



**Приватний вищий навчальний заклад  
«Буковинський університет»  
Факультет інформаційних технологій та економіки  
Кафедра комп'ютерних систем і технологій**

---

СХВАЛЕНО  
на засіданні науково-методичної  
ради факультету  
протокол № 1 від 26 серпня 2025 р.

ЗАТВЕРДЖУЮ  
Декан факультету ІТЕ  
\_\_\_\_\_/ Тетяна ШТЕРМА/  
«\_\_\_» \_\_\_\_\_ 2025 р.

## СИЛАБУС

**обов'язкової навчальної дисципліни**

**«Алгоритми і структури даних»**

<b>Освітньо-професійна програма:</b>	<u>Комп'ютерні науки</u>
<b>Спеціальність:</b>	<u>Комп'ютерні науки</u>
<b>Галузь знань:</b>	<u>Інформаційні технології</u>
<b>Рівень вищої освіти:</b>	<u>перший (бакалаврський)</u>
<b>Факультет:</b>	<u>Інформаційних технологій та економіки</u>
<b>Мова навчання:</b>	<u>українська</u>
<b>Розробник:</b>	Гаць Богдан Миколайович – кандидат технічних наук, доцент
<b>Профайл викладача:</b>	<a href="https://bukuniver.edu.ua/university/faculties-and-departments/ite-faculty/department-of-computer-systems-and-technologies/">https://bukuniver.edu.ua/university/faculties-and-departments/ite-faculty/department-of-computer-systems-and-technologies/</a>
<b>E-mail:</b>	<a href="mailto:gatsbn@gmail.com">gatsbn@gmail.com</a>
<b>Консультації:</b>	четвер з 10.00 до 16.00.

## **1. Анотація (призначення навчальної дисципліни).**

Освітній компонент «Алгоритми і структури даних» є складовою навчального плану підготовки фахівців першого (бакалаврського) рівня вищої освіти галузі знань **Інформаційні технології** спеціальності **Комп'ютерні науки**, яка побудована відповідно до вимог Європейської кредитної трансферно-накопичувальної системи (ECTS) і містить 4 кредити. Форма підсумкового контролю – екзамен.

Предметом вивчення курсу «Алгоритми і структури даних» є теоретичні засади аналізу складності алгоритмів, фундаментальні структури даних та методи розроблення ефективних обчислювальних процесів для розв'язання типових задач у сфері комп'ютерних наук.

## **2. Мета та завдання вивчення дисципліни.**

*Мета курсу* – формування у здобувачів вищої освіти системи теоретичних знань і практичних навичок щодо проєктування та реалізації ефективних алгоритмів, вибору оптимальних структур даних для зберігання й опрацювання інформації, а також проведення аналізу часової та просторової складності програмних рішень для забезпечення їхньої високої продуктивності.

*Завдання дисципліни* – засвоєння здобувачами освіти фундаментальних концепцій оцінювання ефективності алгоритмів, опануванні методів побудови та маніпулювання складними структурами даних (дерева, графи, хеш-таблиці), а також у розвитку вмінь застосовувати стандартні алгоритми сортування, пошуку та обходу структур у практичній розробці програмного забезпечення. Особлива увага приділяється здатності обґрунтовувати вибір алгоритмічних рішень залежно від обмежень ресурсів обчислювальної системи та інтерпретувати результати аналізу складності при розв'язанні прикладних задач моделювання та опрацювання великих обсягів даних.

*Основні знання та вміння, яких набуває студент після опанування цієї дисципліни*

*Основні знання:*

- основні методи аналізу складності алгоритмів (асимптотичний аналіз, нотація «O» велике) та класи складності;
- фундаментальні лінійні структури даних: масиви, динамічні списки, стеки, черги та деки;
- ієрархічні структури даних: бінарні дерева пошуку, збалансовані дерева, піраміди та принципи їхньої побудови;
- теорію графів: способи представлення графів у пам'яті комп'ютера та базові алгоритми на графах;
- принципи хешування, методи розроблення хеш-функцій та способи вирішення колізій у хеш-таблицях;
- класичні алгоритми сортування (внутрішнє та зовнішнє) та пошуку інформації.

*Основні вміння:*

- оцінювати часову та просторову складність програмних реалізацій алгоритмів для визначення їхньої ефективності;
- проєктувати та впроваджувати оптимальні структури даних для зберігання й опрацювання інформації залежно від типу задачі;
- реалізовувати алгоритми сортування та пошуку, адаптуючи їх під конкретні обмеження обчислювальних ресурсів;
- застосовувати сучасні методи розроблення алгоритмів, зокрема динамічне програмування, жадібні алгоритми та стратегію «розділяй і володарюй»;
- використовувати алгоритми на графах для моделювання мережевих структур та пошуку оптимальних шляхів;
- обґрунтовувати вибір алгоритмічного рішення на основі порівняльного аналізу показників результативності;
- реалізовувати складні структури даних та алгоритми опрацювання великих масивів даних засобами мов програмування високого рівня.

## **3. Пререквізити:**

- ОК4 Дискретна математика;
- ОК16 Алгоритмізація та програмування.

## **4. Компетентності та результати навчання.**

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

### **Інтегральна компетентність**

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук

або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов і вимог.

### **Загальні компетентності (ЗК)**

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

### **Спеціальні (фахові, предметні) компетентності**

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: узагальнення, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

*СК17. Здатність здійснювати збір, оброблення та аналіз структурованих та слабоструктурованих даних, а також проектувати й впроваджувати інтелектуальні інформаційні системи та системи підтримки прийняття рішень із використанням сучасних програмних інструментів, методів аналізу даних і технологій штучного інтелекту у прикладних галузях діяльності.*

### **Програмні результати навчання**

РН5. 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.

РН9. 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.

*РН20. Використовувати сучасні програмні інструменти, методи аналізу даних і технології штучного інтелекту для збору, оброблення та аналізу структурованих і слабоструктурованих даних під час розв'язання прикладних задач.*

## **5. Зміст навчальної дисципліни**

### **Змістовий модуль 1. Аналіз складності алгоритмів та лінійні структури даних**

Тема 1. Аналіз складності алгоритмів: асимптотична оцінка та класи Big O.

Тема 2. Лінійні структури даних: масиви, списки, стеки та черги.

Тема 3. Алгоритми сортування та їхня порівняльна ефективність.

Тема 4. Методи пошуку та технології хешування даних.

### **Змістовий модуль 2. Нелінійні структури даних та стратегії розроблення алгоритмів**

Тема 5. Деревоподібні структури даних: бінарні дерева пошуку та балансування.

Тема 6. Графові моделі та базові алгоритми обходу (BFS, DFS).

Тема 7. Оптимізаційні алгоритми на графах та пошук найкоротших шляхів.

Тема 8. Стратегії розроблення: динамічне програмування та жадібні алгоритми.

## **6. Система контролю та оцінювання.**

### **Методи навчання:**

- словесні методи (лекція, дискусія, пояснення, розповідь);
- практичні методи (практичні заняття, практичні завдання);
- наочні методи (демонстрація, ілюстрація, презентація);
- робота з інформаційними ресурсами: з навчально-методичною, науковою, нормативною літературою та інтернет-ресурсами;
- самостійна робота над індивідуальним завданням або за програмою навчальної дисципліни;
- дистанційне навчання з використанням відповідних онлайн-платформ.

### **Форми та методи оцінювання:**

- усне опитування;
- тестування;
- презентація результатів виконаних завдань;
- виконання вправ;
- контрольні роботи;
- підсумковий контроль – екзамен.

*Поточний контроль* проводиться на кожному практичному занятті.

*Об'єктом поточного контролю* знань студентів є:

- систематичність, якість та своєчасність виконання і захисту практичних робіт;
- систематичність та своєчасність виконання завдань самостійної роботи студента;
- якість виконання модульних контрольних робіт.

*Підсумковий контроль знань* проводиться у формі екзамена.

На екзамен виносяться вузлові питання дисципліни та типові задачі, що потребують творчої відповіді та умінь синтезувати отримані знання і застосувати їх при вирішенні практичних задач. Максимально можлива оцінка за екзамен – 30 балів.

Семестрова кількість балів може становити від 0 до 100 балів і визначається як сума балів: отриманих за всі види роботи на практичних заняттях; за виконання самостійної роботи; модульних контрольних робіт, результат підсумкового контролю у формі екзамена.

Оцінювання здійснюється за національною шкалою – «відмінно», «добре», «задовільно», «незадовільно» та за шкалою ECTS.

### Шкала оцінювання: національна та ECTS

Оцінка за шкалою ECTS	Оцінка за шкалою, що використовується у закладі вищої освіти та фахової передвищої освіти	Оцінка за національною шкалою
A	90-100	5 (відмінно)
B	80-89	4 (добре)
C	70-79	
D	60-69	3 (задовільно)
E	50-59	
FX	35-49	2 (незадовільно) з можливістю повторного складання
F	1-34	2 (незадовільно) з обов'язковим повторним вивченням дисципліни

### Розподіл балів з навчальної дисципліни

Поточний контроль (аудиторна та самостійна робота)		Іспит	Загальна кількість балів
Модуль I	Модуль II		
35	35	30	100

### Політика академічної доброчесності

Студент зобов'язаний ознайомитися з Положенням про забезпечення академічної доброчесності у ПВНЗ «Буковинський університет» та неухильно його дотримуватися. Текст документа розміщено у відкритому доступі на офіційному сайті університету. В освітньому процесі студент має виявляти дисциплінованість, ввічливість, доброзичливість, чесність і відповідальність.

Дотримання академічної доброчесності здобувачами освіти передбачає: самостійне виконання навчальних завдань, завдань підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей); посилання на джерела інформації у разі використання ідей, розробок, тверджень, відомостей; дотримання норм законодавства про авторське право і суміжні права; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерела інформації. Списування (копіювання тексту) під час виконання письмових робіт заборонені. Самостійні роботи у вигляді рефератів, доповідей, презентацій повинні мати коректні текстові посилання на використані інформаційні джерела. Дозволяється використання інструментів штучного інтелекту за умови дотримання принципів академічної доброчесності.

## 7.Рекомендована література

1. Крєневич А. П. *Алгоритми і структури даних : підручник*. – Київ : ВПЦ «Київський університет», 2021. – 200 с.

2. Кубій Л. І. *Алгоритми та структури даних. Основи алгоритмізації* : підручник для здобувачів ступеня бакалавра за спеціальністю 121 «Інженерія програмного забезпечення». – Київ : КПІ ім. Ігоря Сікорського, 2022. – 528 с.
3. Марченко О. І., Марченко О. О. *Структури даних та алгоритми. Підручник* : у 2 ч. Ч. 1. – Київ : Просвіта, 2024. – 268 с. – URL: <https://ela.kpi.ua/handle/123456789/66355> (дата звернення: 22.08.2025).
4. Бульба С. С., Бречко В. О., Далека В. Д. *Алгоритми та структури даних : навч. посібник [Електронний ресурс]*. – Харків : NTU «ХПІ», 2021. – 140 с. – URL: <https://repository.kpi.kharkov.ua/handle/KhPI-Press/78262> (дата звернення: 22.08.2025).
5. Коваленко О. О., Ткаченко О. М., Чехмєструк Р. Ю. *Алгоритми та структури даних : навч. посібник [Електронний ресурс]*. – Вінниця : ВНТУ, 2025. – Електрон. текст. дані. – URL: <https://ir.lib.vntu.edu.ua/handle/123456789/44509> (дата звернення: 22.08.2025).

### Додаткова

1. Гаць Б.М. Порівняльний аналіз сучасних технологій зберігання даних у хмарних обчисленнях / Чупринка Н., Яхно В., Гаць Б. // *Наука і техніка сьогодні*, 2024. № 8(36). С. 1202 - 1218.
2. Супрун О. М., Бойко О. Р., Гаць Б. М. Стеганографічні підходи до збереження метаданих у зображеннях для підвищення безпеки архівів // *Зб. наук. пр. НУК*. — 2025. — № 3 (501). — С. 182–189. — DOI: [https://doi.org/10.15589/znp2025.3\(501\).22](https://doi.org/10.15589/znp2025.3(501).22)
3. 1. Laaksonen A. *Competitive Programmer’s Handbook*. – 2-е вид. – San Francisco, CA : Leanpub, 2022. – 441 p. – ISBN 978-1735723410.
4. 2. Kleinberg J., Tardos É. *Algorithm Design and Applications*. – Cambridge : Cambridge University Press, 2021. – 720 p. – ISBN 978-1108738322.
5. 3. Goodrich M. T., Tamassia R., Goldwasser M. H. *Data Structures and Algorithms in Java*. – 7-е вид. – Hoboken, NJ : Wiley, 2021. – 984 p. – ISBN 978-1118771334

### Web-ресурси:

1. ig O Cheat Sheet [Електронний ресурс]. – Режим доступу: <https://www.bigocheatsheet.com> (дата звернення: 22.08.2025).
2. Khan Academy. Algorithms [Електронний ресурс]. – Режим доступу: <https://www.khanacademy.org/computing/computer-science/algorithms> (дата звернення: 22.08.2025).
3. GeeksforGeeks. Data Structures [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/data-structures/> (дата звернення: 22.08.2025).
4. Visualgo – Data Structures and Algorithms Visualization [Електронний ресурс]. – Режим доступу: <https://visualgo.net> (дата звернення: 22.08.2025).
5. Hackerearth. Sorting Algorithms [Електронний ресурс]. – Режим доступу: <https://www.hackerearth.com/practice/algorithms/sorting/> (дата звернення: 22.08.2025).
6. GeeksforGeeks. Sorting Algorithms [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/sorting-algorithms/> (дата звернення: 22.08.2025).
7. Open Data Structures (Python edition) [Електронний ресурс]. – Режим доступу: <https://opendatastructures.org/ods-python/> (дата звернення: 22.08.2025).
8. Massachusetts Institute of Technology OpenCourseWare. Hashing and Hash Tables [Електронний ресурс]. – Режим доступу: <https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-2020/pages/lecture-slides/> (дата звернення: 22.08.2025).
9. USFCA Algorithms Visualization [Електронний ресурс]. – Режим доступу: <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html> (дата звернення: 22.08.2025).
10. Massachusetts Institute of Technology OpenCourseWare. Graph Algorithms [Електронний ресурс]. – Режим доступу: <https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-2020/pages/lecture-slides/> (дата звернення: 22.08.2025).
11. Brilliant.org. Graph Algorithms [Електронний ресурс]. – Режим доступу: <https://brilliant.org/courses/graph-theory/> (дата звернення: 22.08.2025).
12. TopCoder Tutorials. Dynamic Programming [Електронний ресурс]. – Режим доступу: <https://www.topcoder.com/thrive/search?term=dynamic%20programming> (дата звернення: 22.08.2025).
13. CP-Algorithms (e-maxx). Algorithms and Data Structures [Електронний ресурс]. – Режим доступу: <https://cp-algorithms.com/> (дата звернення: 22.08.2025).
14. Massachusetts Institute of Technology OpenCourseWare. 6.006 Introduction to Algorithms [Електронний ресурс]. – Режим доступу: <https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-2020> (дата звернення: 22.08.2025).
15. Princeton University. Coursera. Algorithms, Part 1 [Електронний ресурс]. – Режим доступу: <https://www.coursera.org/learn/algorithms-part1> (дата звернення: 22.08.2025).
16. LeetCode – Algorithms Practice [Електронний ресурс]. – Режим доступу: <https://leetcode.com> (дата звернення: 22.08.2025).
17. HackerRank – Data Structures & Algorithms [Електронний ресурс]. – Режим доступу: <https://www.hackerrank.com/domains/tutorials/10-days-of-javascript> (дата звернення: 22.08.2025).