



Алгоритми та структури даних

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський) /
Галузь знань	F Інформаційні технології
Спеціальність	F1 Прикладна математика
Освітня програма	Наука про дані та математичне моделювання
Статус дисципліни	Нормативна
Форма навчання	очна(денна)
Рік підготовки, семестр	1 курс, весняний семестр
Обсяг дисципліни	3 кредити
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	Лекція – 1 раз на тиждень, практичні заняття 1 раз на 2 тижні
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: канд. техн. наук, доцент, Сирота Сергій Вікторович, syrota.sergiy@ill.kpi.ua Практичні / Семінарські: канд. техн. наук, доцент, Сирота Сергій Вікторович, syrota.sergiy@ill.kpi.ua
Розміщення курсу	Дистанційний ресурс Moodle, https://do.ipk.kpi.ua/course/view.php?id=1681 Далі за текстом платформа «Сікорський»

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою дисципліни є розгляд та вивчення методів і прийомів складання алгоритмів, оцінки їхніх характеристик, вивчення основних структур даних та сфери їх застосування, формування парадигматичного світогляду на програмування, та вирішення послідовних задач професійної діяльності.

Предмет навчальної дисципліни — відомі алгоритми і структури даних, які використовуються в сучасних інформаційних технологіях

Під час вивчення даної дисципліни студенти набудуть такі **компетентності**:

- ЗК1 Здатність учитися і оволодівати сучасними знаннями.
- ЗК3 Здатність генерувати нові ідеї (креативність).
- ЗК4 Здатність бути критичним і самокритичним.
- ЗК6 Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК7 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК8 Знання та розуміння предметної області та розуміння професійної діяльності.
- ФК3 Здатність обирати та застосовувати математичні методи для розв'язання прикладних задач, моделювання, аналізу, проектування, керування, прогнозування, прийняття рішень.
- ФК4 Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.

- *ФК12 Здатність до пошуку, систематичного вивчення та аналізу науково-технічної інформації, вітчизняного й закордонного досвіду, пов'язаного із застосуванням математичних методів для дослідження різноманітних процесів, явищ та систем.*
- *ФК13 Здатність зрозуміти постановку завдання, сформульовану мовою певної предметної галузі, здійснювати пошук та збір необхідних вихідних даних.*
- *ФК14 Здатність сформулювати математичну постановку задачі, спираючись на постановку мовою предметної галузі, та обирати метод її розв'язання, що забезпечує потрібні точність і надійність результату.*

Програмними результатами навчання є такі:

- РН3 Формалізувати задачі, сформульовані мовою певної предметної галузі; формувати їх математичну постановку та обирати раціональний метод вирішення; розв'язувати отримані задачі аналітичними та чисельними методами, оцінювати точність та достовірність отриманих результатів.
- РН4 Виконувати математичний опис, аналіз та синтез дискретних об'єктів та систем, використовуючи поняття й методи дискретної математики та теорії алгоритмів.
- РН5 Уміти розробляти та використовувати на практиці алгоритми, пов'язані з апроксимацією функціональних залежностей, чисельним диференціюванням та інтегруванням, розв'язанням систем алгебраїчних, диференціальних та інтегральних рівнянь, розв'язанням крайових задач, пошуком оптимальних рішень.
- РН7 Вміти проводити практичні дослідження та знаходити розв'язок некоректних задач.
- РН 9 Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.
- РН 10 Володіти методиками вибору раціональних методів та алгоритмів розв'язання математичних задач оптимізації, дослідження операцій, оптимального керування і прийняття рішень, аналізу даних.
- РН 11 Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Вивченню дисципліни «Алгоритми і структури даних» повинне передувати вивчення дисциплін:

- *«Математичний аналіз»,*
- *«Алгебра та геометрія»,*
- *«Дискретна математика»,*
- *«Програмування»*
- *«Програмування на мові Python»,*
- *«Архітектура обчислювальних систем».*

Перелік дисциплін які базуються на результатах навчання з даної дисципліни:

- *Математична логіка та теорія алгоритмів*
- *Чисельні методи*
- *Об'єктно-орієнтоване програмування*
- *Алгоритмічні основи обчислювальної геометрії та комп'ютерної графіки*
- *Методи оптимізації*
- *Бази даних*
- *Основи машинного навчання*
- *Алгоритми і системи комп'ютерної математики*

3. Зміст навчальної дисципліни

ВСТУП

Організаційні питання, режим занять, комунікація РСО

Для чого вивчати алгоритми? Мотиваційна лекція із прикладами.

- Множення Карацуби
- Обчислення біноміальних коефіцієнтів.
- Погрішність обчислень на прикладі метода Гауса
- Парадигми програмування.

ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

- Поняття алгоритму
- Поняття про часову і просторову складність
- Оцінка розміру вхідних даних
- Оцінка часу виконання алгоритмів
- Нотації Ландау
- Класи складності.
- Математичний аналіз нерекурсивних та рекурсивних алгоритмів
- Порівняння класів складності.
- Вплив збільшення швидкодії комп'ютера на максимальний розмір задачі.
- Співвідношення класів EXP Poly NP

ОСНОВНІ МЕТОДИ ТА ПІДХОДИ ДО ПОБУДОВИ АЛГОРИТМІВ

- Метод грубої сили
 - Прості сортування
 - Лінійний та бінарний пошук
 - Вичерпний перебір.
- Метод декомпозиції.
 - Сортування злиттям.
 - Приклади алгоритмів декомпозиції
 - Аналіз складності алгоритмів декомпозиції
- Метод зменшення розміру задачі
 - на прикладі бінарного пошуку,
 - метода Гауса і деяких сортувань.
 - генерація комбінаторних об'єктів
- Метод зменшення розміру задачі на постійний множник
 - задача пошуку фальшивої монети
- Алгоритми зі змінним зменшенням розміру.
 - обчислення медіани і задача вибору
 - інтерполяційний пошук
- Метод перетворення
 - Поліпшення вхідних даних
 - Попереднє сортування
 - задача пошуку квитанцій
 - Перевірка унікальності елементів списку
 - Знаходження моди
 - Обчислення детермінантів
 - Обчислення НСК і НСД
 - Приведення до задач про графи.
- Подолання часових і просторових обмежень. Просторово-часовий компроміс
 - сортування підрахунком
 - Динамічне програмування

- Алгоритм генерації чисел Фібоначчі
- Алгоритм Воршала
- Алгоритм Флойда
- Жадібні алгоритми.
 - Задача про рюкзак із функцією запам'ятовування
 - Алгоритм Дейкстри.
 - Алгоритм Прима.
 - Алгоритм Крускала.
 - Підмножини, що не перетинаються, і алгоритми пошуку об'єднань

СТРУКТУРИ ДАНИХ І ДИСЦИПЛІНИ

- Вказівники і основи роботи з ними.
- Абстрактний тип даних.
- Лінійний список і його різновиди.
- Стек. Черга.
- Черга з пріоритетом. Дек.

Організація складних структур на базі структури вектор.

- Зберігання розряджених матриць.
- Частково заповнених масивів.
- Зв'язаний розподіл, таблиці.
- Купи

ОСОБЛИВОСТІ ПРЕДСТАВЛЕННЯ МАТЕМАТИЧНИХ ОБ'ЄКТІВ В ПРОГРАМАХ.

- Множини як абстрактний тип даних,
- Графи та їх представлення.
- Пошук в глибину, пошук в ширину.
- Дерева як структури даних,
 - визначення та застосування,
 - основні операції,
 - методи представлення зберігання
 - Дерево Фібоначчі.
 - Ідеально збалансоване дерево.

ПОНЯТТЯ ПРО ПОШУКОВІ ІНДЕКСИ І СПОСОБИ ЇХ ОРГАНІЗАЦІЇ

- Дерева двійкового пошуку.
 - Балансування дерева DSW алгоритм.
 - RB-tree
 - AVL-tree
- Інші дерева пошуку
 - Splay- tree
 - 23-tree
 - B-tree
- Хешування основні поняття і застосування
 - Колізії хеш функцій
 - Парадокс днів народження
 - Хеш таблиці
 - з ланцюжками
 - відкритою адресацією
 - з комірками

4. Навчальні матеріали та ресурси

Базова література

1. Aho, Alfred V., Jeffrey D. Ullman, and John E. Hopcroft. *Data structures and algorithms.* / 2. Aho, Alfred V., Jeffrey D. Ullman, and John E. Hopcroft. Reading, Mass.: Addison-Wesley, 1996.
2. Knuth, Donald Ervin. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition.* Addison-Wesley, 2019.
3. Knuth, Donald Ervin. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms, 3rd Edition.* Addison-Wesley, 2019.
4. Knuth, Donald Ervin. *The Art of Computer Programming, Vol. 3: Sorting and Searching, 3rd Edition.* Addison-Wesley, 2019.
5. Levitin, Anany. *Introduction to the Design & Analysis of Algorithms.* Pearson, 2012 ISBN-: 978-0132316811. або
6. Сирота С. В. АЛГОРИТМИ І СТРУКТУРИ ДАНИХ на [платформі «Сікорський»](#) [Електронний ресурс] / С. В. Сирота, В. О. Ліскін // УІІТО. – 2021. – Режим доступу до ресурсу: <https://do.ipk.kpi.ua/course/view.php?id=1681>

Допоміжна література

1. *Introduction to Algorithms, Third Edition.* / by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein — The MIT Press, 2009. — 1292 pages. — ISBN 978-0262033848.
2. Niklaus Wirth. *Algorithms and Data Structures:* / N. Wirth — Prentice Hall, 1985. — 288 pages ISBN : 978-0130220059.

Інформаційні ресурси

1. Платформа «Сікорський» <https://do.ipk.kpi.ua/course/view.php?id=1681>
2. Візуалізатори алгоритмів і структур <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
2. Електронний кампус КПІ ім. Ігоря Сікорського. Матеріали з дисципліни «Алгоритми і структури даних». – Режим доступу : <http://login.kpi.ua>

5. Методика опанування навчальної дисципліни (освітнього компонента)

	Лекційні заняття	Практичні заняття
Тиждень 1	Для чого вивчати алгоритми? Мотиваційна лекція із прикладами. - Організаційні питання режим занять, комунікація, РСО - Множення Карацуби - Обчислення біноміальних коефіцієнтів. - Погрішність обчислень на прикладі метода Гауса - Парадигми програмування.	Перше заняття Організаційні питання Основні типи задач Загальний огляд підходів до складання алгоритмів Основи розв'язку алгоритмічної задачі
Тиждень 2	ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ - Поняття алгоритму - Поняття про часову і просторову складність - Оцінка розміру вхідних даних - Оцінка часу виконання алгоритмів - Нотації Ландау - Класи складності.	
Тиждень 3	Основні методи та підходи до побудови алгоритмів - Метод грубої сили - Аналіз складності нерекурсивних та рекурсивних алгоритмів - Порівняння класів складності. - Вплив збільшення швидкодії комп'ютера на максимальний розмір задачі. - Співвідношення класів EXP Poly NP	Метод Грубої сили Сортування вибором і бульбашкове сортування Послідовний пошук і пошук підрядків методом грубої сили Задача пошуку пари найближчих точок методом грубої сили Обчислення опуклої оболонки з використанням грубої сили
Тиждень 4	СТРУКТУРИ ДАНИХ І ДИСЦИПЛІНИ - Вказівники і основи роботи з ними. - Поняття про абстрактний тип даних. Асоціативний масив - Лінійний список і його різновиди. - Стек. Черга. - Черга з пріоритетом. Дек.	
Тиждень 5	Організація складних структур на базі лінійних. - Зберігання розряджених матриць. - Частково заповнених масивів. - Зв'язаний розподіл, таблиці. - Купи	Метод грубої сили з вичерпним перебором Задача комівояжера методом грубої сили Задача про рюкзак методом грубої сили Задача про призначення методом грубої сили

Тиждень 6	Особливості представлення математичних об'єктів в програмах ▪ Множини як абстрактний тип даних, ▪ Графи та їх представлення. ○ з допомогою матриць ○ з використанням спискових структур ▪ Пошук в глибину, пошук в ширину.	
Тиждень 7	Метод декомпозиції. ▪ Приклади алгоритмів декомпозиції ▪ Аналіз складності алгоритмів декомпозиції	Метод декомпозиції Сортування злиттям Швидке сортування Множення великих цілих чисел Задача пошуку пари найближчих точок методом декомпозиції Обчислення опуклої оболонки методом декомпозиції
Тиждень 8	Дерева як структури даних, ▪ визначення та застосування, ▪ основні операції, ▪ методи представлення зберігання ▪ Дерево Фібоначчі. ▪ Ідеально збалансоване дерево.	
Тиждень 9	Метод зменшення розміру задачі на постійний множник ▪ задача пошуку фальшивої монети Алгоритми зі змінним зменшенням розміру. ▪ Алгоритм Евкліда ▪ генерація комбінаторних об'єктів	Метод зменшення розміру задачі на постійний множник Сортування вставкою, сортування Шела Множення по-російськи Топологічне сортування Задача Йосипа
Тиждень 10	ПОНЯТТЯ ПРО ПОШУКОВІ ІНДЕКСИ І СПОСОБИ ЇХ ОРГАНІЗАЦІЇ ▪ Дерева двійкового пошуку. ○ Балансування дерева DSW алгоритм. ○ RB-tree ○ AVL-tree	
Тиждень 11	▪ Інші дерева пошуку ○ Splay- tree ○ 23-tree ○ B-tree	Метод зменшення розміру задачі Генерація перестановок висхідна Генерація перестановок Джнсона-Тротера Генерація підмножин Генерація підмножин Хіпа (HeapPermute)
Тиждень 12	▪ Хешування основні поняття і застосування ○ Колізії хеш функцій ○ Парадокс днів народження ▪ Хеш таблиці ○ з ланцюжками ○ відкритою адресацією ○ з комірками	
Тиждень 13	Метод перетворення	Алгоритми зі змінним зменшенням розміру

	▪ Поліпшення вхідних даних ▪ Попереднє сортування задача пошуку квитанцій ▪ Перевірка унікальності елементів списку	Обчислення медіани і Задача вибору, Інтерполяційний пошук Сортування підрахунком Алгоритм Хорспула Алгоритм Бойера-Мура
Тиждень 14	▪ Знаходження моди ▪ Обчислення НСК і НСД ▪ Приведення до задач про графи.	
Тиждень 15	Подолання часових і просторових обмежень. Просторово-часовий компроміс Динамічне програмування ▪ Алгоритм генерації чисел Фібоначчі ▪ Алгоритм Воршала ▪ Алгоритм Флойда ▪ Задача про рюкзак із функцією запам'ятовування	Здача індивідуальних завдань
Тиждень 16	Жадібні алгоритми. ▪ Алгоритм Дейкстри. ▪ Алгоритм Прима. ▪ Алгоритм Крускала. ▪ Підмножини, що не перетинаються, і алгоритми пошуку об'єднань	
Тиждень 17-18	Здача заліків	

6. Самостійна робота студента

- *Для засвоєння матеріалів курсу передбачені наступні види самостійної роботи*
- *Проходження тематичного тестування з можливістю перегляду лекційного матеріалу, візуалізаторів та фрагментів відеолекцій.*
- *Підготовка доповіді на практичному занятті з написанням коду алгоритму*

Кожному слухачу пропонується зробити доповідь на практичному занятті або виконати самостійне завдання з програмування.

Теми доповідей і завдань та інструкції щодо підготовки викладені на платформі «Сікорський».

• Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- *Відвідування*
- *Відвідування лекцій обов'язкове, ми заохочуємо студентів активно приймати участь в занятті шляхом проходження онлайн тестів на Платформі Сікорський в реальному часі. За відповіді на такі тести додаються заохочувальні бали.*
- *Відвідування практичних занять обов'язкове. Практичні заняття пропущені з поважних причин можуть бути замінені на виконання додаткових завдань за погодженням із викладачем.*
- *Поточний контроль проходить в режимі тестування і має часові обмеження.*
- *Рейтинг студента, отриманий протягом семестру, складається з обов'язкових та додаткових балів. Додаткові бали можуть бути зараховані до рейтингу і компенсувати недоотримані обов'язкові бали у випадку активного відвідування навчальних ресурсів даного курсу і сумлінного виконання завдань.*
- *Вимоги до слухачів курсу базуються на принципах академічної доброчесності і рівності всіх студентів. У випадку виявлення випадків запозичення без відповідних посилань об'єктів авторського права, як то: програмний код, зображення, креслення, та інший мультимедійний контент або виявлення плагіату — бали за відповідні роботи будуть анульовані і нараховані штрафні бали. Повторні порушення принципів академічної доброчесності можуть призвести до недопуску до складання заліку.*
- *Викладачі можуть перевіряти роботи, виконані у рамках курсу, за допомогою систем виявлення плагіату.*

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

PCO передбачає оцінювання результатів навчальної діяльності здобувача впродовж семестру – проходження або виконання певних видів робіт, передбачених заходами поточного контролю

На семестр пропонуються наступні види діяльності за які нараховуються бали.

Обов'язкові бали:

- *6 тематичних тестів до 5 балів кожний сумарно 30 балів. Необмежена кількість спроб, показ правильних і неправильних відповідей, зараховується краща оцінка:*
 - *Тематичний тест "Представлення множин і графів"*
 - *Тематичний тест "Графи і дерева"*
 - *Тематичний тест "Пошук на графах"*
 - *Тематичний тест "Пошук"*
 - *Тематичний тест "Стратегії (методи) проектування алгоритмів"*
 - *Тематичний тест "Пошукові індекси"*

- Доповідь на практичному занятті присвячена тому чи іншому, обраному зі списку, алгоритму або виконання індивідуального творчого завдання, яке передбачає алгоритмічну реалізацію розв'язку запропонованої задачі і надання звіту (30 балів)

Критерії оцінювання доповіді (творчого завдання):

1. Формулювання постановки задачі до 5 балів;
 - Чітке зрозуміле з визначенням об'єктів та цілей – 5 балів;
 - З незначними неточностями – 4 бали;
 - З помилками – 3 бали;
 - З грубими помилками при загальному розумінні задачі – 2 бали
 - Без розуміння предмета і об'єкта – 1 бал;
 - Відсутнє – 0 балів,
 2. Програмний код алгоритму на Python або C до 5 балів;
 - Структурований, коментований, оригінальний працюючий код 5 балів;
 - Код який містить типові помилки пов'язані із зайвими операціями та викликами 4;
 - Код містить помилки, які на певних даних приводять до невірного результату або збою або запозичений код із посиланням на джерело але пояснений і зрозумілий 3 бали;
 - Код, що не вирішує поставлену задачу в повному обсязі, або запозичений без змоги пояснити роботу – 2 бали;
 - Працездатний код з ознаками запозичення, без змоги пояснити його роботу 1 бал.
 - Відсутність коду 0 балів;
 - Доведена спроба здачі запозиченого коду за власний – мінус 5 балів.
 3. Вміння пояснити алгоритм до 5 балів.
 - 5 – чітке і зрозуміле... 0 відсутнє;
 4. Демонстрація роботи програмної реалізації алгоритму до 5 балів;
 - Інтерактивна демонстрація із зручним інтерфейсом на різних прикладах, включаючи краєві, що демонструють обмеження застосування алгоритму – 4 балів;
 - Демонстрація працездатності на окремих прикладах 3 балів;
 - Демонстрація працюючого коду, але непрацездатного коду на деяких вхідних даних за умови вміти пояснити як працює контрольний приклад – 2 балів;
 - Демонстрація незакінченого або такого що не відповідає повністю поставленій задачі – 1 бал;
 - Відсутність 0 балів;
 5. Аналіз складності алгоритму до 10 балів;
 - Обґрунтований, доведений аналіз з чітким визначенням розміру вхідних даних, класу складності, просторових та часових обмежень, максимального розміру задачі – 10 балів;
 - Наявність незначних неточностей 9 балів
 - Необґрунтоване віднесення до правильного класу складності з розумінням просторових і часових обмежень, максимального розміру задачі – 8 балів;
 - Правильне визначення класу складності без розуміння розміру вхідних даних і максимального розміру задачі 5 балів
 - Невміння обґрунтувати віднесення до класу складності але розуміння максимального розміру задачі 4 бали.
 - Відсутність аналізу складності або нерозуміння розміру вхідних даних і класів складності – 0 балів
- Залікове тестування (30 балів). Кількість спроб 2, час обмежений, зараховується краща оцінка показуються лише результати без правильних відповідей:
 - перший поточний тест макс. 15 балів
 - другий поточний тест макс. 15 балів

Якщо умови PCO виконані, але набрано менше 60 балів, або студент хоче вищу оцінку – оцінку може бути підвищено за рахунок виконання залікової контрольної роботи у вигляді окремого фінального тесту (макс. 10 балів) або за рахунок додаткових балів

Додаткові бали:

Бонус / Штраф (максимально від -10 балів до 10 балів).

- Правильна відповідь на поставлене запитання під час занять, до 10 балів.
- Пропозиція оригінального розв'язку заданої задачі, до 10 балів.
- Виявлення змоги видати запозичений програмний код за свій, віднімання до - 10 балів.
- Виявлення спроби недобросовісного проходження тестування - анулювання результатів тесту.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

- Взаємодія викладача зі студентами відбувається з допомогою MOODLE в рамках Платформи «Сікорський» <https://do.ipk.kpi.ua/mod/page/view.php?id=70423>;
- Дистанційні онлайн заняття проводяться за допомогою Google Meet за запрошеннями, які публікуються в кафедральній платформі Slack та дублюються на надані студентами адреси електронної пошти;
- Офіційні звернення до викладача розглядаються через кафедральну платформу Slack або електронну адресу syrota.sergiy@lil.kpi.ua.

Робочу програму навчальної дисципліни (силабус):

Склав доцент кафедри прикладної математики, канд. техн. Наук, доц. Сирота Сергій Вікторович

Ухвалено кафедрою прикладної математики (протокол №14 від 14.05.2025 року)

Погоджено Методичною комісією факультету¹ (протокол №12 від №11 від 23.05.2025 року)

¹ Методичною радою університету – для загальноуніверситетських дисциплін.