

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра моделювання та програмного забезпечення

ЗАТВЕРДЖУЮ

Перший проректор

_____ Владислав ЧУБАРОВ

“ _____ ” _____ 2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Алгоритми та структури даних

(шифр і назва навчальної дисципліни)

спеціальність

121 – «Інженерія програмного забезпечення»

(шифр і назва напрямку підготовки)

факультет

Інформаційних технологій

(назва інституту, факультету, відділення)

Форма навчання	Курс	Семестр	Всього годин за планом	Кількість національних кредитів	Всього аудиторних годин	Аудиторних годин, (у тому числі КЗ)		Самостійна робота (год.)	Контрольно-модульні роботи	Залік (сем.)	Екзамен (сем.)
						Лекції	Лабораторні				
Денна	2	3	150	5	48	16	32	102	-		*
Денна скорочена	1	1	150	5	48	16	32	102	-		*
Заочна	2	3	150	5	12	6	6	138	-		*
Заочна скорочена	1	1	150	5	12	6	6	138	-		*

Кривий Ріг – 2025 рік

Робочу програму навчальної дисципліни «Алгоритми та структури даних» для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення» розроблено згідно з ОПП галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення».

Розробники: старший викладач кафедри МПЗ Рибальченко О.Г.
старший розробник програмного забезпечення
відділу розробки програмного забезпечення ІТ-продуктів
ТОВ «Автодок Юкрейн» Турчик Б. Л.

Робоча програма затверджена на засіданні кафедри моделювання та програмного забезпечення

Протокол від “ 22 ” листопада 2024 року № 4

Завідувач кафедри МПЗ, доцент, к.п.н. _____ Андрій СТРЮК

Схвалено вченою радою факультету інформаційних технологій

Протокол від “26” грудня 2024 року № 5

Голова вченої ради _____ Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від “ 22 ” листопада 2024 року № 4

Гарант ОПП _____ Андрій СТРЮК

ЗМІСТ

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	4
2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	5
3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	6
4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	7
5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ.....	8
6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ.....	8
7. САМОСТІЙНА РОБОТА.....	9
8. МЕТОДИ НАВЧАННЯ.....	10
9. МЕТОДИ КОНТРОЛЮ.....	11
10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ.....	11
11. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ.	15
12. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ.....	17
13. ІНФОРМАЦІЙНІ РЕСУРСИ.....	17
14. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК	18
Додаток до робочої програми. Робочий план.....	21

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни			
		денна форма навчання	заочна форма навчання	денна скорочена форма навчання	заочна скорочена форма навчання
Кількість кредитів – 5	Галузь знань <u>12 Інформаційні технології</u> (шифр і назва)	Нормативна			
Модулів – 1	Спеціальність <u>121 Інженерія програмного забезпечення</u> (код та найменування спеціальності)	Рік підготовки:			
Змістових модулів – 1		2024/2025 н.р.	2024/2025 н.р.	2024/2025 н.р.	2024/2025 н.р.
Загальна кількість годин - 150		Семестр			
		3-й	3-й	1-й	1-й
Тижневих годин для денної форми навчання: аудиторних – 3 самостійної роботи студента – 6,37	Ступінь вищої освіти: <u>бакалавр</u>	Лекції			
		16 год.	6 год.	16 год.	6 год.
		Практичні, семінарські			
		-	-	-	-
		Лабораторні			
		32 год.	6 год.	32 год.	6 год.
		Самостійна робота			
		102 год.	138 год.	102 год.	138 год.
Вид контролю					
Екзамен – 3 семестр		Екзамен – 1 семестр			

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

- для денної форми навчання – 0,47
- для денної скороченої форми навчання – 0,47
- для заочної форми навчання – 0,087
- для заочної скороченої форми навчання – 0,087

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Метою дисципліни «Алгоритми та структури даних» є ознайомлення студентів із основними класами алгоритмів, оволодіння методикою їх аналізу та розробки; вивчення студентами типових абстрактних структур даних, що мають широке застосування при розробці прикладних програм, та методів їх обробки, вироблення та закріплення навичок роботи з ними.

2.2. Основними завданнями вивчення дисципліни «Алгоритми та структури даних» є

- розвиток теоретичних та практичних навичок розробки, застосування та аналізу алгоритмів при розв'язанні поставлених задач;
- володіння основними прийомами роботи зі структурами даних, реалізація їх на мові програмування.

2.3. В результаті опанування дисципліни студенти освоюють наступні компетентності:

загальні

- ЗК01 Здатність до абстрактного мислення, аналізу та синтезу;
- ЗК02 Здатність застосовувати знання у практичних ситуаціях;

фахові

- СК03 Здатність розробляти архітектури, модулі та компоненти програмних систем;
- СК08 Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- СК14 Здатність до алгоритмічного та логічного мислення.

2.4 **Програмні результати навчання** освітньої програми, яким відповідає дисципліна:

ПР05 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПР06 Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПР07 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПР13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПР15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

В результаті вивчення дисципліни здобувач повинен **знати**:

- основні алгоритми та структури даних;
- базові поняття теорії алгоритмів;
- питання обчислюваності, розв'язності та нерозв'язності масових задач;
- поняття часової та просторової складності алгоритмів при розв'язанні обчислювальних задач;
- принципи проектування алгоритму «зверху-вниз» та покрокового уточнення алгоритму;

вміти:

- використовувати, розробляти та досліджувати математичні методи та алгоритми обробки даних;
- використовувати, розробляти та досліджувати алгоритми розв'язування задач моделювання об'єктів і процесів інформатизації;
- використовувати, розробляти та досліджувати алгоритми функціонування комп'ютеризованих систем методами неперервної, дискретної математики, математичної логіки тощо;
- оцінювати складові ефективності алгоритмів функціонування комп'ютеризованих систем;
- правильно обирати структуру даних для конкретної задачі;
- розробляти алгоритм відповідно до структури даних;
- використовувати рекурсивні структури даних, рекурсивні алгоритми.

2.5. Міждисциплінарні зв'язки

При вивченні дисципліни використовуються знання здобувачів з дисциплін «Алгоритмізація обчислювальних процесів», «Основи програмування», «Вища математика».

Знання, одержані здобувачами при вивченні дисципліни, використовуються при вивченні дисциплін «Дискретні структури», «Об'єктно-орієнтоване програмування» з курсовою роботою, «Розробка програм на платформі .NET», «Програмування на основі JAVA технологій», «Бази даних» з курсовою роботою.

Вимоги до знань та умінь визначаються галузевими стандартами вищої освіти України.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Модуль 1. Структури даних та алгоритми їх обробки

Тема 1 – 2 год. Вступ. Структури даних

Поняття структури даних та абстрактного типу даних. Класифікація структур даних. Способи реалізації структур даних та абстрактного типу даних. Приклади базових структур даних. Масиви.

Тема 2 – 2 год. Математичні основи аналізу алгоритмів

Поняття аналізу. Класифікація методів аналізу. Емпіричний аналіз: допущення по використанню, порівняння декількох алгоритмів, застосування. Асимптотичний аналіз. Швидкість росту функції. Побудова аналітичного виразу, що оцінює ефективність алгоритму. «Тета-нотація» (θ -нотація), O -нотація та Ω -нотація («омега-нотація»). Лінійний ріст, логарифмічний ріст, квадратичний ріст, експоненціальний ріст.

Тема 3 – 4 год. Сортування. Алгоритми сортування масивів

Прості та вдосконалені алгоритми сортування масивів. Сортування простими включеннями, простим вибором, простим обміном. Шейкерне сортування. Сортування Шелла. Швидке сортування. Сортування злиттям. Пірамідальне сортування. Порівняння різних методів сортування.

Тема 4 – 2 год. Пошук. Алгоритми пошуку даних

Поняття пошуку. Класифікація алгоритмів пошуку. Поняття ротації та її використання. Опис алгоритмів пошуку: послідовний, бінарний, інтерполяційний, індексуванням по ключу, дерево бінарного пошуку (BST), порозрядний пошук (DST-дерева та TST-дерева). Область застосування.

Тема 5 – 2 год. Динамічні структури даних

Зв'язні списки. Стеки. Черги. Базові алгоритми для роботи зі списками, стеками, чергами.

Тема 6 – 2 год. Бінарні дерева

Поняття рекурсії. Особливості реалізації рекурсивних алгоритмів. Бінарні дерева. Ідеально збалансовані дерева. Дерева пошуку. Червоно-чорні дерева. Збалансовані AVL-дерева. Основні алгоритми для роботи з бінарними деревами. Приклад використання рекурсії для структури даних "Дерево".

Тема 7 – 2 год. Хеш-таблиці

Поняття хешування та хеш-коду. Методи хешування. Хеш-функції. Способи реалізації хеш-таблиці. Застосування хеш-таблиці. Таблиці з прямою адресацією. Уникнення колізій за допомогою ланцюгів. Відкрита адресація.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	Денна/денна скорочена форма				Заочна/заочна скорочена форма			
	усього	у тому числі			усього	у тому числі		
		лекції	лабораторні	самостійна робота		лекції	лабораторні	самостійна робота
1	2	3	4	5	6	7	8	9
Модуль 1								
Тема 1. Вступ. Структури даних.	6	1	-	5	6	-	-	6
Тема 2. Математичні основи аналізу алгоритмів	6	1	-	5	6	1	-	5
Тема 3. Сортування. Алгоритми сортування масивів	24	4	4	16	24	1	1	22
Тема 4. Пошук. Алгоритми пошуку даних	18	2	4	12	18	1	1	16
Тема 5. Динамічні структури даних	48	4	12	32	48	1	2	45
Тема 6. Бінарні дерева	18	2	4	12	18	1	1	16
Тема 7. Хеш-таблиці	30	2	8	20	30	1	1	28
Усього годин за модуль 1:	150	16	32	102	150	6	6	138

5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено.

6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин	
		Денна/денна скорочена ф. н.	Заочна/заочна скорочена ф. н.
1.	Лабораторна робота №1. Алгоритми сортування. Експериментальне дослідження ефективності різних методів сортування	4	1
2.	Лабораторна робота №2. Алгоритми пошуку. Експериментальне дослідження ефективності різних методів пошуку	4	1
3.	Лабораторна робота №3. Однозв'язні списки, основні функції для роботи з ними. Абстрактний тип даних «Стек»	4	1
4.	Лабораторна робота №4. Двоzv'язні списки, основні функції для роботи з ними	4	-
5	Лабораторна робота №5. Абстрактний тип даних «Черга». Реалізація на основі двозв'язного списку	4	1
6.	Лабораторна робота №6. Абстрактний тип даних «Бінарне дерево»	4	1
7.	Лабораторна робота №7. Бінарні дерева пошуку та хеш-таблиці. Експериментальне дослідження ефективності використання цих АТД.	8	1
РАЗОМ		32	6

7. САМОСТІЙНА РОБОТА

На самостійну роботу студентам денної форми навчання відведено 102 годин, заочної - 132 годин.

Самостійна робота студентів при вивченні дисципліни «Алгоритми та структури даних» залучає такі складові:

- опрацювання теоретичних основ прослуханого лекційного матеріалу;
- вивчення окремих тем або питань, що передбачені для самостійного опрацювання;
- підготовка до виконання, а також до захисту лабораторних робіт;
- підготовка до проведення контрольних заходів.

При виконанні самостійної роботи студент денної форми навчання повинен:

- опрацювати матеріал, що винесено на самостійне вивчення;
- виконати лабораторні роботи та підготуватись до їх захисту;
- оформити звіт з лабораторних робіт.

При виконанні самостійної роботи студент заочної форми навчання повинен:

- опрацювати теоретичний матеріал;
- виконати лабораторні роботи та підготуватись до їх захисту;
- оформити звіт з лабораторних робіт.

Питання для самостійного опрацювання

№	Назва теми	Кількість годин			
		денна	денна скороч.	заочна	заочна скороч.
1.	Тема 1. Представлення спеціалізованих структур за допомогою масивів.	5	5	6	6
2.	Тема 2. Характеристики алгоритмів: швидкість роботи, пам'ять, стійкість. Нижня оцінка алгоритмів сортування, які засновані на порівняннях.	5	5	5	5
3.	Тема 3. Сортування за лінійний час. Сортування підрахунком (counting sort). Сортування за розрядами (radix sort). Рандомізоване швидке сортування.	10	10	15	15
4.	Тема 5. Черги з пріоритетами. Представлення черги за допомогою масивів та зв'язних списків.	12	12	20	20
5.	Тема 6. Реалізація алгоритмів обходу дерев мовами програмування. Реалізація алгоритмів балансування червоно-чорних та AVL-дерев мовами програмування.	12	12	22	22
6.	Тема 7. Метод ланцюгів. Відкрита адресація	14	14	20	20
7.	Опрацювання матеріалу лекцій, робота з рекомендованою літературою	16	16	16	16
8.	Виконання лабораторних робіт	14	14	14	14
9.	Підготовка звітів з лабораторних робіт	14	14	14	14
	Разом	102	102	132	132

8. МЕТОДИ НАВЧАННЯ

Використовуються наступні методи навчання: лекції, лабораторні заняття, самостійна робота.

Навчальна лекція – це логічне, послідовне викладання змісту навчання, яке характеризується судженнями, висновками, підсумком. Вона охоплює основний теоретичний матеріал однієї або кількох тем навчальної дисципліни. Призначенням лекції є формування у здобувачів фундаментальних знань з дисципліни, а також визначає основний зміст і характер усіх інших навчальних занять та самостійної роботи здобувачів із цієї дисципліни.

Лабораторне заняття - форма організації навчання, яку проводять за завданням і під керівництвом НПП. Основні дидактичні цілі – експериментальне підтвердження вивчених теоретичних положень навчальної дисципліни та формування вмінь й навичок їх практичного застосування. Проведення лабораторного заняття ґрунтується на попередньо підготовлених наборах завдань різної складності для розв’язання на занятті. Лабораторне заняття проводиться у навчальних лабораторіях з використанням пристосованого до умов навчального процесу устаткування.

Самостійна робота здобувача є основним способом оволодіння навчальним матеріалом у час, вільний від обов’язкових аудиторних занять. Мета виконання самостійної роботи – поглиблення, узагальнення й закріплення теоретичних знань і практичних умінь здобувачів із дисципліни шляхом вироблення вміння самостійної роботи з навчальною і фаховою літературою та інформацією в мережі Інтернет.

Самостійна робота здобувачів здійснюється у формі: підготовки до лекцій і лабораторних занять, виконанні самостійних проєктів. Самостійну роботу здобувач може виконувати у бібліотеці, комп’ютерних класах, а також у домашніх умовах.

Підготовка до лекцій передбачає самостійне опрацювання теоретичного матеріалу. При цьому необхідно звернути увагу на необхідність чіткого засвоєння основних термінів та визначень, розуміння їх змісту, обов’язкового аналізу використання теоретичних положень для розв’язання наданих прикладів.

Самоперевірку засвоєння навчального матеріалу здобувач здійснює за контрольними запитаннями, що надано після кожної теми у конспекті лекцій та іншій літературі, та після кожного лабораторного заняття у відповідних методичних вказівках. Якщо на деякі запитання здобувач не може надати відповіді, то необхідно повторити вивчення навчального матеріалу, або визначити правильну відповідь з викладачем на консультації.

Під час вивчення даної дисципліни використовуються:

- мультимедійні освітні технології: інтерактивні лекції (презентації) із використанням програми MS Power Point у поєднанні з анімацією та звуковим супроводом; перегляд відеороликів за окремими пунктами тем занять, використання електронних посібників;
- діалогові технології: організація групових обговорень, використання «мозкового штурму».

Лекції проводяться з використанням технічних засобів навчання й супроводжуються демонстрацією презентацій за допомогою проектора.

У разі виникнення необхідності забезпечення навчального процесу в дистанційному режимі супровід та контроль знань реалізовується за допомогою дистанційного курсу, розробленого в Google Classroom. Онлайн лекції, консультації та усні відповіді на питання, захист проєктів проводиться за допомогою Google Meet або Zoom.

9. МЕТОДИ КОНТРОЛЮ

Основними завданнями контролю знань здобувачів вищої освіти з дисципліни є оцінювання засвоєння теоретичних знань і практичних навичок, отриманих під час навчання.

Контрольні заходи мають виконувати наступні функції:

- стимулювати систематичну самостійну роботу над навчальним матеріалом;
- забезпечувати закріплення та реалізацію набутих теоретичних знань при підготовці до лабораторних занять;
- прищеплювати навички відповідального ставлення до своїх обов'язків, самостійного цілеспрямованого пошуку потрібної інформації, чіткої організації свого робочого дня.

Оцінювання знань здобувачів складається з поточного та підсумкового контролю.

Поточний контроль знань здобувачів вищої освіти передбачає оцінювання за наступними основними напрямками:

- перевірка теоретичних знань;
- перевірка підготовки до лабораторних занять.

З даних компонентів складаються загальні бали, які фіксуються в журналі викладача.

Оцінювання рівня засвоєння теоретичних знань здобувачів вищої освіти проводиться під час усної співбесіди зі здобувачами по теоретичним матеріалам, за результатами захисту лабораторних робіт й виконання самостійних робіт. Підсумковим контролем є екзамен.

10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ

Контроль успішності студента з дисципліни «Алгоритми та структури даних» здійснюється у формі поточного та семестрового контролю.

Поточний контроль має за мету перевірку якості засвоєння матеріалу навчальної дисципліни студентами. Він здійснюється під час проведення лабораторних занять. Для цього використовується модульно-рейтингова система оцінювання, яка передбачає розподіл балів за виконання всіх запланованих видів робіт. При цьому максимальна кількість балів за дисципліну при умові бездоганного виконання дорівнює 100. Ця сума складається з балів, що їх накопичив студент у ході поточного контролю та оцінки за екзамен.

Виконання базових завдань лабораторних робіт оцінюється максимально в 75 балів.

Лабораторні роботи №№ 1, 2, 4, 5, 6 містять додаткові завдання різного рівня складності, загалом за їх виконання можна отримати до 25 балів. Виконання завдань не є обов'язковим, студенти самостійно обирають завдання відповідно до своїх бажань та умінь.

Шкала оцінювання лабораторних робіт

№ модуля	№ зан.	Вид роботи	Тема	Максимальна кількість балів	
				основні завдання	додаткові завдання
1	1	Лабораторна робота № 1	Алгоритми сортування. Експериментальне дослідження ефективності різних методів сортування	10	8
	2	Лабораторна робота № 2	Алгоритми пошуку. Експериментальне дослідження ефективності різних методів пошуку	10	6
	3	Лабораторна робота № 3	Лабораторна робота №3. Однозв'язні списки, основні функції для роботи з ними. Абстрактний тип даних «Стек»	10	-
	4	Лабораторна робота № 4	Лабораторна робота №4. Двоzv'язні списки, основні функції для роботи з ними	10	3
	5	Лабораторна робота № 5	Лабораторна робота №5. Абстрактний тип даних «Черга». Реалізація на основі двозв'язного списку	10	3
	6	Лабораторна робота № 6	Лабораторна робота №6. Абстрактний тип даних «Бінарне дерево»	10	5
	7	Лабораторна робота № 7	Бінарні дерева пошуку та хеш-таблиці. Експериментальне дослідження ефективності використання цих АТД.	15	-
	Разом за модуль			75	25

Лабораторні роботи у модулі відображують оволодіння навичками та вміння застосовувати знання на практиці. Таким чином всі лабораторні роботи модуля при бездоганному їх виконанні можуть дати з урахуванням додаткових завдань 100 балів. Ці бали розподіляються поміж лабораторними роботами модуля у відповідності до їх відносної складності. При зниженні якості виконання тієї чи іншої лабораторної роботи, знижується і кількість балів, якою вона оцінюється.

Оцінювання кожної задачі лабораторної роботи ведеться за наступними показниками:

1. Своєчасність практичного виконання лабораторної роботи (у тиждень згідно із графіком робіт) (0-2 бали).
2. Теоретичний захист виконаної лабораторної роботи (у тиждень наступний за тижнем планового виконання роботи) (0-2 бали).
3. Якість знайдених студентом рішень (ефективність алгоритму, доречність використання елементів інтерфейсу, тощо) (4-11 балів).
4. Виконання додаткових завдань (1-8 балів).

Оцінка всієї лабораторної роботи знаходиться підсумовуванням балів за кожний з показників.

Для допуску до підсумкового семестрового контролю студент повинен виконати графік навчального процесу, усі види запланованих завдань і протягом семестру отримати в сумі не менше 50 балів.

Семестровий контроль здійснюється у формі екзамену в третьому семестрі для денної та заочної форм навчання, у першому семестрі для денної скороченої форми навчання. Екзаменаційний білет складається з 3 питань, кожне з яких оцінюється в 5 балів. Для виконання екзаменаційних завдань студенту надається 90 хвилин.

Результати екзамену оцінюються за 15-бальною шкалою. Підсумкова оцінка з дисципліни розраховується як середня арифметична чи зважена з оцінок за лабораторні роботи, додаткові завдання та екзамен.

Успішність студентів заочної форми навчання оцінюється аналогічним чином, за 100-бальною шкалою.

У відомість оцінка проставляється як у балах національної шкали, так і за шкалою ECTS.

При наявності у здобувачів результатів неформального навчання за освітнім компонентом «Алгоритми та структури даних» у повному обсязі, визнання та оцінювання результатів здійснюється відповідно до «Положення про порядок визнання у Криворізькому національному університеті результатів навчання, отриманих в умовах неформальної освіти». У випадку, якщо за підсумками визнання результатів неформального навчання визнається тільки частина результатів навчання, заявнику зараховуються окремі види навчальної роботи за освітнім компонентом «Алгоритми та структури даних».

Нижче наведені окремі види навчальної роботи, які можуть бути зараховані здобувачеві при наявності сертифікату про успішне проходження рекомендованих онлайн курсів.

Тема	Посилання на рекомендовані курси
Математичні основи аналізу алгоритмів	https://pro.foxminded.ua/algorithms-data-structures-1/ https://courses.prometheus.org.ua/courses/course-v1:KPI+Algorithms101+2015_Spring/about https://www.coursera.org/specializations/data-structures-algorithms https://www.coursera.org/learn/algorithms-part1#syllabus
Сортування. Алгоритми сортування масивів	https://pro.foxminded.ua/algorithms-data-structures-1/ https://www.coursera.org/learn/algorithms-searching-sorting-indexing https://www.coursera.org/specializations/data-structures-algorithms https://www.coursera.org/learn/algorithms-part1#syllabus
Пошук. Алгоритми пошуку даних	https://pro.foxminded.ua/algorithms-data-structures-1/ https://www.coursera.org/learn/algorithms-searching-sorting-indexing https://www.coursera.org/specializations/data-structures-algorithms https://www.coursera.org/learn/algorithms-part1#syllabus
Рекурсивні типи даних	https://pro.foxminded.ua/algorithms-data-structures-1/ https://courses.prometheus.org.ua/courses/course-v1:KPI+Algorithms101+2015_Spring/about https://www.coursera.org/learn/graphs https://www.coursera.org/specializations/data-structures-algorithms https://www.coursera.org/learn/algorithms-part2#syllabus

ШКАЛА ОЦІНЮВАННЯ

Національна шкала успішності	Оцінка ECTS	Визначення ECTS	100-бальна система оцінювання
відмінно/ зараховано	A	ВІДМІННО - відмінне виконання лише з незначними помилками	90...100
добре/ зараховано	B	ДУЖЕ ДОБРЕ - вище середнього рівня з кількома помилками	80...89
	C	ДОБРЕ - у цілому правильно робота з певною кількістю помилок і недоліків	71...79
задовільно/ зараховано	D	ЗАДОВІЛЬНО - непогано, але зі значною кількістю грубих помилок	61...70
	E	ДОСТАТНЬО - виконання задовольняє мінімальні потреби	50...60
незадовільно/ не зараховано	FX	НЕЗАДОВІЛЬНО - із можливістю повторного складання	30...49
	F	НЕЗАДОВІЛЬНО - з обов'язковим повторним вивчення дисципліни	0...29

11. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ, УМІНЬ, НАВИЧОК

1. Поняття обчислювальної складності алгоритму. Асимптотичний аналіз складності, O-нотація. Верхня та середня оцінки складності.
2. Складність алгоритмів. Часова і ємнісна складність. Асимптотична складність. Основні класи часової складності алгоритмів.
3. Емпіричні вимірювання ефективності алгоритмів.
4. Поняття сортування масиву. Класифікація сортувань. Характеристики сортувань.
5. Бульбашкове сортування. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків сортування.
6. Сортування вставками. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків сортування.
7. Сортування Шелла. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків сортування.
8. Пірамідальне сортування. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків сортування.
9. Сортування злиттям. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків сортування.
10. Швидке сортування. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків сортування.
11. Лінійний пошук у масиві. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків пошуку.
12. Бінарний пошук у масиві. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків пошуку.
13. Інтерполяційний пошук у масиві. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків пошуку.
14. Пошук «від краю» у масиві. Основна ідея та загальна схема методу. Аналіз найгіршого та середнього випадків пошуку.
15. Порівняння методів пошуку у масивах, області застосування лінійного, бінарного, інтерполяційного пошуку.
16. Динамічні структури даних та основні операції над ними.
17. Зв'язні списки, однозв'язні і двозв'язні списки, циклічні списки.
18. Створення однозв'язного списку. Прохід однозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку.
19. Створення двозв'язного списку. Прохід двозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку.
20. Принципи реалізації списків за допомогою вказівників.
21. Стек, алгоритми роботи стека, галузі застосування, приклади реалізації.
22. Наведіть приклади початкового формування стеку, занесення даних у стек, вибірки даних із стеку.
23. Реалізація стеків на базі масивів.
24. Реалізація стеків на базі динамічних структур.

25. Черга, алгоритми роботи черги, галузі застосування, приклади реалізації.
26. Наведіть приклади початкового формування черги, додавання даних у кінець черги, вибірки даних із черги.
27. Реалізація черг на базі динамічних структур.
28. Реалізація черг на базі масивів.
29. Черги з пріоритетами, приклади додавання даних до черги з пріоритетами, вибірки даних з черги з пріоритетами.
30. Порівняння часової та просторової ефективності реалізації черг на базі масивів та на базі динамічних структур. Які особливості в цих питаннях мають черги з пріоритетами?
31. Дерева. Основна термінологія. Порядок вузлів.
32. Прямий, зворотній та симетричні обходи дерев.
33. Абстрактний тип даних «Дерево». Реалізація дерев.
34. Бінарні дерева пошуку, визначення, основні операції, використання.
35. Визначення бінарних впорядкованих дерев, випадкові, оптимальні та збалансовані по висоті дерева. Навести приклади.
36. Поняття про збалансовані дерева. Червоно-чорне дерево, AVL-дерево.
37. Основні алгоритми оброблення даних на деревах.
38. Способи реалізації абстрактного типу даних «Словник». Реалізація на базі масивів. Обчислювальна складність операцій додавання, пошуку, вилучення.
39. Способи реалізації абстрактного типу даних «Словник». Реалізація на базі списків. Обчислювальна складність операцій додавання, пошуку, вилучення.
40. Способи реалізації абстрактного типу даних «Словник». Реалізація на базі бінарних дерев пошуку. Обчислювальна складність операцій додавання, пошуку, вилучення.
41. Способи реалізації абстрактного типу даних «Словник». Реалізація на базі хеш-таблиць. Обчислювальна складність операцій додавання, пошуку, вилучення.
42. Хеш-таблиці. Основні визначення та алгоритми побудови.
43. Різновиди хеш-функцій. Основні вимоги до хеш-функцій.
44. Поняття колізії у хеш-таблицях. Методи вирішення колізій.
45. Методи вирішення колізій: метод ланцюжків, метод відкритої адресації.

Приклад екзаменаційного білету

1. Поняття обчислювальної складності алгоритму. Асимптотичний аналіз складності, O-нотація. Верхня та середня оцінки складності.
2. Методи вирішення колізій: метод ланцюжків, метод відкритої адресації.
3. Динамічні структури даних та основні операції над ними.

12. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ

12.1 Навчальна та довідкова література

1. Крєневич А.П. Алгоритми і структури даних: підруч. Київ: ВПЦ "Київський Університет", 2021. 200 с.
2. Матвієнко М.П. Алгоритми та структури даних: навч. посіб. Київ: Ліра-К, 2014. 340 с.
3. Cormen Th.H., Leiserson Ch.E., Rivest R.L., Stein Cl. Introduction to Algorithms. 4th Edition. The MIT Press, 2022. 1312 p.
4. Skiena S. The Algorithm Design Manual. 3rd Edition. Springer, 2021. 810 p.
5. Cormen T.H. Algorithms Unlocked. MIT Press, 2013. 240 p.
6. Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms [Електронний ресурс] – Режим доступу до ресурсу: http://lib.ysu.am/disciplines_bk/f095abf306249866fdbf8194ea788d2c.pdf. 372 p.
7. Wirth N. Algorithms and Data Structures. [Електронний ресурс] – Режим доступу до ресурсу: [https://github.com/aforarup/interview/blob/master/Data Structures and Algorithm/Algorithm Books/Algorithms and Data Structures - Thin Book- Niklaus Wirth.pdf](https://github.com/aforarup/interview/blob/master/Data%20Structures%20and%20Algorithm/Algorithm%20Books/Algorithms%20and%20Data%20Structures%20-%20Thin%20Book-%20Niklaus%20Wirth.pdf). 288 p.

12.2. Методична література

1. Рибальченко О.Г., Білашенко С.В. Методичні вказівки до виконання лабораторних робіт з дисципліни «Алгоритми та структури даних» для студ. всіх форм навч. за спец. 121 «Інженерія програмного забезпечення». Кривий Ріг, 2023. 36 с.

13. ІНФОРМАЦІЙНІ РЕСУРСИ

До складу інформаційних ресурсів навчальної дисципліни входять:

1. Бібліотека Криворізького національного університету. URL: <http://lib.knu.edu.ua/> (дата звернення 20.12.2024).
Internet-ресурси:
2. Рибальченко О.Г. Алгоритми та структури даних: консп. лекцій. URL: <https://classroom.google.com/c/MTYzMzMxMDIwNTgy?hl=ru&cjc=jxrkxy3> (дата звернення 20.12.2024).

14. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

Алгоритм (Algorithm) - is a set of instructions that describe the order of actions of the performer to achieve the result of solving the problem in a finite number of actions; a system of rules for the execution of a discrete process that achieves the set goal in a finite time. Block diagrams are often used to visualize algorithms .

Математична модель (Mathematical model) - is a system of mathematical relationships that describe the process or phenomenon being studied . The mathematical model is important for such sciences as: economics , ecology , sociology , physics , chemistry , mechanics , computer science , biology , etc. Any mathematical tools can be used to create mathematical models - differential or integral equations, set theory , abstract algebra , mathematical logic , probability theory , graphs , etc. The process of creating a mathematical model is called mathematical modeling. This is the most general and most widely used research method in science, in particular, in cybernetics .

Структура даних (Data structure) - in programming and computer science, a data structure is a way of organizing data in computers . Often, a specific list of operations that can be performed on data organized into such a structure is associated with the data structure .The correct selection of data structures is extremely important for the effective functioning of the relevant algorithms for their processing. Well-constructed data structures allow you to optimize the use of machine time and computer memory to perform the most critical operations.

Масив (Array) - is an ordered set of a fixed number of elements of the same type, stored in sequentially located RAM cells, having a sequence number and a common name provided by the user.

Абстрактний тип даних (Abstract data type (ADT)) - is a mathematical model for data types, defined by its behavior (semantics) from the point of view of a user of the data, specifically in terms of possible values, possible operations on data of this type, and the behavior of these operations. This mathematical model contrasts with data structures, which are concrete representations of data, and are the point of view of an implementer, not a user. ADTs are a theoretical concept, used in formal semantics and program verification and, less strictly, in the design and analysis of algorithms, data structures, and software systems. Most mainstream computer languages do not directly support formally specifying ADTs.

Зв'язаний список (Linked list) - is a linear collection of data elements whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a collection of nodes which together represent a sequence. In its most basic form, each node contains data, and a reference (in other words, a link) to the next node in the sequence.

Стек (Stack) - is an abstract data type that serves as a collection of elements with two main operations: Push, which adds an element to the collection, and Pop, which removes the most recently added element. Additionally, a peek operation can, without modifying the stack, return the value of the last element added. The name stack is an analogy to a set of physical items stacked one atop another, such as a stack of plates. The order in which an element added to or removed from a stack is described as last in, first out, referred to by the acronym LIFO. As with a stack of physical objects, this structure makes it easy to take an item off the top of the stack, but accessing a datum deeper in the stack may require removing multiple other items first.

Черга (Queue) - is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is

called the back, tail, or rear of the queue, and the end at which elements are removed is called the head or front of the queue, analogously to the words used when people line up to wait for goods or services. The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing it. The operations of a queue make it a first-in-first-out (FIFO) data structure. In a FIFO data structure, the first element added to the queue will be the first one to be removed.

Асоціативний масив (Associative Array), map, symbol table, or dictionary is an abstract data type that stores a collection of (key, value) pairs, such that each possible key appears at most once in the collection. In mathematical terms, an associative array is a function with finite domain. It supports 'lookup', 'remove', and 'insert' operations.

Дерево (Tree) - is a widely used abstract data type that represents a hierarchical tree structure with a set of connected nodes. Each node in the tree can be connected to many children (depending on the type of tree), but must be connected to exactly one parent, except for the root node, which has no parent (i.e., the root node as the top-most node in the tree hierarchy). These constraints mean there are no cycles or "loops" (no node can be its own ancestor), and also that each child can be treated like the root node of its own subtree, making recursion a useful technique for tree traversal. In contrast to linear data structures, many trees cannot be represented by relationships between neighboring nodes (parent and children nodes of a node under consideration, if they exist) in a single straight line (called edge or link between two adjacent nodes).

Бінарне дерево (Binary Tree) - is a tree data structure in which each node has at most two children, referred to as the left child and the right child. That is, it is a k-ary tree with $k = 2$. A recursive definition using set theory is that a binary tree is a tuple (L, S, R) , where L and R are binary trees or the empty set and S is a singleton set containing the root.

Бінарне дерево пошуку (Binary Search Tree), also called an ordered or sorted binary tree, is a rooted binary tree data structure with the key of each internal node being greater than all the keys in the respective node's left subtree and less than the ones in its right subtree. The time complexity of operations on the binary search tree is linear with respect to the height of the tree. Binary search trees allow binary search for fast lookup, addition, and removal of data items. Since the nodes in a BST are laid out so that each comparison skips about half of the remaining tree, the lookup performance is proportional to that of binary logarithm.

Хеш таблиця (Hash Table) - is a data structure that implements an associative array, also called a dictionary or simply map; an associative array is an abstract data type that maps keys to values. A hash table uses a hash function to compute an index, also called a hash code, into an array of buckets or slots, from which the desired value can be found. During lookup, the key is hashed and the resulting hash indicates where the corresponding value is stored. A map implemented by a hash table is called a hash map. Most hash table designs employ an imperfect hash function. Hash collisions, where the hash function generates the same index for more than one key, therefore typically must be accommodated in some way. In a well-dimensioned hash table, the average time complexity for each lookup is independent of the number of elements stored in the table. Many hash table designs also allow arbitrary insertions and deletions of key-value pairs, at amortized constant average cost per operation.

Хеш функція (Hash Function) - is any function that can be used to map data of arbitrary size to fixed-size values, though there are some hash functions that support variable-length output. The

values returned by a hash function are called hash values, hash codes, hash digests, digests, or simply hashes. The values are usually used to index a fixed-size table called a hash table. Use of a hash function to index a hash table is called hashing or scatter-storage addressing.

Колізія (Hash Collision or hash clash) - is when two distinct pieces of data in a hash table share the same hash value. The hash value in this case is derived from a hash function which takes a data input and returns a fixed length of bits. In hash tables, since hash collisions are inevitable, hash tables have mechanisms of dealing with them, known as collision resolutions. Two of the most common strategies are open addressing and separate chaining. The cache-conscious collision resolution is another strategy that has been discussed in the past for string hash tables.

Відкрита адресація (Open Addressing) - cells in the hash table are assigned one of three states in this method – occupied, empty, or deleted. If a hash collision occurs, the table will be probed to move the record to an alternate cell that is stated as empty. There are different types of probing that take place when a hash collision happens and this method is implemented. Some types of probing are linear probing, double hashing, and quadratic probing. Open Addressing is also known as closed hashing.

Метод ланцюжків (Separate Chaining) - this strategy allows more than one record to be "chained" to the cells of a hash table. If two records are being directed to the same cell, both would go into that cell as a linked list. This efficiently prevents a hash collision from occurring since records with the same hash values can go into the same cell, but it has its disadvantages. Keeping track of so many lists is difficult and can cause whatever tool that is being used to become very slow. Separate chaining is also known as open hashing.

Рекурсія (Recursion) - in computer science, recursion is a method of solving a computational problem in which the solution relies on the solutions of smaller cases of the same problem. Recursion solves such recursive problems using functions or procedures that call themselves. This approach can be applied to many problems, and recursion is one of the central ideas of computer science.

Алгоритм сортування (A sorting algorithm) - is an algorithm that solves the sorting problem, that is, arranges a linear list (array) of elements. For the sorting algorithm (as for any other modern algorithm), the main characteristics are: the time required to arrange an n-element array, the need for additional memory for sorting, stability — stable sorting does not change the relative arrangement of elements with the same keys.

Алгоритм пошуку (A search algorithm) - is an algorithm that solves a search problem, that is, finds information that is stored in a certain data structure. Data structures can be implemented using linked lists, arrays, search trees, hash tables, or other information storage methods. The direct search algorithm depends on the data structure for which it is implemented. Very often, the search algorithm includes special commands that specify the data structure.

Додаток до робочої програми
Робочий план
з дисципліни «Алгоритми та структури даних»
для здобувачів вищої освіти першого (бакалаврського) рівня вищої освіти
за спеціальністю 121 «Інженерія програмного забезпечення»

Вид навчальної роботи	Годин у семестрі/кредитів	Розподіл годин по тижнях																Вид підсумкового контролю
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Лекційні заняття	16	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	екзамен
Лабораторні заняття	32	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	
Самостійна робота	102	5	5	6	7	6	7	6	7	6	7	6	7	6	7	6	8	
Всього годин/кредитів	150/5	9	7	10	9	10	9	10	9	10	9	10	9	10	9	10	10	

Позначки:

ПК - поточний контроль
ЗМ - складання змістових модулів

ЗМІНИ ТА ДОПОВНЕННЯ ДО РОБОЧОЇ ПРОГРАМИ

[illegible]

Схвалено на засіданні кафедри

Протокол № _____ від “_____” _____ 20__ р.

Завідувач кафедри _____

Схвалено на засіданні кафедри

Протокол № _____ від “_____” _____ 20__ р.

Завідувач кафедри _____

Схвалено на засіданні кафедри

Протокол № _____ від “_____” _____ 20__ р.

Завідувач кафедри _____

Схвалено на засіданні кафедри

Протокол № _____ від “_____” _____ 20__ р.

Завідувач кафедри _____