

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ:
Декан факультету комп'ютерних
інформаційних технологій

Ігор ЯКИМЕНКО

“ _____ ” _____ 2024 р.

ЗАТВЕРДЖУЮ:
Проректор з
науково-педагогічної роботи

Віктор ОСТРОВЕРХОВ

“ _____ ” _____ 2024 р.

РОБОЧА ПРОГРАМА

з дисципліни «Алгоритми та структури даних»

ступінь вищої освіти – бакалавр

галузь знань – 12 “Інформаційні технології”

спеціальність – 122 „Комп’ютерні науки”

освітньо-професійна програма – „Штучний інтелект”

КАФЕДРА ІНФОРМАЦІЙНО-ОБЧИСЛЮВАЛЬНИХ СИСТЕМ І УПРАВЛІННЯ

| Форма навчання | Курс | Семестр | Лекції (год.) | Лабораторні заняття (год.) | ІРС (год.) | Тренінг (год.) | Самост. робота студ. (год.) | Разом (год.) | Екз. (сем.) |
|----------------|------|---------|---------------|----------------------------|------------|----------------|-----------------------------|--------------|-------------|
| Денна | 2 | 3 | 30 | 30 | 4 | 8 | 78 | 150 | 3 |

Тернопіль – ЗУНУ
2024

Робоча програма складена на основі освітньо–професійної програми «Штучний інтелект» підготовки бакалавра галузі знань 12 «Інформаційні технології» спеціальності 122 – «Комп’ютерні науки», затвердженої Вченою Радою ЗУНУ (протокол № 10 від 23.06.2023 р.).

Робочу програму складено доцентом кафедри інформаційно-обчислювальних систем і управління, к.т.н. Василем КОВАЛЕМ

Робоча програма затверджена на засіданні кафедри інформаційно-обчислювальних систем і управління, протокол №1 від 27 серпня 2024 р.

В.о. завідувача кафедри,
канд. техн. наук, доцент

Надія ВАСИЛЬКІВ

Розглянуто та схвалено групою забезпечення спеціальності «Комп’ютерні науки», протокол №1 від 30 серпня 2024 р.

Голова групи
забезпечення спеціальності,
д-р техн. наук, професор

Мирослав КОМАР

Гарант освітньо-професійної
програми "Штучний інтелект",
канд. техн. наук, доцент

Василь КОВАЛЬ

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ "АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ"

1. Опис дисципліни "Алгоритми та структури даних"

| Дисципліна «Алгоритми та структури даних» | Галузь знань, спеціальність, СВО | Характеристика навчальної дисципліни |
|---|--|--|
| Кількість кредитів – 5 | Галузь знань – 12 “Інформаційні технології” | Статус дисципліни: обов’язкова дисципліна циклу професійної підготовки Мова навчання: Українська |
| Кількість залікових модулів – 5 | Спеціальність – 122 «Комп’ютерні науки» | Рік підготовки: 2 Семестр: 3 |
| Кількість змістових модулів – 4 | Освітньо- професійна програма «Штучний інтелект» | Лекції: 30 год. Лабораторні заняття: 30 год. |
| Загальна кількість годин – 150 | Ступінь вищої освіти – бакалавр | Самостійна робота: 78 год., Тренінг: 8 год. Індивідуальна робота: 4 год. |
| Тижневих годин – 10, з них аудиторних – 4 год. | | Вид підсумкового контролю – екзамен |

2. Мета і завдання дисципліни "Алгоритми та структури даних"

2.1. Мета вивчення дисципліни

Мета вивчення дисципліни "Алгоритми та структури даних" – освоєння фундаментальних принципів побудови та аналізу алгоритмів, сформулювати практичні навички розробки алгоритмів розв'язання прикладних задач їх програмування та математичному обґрунтуванню їх застосування. Вивчення цієї дисципліни передбачає розуміння та засвоєння студентами основних принципів розробки алгоритмів і програм, а також виконання практичної роботи в галузях інформаційних технологій і автоматизації та приладобудування.

Метою проведення лекційних занять є ознайомлення студентів з теоретичними питаннями щодо теорії алгоритмів. В рамках лекційного курсу досліджуються питання обчислювальної складності, структур даних, ефективності алгоритмів, технологій проектування алгоритмів.

Мета проведення лабораторних занять полягає у тому, щоб виробити у студентів практичні навички при дослідженні алгоритмів, включаючи дослідження часової складності, технологій проектування та ефективності алгоритмів.

2.2. Завдання вивчення дисципліни

В даному курсі розглядається вступ в теорію алгоритмів для студентів освітнього ступеню бакалавр. Дисципліна включає розгляд методів побудови та аналізу алгоритмів, оцінки коректності та ефективності алгоритмів. Даний курс починається із розгляду поняття алгоритму, аналізу парадигм для побудови алгоритмів, аналізу складності алгоритмів, також розглядаються сучасні засоби міжоперабельності, рандомізації та узагальнення. В рамках даної дисципліни студенти набудуть практичних навичок в освоєнні принципів роботи основних алгоритмів сортування, пошуку тощо.

2.3. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни.

СК3. 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.

СК8. 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.

2.4. Передумови для вивчення дисципліни.

Для засвоєння дисципліни студентам необхідні знання і навички, отримані при вивченні дисциплін: «Вища математика», «Основи комп'ютерних наук».

2.5. Результати навчання.

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.

ПР16. Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення.

3. Програма навчальної дисципліни «Алгоритми та структури даних»

ЗМІСТОВИЙ МОДУЛЬ 1. Аналіз алгоритмів.

Тема 1. Алгоритм, його зміст і основні властивості. Визначення інформації, алгоритму. Способи описання алгоритмів. Властивості алгоритмів. Математичні основи аналізу алгоритмів. Алгебра висловлювань. Таблиці істинності.

Тема 2. Алгоритмічні стратегії. Комбінування декількох алгоритмічних стратегій. Ітераційні алгоритми. Рекурсивні алгоритми. Зворотні алгоритми. Стратегія гілок і границь (Branch and bound algorithms). Стратегія розподіляй і володарюй. Динамічне програмування. Стратегія жадності (Greedy algorithms). Стратегія грубої сили (Brute force algorithms). Використання евристики.

Тема 3. Структури алгоритмів. Класи алгоритмів. Лінійні, розгалужені та циклічні алгоритми.

Тема 4. Поняття обчислювальної складності.

Тимчасова та просторова складності: асимптотична складність, приклади тимчасової та просторової складності. Класи складності: клас P, клас NP, проблема рівності класів P і NP.

ЗМІСТОВИЙ МОДУЛЬ 2. Структури даних.

Тема 5. Поняття структури даних. Структурні та лінійні типи даних. Поняття структури даних. Рівні описування даних. Класифікація структур даних у програмах користувача та у пам'яті комп'ютера. Основні види складених типів даних. Структури даних у пам'яті комп'ютера.

Тема 6. Статичні напівстатичні та динамічні структури даних. Масиви, множини, структури, бітові структури, таблиці, стеки, черги, лінійні списки, стрічки, нелінійні розгалужені списки

Тема 7. Нелінійні структури даних. Визначення дерева. Бінарне дерево. Подання дерев у пам'яті комп'ютера. Поняття графу. Подання графу у пам'яті комп'ютера.

ЗМІСТОВИЙ МОДУЛЬ 3. Фундаментальні алгоритми та їх побудова.

Тема 8. Алгоритми сортування, злиття та пошуку

Алгоритми сортування: бульбашковий, вставки, вибірки, Shellsort, злиття

(mergesort), Radix-sort.

Алгоритми пошуку: послідовний пошук, бінарний пошук, Brute Force, алгоритм Рабін сортування стрічок, Кнут–Моріс–Прет.

Тема 9. Фундаментальні алгоритми на графах і деревах. Види графів. Представлення графів. Алгоритм пошуку на графах. Визначення дерева. Алгоритми деревовидного пошуку.

ЗМІСТОВИЙ МОДУЛЬ 4. Алгоритми криптографії та кодування даних

Тема 10. Основні криптографічні системи. Прості підстановочні шифри. Криптографічні системи із відкритим ключем. Криптографічні системи із симетричним ключем. Гібридні криптографічні системи.

Тема 11. Криптосистема RSA. Криптографічний алгоритм з відкритим ключем RSA.

Тема 12. Алгоритми кодування даних. Поняття кодування даних. Код Хаффмана. Алгоритми стиснення без втрати інформації.

**4. Структура залікового кредиту з дисципліни
"Алгоритми та структури даних"
Денна форма навчання**

| Тема | <i>Кількість годин</i> | | | | | |
|--|------------------------|---------------------|----------------------|---------|-------------------|----------------------------|
| | Лекції | Лабораторні заняття | Індивідуальна робота | Тренінг | Самостійна робота | Контрольні заходи |
| <i>Змістовий модуль 1 – Аналіз алгоритмів</i> | | | | | | |
| Тема 1. Алгоритм, його зміст і основні властивості | 2 | 2 | 1 | 2 | 2 | Опитування під час заняття |
| Тема 2. Алгоритмічні стратегії | 2 | - | | | 6 | Опитування під час заняття |
| Тема 3. Структури алгоритмів | 2 | 8 | | | 6 | Опитування під час заняття |
| Тема 4. Поняття обчислювальної складності | 4 | 4 | | | 6 | Опитування під час заняття |
| <i>Змістовий модуль 2 – Структури даних</i> | | | | | | |
| Тема 5. Поняття структури даних | 2 | 4 | 1 | 2 | 6 | Опитування під час заняття |
| Тема 6. Статичні напівстатичні та динамічні структури даних | 2 | - | | | 8 | Опитування під час заняття |
| Тема 7. Нелінійні структури даних | 2 | - | | | 8 | Опитування під час заняття |
| <i>Змістовий модуль 3 – Фундаментальні алгоритми та їх побудова</i> | | | | | | |
| Тема 8. Алгоритми сортування, злиття та пошуку | 4 | 8 | 1 | 2 | 6 | Опитування під час заняття |
| Тема 9. Фундаментальні алгоритми на графах і деревах | 4 | - | | | 6 | Опитування під час заняття |
| <i>Змістовий модуль 4 – Алгоритми криптографії та кодування даних</i> | | | | | | |
| Тема 10. Основні криптографічні системи | 2 | 4 | 1 | 2 | 8 | Опитування під час заняття |
| Тема 11. Криптосистема RSA | 2 | - | | | 8 | Опитування під час заняття |
| Тема 12. Алгоритми кодування даних | 2 | - | | | 8 | Опитування під час заняття |
| Разом | 30 | 30 | 4 | 8 | 78 | |

5. Тематика лабораторних занять.

1. Математичні основи аналізу алгоритмів.
2. Побудова лінійних алгоритмів.
3. Розгалужені алгоритмічні структури.
4. Циклічні алгоритмічні структури.
5. Комбіновані алгоритмічні структури.
6. Алгоритми сортування вибором.
7. Алгоритм сортування методом Шелла.
8. Алгоритм сортування злиттям.
9. Алгоритми пошуку.
10. Аналіз складності алгоритмів.
11. Імплементация алгоритмічних структур на мові програмування.
12. Алгоритми в криптосистемах.

6. Самостійна робота.

Самостійна робота студентів з дисципліни «Алгоритми та структури даних» передбачає роботу над розширеним завданням у формі проекту, що охоплює вивчення матеріалів обраної теми, представлення стратегії рішення та алгоритмічного розв'язку практичної задачі. Тема завдання обирається студентом за погодженням із викладачем. Нижче зазначено пропонований перелік тем. Результати виконання проекту оформляються студентами у вигляді звіту, що представляється викладачу до захисту у визначені терміни. Захист повинен супроводжуватись демонстрацією із використанням технічних засобів.

Оцінювання проекту виконується комплексно із врахуванням наступних критеріїв: глибина дослідження та аналізу теми, оригінальність підходу до розв'язання, побудова схеми алгоритму, перевірка алгоритму тестування.

Перелік тем:

1. Аналіз часової складності алгоритмів сортування
Завдання: Пояснити та порівняти часову складність алгоритмів сортування (бульбашкою, швидке сортування, сортування злиттям, сортування вибором). Зробити висновки щодо їх ефективності для різних типів вхідних даних (вже відсортованих, майже відсортованих, випадкових).
2. Рекурсивні алгоритми і їх оптимізація
Завдання: Дослідити принцип роботи рекурсивних алгоритмів (наприклад, алгоритм Ханойської вежі або алгоритм Евкліда). Проаналізувати їх часову та просторову складність, а також можливі методи оптимізації за допомогою мемоізації або переведення рекурсії у цикл.
3. Алгоритми з розділяй та володарюй
Завдання: Розглянути основні принципи алгоритмів "розділяй та володарюй" (divide and conquer). Пояснити на прикладі алгоритмів сортування злиттям та пошуку максимального підмасиву, як цей підхід допомагає покращити ефективність алгоритмів.
4. NP-повні задачі
Завдання: Визначити клас NP-повних задач і пояснити їх основні характеристики. Розглянути декілька класичних NP-повних задач (наприклад, задача комівояжера,

задача упаковки рюкзака) та дослідити, чому не існує відомих ефективних рішень для цих задач.

5. Порівняння жадібних алгоритмів та динамічного програмування

Завдання: Пояснити різницю між жадібними алгоритмами та алгоритмами динамічного програмування. На прикладі задачі про рюкзак і задачі про знаходження найкоротшого шляху проаналізувати, в яких випадках жадібні алгоритми не можуть дати оптимальне рішення.

6. Алгоритми пошуку на графах: BFS vs DFS

Завдання: Пояснити основні відмінності між алгоритмами пошуку в ширину (BFS) та пошуку в глибину (DFS). Навести приклади задач, для яких один із цих підходів буде ефективнішим за інший.

7. Аналіз складності алгоритмів з використанням оціночних символів

Завдання: Розглянути методи оцінки часової та просторової складності алгоритмів за допомогою "O-велике", "o-мале", " Ω ", " θ ". Пояснити, як ці символи використовуються для оцінки продуктивності алгоритмів.

8. Розбір жадібного алгоритму для задачі про рюкзак

Завдання: Розглянути жадібний алгоритм для задачі про рюкзак з дробовими предметами. Пояснити, чому цей підхід працює для дробової версії задачі, але не підходить для цілочислового рюкзака.

9. Розбір алгоритму Крускала та Пріма для пошуку мінімального кістякового дерева

Завдання: Порівняти алгоритми Крускала та Пріма для побудови мінімального кістякового дерева. Пояснити принципи роботи обох алгоритмів, їх часову складність і випадки, в яких кожен із них буде ефективнішим.

10. Задачі на графах: обхід Ейлера та Гамільтона

Завдання: Пояснити відмінність між Ейлеровим та Гамільтоновим шляхами в графах. Розглянути алгоритми для знаходження Ейлерових шляхів та розібрати, чому знаходження Гамільтонових шляхів є значно складнішою задачею.

11. Вивчення складності алгоритмів (часова та просторова).

Завдання: Провести порівняння ефективності алгоритмів сортування (порівняльне сортування, сортування вставками, сортування злиттям) на базі різних вхідних даних. Написати звіт із графіками продуктивності алгоритмів.

12. Реалізація класичних структур даних: стек, черга, список, дерево.

Завдання: Реалізувати одну з структур даних (наприклад, бінарне дерево пошуку або зв'язаний список) на вибраній мові програмування. Додати методи пошуку, вставки, видалення елементів.

13. Пошукові алгоритми (лінійний та бінарний пошук) і алгоритми сортування.

Завдання: Реалізувати та порівняти алгоритми бінарного і лінійного пошуку на масиві. Проаналізувати їх ефективність на масивах різних розмірів.

14. Алгоритми на графах (обхід у глибину, обхід у ширину, алгоритм Дейкстри).

Завдання: Реалізувати один із алгоритмів (наприклад, обхід графа в ширину або глибину). Протестувати алгоритм на реальних або згенерованих графах.

15. Принципи роботи хеш-таблиць.

Завдання: Реалізувати хеш-таблицю з відкритою адресацією або методом ланцюгів. Додати функції для вставки, видалення і пошуку елементів.

16. Поліпшення ефективності алгоритмів за допомогою різних підходів.

Завдання: Оптимізувати існуючий алгоритм, наприклад, сортування або пошуку, та

описати результат перед і після оптимізації.

17. Розв'язання задач з використанням динамічного програмування.

Завдання: Реалізувати алгоритм для вирішення задачі, що піддається динамічному програмуванню (наприклад, задача про рюкзак або найкоротший шлях у графі).

18. Використання алгоритмів для комбінаторних задач.

Завдання: Написати алгоритм для генерації всіх перестановок або підмножин із даного набору елементів. Описати ефективність алгоритму.

19. Алгоритми для паралельної обробки даних.

Завдання: Розробити або дослідити алгоритм для розподілених систем, наприклад, MapReduce.

20. Дослідження алгоритмів штучного інтелекту. Алгоритми пошуку шляху або ігрові алгоритми.

Завдання: Реалізувати алгоритм A^* для пошуку шляху в графі або мінімакс для гри з нульовою сумою.

21. Аналіз амортизованої складності

Завдання: Пояснити концепцію амортизованої складності алгоритмів. Розглянути приклад структури даних (наприклад, динамічний масив) і проаналізувати її амортизовану вартість операцій вставки та видалення.

22. Доказ коректності алгоритмів

Завдання: Вибрати один з алгоритмів (наприклад, алгоритм сортування злиттям або алгоритм пошуку найкоротшого шляху за Беллманом-Фордом) і надати формальний доказ його коректності.

23. Розбір жадібного алгоритму для задачі розфарбовування графа

Завдання: Описати жадібний алгоритм для задачі розфарбовування графа. Дослідити, чому цей алгоритм не завжди дає оптимальне рішення, і в яких випадках він може бути ефективним.

24. Алгоритми планування: задача про оптимальний розклад завдань

Завдання: Дослідити класичну задачу планування завдань (наприклад, задача з дедлайнами або задача мінімізації часу виконання). Пояснити, які алгоритми використовуються для її розв'язання та які їх часові складності.

25. Аналіз розбиття на підзадачі у динамічному програмуванні

Завдання: Дослідити метод розбиття складних задач на підзадачі у динамічному програмуванні. На прикладі задачі про найдовшу спільну підпоследовність або пошук оптимального розбиття матриць пояснити, як це підвищує ефективність алгоритму.

26. Алгоритм Флойда-Воршелла: аналіз і застосування

Завдання: Пояснити алгоритм Флойда-Воршелла для знаходження всіх найкоротших шляхів у зваженому графі. Розглянути його часову складність і порівняти з іншими алгоритмами для графів (наприклад, з алгоритмом Дейкстри).

27. Алгоритми обходу графів: переваги та недоліки різних підходів

Завдання: Порівняти різні алгоритми обходу графів (BFS, DFS, алгоритм A^*). Пояснити, в яких задачах кожен із цих підходів буде найбільш ефективним, а в яких — менш ефективним.

28. Алгоритми пошуку максимального потоку в мережі

Завдання: Пояснити алгоритм Форда-Фалкерсона для знаходження максимального потоку в мережі. Описати його застосування в реальних задачах, таких як задача про транспортні мережі або задача про призначення.

29. Аналіз конкурентних алгоритмів (онлайн алгоритми)

Завдання: Дослідити поняття онлайн алгоритмів, які працюють в реальному часі з неповними даними. На прикладі задачі про бінарне пошукове дерево або задачі пошуку сторінки в кеші, пояснити, як працюють конкурентні алгоритми.

30. Алгоритми збалансованих дерев

Завдання: Пояснити роботу збалансованих дерев (AVL-дерево, червоно-чорне дерево). Порівняти їх із бінарними деревами пошуку з точки зору вставки, видалення і пошуку елементів. Дослідити переваги та недоліки кожної структури.

7. Тренінг з дисципліни.

Тема тренінгу: Програмно-технічна імплементація алгоритмічних структур.

У рамках тренінгу необхідно програмно, або з допомогою технічних засобів імплементувати алгоритм рішення задач та здійснити аналіз ефективності. Тренінг виконується в напрямку розвитку завдання та за темою самостійної роботи, яку погоджується із викладачем.

Виконання завдання тренінгу передбачає наступне:

1. Обґрунтування вибору програмно-технічного середовища реалізації алгоритму рішення задачі.
2. Написання програмного коду, що реалізує схему алгоритму.
3. Проведення процедур відлагодження та тестування.
4. Оцінка складності.

Порядок проведення тренінгу:

1. Вступна частина: ознайомлення студентів з темою тренінгового заняття та погодження варіанту із викладачем.
2. Практична частина: виконання завдань тренінгу згідно з індивідуальним варіантом, що визначений під час виконання самостійної роботи у розділі 6; оформлення короткого звіту.
3. Підведення підсумків: обговорення результатів виконаних завдань.

8. Критерії, форми поточного та підсумкового контролю.

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Алгоритми та структури даних» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

| Модуль 1 | | Модуль 2 | | Модуль 3 | Модуль 4 | Модуль 5 |
|--|---|---|---|---|---|--|
| 10 % | 10 % | 10 % | 10 % | 5% | 15% | 40% |
| Поточне оцінювання | Модульний контроль 1 | Поточне оцінювання | Модульний контроль 2 | Тренінг. | Самостійна робота. | Екзамен. |
| Оцінювання виконання лабораторних робіт (4 роботи) | - 10 тестових питань - одне практичне завдання | Оцінювання виконання лабораторних робіт (8 робіт) | - 20 тестових питань - Одне практичне завдання | Виконання завдань тренінгу (4 завдання) | Виконання завдання для самостійної роботи | - 24 тестових питання - Одне практичне завдання |

Оцінка за «Поточне оцінювання» визначається як середнє арифметичне з оцінок, отриманих під час лабораторних занять. Пропуски лабораторних занять обов'язково відпрацьовуються в години консультацій, в іншому випадку вони

вважаються оцінкою «0» та враховуються при визначенні середнього арифметичного. Для здобувачів, які навчаються за індивідуальним графіком, поточне оцінювання проводиться під час консультацій, та шляхом виконання лабораторних робіт та демонстрації їх в системі Moodle.

Модуль «Тренінг» визначається як середнє арифметичне з оцінок, отриманих під час тренінгу.

Модуль «Самостійна робота» оцінюється як середнє арифметичне з двох оцінок, отриманих під час виконання наскрізного проекту: за якість виконаного звіту виконаного завдання та рівень представлення і захисту результатів, відповідно до критеріїв, визначених у розділі 6 «Самостійна робота» цієї робочої програми.

Шкала оцінювання:

| За шкалою Університет | За національною | За шкалою ECT8 |
|-----------------------|-----------------|--|
| 90-100 | відмінно | A (відмінно) |
| 85-89 | добре | B (дуже добре) |
| 75-84 | | C (добре) |
| 65-74 | задовільно | D (задовільно) |
| 60-64 | | E (достатньо) |
| 35-59 | незадовільно | FX (незадовільно з можливістю повторного |
| 1-34 | | F (незадовільно з обов'язковим повторним |

9. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

| № | Найменування |
|---|--|
| 1 | Комп'ютери з доступом до мережі Інтернет |
| 2 | Середовище Python |
| 3 | Пакет програм MS Office |

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Sergienko Ivan V. Elements of the General Theory of Optimal Algorithms / Ivan V. Sergienko, Valeriy K. Zadiraka, Oleg M. Lytvyn, - Springer International Publishing, 2022. – 378 с.
2. Donald Ervin Knuth. Art of Computer Programming, Volume 4B, The: Combinatorial Algorithms, Addison-Wesley Professional; 1st edition, 2022. – 732 pp.
3. Крєневич Андрій. Алгоритми та структури даних / підручник. — Київ: ВПЦ "Київський Університет", 2021. — 200 с.
4. Steven S. Skiena. The Algorithm Design Manual. Third Edition, - Springer Cham, 2020. – 793pp.
5. Tor Lattimore. Bandit Algorithms / Tor Lattimore, Csaba Szepesvari, - Cambridge University Press, 2020. – 536 pp.
6. Jay Wengrow. A Common-Sense Guide to Data Structures and Algorithms: Level Up Your Core Programming Skills [2 ed.], - The Pragmatic Programmers, LLC, 2020. – 481 pp.

7. Шаховська Н.Б., Голощук Р.О. Алгоритми і структури даних: посібник. – Львів: Магнолія-2006, 2020. - 215 с.
8. Кормен, Томас Г. Вступ до алгоритмів : Переклад з англійської третього видання : [укр.] = Introduction to Algorithms : Third Edition : [пер. з англ.] / Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн, — К.: К. І. С., 2019. - 1288 с.
9. Rod Stephens. Essential Algorithms: A Practical Approach to Computer Algorithms Using Python® and C# - John Wiley & Sons, Inc., 2019. – 800 pp.
- 10.Бородкіна І.Л. Теорія алгоритмів: посібник для студентів вищих навчальних закладів., – К. ТОВ "Видавничий дім "Центр учбової літератури" (ЦУЛ), 2019. – 184 с.
- 11.Rex Porbasas Flejoles. Introduction to Search Algorithms, - Canada: Arcler Press, 2019. – 256 pp.
- 12.Sriraman Sridharan. Foundations of Discrete Mathematics with Algorithms and Programming / R. Balakrishnan, Sriraman Sridharan. – NW: CRC Press, 2019. – 535 pp.