



АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ.

Частина 1. Базові алгоритми

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Системи і методи штучного інтелекту
Статус дисципліни	Нормативна
Форма навчання	очна(денна)/дистанційна/змішана
Рік підготовки, семестр	1 курс, осінній семестр
Обсяг дисципліни	2 кредити ЄКТС
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	Четвер 1 пара через тиждень (лекція), вівторок 2,3 пари (лабораторна робота)
Мова викладання	Українська (матеріали також надаються на англійській мові)
Інформація про керівника курсу / викладачів	Лектор: Малишевський Олексій Григорович malyshevskiyi.oleksii@kpi.ua Лабораторні: Клещ Кирило Олегович та Яковчук Олег Костянтинівич iasa_asd_2020_da_labs@ukr.net , Куб'юк Євгеній Юрійович та Касьянчук Ігор Вячеславович iasa.ads.20@gmail.com
Розміщення курсу	https://classroom.google.com/c/MTQ1MDU0MzcwMzE5?cjc=qndj7jg

Програма навчальної дисципліни

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

Поняття алгоритму – це одне з головних понять математики, обчислювальної математики, техніки, оскільки знаходження прийнятного алгоритму для рішення різних класів задач є одним із завдань цих наук.

Успіхи сучасної обчислювальної математики в значній мірі обумовлені принципом поділу складних обчислювальних процесів на елементарні операції – арифметичні та логічні. Це робиться за допомогою послідовних дій, які задаються алгоритмом. Оскільки складність задач обчислювального характеру постійно зростає, актуальною також є задача розробки ефективних алгоритмів обчислювального процесу. В залежності від характеру задач, що вирішуються, використовуються дані різних типів. Це істотно впливає на структуру алгоритмів.

В дисципліні розглядаються алгоритми та структури даних, які є фундаментом сучасної методології розробки програм. В ній вивчаються основні поняття, які визначають суть алгоритмізації процесів обробки інформації, засоби представлення алгоритмів,

викладаються можливі варіанти реалізації алгоритмів та структур даних. Алгоритми записуються на мові C/C++, або у вигляді псевдокоду, з якого вони легко перекладаються на будь-яку мову програмування.

В дисципліні також розглядається поняття ефективності алгоритмів, їх алгоритмічної складності та методологія оцінки часу їх роботи.

Загальні компетентності, а саме ЗК 1 Здатність до абстрактного мислення, аналізу та синтезу, ЗК 2 Здатність застосовувати знання у практичних ситуаціях, ЗК 3 Знання та розуміння предметної області та розуміння професійної діяльності, ЗК 6 Здатність вчитися і оволодівати сучасними знаннями, ФК 3 Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем, ФК 8 Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Програмні результати навчання: ПР1 Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук; ПР5 Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій; ПР9 Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

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

Забезпечуючі дисципліни: "Алгоритмізація та програмування", "Дискретна математика", "Аналітична геометрія та лінійна алгебра".

Дисципліни, які базуються на результатах навчання з даної дисципліни: "Алгоритми та структури даних. Частина 2. Структури даних і структури зберігання", "Паралельні обчислення", "Теорія інформації і кодування", "Інтелектуальний аналіз даних".

3. Зміст навчальної дисципліни

1. Історія алгоритмів. Обчислювальна складність, складні проблеми. Моделі обчислення. Модель RAM.
2. Ефективність алгоритмів: часова та просторова. Інваріанти та доказ коректності алгоритмів.
Концепція дизайну алгоритмів. Підхід розділяй та володарюй (divide-and-conquer). Аналіз часової складності алгоритмів типу розділяй та володарюй.
3. Ріст функцій та асимптотичні нотації: найкраща та найгірша часова складності алгоритмів, Θ -нотація, O -нотація, Ω -нотація, o -нотація, ω -нотація.
4. Структури даних: стеки, черги, зв'язані списки (linked lists), дерева, купи (heaps), черги з пріоритетом (priority queues). Основні операції над структурами даних.

5. Сортування: сортування вставкою (insertion sort), сортування бульбашкою (bubble sort), сортування вибором (selection sort), сортування злиттям (merge sort), сортування купою (heap sort).
6. Множення матриць: ітеративний алгоритм множення матриць, застосування підходу розділяй та володарюй до задачі множення матриць, наївний рекурсивний алгоритм множення матриць, алгоритм Штрассена. Решето Ератосфена.
7. Пошук: лінійний пошук, бінарний пошук, пошук інтерполяцією (interpolation search).

4. Навчальні матеріали та ресурси

Базова література:

1. Т. Кормен, Ч. Лейзерсон, Р. Рівест, К. Стайн, Вступ до Алгоритмів, 3-е вид. : Переклад з англ. — К. : К. І. С., 2019. — 1288 с.

Додаткова література:

2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, Third Edition, The MIT Press, 2009, Cambridge, Massachusetts/London, England, 1292 p.
3. Alfred Aho, Jeffrey Ullman, John Hopcroft, Data Structures and Algorithms, Pearson, 1983. – 448 p.
4. Alfred Aho, Jeffrey Ullman, John Hopcroft, Design and Analysis of Computer Algorithms, Addison-Wesley; 1st edition, 1974, 470 p.
5. Steven S. Skiena, The Algorithm Design Manual, Second Edition, Springer-Verlag London Limited, 2008, 730 p.
6. Udi Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley Publishing Company Inc., 1989, 478 p.
7. Adam Drozdek, Data Structures and Algorithms in C++, Fourth Edition, Cengage Learning, 2012, 784 p.
8. Robert Sedgewick, Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, Third Edition, Addison-Wesley Professional, 1998, 738 p.
9. Robert Sedgewick, Algorithms in C++ Part 5: Graph Algorithms, Third Edition, Addison-Wesley Professional, 2001, 528 p.
10. Donald E. Knuth, Art of Computer Programming, Volume 1: Fundamental Algorithms, Addison-Wesley Professional; 3rd edition, 1997, 650 p.
11. Donald E. Knuth, Art of Computer Programming, Volume 2: Seminumerical Algorithms, Addison-Wesley Professional; 3rd edition, 1997, 762 p.
12. Donald E. Knuth, Art of Computer Programming, Volume 3: Sorting and Searching, Addison-Wesley Professional; 3rd edition, 1998, 780 p.
13. Donald E. Knuth, Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Addison-Wesley Professional; 1st edition, 2011, 3168 p.
14. Jeffrey J. McConnell, The Analysis of Algorithms: An Active Learning Approach, Jones & Bartlett Learning; 1st edition, 2001, 297 p.
15. Richard Heathfield, Lawrence Kirby, C Unleashed, Sams, 2000, 1200 p.
16. Niklaus Wirth, Algorithms + Data Structures = Programs, Prentice-Hall Inc., 1976, 381 p.
17. Chang, Shi-Kuo (2003). Data structures and algorithms. Software Engineering and Knowledge Engineering. 13. Singapore: World Scientific. ISBN 978-981-238-348-8.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, завдання на СРС з посиланням на літературу)
1	<p>Розділ 1. Основи аналізу алгоритмів.</p> <p>Тема 1.1 Вступ. Історія алгоритмів. Основні поняття.</p> <p>Цілі та задачі курсу, його структура, зміст і методичні рекомендації по вивченню. Терміни, означення, основні поняття. Поняття коректності. Обчислювальна складність, складні проблеми. Паралелізм. Моделі обчислення. Модель RAM. Ефективність алгоритмів.</p> <p>Л2, стор. 5-14, Л1; стор. 24-48</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
2	<p>Розділ 2. Ріст функцій, асимптотичні нотації.</p> <p>Тема 2.1 Ріст функцій, асимптотичні нотації.</p> <p>Поняття найкращого та найгіршої часової складності алгоритмів. Порядок росту. Ріст функцій. Асимптотичні нотації. Θ-нотація: визначення, приклади. O-нотація: визначення, приклади. Ω-нотація: визначення, приклади. o-нотація: визначення, приклади. ω-нотація: визначення, приклади. Порівняння нотацій.</p> <p>Л2, стор. 43-53; Л1, стор. 62-72.</p> <p>Завдання на СРС: Повторення лекційного матеріалу. Виконання домашньої роботи з теми "асимптотичні нотації".</p>
3	<p>Розділ 3. Структури даних.</p> <p>Тема 3.1 Динамічні множини. Примітивні структури даних: стеки, черги, зв'язані списки (linked lists), дерева.</p> <p>Основні концепції структур даних. Динамічні множини: концепція, основні операції. Стеки: визначення, операції, приклади, реалізація, часова складність операцій. Черги: визначення, операції, приклади, реалізація, часова складність операцій. Зв'язані списки: визначення, операції, приклади, реалізація, часова складність операцій. Вказівники: поняття, реалізація. Деревя: визначення, операції, приклади, реалізація, часова складність операцій.</p> <p>Л2, стор. 229-248; Л1, стор. 244-267.</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
4	<p>Тема 3.2 Купи (heaps).</p> <p>Купи (heaps): основні поняття, приклади, властивості, часова складність операцій.</p> <p>Л2, стор. 151-159; Л1, стор. 168-176.</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
5	<p>Тема 3.3 Черги з пріоритетом (priority queues).</p> <p>Черги з пріоритетом (priority queues): основні поняття, операції, приклади, властивості, часова складність операцій.</p> <p>Л2, стор. 162-165; Л1, стор. 177-183.</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
6	<p>Розділ 4. Сортування.</p>

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, завдання на СРС з посиланням на літературу)
	<p>Тема 4.1 Сортування. Сортування вставкою (insertion sort). Сортування бульбашкою (bubble sort).</p> <p>Задача сортування, приклади. Сортування вставкою (insertion sort): основна концепція, приклади. Коректність алгоритму сортування вставкою: інваріант циклу, доказ коректності індукцією. Аналіз алгоритмів. Аналіз часової складності алгоритму сортування вставкою: найкращий та найгірший випадки.</p> <p>Сортування бульбашкою (bubble sort). Коректність алгоритму сортування бульбашкою: інваріант циклу, доказ коректності індукцією. Аналіз часової складності алгоритму сортування бульбашкою.</p> <p>Л2, стор. 16-29; Л1, стор. 35-48.</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
7	<p>Тема 4.2 Сортування вибором (selection sort). Сортування злиттям (merge sort).</p> <p>Сортування вибором (selection sort): основна концепція, приклади. Коректність алгоритму сортування вибором: інваріант циклу, доказ коректності індукцією. Аналіз часової складності алгоритму сортування вибором.</p> <p>Концепція дизайну алгоритмів. Підхід розділяй та володарюй (divide-and-conquer): основна концепція, кроки. Аналіз часової складності алгоритмів типу розділяй та володарюй.</p> <p>Сортування злиттям (merge sort): основна концепція, приклади. Процедура злиття (merge): алгоритм, приклади. Коректність процедури злиття: інваріант циклу, доказ коректності індукцією. Аналіз часової складності алгоритму сортування злиттям: рекуренція, дерево рекурсії.</p> <p>Л1, стор. 29-38; Л2, стор. 48-58.</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
8	<p>Тема 4.3 Сортування купою (heap sort).</p> <p>Сортування купою (heap sort): основна концепція, алгоритм, приклади. Аналіз часової складності алгоритму сортування купою.</p> <p>Л2, стор. 159-161; Л1, стор. 159-162.</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
9	<p>Розділ 5. Базові алгоритми.</p> <p>Тема 5.1. Пошук.</p> <p>Лінійний пошук. Бінарний пошук. Пошук інтерполяцією (interpolation search). Умови використання. Час виконання.</p> <p>Л12; Л17.</p> <p>Завдання на СРС: Повторення лекційного матеріалу.</p>
10	<p>Тема 5.2 Множення матриць. Решето Ератосфена.</p> <p>Ітеративний алгоритм множення матриць: алгоритм, часова складність. Застосування підходу розділяй та володарюй до задачі множення матриць. Наївний рекурсивний алгоритм множення матриць: алгоритм, рекуренція, часова складність. Алгоритм Штрассена: алгоритм, рекуренція, часова складність. Асимптотично найшвидші алгоритми множення матриць. Решето Ератосфена: алгоритм, приклади, часова та просторова складність.</p> <p>Л2, стор. 75-82; Л1, стор. 94-101.</p>

№ з/п	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, завдання на СРС з посиланням на літературу)
	Завдання на СРС: Повторення лекційного матеріалу.
11	Контрольна робота

Лабораторні заняття

№ з/п	Назва лабораторної роботи
1	Дослідження базових конструкцій мови програмування С/С++
2	Дослідження базових алгоритмів
3	Дослідження елементарних алгоритмів сортування та пошуку
4	Дослідження структур даних стек та черга
9	Залік

6. Самостійна робота студента

№ з/п	Види робіт, що виносяться на самостійне опрацювання	Кількість годин СРС
1	Підготовка до лекційних занять	12
2	Підготовка до лабораторних робіт	20
3	Повторення лекційного матеріалу та вивчення питань, що винесені на самостійну роботу	10
4	Підготовка до контрольної роботи	9

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Вимоги, яких має дотримуватися студент в рамках даної дисципліни:

- відвідування лекційних та лабораторних занять є бажаним, але не обов'язковим;
- під час проведення занять мобільні телефони мають бути переведені у беззвучний режим;
- дозволяється, при необхідності, використання засобів зв'язку для пошуку потрібної інформації на платформі дистанційного навчання та/або в інтернеті;
- виконання та захист лабораторних робіт є обов'язковим;
- лабораторні роботи мають бути виконані та захищені особисто, під час захисту студент повинен відповісти на питання викладача, що стосуються як самої лабораторної роботи, так і теоретичного матеріалу, на якому вона базується;
- лабораторні роботи повинні бути виконані у відповідності до свого варіанту, який відповідає номеру студента у списку групи, звіт по виконаній роботі повинен бути оформлений згідно вимог, які вказуються викладачами та описані в протоколі до завдання;
- заохочувальні бали можуть призначатися за активність на лабораторних роботах;
- штрафні бали можуть призначатися за несвоєчасне виконання лабораторних робіт, виконання не свого варіанту, списування лабораторних робіт або МКР;
- при виконанні лабораторних робіт потрібно дотримуватися графіка, який доводиться до відома студентів викладачами на початку семестру;
- обов'язковим є дотримання академічної доброчесності.

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

Поточний контроль: лабораторна робота № 1 (15 балів), лабораторна робота № 2 (15 балів), лабораторна робота № 3 (15 балів), лабораторна робота № 4 (15 балів), МКР (40 балів).

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силлабусу.

Семестровий контроль: залік.

Умова допуску до семестрового контролю: зарахування усіх лабораторних робіт, рейтинг за виконання лабораторних робіт не менше 36 балів (60% від усіх можливих балів за лабораторні роботи).

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)

Перелік питань, які виносяться на модульну контрольну роботу:

Основні концепції дизайну алгоритмів,

Підхід розділяй та володарюй (divide-and-conquer),

Аналіз часової складності алгоритмів типу розділяй та володарюй,

ріст функцій та асимптотичні нотації,

Θ-нотація, O-нотація, Ω-нотація, o-нотація, ω-нотація.

стеки,

черги,

зв'язані списки (linked lists),

дерева,

купи (heaps),

черги з пріоритетом (priority queues),

сортування вставкою (insertion sort),

сортування бульбашкою (bubble sort),

сортування вибором (selection sort),

сортування злиттям (merge sort),

сортування купою (heap sort),

ітеративний алгоритм множення матриць,

застосування підходу розділяй та володарюй до задачі множення матриць,

решето Ератосфена.

За погодженням з викладачем, студент має можливість пройти дистанційні чи онлайн курси за відповідною тематикою та зарахувати отримані сертифікати як додаткові бали до рейтингу (не більше 10 балів).

Лекції з курсу “Алгоритми та структури даних” ведуться в системі відеоконференцій Zoom та Google meet відповідно розкладу занять. Студенти мають можливість задавати питання під час відеоконференції та обговорювати питання для уточнення.

Лабораторні роботи кожен студент виконує відповідно індивідуальному завданню. Захист лабораторної роботи проходить індивідуально з кожним студентом в системі Zoom у порядку черги яка задається викладачами. Звіт по лабораторній роботі необхідно надіслати на пошту не пізніше, ніж за день до захисту.

Поточний контроль ведеться у системі Google Sheets, яка доступна для перегляду студентам. Для підтримання комунікацій зі студентами в режимі дистанційної освіти створено Telegram канал, де розміщуються оголошення з організаційних питань, також створено Telegram чати, де студенти задають свої питання.

Робочу програму навчальної дисципліни (силабус):

Складено старший викладач, к. т. н., Малишевський Олексій Григорович

Ухвалено кафедрою системного проектування (протокол № 2 від 30 вересня 2020 р.)

Погоджено Методичною комісією Інституту прикладного системного аналізу (протокол № 2 від 15 жовтня 2020 р.)