2	7
Модуль 4	Тема 4.1 — Методи оптимізації	2	2	7
	Тема 4.2 — Чисельне розв'язання 3ДР	2	2	7
	Тема 4.3 — Алгоритми на рядках та хешування	2	2	7
	Тема 4.4 — Рандомізовані алгоритми та Big Data	2	6	4
Разом	16 тем	30	30	60

 **Загальний обсяг: 120 годин = 4 кредити ЄКТС.** Розподіл є орієнтовним і може коригуватися відповідно до специфіки навчального процесу.

Структура навчального часу



Коментар до розподілу

Загальний обсяг — **120 годин (4 кредити ЄКТС)** — розподілено між трьома формами роботи для забезпечення балансу між теорією та практикою.

Лекції — 30 год. (25%)

Системне подання теоретичного матеріалу з використанням наочних схем, таблиць, прикладів алгоритмів та фрагментів програмного коду на C++ і Python.

Практичні — 30 год. (25%)

Реалізація алгоритмів, аналіз складності, розв'язання обчислювальних задач, захист міні-проектів. Рівна частка з лекціями підкреслює практичну спрямованість курсу.

Самостійна робота — 60 год. (50%)

Поглиблене вивчення, виконання індивідуальних розрахункових робіт, робота з науковою літературою та відкритими алгоритмічними ресурсами. СРС становить **50% обсягу дисципліни**.

Самостійна робота студентів — Детальний опис



Опрацювання літератури

Поглиблене вивчення теоретичного матеріалу за підручниками з теорії алгоритмів та методів обчислень (Марченко, Квітний, Кублій, Грудзинський та ін.), науковими статтями у фахових виданнях. Студент конспектує ключові положення, виділяє головні алгоритми та порівнює підходи різних авторів.



Робота з інформаційними ресурсами

Опрацювання репозиторію КПІ (ela.kpi.ua), ВНТУ (pdf.lib.vntu.edu.ua), Національної бібліотеки (nbuv.gov.ua), відкритих алгоритмічних платформ. Аналіз наукових статей, пошук актуальних реалізацій алгоритмів у відкритих джерелах.



Індивідуальні завдання

Виконання розрахункових та аналітичних завдань: реалізація алгоритмів мовами C++ та Python, порівняльний аналіз методів, розробка звітів з чисельних експериментів; підготовка рефератів та аналітичних записок з актуальних питань теорії обчислень.



Підготовка до занять

Підготовка до практичних занять (реалізація базових алгоритмів, повторення теорії); підготовка до тестувань (ключові терміни, алгоритми, складності); підготовка до контрольних модульних робіт (систематизація знань, типові задачі).

Форми контролю знань

1

Поточний контроль

Завдання: аналітичні та тестові вправи за темами змістовного модуля.

Оцінюються регулярність виконання завдань, якість реалізації алгоритмів та рівень засвоєння теоретичного матеріалу. Поточний контроль формує накопичувальну оцінку протягом семестру. Проводиться після кожної теми у формі тестування (10–15 питань) або усного опитування.

2

Проміжний контроль

Розрахункова або контрольна робота — комплексне завдання, що охоплює теми змістовного модуля (теоретичні питання, аналіз складності алгоритмів, реалізація або трасування алгоритму).

Оцінюються повнота відповіді, коректність реалізації та обґрунтованість вибору методу.

Проводиться після завершення кожного змістовного модуля.

3

Підсумковий контроль

Іспит відповідно до навчального плану. Форма проведення — тестування та практичні задачі на реалізацію/аналіз алгоритмів. Підсумкова оцінка враховує результати поточного, проміжного та підсумкового контролю відповідно до встановленої шкали: **100-бальна система** з переведенням у національну шкалу та ЄКТС (A, B, C, D, E, FX, F).

Методи викладання

Викладання дисципліни «Алгоритми та методи обчислень» поєднує класичні академічні та інтерактивні методи, що забезпечує формування не лише міцної теоретичної бази, а й практичних навичок проектування та реалізації ефективних алгоритмів, необхідних для майбутньої професійної діяльності у сфері комп'ютерної інженерії.



Лекції

Системне подання теоретичного матеріалу з використанням наочних схем, псевдокоду, порівняльних таблиць складності та фрагментів коду мовами C++ і Python.



Практичне програмування

Реалізація алгоритмів та структур даних; тестування на граничних значеннях; замірювання часу виконання; порівняння теоретичної та фактичної складності.



Кейс-стаді

Аналіз реальних обчислювальних задач із сфери комп'ютерної інженерії; студенти виявляють проблему, обирають алгоритм і обґрунтовують вибір на основі аналізу складності.



Дискусії

Обговорення вибору алгоритмів та методів для конкретних задач; розвиток критичного мислення, аргументації та поваги до альтернативних підходів.



Обчислювальний експеримент

Проведення чисельних експериментів: порівняння алгоритмів, оцінка похибок, побудова графіків залежностей; формування навичок наукового підходу до аналізу даних.



Аналіз складності

Математичний аналіз алгоритмів: підрахунок операцій, побудова рекурентних співвідношень, застосування теореми Мастера; формування навичок точного оцінювання ефективності.



Усі методи спрямовані на формування компетентностей, передбачених освітньо-професійною програмою спеціальності 123 «Комп'ютерна інженерія».

Рекомендована література — Підручники та посібники (2022–2025 рр.)

1. Марченко О. І., Марченко О. О.

Структури даних та алгоритми. Підручник у 2-х частинах. Частина 1 [Електронний ресурс] / КПІ ім. Ігоря Сікорського. — Київ : Просвіта, 2024. — 9,88 Мбайт.

2. Кветний Р. Н., Іванчук Я. В., Богач І. В., Софіна О. Ю., Барабан М. В.

Методи та алгоритми комп'ютерних обчислень. Теорія і практика : підручник. — Вінниця : ВНТУ, 2023. — 280 с. — ISBN 978-966-641-924-1.

3. Кублій Л. І.

Алгоритми та структури даних. Основи алгоритмізації : підручник / КПІ ім. Ігоря Сікорського. — Київ : КПІ ім. Ігоря Сікорського, 2022. — 528 с.

4. Грудзинський Ю. Є.

Алгоритми та структури даних : навчальний посібник / КПІ ім. Ігоря Сікорського. — Київ : НТУУ «КПІ ім. Ігоря Сікорського», 2022. — 215 с.

5. Бульба С. С., Бречко В. О., Лисиця Д. О., Бельорін-Еррера О. М.

Навчально-методичний посібник «Алгоритми та структури даних» для студентів спеціальності 123 «Комп'ютерна інженерія». — Харків : НТУ «ХПІ», 2024. — 142 с.

6. Назаренко О. А., Третьяк О. І.

Методи обчислень у системі комп'ютерної математики Maple : монографія. — Одеса : Астропринт, 2023. — 348 с.

7. Новотарський М. А., Порєв В. М.

Алгоритми та методи обчислень. Лабораторний практикум [Електронний ресурс] / КПІ ім. Ігоря Сікорського. — Київ : КПІ ім. Ігоря Сікорського, 2025.

8. Бугаєва Л. М., Ковалюк Д. О.

Алгоритми та структури даних. Комп'ютерний практикум [Електронний ресурс] : навч. посіб. / КПІ ім. Ігоря Сікорського. — Київ : КПІ ім. Ігоря Сікорського, 2022.

Додаткова література та наукові статті

9. Цибульник С. О., Барандич К. С.

Технології розроблення програмного забезпечення : підручник / КПІ ім. Ігоря Сікорського. — Київ : КПІ ім. Ігоря Сікорського, 2022. — 270 с. *Рекомендується для тем 2.3–2.4 (алгоритмічні стратегії та патерни).*

11. Долголенко О. М. та ін.

Проектування складних паралельних систем : навч. посіб. / КПІ ім. Ігоря Сікорського. — Київ : КПІ ім. Ігоря Сікорського, 2023. *Рекомендується для теми 4.4 (паралельні обчислення).*

10. Васильєв О. М.

Програмування в Python. Теорія і практика : навчальний посібник. — Київ : Видавництво Ліра-К, 2023. — 462 с. *Рекомендується для практичних занять усіх модулів.*

12. Волокита А. М., Селіванов В. Л. та ін.

Основи теорії планування експерименту: розділ дисципліни «Методика та організація наукових досліджень» / КПІ ім. Ігоря Сікорського. — Київ : КПІ, 2022. *Рекомендується для тем 3.1–3.4 (чисельний експеримент).*

Нормативно-правова база

→ Конституція України

Прийнята 28 червня 1996 р. Основний Закон держави. zakon.rada.gov.ua

→ Закон «Про освіту» від 05.09.2017 № 2145–VIII

Визначає засади освітньої діяльності в Україні, у т. ч. підготовку фахівців у сфері комп'ютерних наук та інженерії.

zakon.rada.gov.ua

→ Закон «Про вищу освіту» від 01.07.2014 № 1556–VII

Регулює відносини у сфері вищої освіти; визначає кредитну систему ЄКТС, форми контролю знань та вимоги до освітніх програм.

zakon.rada.gov.ua

→ Стандарт вищої освіти України — бакалавр, спеціальність 123 «Комп'ютерна інженерія»

Визначає загальні та фахові компетентності, результати навчання та вимоги до освітньо-професійних програм підготовки бакалаврів. mon.gov.ua

→ Наказ МОН України «Про затвердження Положення про організацію освітнього процесу»

Регулює порядок організації навчання, проведення контрольних заходів та нарахування навчального навантаження. mon.gov.ua

Офіційні ресурси та фахові видання

Електронний архів КПІ

ela.kpi.ua — навчально-методичні матеріали, підручники та посібники кафедр спеціальності 123 «Комп'ютерна інженерія».

Верховна Рада України

zakon.rada.gov.ua — нормативно-правова база, стандарти вищої освіти та законодавчі акти у сфері ІТ.

Національна бібліотека України ім. В. І. Вернадського

nbuv.gov.ua — електронні ресурси, наукові статті, монографії з теорії алгоритмів та обчислювальних методів.

Бібліотека ВНТУ

pdf.lib.vntu.edu.ua — підручники та навчальні посібники, зокрема з методів та алгоритмів комп'ютерних обчислень.

Журнал «Комп'ютерні засоби, мережі та системи»

nbuv.gov.ua — фахове видання НАН України з питань комп'ютерних систем та алгоритмів.

Вісник НТУ «ХПІ»

repository.kpi.kharkov.ua — серія «Інформатика та моделювання»; наукові статті з алгоритмів та структур даних.

Міністерство освіти і науки України

mon.gov.ua — освітні стандарти, навчальні програми, методичні матеріали для спеціальності 123.

Open Courseware КПІ

ocw.kpi.ua — відкриті навчальні курси, відеолекції та матеріали з дисциплін комп'ютерної інженерії.

Зв'язок дисципліни зі спеціальністю 123 «Комп'ютерна інженерія»

Чому «Алгоритми та методи обчислень» для інженерів?

Дисципліна є фундаментальною складовою підготовки бакалаврів зі спеціальністю 123 «Комп'ютерна інженерія». Розуміння алгоритмів та методів обчислень формує у майбутніх інженерів здатність проектувати ефективні програмні рішення, оцінювати їх ресурсоємність та обирати оптимальні підходи до розв'язання обчислювальних задач.

Сучасний фахівець у сфері комп'ютерної інженерії щоденно стикається із задачами обробки даних, проектування систем та оптимізації, де знання алгоритмів є критично важливою конкурентною перевагою.

Практичне значення для майбутньої кар'єри

- Проектування ефективних програмних алгоритмів для систем реального часу та вбудованих систем
- Оцінка та оптимізація продуктивності програмного забезпечення на базі аналізу складності
- Застосування чисельних методів у задачах моделювання комп'ютерних систем та цифрової обробки сигналів
- Вирішення NP-складних задач методами наближеної та евристичної оптимізації
- Робота з великими масивами даних та розробка масштабованих алгоритмів
- Підготовка до технічних співбесід у провідних IT-компаніях, де знання алгоритмів є обов'язковою вимогою

Ключові компетентності курсу — Підсумок



Курс охоплює всі ключові розділи теорії алгоритмів та чисельних методів — від математичних основ до практичної реалізації мовами C++ та Python — і формує у студентів цілісне розуміння алгоритмічного підходу до розв'язання інженерних задач.

Бажаємо успіхів у вивченні дисципліни «Алгоритми та методи обчислень»!

120

Годин

Загальний обсяг дисципліни

4

Кредити ЄКТС

Відповідно до стандарту

16

Тем

У 4 змістовних модулях

8

Джерел

Рекомендованої літератури 2022–2025 рр.

«Алгоритм — це точний рецепт, що описує послідовність дій для розв'язання певного класу задач.» — Андрій Єршов

Спеціальність 123 · Комп'ютерна інженерія · Бакалавр · 4 кредити ЄКТС