

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
/назва факультету/

Кафедра комп'ютерних наук
/назва кафедри/

“ЗАТВЕРДЖУЮ”
Декан факультету
Баран І.О.
“ 02 ” 09 ” 2024 року



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ТЕОРІЯ АЛГОРИТМІВ

/назва дисципліни/

галузь знань 12 Інформаційні технології

/шифр і назва галузі знань/

рівень вищої освіти Перший (бакалаврський рівень)

/назва/

спеціальність 122 «Комп'ютерні науки»

/шифр і назва/

освітня програма Комп'ютерні науки

/назва/

спеціалізація назва/

вид дисципліни обов'язкова

/обов'язкова / вибіркова/

Тернопіль – 2024 рік

Робоча програма з навчальної дисципліни Теорія алгоритмів

/назва дисципліни/

для студентів факультету комп'ютерно-інформаційних систем і програмної інженерії

/назва факультету(ів)/

Розробник:

професор каф. КН, д.т.н., проф.

/посада, науковий ступінь та вчене звання/

/підпис/

/ Ярослав ЛИТВИНЕНКО /

/ініціали та прізвище/

Робоча програма розглянута та схвалена

на засіданні кафедри комп'ютерних наук

/назва/

Протокол від « 26 » серпня 2024 року № 1

Завідувач кафедри

/підпис/

/ Ігор БОДНАРЧУК /

/ініціали та прізвище/

Робоча програма розглянута та схвалена НМК

факультету комп'ютерно-інформаційних систем і програмної інженерії

Протокол від « 2 » вересня 2024 року № 1

Секретар НМК

/підпис/

/ Б.Б. Млишко /

/ініціали та прізвище/

Робоча програма погоджена:

Спеціальність 122 «Комп'ютерні науки»

/шифр і назва/

освітня програма Комп'ютерні науки

/назва/

Завідувач випускової кафедри

/підпис/

/ Ігор БОДНАРЧУК /

/ініціали та прізвище/

Гарант освітньої програми

/підпис/

/ Леся ДМИТРОЦА /

/ініціали та прізвище/

1. Структура навчальної дисципліни

Показник	Всього годин	
	Денна форма навчання (ДФН)	Заочна (дистанційна) форма навчання (ЗФН)
Кількість кредитів/год	4,0/120	4,0/120
Аудиторні заняття, год.	54	12
Самостійна робота, год.	66	108
Аудиторні заняття:		
– лекції, год.	18	6
– лабораторні заняття, год.	36	6
– практичні заняття, год.	–	–
– семінарські заняття, год.	–	–
Самостійна робота:		
– підготовка до лабораторних (практичних – семінарських) занять, год.	18	30
– опрацювання окремих розділів програми, які не виносяться на лекції, год.	8	38
– виконання контрольних завдання, год.	–	–
– виконання індивідуальних завдань, год.	–	–
– виконання курсових проектів (робіт), год.	–	–
– підготовка та складання заліків, <u>екзаменів</u> , контрольних робіт, рефератів, есе, <u>тестування</u> , год.	40	40
Екзамен	+, IV с	+, IV с
Залік	–	–

Частка годин самостійної роботи студента:

денна форма навчання - 55 %;

заочна (дистанційна) форма навчання - 90 %.

2. Мета та завдання навчальної дисципліни

2.1. Мета вивчення навчальної дисципліни

Мета дисципліни «Теорія алгоритмів» — надати студентам знання в сфері реалізації задач автоматизації обробки інформації, автоматизації керування об'єктами за допомогою комп'ютерної техніки, формальних моделей алгоритмів та алгоритмічно обчислюваних функцій. Такі знання майбутній спеціаліст може застосовувати як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.

2.2. Завдання навчальної дисципліни

Завданням дисципліни є набуття студентами компетенцій, знань, умінь та навиків на рівні новітніх досягнень у теорії алгоритмів. Вивчення дисципліни дозволяє розширити професійні компетенції майбутніх фахівців в галузі інформаційних технологій щодо проектування та розробки програмного забезпечення в цілому.

По завершенню вивчення дисципліни студент повинен набути таких знань та вмінь:

Знати:

1. Визначення та призначення алгоритмів;
2. Технології розробки алгоритмів;
3. Основні поняття, засоби і методи теорії алгоритмів та їх застосування;
4. Основні формальні моделі алгоритмів та обчислювальних функцій.

Вміти:

1. Будувати формальні моделі алгоритмів та обчислювальних функцій;
2. Розробляти алгоритми лінійної, розгалуженої та циклічної структур;
3. Розробляти рекурсивні та ітераційні алгоритми;
4. Розробляти алгоритми для роботи з масивами;
5. Розробляти алгоритми сортування масивів;

6. Оцінювати ефективність розроблених алгоритмів.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів **загальних компетентностей (ЗК)** та **спеціальних (фахових) компетентностей (СК)** згідно освітньої програми.

Загальні:

ЗК 1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 2. Здатність застосовувати знання у практичних ситуаціях.

ЗК 3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК 6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК 7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК 8. Здатність генерувати нові ідеї (креативність).

Спеціальні (фахові):

СК 3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

Вивчення навчальної дисципліни передбачає формування у студента таких прикладних **результатів навчання (ПР)** згідно освітньої програми:

ПР 5. 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.

3. Опис навчальної дисципліни

3.1. Лекційні заняття

№	Тема заняття та короткий зміст	Кількість годин	
		ДФН	ЗФН
1	2	3	4
1.	Тема 1. Алгоритми та обчислення. Що таке алгоритм? Для чого вивчати алгоритми? Ефективність алгоритмів. Золоте правило розробників алгоритмів. Поняття часової та просторової (за обсягом пам'яті) складності алгоритму.	1	0,5
2.	Тема 2. Аналіз алгоритмів. Сортування включенням. Машина з довільним доступом до пам'яті. Аналіз алгоритму сортування методом включення. Порядок зростання. Асимптотичні позначення. Порівняння функцій.	2	1
3.	Тема 3. Метод декомпозиції. Метод декомпозиції. Аналіз алгоритму сортування злиттям. Підрахунок інверсій. Добуток матриць. Стратегія «розділяй та володарюй» приклади застосування. Стратегія балансування та приклади застосування.	2	2
4.	Тема 4. Рекурентні співвідношення. Метод підстановки. Метод дерев рекурсії. Основний метод. Доведення основної теореми.	1	1
5.	Тема 5. Швидке сортування. Опис швидкого сортування. Ефективність швидкого сортування. Випадкова версія швидкого сортування. Аналіз швидкого сортування. Порядкові статистики. Вибір за лінійний час. Інші алгоритми пошуку, сортування прості та удосконалені, сортування деревом, сортування Шелла.	1	1

6.	Тема 6. Сортування за лінійний час. Нижня оцінка алгоритмів сортування. Сортування підрахунком. Сортування за розрядами. Поняття абстрактного типу даних. Абстрактні типи даних: стеки, списки, вектори, словники, множини, мультимножини, черги, черги з пріоритетами. Кортежі множини, словники одно- та двобічнозв'язані списки. Реалізація абстрактних типів даних з оцінюванням складності операцій.	1	0,5
7.	Тема 7. Піраміди. Піраміди. Підтримка властивості піраміди. Створення піраміди. Алгоритм пірамідального сортування. Черги з пріоритетами.	1	—
8.	Тема 8. Хеш-таблиці. Таблиці з прямою адресацією. Хеш-таблиці. Уникнення колізій за допомогою ланцюгів. Хеш-функції. Відкрита адресація.	2	—
9.	Тема 9. Бінарні дерева пошуку. Бінарні дерева пошуку. Робота з бінарними деревами пошуку. Вставка та видалення.	2	—
10.	Тема 10. Графи. Пошук в графах. Означення графів. Застосування графів. Представлення графів. Задача пошуку в графі. Пошук вшир. Пошук найкоротших шляхів. Пошук компонент зв'язності в неорієнтованому графі.	2	—
11.	Тема 11. Пошук вглиб у графах. Обхід графу вглиб. Обхід графу вглиб (алгоритм 2). Топологічне сортування. Алгоритм для топологічного сортування. Компоненти сильної зв'язності. Алгоритм для пошуку компонент сильної зв'язності.	2	—

12.	Тема 12. Алгоритм Дейкстри. Пошук найкоротших шляхів у графі. Алгоритм Дейкстри. Приклад роботи алгоритму. Доведення коректності алгоритму. Складність алгоритму. Моделі обчислень. Динамічне програмування та приклади застосування.	1	–
Усього годин		18	6

3.2. Лабораторні заняття

№	Тема заняття	Кількість годин	
		ДФН	ЗФН
1	2	3	4
1.	Метод Карацуби*.	4	–
2.	Підрахунок інверсій.	4	1
3.	Швидке сортування.	4	2
4.	Сортування за розрядами.	4	2
5.	Пошук медіан з допомогою пірамід.	4	1
6.	Пошук сум в хеш-таблиці.	4	–
7.	Пошук сум у бінарному дереві.	4	–
8.	Пошук компонент сильної зв'язності*.	4	–
9.	Алгоритм Дейкстри пошуку найкоротшого шляху в графі*.	4	–
Усього годин		36	6

3.3. Самостійна робота

№	Найменування робіт	Кількість годин	
		ДФН	ЗФН
1	2	3	4
1.	Тема 1. Алгоритми та обчислення. Самоконтроль. Лабораторна робота*.	2	2

2.	Тема 2. Аналіз алгоритмів. Самоконтроль.	2	4
3.	Тема 3. Метод декомпозиції. Самоконтроль.	2	4
4.	Тема 4. Рекурентні співвідношення. Самоконтроль.	2	6
5.	Тема 5. Швидке сортування. Самоконтроль. Лабораторна робота.	4	6
6	Тема 6. Сортування за лінійний час. Самоконтроль. Лабораторна робота.	2	6
7.	Підготовка до тестування	20	20
8.	Тема 7. Піраміди. Самоконтроль. Лабораторна робота.	2	6
9.	Тема 8. Хеш-таблиці. Самоконтроль. Лабораторна робота.	2	6
10.	Тема 9. Бінарні дерева пошуку. Самоконтроль.	2	6
11.	Тема 10. Графи. Пошук в графах. Самоконтроль. Лабораторна робота.	2	6
12.	Тема 11. Пошук вглиб у графах. Самоконтроль.	2	8
13.	Тема 12. Алгоритм Дейкстри. Самоконтроль. Лабораторна робота.	2	8
14.	Підготовка до тестування	20	20
Усього годин		66	108

4. Критерії оцінювання результатів навчання студентів

Форма підсумкового семестрового контролю – **екзамен**

Підсумкова семестрова оцінка складається з суми балів, отриманих студентом при проведенні проміжних (модульних) та підсумкового контролів (фінальний екзаменаційний тест) рівня засвоєння теоретичних знань та отриманих балів за лабораторні роботи.

Захист звіту з лабораторної роботи оцінюється відповідною кількістю балів поданою в таблиці.

Після проходження теоретичного матеріалу проводиться електронне тестування його засвоєння у вигляді проміжного (модульного) контролю. Контроль здійснюється засобами електронного навчального курсу (ЕНК) на сервері дистанційного навчання <http://dl.tntu.edu.ua>, ідентифікатор дисципліни 4436.

МОДУЛЬ 1		МОДУЛЬ 2			МОДУЛЬ 3			ПІДСУМКОВА СЕМЕСТРОВА ОЦІНКА	РАЗОМ 3 ДИСЦИПЛІНИ
АУДИТОРНА ТА САМОСТІЙНА РОБОТА		АУДИТОРНА ТА САМОСТІЙНА РОБОТА			АУДИТОРНА ТА САМОСТІЙНА РОБОТА				
ТЕОРЕТИЧНИЙ КУРС (ТЕСТУВАННЯ)		ТЕОРЕТИЧНИЙ КУРС	ЛАБОРАТОРНА РОБОТА		ТЕОРЕТИЧНИЙ КУРС	ЛАБОРАТОРНА РОБОТА			
СЕМЕСТРОВА ОЦІНКА									
15		30			30			25	100
№ ЛЕКЦІЙ	БАЛ	№ ЛЕКЦІЙ	ВИД РОБІТ	БАЛ	№ ЛЕКЦІЙ	ВИД РОБІТ	БАЛ	Іспит	
Лекція 1	2	Лекція 5	Лаб. роб. №2	10	Лекція 8	Лаб. роб. №5	10		
Лекція 2	2	Лекція 6	Лаб. роб. №3	10	Лекція 9				
Лекція 3	3	Лекція 7	Лаб. роб. №4	10	Лекція 10	Лаб. роб. №6	10		
Лекція 4	3				Лекція 11				
Основи Python (самостійно)	5				Лекція 12	Лаб. роб. №7	10		
Поглиблений рівень (додаткові бали):									
Лаб. роб. №1*									

Лаб. роб. №8*
Лаб. роб. №9*

До підсумкового семестрового контролю (складання семестрового екзамену) допускаються студенти, які протягом семестру виконали всі види навчальної роботи, успішно пройшли проміжні (модульні) контролю і набрали не менше 45 балів семестрової бальної оцінки та за умови отримання не менше 60% (15) балів за результатами кожного проміжного (модульного) контролю рівня знань.

Підсумкова оцінка записується за 100-бальною шкалою із подальшим переведенням її у шкалу Європейської кредитно-трансферної системи (ECTS) відповідно A, B, C, D, E, F, FX при цьому чотирибальна шкала оцінок (з записом семестрової оцінки «відмінно» - A, «добре» - B,C, «задовільно» - D, E відповідають підсумковому результату «складено», «незадовільно» - F, FX відповідає підсумковому результату «не складено»).

5. Навчально-методичне забезпечення

1. Електронний навчальний курс «Теорія алгоритмів» для студентів спеціальностей: 122 «Комп'ютерні науки»; 126 «Інформаційні системи та технології».

2. Методичні вказівки до виконання лабораторних робіт з дисципліни «Теорія алгоритмів» для здобувачів освітнього ступеня «бакалавр», спеціальностей: 122 «Комп'ютерні науки», 126 «Інформаційні системи та технології», денної форми навчання. Укладач Я.В. Литвиненко: ТНТУ, 2023 р. 25с.

3. Конспект лекцій з курсу “Теорія алгоритмів” для здобувачів освітнього ступеня «бакалавр», спеціальностей: 122 «Комп'ютерні науки», 126 «Інформаційні системи та технології», денної форми навчання. Укладач Я.В. Литвиненко: ТНТУ, 2023 р. 80 с.

6. Рекомендована література

Базова

1. Книга Теорія алгоритмів. Посібник для студентів вищих навчальних закладів / Ірина Бородкіна. Центр навчальної літератури. 2019. 184 с.
2. Л. М. Клакович, С. М. Левицька. Теорія алгоритмів: Навчальний посібник. Друге видання, доповнене. Львів : Видавничий центр ЛНУ ім. Івана Франка, 2015. 161 с.
3. Глибовець Микола Миколайович. Основи комп'ютерних алгоритмів / М. М. Глибовець. К.: Видавничий дім "КМ Академія", 2003. 450 с.
4. Кормен, Томас Г. Вступ до алгоритмів : Переклад з англійської третього видання : [укр.] = Introduction to Algorithms : Third Edition : [пер. з англ.] / Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліффорд Стайн, К. : К. І. С., 2019. 1288 с.
5. Problem Solving with Algorithms and Data Structures using Python [Електронний ресурс] – Режим доступу до ресурсу: <https://runestone.academy/runestone/books/published/pythonds/index.html>.
6. Вихідний код програм [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/krenevych/algo>.
7. Kent D. Lee. Data Structures and Algorithms with Python / Kent D. Lee, Steve Hubbard. Springer, 2015. 363 с.
8. Algorithms / Christos; vazirani Umesh Dasgupta, Sanjoy; Papadimitriou, Christos H. Papadimitriou, Umesh Vazirani P. 336.
9. Competitive Programming 3 / Steven Halim, Felix Halim. Hardcover, P. 447.
10. Cracking the Coding Interview: 189 Programming Questions and Solutions/ Gayle Laakmann McDowell. Paperback. P. 551.
11. Grokking Algorithms An illustrated guide for programmers and other

curious people. Aditya Y. Bhargava 2016 P 256

12. Algorithms / Robert Sedgewick, Kevin Wayne. 2011 Hardcover. P. 976.

Допоміжна

1. Algorithms / Christos; vazirani Umesh Dasgupta, Sanjoy; Papadimitriou, Christos H. Papadimitriou, Umesh Vazirani P. 336.
2. Competitive Programming 3 / Steven Halim, Felix Halim. Hardcover, P. 447.
3. Cracking the Coding Interview: 189 Programming Questions and Solutions / Gayle Laakmann McDowell. Paperback. P. 551.

7. Інформаційні ресурси

1. <http://dl.tntu.edu.ua> Електронні навчальні курси ТНТУ імені І. Пулюя.
2. https://edx.prometheus.org.ua/courses/KPI/Algorithms101/2015_Spring/course/

Розробка та аналіз алгоритмів. Частина 1.

3. https://edx.prometheus.org.ua/courses/KPI/Programming101/2015_T1/course/

Основи програмування.

8. Зміни та доповнення до робочої програми навчальної дисципліни

№	Зміст внесених змін (доповнень)	Дата і № протоколу засідання кафедри	Примітки