

**Чернівецький національний університет імені Юрія Федьковича**

(повне найменування закладу вищої освіти)

**Факультет математики та інформатики**

(назва факультету/навчально-наукового інституту)

**Кафедра математичного моделювання**

(назва кафедри, що забезпечує викладання)



**“ЗАТВЕРДЖУЮ”**

**Декан факультету**

**математики та інформатики**

**Ольга МАРТИНЮК**

**2025 року**

**РОБОЧА ПРОГРАМА**

**навчальної дисципліни**

**Теорія алгоритмів**

(назва навчальної дисципліни)

**обов'язкова**

(вказати: обов'язкова)

**Освітньо-професійна програма «Системний аналіз»**

(назва програми)

**Спеціальність 124 Системний аналіз**

(вказати: код, назва)

**Галузь знань 12 Інформаційні технології**

(вказати: шифр, назва)

**Рівень вищої освіти перший (бакалаврський)**

(вказати: перший (бакалаврський) / другий (магістерський) / третій (освітньо-науковий))

**Факультет математики та інформатики**

(назва факультету/ навчально-наукового інституту, на якому здійснюється підготовка фахівців за вказаною освітньою програмою)

**Мова навчання українська**

(вказати: на якій мові читасться дисципліна)

**Чернівці 2025 рік**

Робоча програма навчальної дисципліни «Теорія алгоритмів» складена відповідно до освітньо-професійної програми «Системний аналіз»

**Розробники:**

Караванова Тетяна Петрівна, асистент кафедри математичного моделювання;  
Косован Василь Михайлович, асистент, кандидат фіз.-мат. наук

**Викладачі**, що забезпечують читання даної навчальної дисципліни:

Косован Василь Михайлович, асистент, кандидат фіз.-мат. наук;  
Івасюк Галина Петрівна, доцент кафедри математичного моделювання,  
кандидат фіз.-мат. наук, доцент

Погоджено з гарантом ОП  Андрій ПЕРЦОВ

**Затверджено** на засіданні кафедри математичного моделювання  
Протокол № 15 від «24» червня 2025 року

Завідувач кафедри  Ігор ЧЕРЕВКО

**Схвалено** методичною радою факультету математики та інформатики  
Протокол № 12 від «25» червня 2025 року

Голова методичної ради  Віра СІКОРА

### **Мета навчальної дисципліни:**

Метою освітньої програми є формування у здобувачів вищої освіти системного мислення, здатності до логічного аналізу та аналітичного опрацювання інформації, зокрема шляхом застосування індуктивних і дедуктивних методів, аналізу, синтезу, узагальнення та формулювання обґрунтованих висновків; засвоєння теоретичних основ структур даних і алгоритмів, а також набуття практичних навичок їх реалізації засобами сучасних мов програмування; здатності до розв'язання алгоритмічних задач із використанням математичного апарату, класичних алгоритмів, спеціалізованої літератури та програмного забезпечення; розуміння міждисциплінарної значущості здобутих знань і їх застосування в професійній діяльності у сфері системного аналізу.

### **Пререквізити:**

Для вивчення освітньої компоненти будуть базою такі дисципліни: *Алгебра і геометрія, Математичний аналіз, Дискретна математика, Програмування.*

### **Завдання вивчення навчальної дисципліни:**

*Після вивчення даної дисципліни студент повинен знати:*

- сутність поняття алгоритму, взаємодію із базовими структурами даних, принципи організації, пошуку та впорядкування інформації;
- фундаментальні алгоритмічні стратегії, зокрема жадібні методи, динамічне програмування, алгоритми на графах та підходи обчислювальної геометрії;
- принципи оцінювання ефективності алгоритмів, включаючи часову та просторову складність, а також межі їх застосовності;
- загальні етапи розв'язання алгоритмічних задач: постановка, побудова, реалізація та тестування алгоритмів;
- методи оптимізації та їх застосування для розв'язання прикладних задач у сфері системного аналізу.

*Після вивчення даної дисципліни студент повинен уміти:*

- формулювати прикладні алгоритмічні задачі та обирати ефективні методи їх розв'язання з урахуванням специфіки предметної області;
- проектувати алгоритми з використанням відповідних структур даних, орієнтуючись на вимоги до продуктивності та масштабованості;
- реалізовувати алгоритмічні рішення мовами програмування, забезпечуючи їх функціональність, надійність і зручність у використанні;
- проводити аналіз коректності, ефективності та обчислювальної складності алгоритмів, з урахуванням часових і просторових обмежень;
- застосовувати алгоритмічні підходи для розв'язання реальних задач у сфері системного аналізу та їх впровадженням у практичну діяльність.

### Результати навчання:

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

#### *загальні компетентності за ОП «Системний аналіз»:*

**ЗК01.** Здатність до абстрактного мислення, аналізу та синтезу;

**ЗК10.** Здатність працювати автономно;

**ЗК08.** Здатність бути критичним і самокритичним

#### *фахові компетентності за ОП «Системний аналіз» :*

**ФК6.** Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних;

#### *програмні результати навчання за ОП «Системний аналіз»:*

**ПР2.** Вміти використовувати стандартні схеми для розв'язання комбінаторних та логічних задач, що сформульовані природною мовою, застосовувати класичні алгоритми для перевірки властивостей та класифікації об'єктів, множин, відношень, графів, груп, кілець, решіток, булевих функцій тощо.

**ПР9.** Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.

### Опис навчальної дисципліни

#### Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	3	5	150	30	–	–	46	74	–	Екзамен

## Структура змісту навчальної дисципліни

Назви змістових модулів і тем навчальних занять	Кількість годин											
	денна форма						заочна форма					
	усьог о	у тому числі					усьог о	у тому числі				
		лекц.	прак./ семін	лаб.	інд.	с.р.		лекц.	прак./ сем.	лаб.	інд.	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Змістовий модуль 1. Базові структури даних та алгоритми обробки їх елементів</b>												
<b>Тема 1. Теорія алгоритмів. Основні поняття.</b>	6	2				4						
<b>Тема 2. Базові структури даних. Проста змінна. Масив. Стек. Черга.</b>	13	2		6		5						
<b>Тема 3. Динамічна структура даних зв'язний список</b>	7	2				5						
<b>Тема 4. Деревя. Бінарні дерева.</b>	13	2		6		5						
<b>Тема 5. Хеш-функції та хеш-таблиці.</b>	7	2				5						
<b>Тема 6. Пошукові алгоритми на деревах та у мережах.</b>	15	2		8		5						
<b>Разом за ЗМ1</b>	<b>61</b>	<b>12</b>		<b>20</b>		<b>29</b>						
<b>Змістовий модуль 2. Оптимізаційні алгоритми з використанням структур даних</b>												
<b>Тема 7. Основи динамічного програмування. Базові задачі ДП.</b>	7	2				5						
<b>Тема 8. Жадібні алгоритми.</b>	13	2		6		5						
<b>Тема 9. Основні поняття теорії графів. Пошук в глибину та в ширину.</b>	7	2				5						
<b>Тема 10. Ейлерів та Гамільтонів графи.</b>	13	2		6		5						
<b>Тема 11. Остовне дерево. Мінімальне остовне дерево.</b>	7	2				5						

Тема 12. Пошук найкоротшого шляху між двома вершинами у зваженому графі.	13	2		6		5						
Тема 13. Топологічне сортування.	7	2				5						
Тема 14. Основи обчислювальної геометрії. Базові алгоритми обчислювальної геометрії.	7	2				5						
Тема 15. Побудова опуклої оболонки.	15	2		8		5						
Разом за ЗМ 2	89	18		26		45						
Усього годин	150	30		46		74						

### Тематика лекційних занять

#### № з/п

№ з/п	Назва теми
1.	<b>Теорія алгоритмів. Основні поняття.</b> Ключові питання: 1. Опишіть властивості алгоритму та способів їх представлення; 2. Методика розробки алгоритмів, оцінка їх ефективності; 3. Описати основні напрямки дослідження алгоритмів.
2.	<b>Базові структури даних. Проста змінна. Масив. Стек. Черга.</b> Ключові питання: 1. Поняття структур даних, їх класифікація; 2. Особливості роботи зі структурою даних стек; 1. Особливості роботи зі структурою даних черга.
3.	<b>Динамічна структура даних зв'язний список.</b> Ключові питання: 1. Опис лінійного однозв'язного циклу; 2. Узагальнення лінійних списків: кідьцеві, двозв'язні.
4.	<b>Дерева. Бінарні дерева.</b> Ключові питання: 1. Опис та реалізація бінарних дерев пошуку; 2. Опис та реалізація збалансованих бінарних дерев.
5.	<b>Хеш-функції та хеш-таблиці.</b> Ключові питання: 1. Як проводиться хешування даних;

	2. Особливості роботи з хеш-таблицями: створення динамічних таблиць, редагування вмісту, реструктуризація.
6.	<b>Пошукові алгоритми на деревах та у мережах.</b> Ключові питання: 1. Побудова частотного словника на бінарних деревах; 2. Пошук в мережі максимального потоку
7.	<b>Основи динамічного програмування. Базові задачі динамічного програмування.</b> Ключові питання: 1. Загальна характеристика побудови розв'язку задачі динамічного програмування; 2. Реалізація задач динамічного програмування: випадок задачі лінійного програмування, динамічне програмування на площині.
8.	<b>Жадібні алгоритми.</b> Ключові питання: 1. Характеристика задач, які розв'язуються методом “жадібного відбору”; 2. Критерії застосовності жадібних алгоритмів та методу динамічного програмування
9.	<b>Основні поняття теорії графів. Пошук в глибину та в ширину.</b> Ключові питання: 1. Загальна постановка пошукових задач у графі; 2. Критерії оптимального застосування кожного із описаних пошукових алгоритмів; 3. Які структури даних слід використовувати для оптимізації роботи пошукових алгоритмів?
10.	<b>Ейлерів та Гамільтонів графи.</b> Ключові питання: 1. Опишіть загальні властивості ейлерових та гамільтонових графів; 2. Критерії пошуку ейлерового шляху та ейлерового циклу в графі; 3. Оптимізація пошуку шляху в гамільтоновому графі
11.	<b>Остовне дерево. Мінімальне остовне дерево.</b> Ключові питання: 1. Загальне формулювання задачі про мінімальне кістякове дерево; 2. Опис та реалізація алгоритму Прима для зваженого неорієнтованого графу; 3. Опис та реалізація алгоритму Прима для зваженого неорієнтованого графу;
12.	<b>Пошук найкоротшого шляху між двома вершинами у зваженому графі.</b> Ключові питання: 1. Загальне формулювання задачі про мінімальних шлях в зважених графах;

	<p>2. Опис та реалізація алгоритму Дейкстри;</p> <p>3. Опис та реалізація алгоритму Флойда-Воршалла;</p>
13.	<p><b>Топологічне сортування.</b></p> <p>Ключові питання:</p> <ol style="list-style-type: none"> <li>1. Опис задач, при розв'язанні яких потрібно впорядкувати вершини графа за залежностями;</li> <li>2. Як зрозуміти, з якої вершини можна почати топологічне сортування?</li> </ol>
14.	<p><b>Основи обчислювальної геометрії. Базові алгоритми обчислювальної геометрії.</b></p> <p>Ключові питання:</p> <ol style="list-style-type: none"> <li>1. Побудова математичних моделей базових геометричних об'єктів;</li> <li>2. Оптимальне знаходження площі многокутників, заданих послідовними координатами своїх координат на площині;</li> <li>3. Розв'язання та реалізація задачі про перетин відрізків;</li> <li>4. Розв'язання та реалізація задачі про положення точки відносно многокутника.</li> </ol>
15.	<p><b>Побудова опуклої оболонки.</b></p> <p>Ключові питання:</p> <ol style="list-style-type: none"> <li>1. Загальне формулювання задач, розв'язання яких зводиться до побудови опуклої оболонки точок на площині;</li> <li>2. Реалізація алгоритму побудови опуклої оболонки методом Грехема;</li> <li>3. Реалізація алгоритму побудови опуклої оболонки методом Джарвіса.</li> </ol>

### Тематика лабораторних занять

№	Назва теми (завдання)
1	<p><b>Тема:</b> Структури даних «Стек», «Черга».</p> <p><b>Завдання:</b> <i>Написати програму, яка в діалоговому режимі реалізує дії:</i></p> <ol style="list-style-type: none"> <li>1. створення стеку/черги (на базі масиву);</li> <li>2. додавання елементу до стеку/черги;</li> <li>3. зчитування елементу зі стеку/черги;</li> <li>4. перегляд вмісту стеку/черги;</li> <li>5. виконання завдання за варіантом;</li> <li>6. вихід із програми.</li> </ol>
2	<p><b>Тема:</b> Структури даних "Зв'язний список", "Бінарні дерева"</p> <p><b>Завдання:</b></p> <ol style="list-style-type: none"> <li>1. Розробити програму, яка створює список, елементами якого є цілі числа. У програмі передбачити розміщення в оберненому порядку всіх елементів між першим і останнім входженням заданого елемента E, якщо E входить у не менше двох разів, у протилежному випадку список залишити без змін.</li> <li>2. Створити збалансоване бінарне дерево, елементами якого є випадкові цілі числа. Використавши алгоритм обходу цього дерева, побудувати бінарне</li> </ol>

	<i>дерево пошуку, утворене з тієї ж послідовності елементів, що й збалансоване бінарне дерево. Передбачити виведення вмісту вершин обох побудованих дерев.</i>
3	<p><b>Тема:</b> Структура даних «хеш-таблиця». Пошукові алгоритми на деревах.</p> <p><b>Завдання:</b></p> <ol style="list-style-type: none"> <li><i>1. Реалізувати у вигляді меню-орієнтованої програми з можливістю перегляду поточного стану та внесення змін хеш-таблицю «Таксі», елементами якої є номер машини (ціле число), марка машини, прізвище водія, використавши хеш-функцію ділення з остачею для розбиття інформації на сегменти за номером машини. Передбачити видалення з хеш-таблиці за номером машин, які виїхали на виклик, вставлення в хеш-таблицю машин, які повернулися з виклику, виведення поточного стану хеш-таблиці, її реструктуризацію.</i></li> <li><i>2. Розробити меню-орієнтовану програму, що реалізує побудову частотних словників для підтримки інформаційної системи «Бібліотека» (прізвище/ім'я автора книжки, назва книжки, видавництво, рік видання). У програмі передбачити виведення інформації на запити про кількість книжок вказаного автора, вказаного видавництва, вказаного року видання</i></li> </ol>
4	<p><b>Тема:</b> Пошукові алгоритми на графах. Ейлереві та Гамільтонові графи.</p> <p><b>Завдання:</b></p> <p><i>Виконати завдання за варіантом, розробити програмну реалізацію розв'язання завдання. Самостійно обрати спосіб введення відповідних даних (кількість вершин, ребер графа, матрицю суміжності, списки суміжних вершин, інших даних з умови), здійснити перевірку коректності вхідних даних до умови задачі. Здійснити дослідження ефективності програмної реалізації розв'язку задачі.</i></p>
5	<p><b>Тема:</b> Алгоритми на зважених та орієнтованих графах.</p> <p><b>Завдання:</b></p> <p><i>Виконати завдання за варіантом, розробити програмну реалізацію розв'язання завдання. Самостійно обрати спосіб введення відповідних даних (кількість вершин, ребер графа, матрицю суміжності, списки суміжних вершин, інших даних з умови), здійснити перевірку коректності вхідних даних до умови задачі. Здійснити дослідження ефективності програмної реалізації розв'язку задачі.</i></p>
6	<p><b>Тема:</b> Динамічне програмування. Жадібні алгоритми.</p> <p><b>Завдання:</b></p> <p><i>Реалізувати програмне розв'язання кожного із двох завдань на використання методу динамічного програмування та жадібних алгоритмів. Знайти оптимальний алгоритм для розв'язання кожного із завдань. Провести тестування програмних реалізацій алгоритмів на оцінку часової та просторової складності, проаналізувати поведінку алгоритму на граничних випадках.</i></p>
7	<p><b>Тема:</b> Реалізація алгоритмів побудови опуклої оболонки.</p> <p><b>Завдання:</b></p> <p><i>Побудувати математичну модель завдання за варіантом завдання на використання методу побудови площ та опуклих оболонок, реалізувати його програмно. Провести оцінку часової та просторової складності, проаналізувати поведінку алгоритму на граничних випадках, попередньо підготувавши відповідні набори вхідних даних.</i></p>

### Завдання для самостійної роботи студентів

Самостійна робота складає 74 години. Розподіл самостійної роботи за видами навчальних робіт:

- підготовка до лекційних занять – 20 годин;
- виконання лабораторних робіт – 40 годин;
- підготовка до екзамену – 14 годин.

Самостійна робота студентів використовується при вивченні наступних тем і передбачає опрацювання теоретичного матеріалу, результати якого застосовуються під час виконання лабораторних робіт.

№	Теми і завдання для самостійної роботи	К-ть годин
1.	Теорія алгоритмів. Основні поняття. Ключові питання: 1. Аналіз властивостей машини Тьюринга; 2. Аналіз ефективності алгоритмів за часом та пам'яттю.	4
2.	Базові структури даних. Проста змінна. Масив. Стек. Черга. 1. Реалізувати програмну реалізацію стеків; 2. Реалізувати програмну реалізацію черг.	5
3.	Динамічна структура даних зв'язний список. 1. Реалізувати програмну реалізацію структур лінійних списків; 2. Реалізувати програмну реалізацію двозв'язних та циклічних списків.	5
4.	Дерева. Бінарні дерева. 1. Реалізувати класичні алгоритми роботи з бінарними деревами; 2. Реалізувати алгоритм візуалізації структури бінарних дерев.	5
5.	Хеш-функції та хеш-таблиці. 1. Здійснити порівняльний аналіз методів множення, ділення з остачею тощо 2. Опрацювати теоретичні викладки про хешування в сучасних БД	5
6.	Пошукові алгоритми на деревах та у мережах. 1. Дослідити застосування структури даних при розгортанні комп'ютерних мереж; 2. Реалізувати програмні реалізації пошуку максимального потоку в мережах	5
7.	Основи динамічного програмування. Базові задачі ДП. 1. Створити програмні реалізації розв'язання класичних	5

	<p>задач: пошук найбільшої зростаючої підпоследовності 2-х та більше последовностей;</p> <p>2. Дослідити використання методів динамічного програмування при розв'язуванні комбінаторних задач.</p>	
8.	<p>Жадібні алгоритми.</p> <p>1. Опрацювати критерії застосовності принципів “жадібних алгоритмів” для розв'язання задач;</p> <p>2. Реалізувати програмні реалізації задач про максимальні знижки.</p>	5
9.	<p>Основні поняття теорії графів. Пошук в глибину та в ширину.</p> <p>1. Здійснити меню орієнтовані програмні реалізації для ілюстрації базових властивостей графів;</p> <p>2. Реалізувати розв'язки задач на використання пошуку в ширину та в глибину в онлайн-системах перевірки</p>	5
10.	<p>Ейлерів та Гамільтонів графи.</p> <p>1. Реалізувати розв'язки задач на використання пошуку в ширину та в глибину в онлайн-системах перевірки;</p> <p>2. Дослідити методи оптимізації пошуку шляху максимальної довжини у гамільтоновому графі;</p>	5
11.	<p>Остовне дерево. Мінімальне остовне дерево.</p> <p>1. Дослідити можливість заміни умов мінімальності на умови максимальності в класичних алгоритмах;</p> <p>2. Модифікувати алгоритми побудови мінімальних кістякових дерев для незважених графів;</p>	5
12.	<p>Пошук найкоротшого шляху між двома вершинами у зваженому графі.</p> <p>1. Дослідити питання оптимальності кратного застосування алгоритмів Дейкстри та Флойда-Уоршалла.</p> <p>2. Дослідити модифікації запропонованих алгоритмів для графів із від'ємними вагами ребер</p>	5
13.	<p>Топологічне сортування.</p> <p>1. Реалізувати програмно алгоритм топологічного сортування;</p> <p>2. Реалізувати розв'язки задач на використання топологічного сортування системах онлайн перевірки;</p>	5
14.	<p>Основи обчислювальної геометрії. Базові алгоритми обчислювальної геометрії.</p> <p>1. Повторити теоретичні викладки з аналітичної геометрії з тем вектори, розміщення прямих та площин на</p>	5

	площині та в просторі, рівняння ліній другого порядку. 2. Реалізувати програмно візуалізацію геометричних об'єктів;	
15.	Побудова опуклої оболонки. 1. Реалізувати розв'язки задач на побудову опуклої оболонки в системах онлайн перевірки; 2. Дослідити можливість узагальнення запропонованих методів побудови опуклої оболонки для випадку розміщення точок у просторі.	5

### Методи навчання

Цей курс представляє собою розроблений комплекс презентацій лекцій, варіантів лабораторних робіт до курсу, методичних рекомендацій до їх виконання, інтерактивних можливостей здачі лабораторних робіт, консультування, що розміщений на платформі <https://moodle.chnu.edu.ua/>

Під час викладання курсу застосовуються різноманітні методи навчання. Лекційні заняття проводяться аудиторно з використанням подання матеріалу студентам у вигляді лекції з використанням відповідних програмних та технічних засобів.

Під час знайомства з лекційним матеріалом використовуються інтерактивні форми навчання для актуалізації опорних знань.

Ключовим елементом курсу є індивідуальний захист лабораторних робіт. Усі завдання лабораторних є авторськими і передбачають:

- реалізацію коду програми відповідно до розробленого оптимізаційного алгоритму,
- підготовку комплексу власних тестів, що демонструють коректність і оптимальність розв'язаної задачі,
- перевірку ефективності реалізованих алгоритмів у онлайн системах тестування алгоритмічних задач.

Під час захисту студенти мають продемонструвати достатній і високий рівень програмування, розуміння матеріалу, якого стосується дане лабораторне завдання, довести достатність і повноту комплексу підібраних тестів.

Програмою курсу передбачено надання онлайн консультацій студентам. Також у разі непередбачуваних ситуацій (повітряна тривога тощо) заняття можуть проводитися онлайн за вказаним на сайті курсу посиланням.

### Методи навчання:

- вербальні методи (лекція, диспут, пояснення, розповідь тощо);
- наочні методи (демонстрація, ілюстрація);
- практичні методи (виконання завдань лабораторних робіт, розробка тестових даних для оцінки їх реалізації);
- робота з інформаційними ресурсами: з навчально-методичною, науковою, нормативною літературою; робота з інтернет ресурсами, зокрема автоматизованими системами тестування алгоритмічних задач

(ресурси проєкту "Буковинські інформатичні олімпіадні студії", EOlymp, Algotester тощо);

- самостійна робота над індивідуальним завданням та за програмою навчальної дисципліни;
- навчання з використанням відповідних онлайн-платформ.

### **Система контролю та оцінювання**

Критерієм підсумкового оцінювання є досягнення студентом мінімальних порогових рівнів оцінок (балів) за кожним запланованим результатом навчання навчальної дисципліни.

Поточний контроль знань відбувається впродовж лабораторних занять шляхом усного та письмового опитування теоретичних основ теми, виконання лабораторних завдань, а також у вигляді додаткових індивідуальних завдань.

Теоретичне опитування може здійснюватися на лекціях та лабораторних заняттях.

Впродовж вивчення дисципліни студент за виконання та захист лабораторних робіт повинен набрати не менше 50% можливих балів (35 балів з 70).

Оцінювання лабораторних робіт здійснюється з урахуванням правильності виконання завдань, оформлення звіту та розуміння отриманих результатів:

- 100 % балів — лабораторна робота виконана повністю та правильно; звіт оформлений відповідно до вимог; студент аргументовано пояснює хід роботи та результати.
- 80 % балів — лабораторна робота виконана з незначними помилками; звіт оформлений коректно; студент у цілому розуміє зміст роботи.
- 60 % балів — лабораторна робота виконана частково; наявні помилки в обчисленнях або оформленні; пояснення результатів неповні.
- 40 % балів — лабораторна робота виконана фрагментарно; значні помилки в розрахунках або висновках; слабе розуміння матеріалу.
- 20 % балів — лабораторна робота виконана формально; відсутні ключові етапи або висновки.
- 0 % балів — лабораторна робота не виконана або звіт не подано.

На підсумковому модулі (іспиті) студентам пропонуються білети, які містять три завдання. Два з них – теоретичні (кожне оцінюється в 10 балів), третє завдання передбачає розв’язання задачі на одну з тем «Структури даних», «Алгоритми на графах», «Алгоритми динамічного програмування», «Жадібні алгоритми», «Алгоритми обчислювальної геометрії» яке також оцінюється у 10 балів.

Оцінювання відповідей студентів здійснюється за таким алгоритмом:

- 100 % балів — студент надає всі правильні та повні теоретичні відповіді, демонструє глибоке розуміння навчального матеріалу, використовує коректну термінологію; практичне завдання (задачу) розв’язано повністю й правильно, з обґрунтуванням усіх етапів і коректними висновками.
- 80 % балів — студент надає більшість правильних теоретичних

відповідей, допускає незначні неточності, загалом розуміє навчальний матеріал; практичне завдання розв'язано правильно, проте можливі незначні помилки в обґрунтуванні або оформленні.

- 60 % балів — студент надає приблизно половину правильних теоретичних відповідей, демонструє фрагментарне розуміння теми; практичне завдання розв'язано частково, наявні помилки в окремих етапах або розрахунках.
- 40 % балів — студент надає менше половини правильних теоретичних відповідей, відповіді поверхові або неповні; практичне завдання виконано фрагментарно, з істотними помилками або без належного обґрунтування.
- 20 % балів — студент надає окремі правильні елементи теоретичних відповідей, але загалом не володіє матеріалом; практичне завдання розпочато, проте виконано неправильно або без розуміння алгоритму розв'язання.
- 0 % балів — студент не надає жодної правильної теоретичної відповіді та не виконує практичне завдання, або відмовляється від відповіді.

Підсумкова оцінка, як показник результатів вивчення навчальної дисципліни, складається із сумарної кількості балів за поточне оцінювання – **70 балів** та підсумкового модуль – контролю (екзамен) – **30 балів**, за **100-бальною університетською шкалою**.

#### Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	Оцінка за шкалою ECTS	
	Оцінка (бали)	Пояснення за розширеною шкалою
<b>Відмінно</b>	A (90-100)	відмінно
<b>Добре</b>	B (80-89)	дуже добре
	C (70-79)	добре
<b>Задовільно</b>	D (60-69)	задовільно
	E (50-59)	достатньо
<b>Незадовільно</b>	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим самостійним опрацюванням освітнього компоненту до перескладання

#### Розподіл балів, які отримують студенти

Система оцінювання рівня навчальних досягнень ґрунтується на принципах ECTS та є накопичувальною.

Протягом семестру студенти виконують 7 лабораторних робіт. Лабораторні роботи (ЛР) оцінюються по 10 балів, всього протягом семестру можна набрати 70 балів (див. таблицю нижче).

Поточне оцінювання (аудиторна та самостійна робота)							Кількість балів (екзамен)	Сумарна к-ть балів
Змістовий модуль №1			Змістовий модуль №2					
ЛР1	ЛР2	ЛР3	ЛР4	ЛР5	ЛР6	ЛР7	30	100
10	10	10	10	10	10	10		

**Перелік питань для самоконтролю й контролю навчальних досягнень студентів з навчальної дисципліни:**

1. Якими властивостями повинен володіти алгоритм?
2. Що таке обчислювальна складність алгоритму? У чому різниця між часовою та просторовою складністю?
3. Які існують способи подання алгоритмів?
4. У чому різниця між масивом, стеком і чергою за принципом доступу до елементів?
5. Як реалізувати стек за допомогою масиву або зв'язного списку?
6. Які основні операції виконуються над чергою?
7. Чим відрізняється однов'язний список від двов'язного?
8. Як здійснюється вставка та видалення елементів у зв'язному списку?
9. Які переваги та недоліки зв'язного списку порівняно з масивом?
10. Які властивості має бінарне дерево пошуку?
11. Як здійснюється обхід дерева в глибину (префіксний, інфіксний, постфіксний)?
12. У чому переваги використання дерев для зберігання даних?
13. Що таке хеш-функція та які її вимоги?
14. Які існують методи вирішення колізій у хеш-таблицях?
15. У яких випадках використання хеш-таблиці є найбільш доцільним?
16. У чому полягає різниця між пошуком у глибину та пошуком у ширину?
17. Як працює пошук у графі, представленому списком суміжності?
18. Які задачі можна ефективно розв'язати за допомогою пошуку на графах?
19. Якими способами можна представити граф у програмі?
20. Які основні відмінності між DFS і BFS?
21. Які задачі можна розв'язати за допомогою пошуку в графі?
22. Які умови існування Ейлерового циклу або шляху в графі?
23. У чому полягає відмінність між Ейлеровим та Гамільтоновим шляхом?
24. Чому задача знаходження Гамільтонового циклу належить до NP-повних?
25. Що таке остовне дерево графа?
26. Як працює алгоритм Крускала для пошуку мінімального остовного дерева?
27. У чому відмінність між алгоритмами Прима та Крускала?
28. Як працює алгоритм Дейкстри і в яких випадках він ефективний?
29. Коли доцільно використовувати алгоритм Беллмана-Форда?
30. Які проблеми виникають при наявності від'ємних ваг на ребрах?
31. Для яких графів можливе топологічне сортування?

32. Як здійснюється топологічне сортування за допомогою DFS?
33. Які практичні задачі можна розв'язати за допомогою топологічного сортування?
34. Що таке принцип оптимальності в динамічному програмуванні?
35. Які задачі класично розв'язуються методом динамічного програмування?
36. У чому перевага використання динамічного програмування над рекурсією?
37. У чому полягає суть жадібного підходу до побудови алгоритмів?
38. Які задачі ефективно розв'язуються жадібними методами?
39. Які умови необхідні для застосування жадібного алгоритму?
40. Які основні задачі вирішує обчислювальна геометрія?
41. Як визначити, чи перетинаються два відрізки на площині?
42. Які алгоритми використовуються для перевірки належності точки до многокутника?
43. Що таке опукла оболонка множини точок?
44. Як працює алгоритм Грехема для побудови опуклої оболонки?
45. Які інші алгоритми побудови опуклої оболонки вам відомі?

## Рекомендована література

### Основна

1. Караванова Т.П. Теорія алгоритмів. Частина 1. Необчислювальні алгоритми : навч. посіб. / Т.П. Караванова. Чернівці : Чернів. нац. ун-т. ім. Ю. Федьковича. 2022. 268 с.
2. Караванова Т.П. Теорія алгоритмів. Частина 2. Обчислювальні алгоритми.: навч. посіб., 2-ге вид. випр. Чернівці: Чернів. нац. ун-т ім. Ю. Федьковича. 2024. 288 с.
3. Караванова Т.П. Теорія алгоритмів. Частина 3. Збірник завдань з рекомендаціями та прикладами. Чернівці : Чернів. нац. ун-т ім. Ю. Федьковича. 2025. 224 с.
4. Кормен Т. Г., Лейзерсон Ч. Е., Рівест Р. Л., Стайн К. *Вступ до алгоритмів* : пер. з англ. 3-го вид. — К. : К. І. С., 2023. — 1288 с.
5. Бхаргава Адітья. Грокаємо алгоритми. Ілюстрований посібник для програмістів і допитливих. — Київ: ArtHuss, 2023. — 256 с.
6. Niklaus Wirth Algorithms and Data Structures — Publisher: Pearson Education, 2004. — 179 pages.
7. Donald E. Knuth The Art of Computer Programming, Volumes 1-4b — Publisher: Addison-Wesley Professional, 2022.
8. Tim Roughgarden Algorithms Illuminated: (Part 2): Graph Algorithms and Data Structures — Publisher: Soundlikeyourself Publishing; Illustrated edition, 2018. — 226 pages.
9. Tim Roughgarden Algorithms Illuminated: (Part 3): Greedy Algorithms and Dynamic Programming — Publisher: Soundlikeyourself Publishing, LLC; Illustrated edition, 2019. — 232 pages.
10. Tim Roughgarden Algorithms Illuminated: (Part 4): Algorithms for NP-Hard Problems — Publisher: Soundlikeyourself Publishing, LLC, 2020. — 273 pages.

### Допоміжна

1. Караванова Т.П. Інформатика: основи алгоритмізації та програмув.: 777 задач з рек. та прикл.: Навч. посіб. для 8-9 кл. із поглибленим вивч. інф-ки / За заг. ред. М.З. Згуровського — К.: Генеза, 2012. — 286 с.
2. Караванова Т.П. Методи побудови алгоритмів та їх аналіз: необчислювальні алгоритми: Навч. посіб. — К.: Генеза, 2007. — 224 с.
3. Караванова Т.П. Методи побудови алгоритмів та їх аналіз: обчислювальні алгоритми: Навч. посіб. — К.: Генеза, 2008. — 336 с.
4. Караванова Т.П. Методика розв'язування алгоритмічних задач. Основи алгоритмізації та програмування: Навчально-методичний посібник для вчителів / Т.П. Караванова. — Кам'янець-Подільський: Аксіома, 2013. — 460с.
5. Караванова Т.П. Методика розв'язування алгоритмічних задач. Побудова алгоритмів: Навчально-методичний посібник для вчителів / Т.П. Караванова. — Кам'янець-Подільський: Аксіома, 2014. — 344 с.

### Інформаційні ресурси:

<https://basecamp.eolymp.com/uk>

<https://algotester.com/uk>

<https://leetcode.com/>

<http://www.project-bios.chnu.edu.ua>

### Зарахування результатів неформальної/інформальної освіти

Здобувачі вищої освіти має право на участь у неформальній/інформальній освіті.

У межах поточного контролю можуть визнаватися результати неформальної/інформальної освіти за умови наявності сертифікату або освітньої декларації про результати неформальної/інформальної освіти з питань, що відповідає тематиці курсу («Порядок визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти», <https://www.chnu.edu.ua/media/4g5fzssb/poriadok-vyznannia-rezultativ-navchannia-zdobutykh-shliakhom-neformalnoi-ta-abo-informalnoi-osvity.pdf> ).

Студентам можуть бути зараховані додаткові бали, отримані через неформальну освіту, до загальної суми балів, набраної з освітньої компоненти, за умови, що результати з проблеми, за якою відбувалося навчання, відповідають тематиці курсу.

### Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні навчальної дисципліни регламентовано такими документами:

✓ «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича» <https://www.chnu.edu.ua/universytet/normatyvni->

[dokumenty/etychnyi-kodeks-chernivetsko-ho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/](#)

✓ «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університету імені Юрія Федьковича» <https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyivlennia-ta-zapobihannia-akademichnomu-plahiatu/>