

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

**Робоча програма
навчальної дисципліни
"ТЕОРІЯ АЛГОРИТМІВ"
для студентів напряму підготовки
6.050101 "Комп'ютерні науки"
всіх форм навчання**

Харків. Вид. ХНЕУ, 2012

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 1 від 27.08.2011 р.

Укладачі: Щербаков О. В.
Лосєв М. Ю.

P58 Робоча програма навчальної дисципліни "Теорія алгоритмів"
для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх
форм навчання / укл. Щербаков О. В., Лосєв М. Ю. – Х. : Вид. ХНЕУ,
2012. – 44 с. (Укр. мов.)

Подано тематичний план навчальної дисципліни та її зміст за модулями й те-
мами, вміщено плани лекцій та лабораторних занять, матеріал щодо закріплення
знань (самостійна робота, індивідуально-консультативна робота, контрольні запи-
тання, тести) та методику оцінювання знань студентів.

Рекомендовано для студентів напряму підготовки 6.050101 "Комп'ютерні науки".

Вступ

Широке розповсюдження інформаційних технологій, науково-технічний прогрес, проникнення інформаційно-комунікаційних технологій в усі сфери людської діяльності висувають нові, підвищені вимоги до підготовки фахівців з напрямку підготовки "Комп'ютерні науки". Сучасний професіонал у цій галузі повинен володіти цілим рядом компетенцій, серед яких особливе місце займають загальнонаукові та загально технічні компетенції, або, іншими словами – фундаментальні знання.

Саме тому, в новому Галузевому стандарті вищої освіти України з напрямку підготовки 6.050101 "Комп'ютерні науки" серед нормативних навчальних дисциплін поряд з такими дисциплінами, як "Дискретна математика" або "Чисельні методи" з'явилася дисципліна "Теорія алгоритмів".

У загальному випадку теорія алгоритмів розглядає такі питання як формалізація поняття "алгоритм" та дослідження формальних алгоритмічних систем; формальне доведення алгоритмічної нерозв'язуваності ряду завдань; класифікація завдань, визначення і дослідження класів складності; асимптотичний аналіз складності алгоритмів; дослідження та аналіз рекурсивних алгоритмів; отримання явних функцій трудомісткості для порівняльного аналізу алгоритмів; розробка критеріїв порівняльного оцінювання якості алгоритмів.

Одержані в теорії алгоритмів теоретичні результати знаходять достатньо широке практичне застосування. При цьому під час дослідження деякого завдання результати теорії алгоритмів дозволяють відповісти на ряд важливіших питань. Чи є це завдання принципово алгоритмічно розв'язним? І якщо так – то чи не належить воно до класу NP-повних завдань, при ствердній відповіді на яке можна говорити про істотні часові затрати для отримання точного розв'язку у випадку значних розмірностей вихідних даних. Крім цього, завдяки методам теорії алгоритмів стає можливим раціональний вибір алгоритму (з відомої множини алгоритмів) розв'язання даної задачі з урахуванням особливостей їх застосування; отримання часових оцінок розв'язання складних задач; отримання вірогідних оцінок неможливості розв'язання деякої задачі за певний час; розробку та вдосконалення ефективних алгоритмів на основі практичного аналізу тощо.

Вивчення дисципліни "Теорія алгоритмів" ґрунтується на знаннях та вміннях, які студенти отримали під час вивчення дисциплін: "Вступ до комп'ютерних наук", "Алгоритмізація та програмування", "Вища математика", "Дискретна математика".

Вона забезпечує дисципліни: "Об'єктно-орієнтоване програмування", "Технологія створення програмних продуктів", "Крос-платформне програмування", "Проектування інформаційних систем", "Технології захисту інформації", "Мобільні технології".

Мета навчальної дисципліни: отримання студентами ґрунтовної математичної підготовки та знань теоретичних, методичних і алгоритмічних основ інформаційних технологій для їх використання під час розв'язання прикладних і наукових завдань в області інформаційних систем і технологій, забезпечення теоретичної та інженерної підготовки фахівців у галузі проектування, впровадження та використання інформаційних систем в бізнесі.

Програма дисципліни передбачає навчання в формі лекцій та лабораторних робіт. Для практичного засвоєння основних тем дисципліни лабораторні роботи проводяться з застосуванням комп'ютерів, локальних мереж та мережі Internet у комп'ютерних класах ХНЕУ.

Завданням вивчення дисципліни "Теорія алгоритмів" є теоретична та практична підготовка майбутніх фахівців з таких питань:

- Математичні основи аналізу алгоритмів.
- Алгоритмічні стратегії.
- Основи теорії обчислюваності.
- Класи складності P і NP.
- Алгоритми сортування, злиття та пошуку.
- Комбінаторні алгоритми.
- Рекурсивні алгоритми.
- Фундаментальні алгоритми на графах і деревах.
- Геометричні алгоритми.
- Криптографічні алгоритми.
- Евристичні алгоритми.

Предметом навчальної дисципліни є сучасні та ефективні алгоритми комп'ютерної обробки інформації, а також методи їх дослідження та аналізу.

Структура програми навчальної дисципліни "Теорія алгоритмів" наведена в табл. 1.

Структура програми навчальної дисципліни

Навчальна дисципліна: підготовка бакалаврів	Галузь знань, напрям підготовки, спеціалізація, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів відповідних ECTS – 3,5; у тому числі: залікових модулів – 2; самостійна робота	Шифр та назва галузі знань: 0501 "Інформатика та обчислювальна техніка"	Нормативна. Рік підготовки: 1. Семестр: 2
Кількість годин: усього – 126; за заліковими модулями: модуль 1 – 54 години; модуль 2 – 72 години	Назва напрямку підготовки 6.050101: "Комп'ютерні науки"	Лекції: кількість годин – 17. Лабораторні заняття: кількість годин – 34. Самостійна робота: кількість годин – 75. (64+7 консульт.+ 4 іспит)
Кількість тижнів викладення навчальної дисципліни: 17. Кількість годин на тиждень: 3	Освітньо-кваліфікаційний рівень: бакалавр	Вид контролю: іспит

1. Кваліфікаційні вимоги до студентів

1.1. Вивчення дисципліни спрямовано на отримання студентами компетенції щодо сучасних теоретичних, методичних і алгоритмічних основ інформаційних технологій для їх використання під час розв'язання прикладних і наукових завдань в області інформаційних систем і технологій у майбутній професійної діяльності.

Студент повинен знати:

1. Теоретичні, методичні і алгоритмічні основи сучасних інформаційних технологій (понятійно-аналітичний).

2. Загальні принципи побудови ефективних алгоритмів (понятійно-аналітичний).

3. Сучасні методи дослідження та аналізу алгоритмів (понятійно-аналітичний).

4. Способи та механізми реалізації ефективних алгоритмів у конкретних застосуваннях (понятійно-аналітичний).

Студенти повинні вміти:

1. Реалізовувати основні алгоритми засобами алгоритмічної мови.
2. Розробляти нові математичні методи, ефективні алгоритми і методи реалізації функцій інформаційних систем і технологій у прикладних областях, зокрема під час розробки методів і систем штучного інтелекту.
3. Аналізувати, теоретично та експериментально досліджувати методи, алгоритми, програми апаратно-програмних комплексів і систем.
4. Створювати та досліджувати математичні та програмні моделі обчислювальних та інформаційних процесів, пов'язаних з функціонуванням об'єктів професійної діяльності.
5. Аналізувати та вибирати обчислювальні методи розв'язання задач проектування інформаційних систем за критеріями мінімізації обчислювальних витрат, стійкості, складності тощо.
6. Проектувати елементи математичного та лінгвістичного забезпечення обчислювальних систем.

Студенти повинні мати уявлення про:

1. Перспективи розвитку та використання сучасних ефективних алгоритмів обробки інформації в корпоративних інформаційно-аналітичних системах.
2. Основні способи та методи розробки нових перспективних алгоритмів для широкого кола задач.

1.2. Професійні компетенції та відповідні їм вміння, якими мають володіти випускники напряму підготовки "Комп'ютерні науки" освітньо-кваліфікаційного рівня "бакалавр" після вивчення навчальної дисципліни "Теорія алгоритмів", наведені в табл. 2.

2. Тематичний план навчальної дисципліни

При вивченні дисципліни "Теорія алгоритмів" кожен студент має бути ознайомлений як з програмою дисципліни і формами організації на-

вчання, так і з структурою , змістом та обсягом кожного із навчальних модулів, а також з усіма видами контролю та методикою оцінювання знань.

Таблиця 2

Компетенції, якими має оволодіти студент

Зміст функції	Компетенції фахівця	Вміння фахівців освітньо-професійного рівня "бакалавр"
Організаційна	Здатність до організації роботи з використанням сучасних інформаційних систем та технологій діяльності створення комп'ютерних програм, та розробки заходів щодо підвищення їх ефективності	Розробляти рекомендації та заходи, організувати та професійно брати участь у реалізації ІТ-рішень в діяльності створення комп'ютерних програм
Мотиваційна	Вміння розробляти ефективні системи мотивації з використання інформаційних технологій діяльності створення комп'ютерних програм	Розробляти систему мотивації персоналу щодо здійснення заходів із створення комп'ютерних програм та їхнього впровадження в різноманітні галузі економіки
Маркетингова	Здатність до пошуку сфер діяльності, в яких можуть бути використані інформаційні системи та технології для більш ефективної діяльності бізнесу	Розробляти проекти, які забезпечують підвищення ефективності інформаційної діяльності в бізнесі завдяки сучасним ефективним алгоритмам
Технологічна	Здатність розробляти технології створення та впровадження засобів комп'ютерної системи обробки інформації в діяльності бізнесу	Розробляти та реалізовувати алгоритми автоматизованої обробки інформації на основі сучасних інформаційних технологій

Тематичний план дисципліни складається з двох змістовних модулів, кожний з яких об'єднує в собі відносно окремий самостійний блок дисципліни, який логічно пов'язує кілька навчальних елементів дисципліни за змістом та взаємозв'язками (табл. 3).

Структура залікового кредиту навчальної дисципліни

Тема	Кількість годин, відведених на		
	Лекції	лабораторні роботи	самостійну роботу
Модуль 1. Аналіз алгоритмів та алгоритмічні стратегії			
Тема 1. Математичні основи аналізу алгоритмів. Алгоритмічні стратегії	2	4	10
Тема 2. Основи теорії обчислюваності. Класи складності P і NP	2	4	10
Разом годин за модулем 1	4	8	20
Модуль 2. Фундаментальні алгоритми та їх побудова			
Тема 3. Алгоритми сортування, злиття та пошуку	2	4	8
Тема 4. Комбінаторні та рекурсивні алгоритми	2	4	9
Тема 5. Фундаментальні алгоритми на графах і деревах	2	4	9
Тема 6. Алгоритми пошуку в рядках та їх обробки	2	4	9
Тема 7. Геометричні алгоритми	2	4	9
Тема 8. Евристичні та криптографічні алгоритми	3	6	11
Разом годин за модулем 2	13	26	55
Всього за дисципліну	17	34	75

3. Зміст навчальної дисципліни за модулями, розділами та темами

Модуль 1. Аналіз алгоритмів та алгоритмічні стратегії

Тема 1. Математичні основи аналізу алгоритмів. Алгоритмічні стратегії

Вступ. Мета та завдання дисципліни, її місце у навчальному процесі. Структура дисципліни, рекомендації щодо її вивчення. Організаційно-методичне забезпечення дисципліни.

Основи аналізу алгоритмів. Асимптотичний аналіз верхньої та середньої оцінок складності алгоритмів; порівняння найкращих, середніх і найгірших оцінок; O -, o -, ω - та θ -нотації; емпіричні вимірювання ефективності алгоритмів; накладні витрати алгоритмів за часом і пам'яттю; рекурентні співвідношення та аналіз рекурсивних алгоритмів.

Порівняння алгоритмів, вплив структур даних і особливостей реалізації на ефективність алгоритмів: сортування файлів.

Методи розробки алгоритмів: структурне програмування, рекурсія, обходи дерев, "поділяй і пануй", балансування, динамічне програмування, програмування з відходом назад, метод "гілок і меж", евристичні та наближені алгоритми.

Тема 2. Основи теорії обчислюваності. Класи складності P і NP

Поняття обчислюваною функції. Приклади. Властивість покрокового виконання алгоритму. Розв'язані множини і їх властивості. Перераховані множини і їх властивості. Перераховувана множина, як множина визначення обчислюваної функції. Перераховувана множина, як множина значень обчислюваної функції. Теорема Поста. Теорема про графік обчислюваної функції.

Поняття складності обчислення. Функція складності обчислень (за часом). Аксиоми Блюма. Теорема про прискорення. Класи складності. Опис класів P і NP. Приклади завдань, що належать цим класам. Ототожнення класу P з класом реально обчислювальних функцій. Поліноміальне зведення NP-повні задачі. Теорема Кука. Приклади NP-повних задач. Проблема перебору ($P = NP?$). Застосування теорії NP-повноти для аналізу складності завдань.

Модуль 2. Фундаментальні алгоритми та їх побудова

Тема 3. Алгоритми сортування, злиття та пошуку

Значення сортувань при реалізації алгоритмів. Класифікація сортувань. Характеристики сортувань. Прості сортування як спосіб швидкої реалізації алгоритму. Приклади простих сортувань – метод простого включення, метод простого обміну (бульбашкове сортування), шейкерне сортування, сортування вставками, сортування підрахунком, цифрове сортування. Переваги і недоліки простих сортувань.

Складні сортування як спосіб створення ефективних алгоритмів. Приклади складних сортувань – сортування Шелла, сортування Хоара (швидке сортування), сортування злиттям. Переваги і недоліки складних сортувань. Порівняння простих та складних сортувань.

Тема 4. Комбінаторні та рекурсивні алгоритми

Основні поняття комбінаторики. Поняття комбінаторної задачі. Перестановки. Підрахунок кількості можливих перестановок. Організація

перестановок. Розміщення та сполучення. Підрахунок кількості. Організація знаходження всіх можливих розміщень і сполучень. Методи організації повного перебору.

Метод гілок і границь. Обмеження варіантів перебору. Алгоритми пошуку з повертанням. Задача про розстановку дужок.

Тема 5. Фундаментальні алгоритми на графах і деревах

Основні поняття теорії графів. Матричне подання графів. Матриця зв'язності та матриця відстаней на графі. Пошук найкоротших шляхів та оптимальних маршрутів у графах. Алгоритм Дейкстри. Метод Беллмона. Знаходження мінімального остовного дерева графа за алгоритмом Прима-Краскала. Перевірка зв'язності графів. Алгоритм Тар'яна знаходження найменшого спільного пращура. Задача про найменше вершинне покриття.

Задача про гамільтонові шляхи на графі. Пошук у ширину на графах. Пошук у глибину на графах.

Тема 6. Алгоритми пошуку в рядках та їх обробки

Алгоритми пошуку в рядках: бінарний пошук, алгоритм Бойера – Мура, алгоритм Кнута – Морріса – Пратта, алгоритм Карпа – Рабіна, наближений пошук.

Прості алгоритми побудови дерева суфіксів. Алгоритм Укконена. Масиви суфіксів. Задача про найбільший спільний підрядок двох рядків.

Основні алгоритми обробки рядків – розбиття рядків, об'єднання рядків, алгоритми вставки, видалення, заміни.

Тема 7. Геометричні алгоритми

Основні формули обчислювальної геометрії. Знаходження довжини відрізка в n -вимірному просторі. Відстань від точки до прямої. Координати точок перетину відрізків і прямих. Рівняння прямої, кола, площини. Знаходження площі багатокутника. Метод триангуляції. Метод трапецій. Перевірка опуклості багатокутника.

Векторна геометрія. Колінеарність векторів. Перевірка належності точок прямій. Ліві та праві трійки векторів. Знаходження порядку обходу вершин опуклого багатокутника. Задачі мінімізації в геометричній інтерпретації.

Тема 8. Евристичні та криптографічні алгоритми

Поняття "жадібного" алгоритму. Теоретичні основи "жадібних" алгоритмів. Переваги та недоліки "жадібних" алгоритмів.

Класичні приклади "жадібних" алгоритмів. Задача про вкладання рюкзака. Розв'язання задач із застосуванням "жадібних" алгоритмів. Геометричні, транспортні, економічні задачі.

Поняття про динамічне програмування. Основні підходи до розв'язання задач методом динамічного програмування. Матричне числення. Перемноження декількох матриць. Знаходження найбільшої спільної підпоследовності множин. Визначення оптимальної триангуляції багатокутника. Задачі лінійного програмування. Симплекс-метод розв'язання задач економічного планування. Криптографічні алгоритми та їх застосування для захисту інформації.

4. Плани лекцій

Модуль 1. Аналіз алгоритмів та алгоритмічні стратегії

Тема 1. Математичні основи аналізу алгоритмів.

Алгоритмічні стратегії

1.1. Вступ. Мета та завдання дисципліни, її місце у навчальному процесі.

1.2. Основи аналізу алгоритмів.

1.3. Основні способи вимірювання ефективності алгоритмів.

1.4. Алгоритмічні стратегії та методи розробки алгоритмів

1.5. Обґрунтування раціонального вибору алгоритму (з відомої множини алгоритмів).

Література: основна [1 – 3]; додаткова [5; 8].

Тема 2. Основи теорії обчислюваності. Класи складності P і NP

2.1. Поняття складності обчислення.

2.2. Опис класів P і NP.

2.3. Приклади NP-повних задач.

Література: основна [1 – 3]; додаткова [4 – 6].

Модуль 2. Фундаментальні алгоритми та їх побудова

Тема 3. Алгоритми сортування, злиття та пошуку

3.1. Прості алгоритми сортування: обмін, вибір, вставка.

3.2. Покращені алгоритми сортування – сортування Шелла, сортування Хоара (швидке сортування), сортування злиттям.

3.3. Аналіз обчислювальної складності алгоритмів сортування.

3.4. Алгоритми лінійного та бінарного пошуку.

Література: основна [1 – 3]; додаткова [4; 8].

Тема 4. Комбінаторні та рекурсивні алгоритми

4.1. Алгоритми перебору з поверненням.

4.2. Алгоритм підрахування комбінаторних об'єктів.

4.3. Основні рекурсивні алгоритми.

Література: основна [1 – 3]; додаткова [6; 8].

Тема 5. Фундаментальні алгоритми на графах і деревах

5.1. Алгоритми пошуку найкоротших шляхів та оптимальних маршрутів у графах.

5.2. Алгоритми пошуку у ширину на графах та деревах.

5.3. Алгоритми пошуку в глибину на графах та деревах.

5.4. Аналіз складності алгоритмів на графах та деревах.

Література: основна [1 – 3]; додаткова [4; 7; 8].

Тема 6. Алгоритми пошуку в рядках та їх обробки

6.1. Огляд та класифікація основних алгоритмів пошуку в рядках.

6.2. Алгоритм пошуку підрядка Кнута – Морріса – Пратта (на основі префікс-функції).

6.3. Алгоритм пошуку підрядка Бойера – Мура (на основі стоп-символів).

6.4. Алгоритми обробки рядків.

Література: основна [1 – 3]; додаткова [4; 7; 8].

Тема 7. Геометричні алгоритми

7.1. Основні формули обчислювальної геометрії.

7.2. Алгоритми знаходження відстані між геометричними об'єктами.

7.3. Алгоритм побудови опуклої оболонки.

7.4. Задачі мінімізації в геометричній інтерпретації.

Література: основна [1 – 3]; додаткова [4; 7; 8].

Тема 8. Евристичні та криптографічні алгоритми

8.1. Основні евристичні алгоритми.

8.2. "Жадібні алгоритми".

8.3. Динамічне програмування.

8.4. Криптографічні алгоритми.

8.4. Аналіз складності алгоритмів.

Література: основна [1; 2]; додаткова [4; 7; 8].

5. Плани лабораторних занять

Лабораторна робота – форма навчального заняття, при якій студенти під керівництвом викладача досліджують можливості застосування сучасних комп'ютерних технологій для виконання різних завдань економічного характеру. При цьому у студентів формуються вміння й практичні навички використання різних програмних засобів ПК для розв'язання конкретних економічних задач відповідно до індивідуального завдання. Перелік тем лабораторних занять наведено в табл. 4.

Проведення лабораторних занять ґрунтується на попередньо підготовлених методичних матеріалах: визначення підготовленості студентів до виконання завдань лабораторного заняття на основі тестового контролю знань основних положень теорії досліджуваної теми, усного контролю виконання домашнього завдання, пов'язаного з розробкою макетів документів, які необхідно розробити програмно під час заняття.

Індивідуальні завдання до кожної лабораторної роботи мають чітко виражену прикладну спрямованість, що враховує профіль підготовки студентів, тобто охоплюють питання автоматизації рішення різних завдань економіки і підприємництва.

Лабораторні роботи виконуються в такій послідовності:

вивчення навчального матеріалу з теми лабораторної роботи з використанням конспекту лекцій, рекомендованих підручників і навчальних посібників;

самостійна підготовка студентами макетів інтерфейсів програм, які мають бути практично створені на занятті;

виконання завдання на ПК відповідно до виданого варіанта й подання результатів викладачеві.

По завершенню кожної роботи студенти готують і оформлюють звіт й захищають отримані результати. Звіт повинен містити:

тему й мету роботи;

зміст завдання й короткий опис порядку його виконання;

аналіз отриманих результатів та висновки

роздруківку основних результатів виконання індивідуального завдання.

Таблиця 4

Перелік тем лабораторних занять

Назва залікового модуля	Теми лабораторних занять (за модулями)	Кількість годин	Література
1	2	3	4
Модуль 1. Аналіз алгоритмів та алгоритмічні стратегії	1. Прості цілочисельні алгоритми	4	[1, с. 19–75; 2, с. 17–75; 3, с. 9–32]
	2. Динамічні структури даних	4	[1, с. 187–220; 2, с. 123–129; 3, с. 54–95]
Разом годин за модулем 1		8	
Модуль 2. Фундаментальні алгоритми та їх побудова	3. Основні методи сортування масивів	4	[1, с. 471–524; 2, с. 295–320]
	4. Комбінаторні та рекурсивні алгоритми	4	[1, с. 525–568; 2, с. 321–344; 3, с. 94–95; с. 626–658]
	5. Алгоритми на графах та деревах	4	[1, с. 525–568; 2, с. 321–344]
	6. Алгоритми пошуку в рядках	4	[1, с. 525–568; 2, с. 321–344]
	7. Алгоритми обчислювальної геометрії	4	[1, с. 525–568; 2, с. 321–344]
	8. Евристичні та криптографічні алгоритми	6	[1, с. 611–688; 2, с. 310–325; 3, с. 164–176]
Разом годин за модулем 2		26	
Разом годин за модулями		34	

6. Самостійна робота студентів

6.1 Загальні положення

Одним з основних напрямів успішного засвоєння матеріалів навчальної дисципліни є самостійна робота студентів над основною й додатковою літературою з вивчення й використання сучасних комп'ютерних технологій при вирішенні економічних задач.

Основними видами самостійної роботи є:

1. Вивчення лекційного матеріалу.
2. Вивчення рекомендованої літератури.
3. Вивчення термінів і основних понять з тем навчальної дисципліни.
4. Підготовка до лабораторних занять і розробка ескізів документів з кожної лабораторної роботи.
5. Підготовка до тестового контролю з модулів навчальної дисципліни.
6. Підготовка до виконання контрольних робіт з модулів навчальної дисципліни.
7. Робота з опрацювання та вивчення рекомендованої літератури.
8. Систематизація вивченого матеріалу перед іспитом.

6.2. Питання для самостійного опрацювання

Модуль 1. Аналіз алгоритмів та алгоритмічні стратегії

Тема 1. Математичні основи аналізу алгоритмів.

Алгоритмічні стратегії

1. Поняття алгоритму.
2. Поняття обчислювальної складності алгоритму.
3. Асимптотичний аналіз складності, O-нотація.
4. Верхня та середня оцінка складності.
5. Емпіричні вимірювання ефективності алгоритмів.
6. Вплив структур даних і особливостей реалізації на ефективність алгоритмів.
7. Основні алгоритмічні стратегії.
8. Методи розробки алгоритмів.

Література: основна [1 – 3]; додаткова [5; 6; 8].

Тема 2. Основи теорії обчислюваності. Класи складності P і NP

1. Поняття обчислювальної функції.
2. Властивості обчислювальних функцій.
3. Розв'язні множини і їх властивості.
4. Поняття складності обчислення.
5. Теорема Поста.
6. Теорема про графік обчислювальної функції.
7. Класи складності.
8. Опис класів P і NP.
9. Ототожнення класу P з класом реально обчислювальних функцій.
10. Приклади NP-повних задач.
11. Застосування теорії NP-повноти для аналізу складності завдань.

Література: основна [1 – 3]; додаткова [4; 5; 6].

Модуль 2. Фундаментальні алгоритми та їх побудова

Тема 3. Алгоритми сортування, злиття та пошуку

1. Поняття сортування масиву.
2. Класифікація сортувань.
3. Характеристики сортувань.
4. Прості алгоритми сортування.
5. Алгоритм простого обміну.
6. Алгоритм простого вибору.
7. Алгоритм простого включення.
8. Алгоритм шейкерного сортування.
9. Переваги і недоліки простих сортувань.
10. Покращені алгоритми сортування.
11. Алгоритм сортування Шелла.
12. Алгоритм сортування Хоара (швидке сортування).
13. Алгоритм сортування злиттям.
14. Переваги і недоліки складних сортувань.
15. Порівняння простих та складних сортувань.
16. Лінійний пошук у масиві.
17. Бінарний пошук у масиві.

Література: основна [1 – 3]; додаткова [4; 8].

Тема 4. Комбінаторні та рекурсивні алгоритми

1. Основні поняття комбінаторики.
2. Підрахування комбінаторних об'єктів.
3. Алгоритм повного перебору.
4. Алгоритм перебору з поверненням.
5. Алгоритм "гілок та границь".
6. Поняття рекурсивного алгоритму.
7. Обчислювальна складність рекурсивних алгоритмів.
8. Порівняння рекурсивних та не рекурсивних алгоритмів.

Література: основна [1 – 3]; додаткова [6; 8].

Тема 5. Фундаментальні алгоритми на графах і деревах

1. Основні поняття теорії графів.
2. Матричне подання графів.
3. Матриця зв'язності графу.
4. Матриця відстаней на графі.
5. Алгоритми пошуку найкоротших шляхів та оптимальних маршрутів у графах.
6. Алгоритм Дейкстри.
7. Знаходження мінімального остовного дерева графа за алгоритмом Прима – Краскала.
8. Гамільтонові шляхи на графі.
9. Дерева, способи подання.
10. Пошук у ширину на графах та деревах.
11. Пошук в глибину на графах та деревах.
12. Задача комівояжера.
13. Задача про найменше вершинне покриття.

Література: основна [1 – 3]; додаткова [4; 7; 8].

Тема 6. Алгоритми пошуку в рядках та їх обробки

1. Основні поняття.
2. Методи попереднього аналізу рядків.
3. Алгоритм Кнута – Морріса – Пратта.
4. Алгоритм Бойера – Мура.
5. Алгоритм Карпа – Рабіна.
6. Алгоритм Мейна – Лоренца.

7. Алгоритм Крочемора.
 8. Прості алгоритми побудови дерева суфіксів.
 9. Алгоритм Укконена.
 10. Масиви суфіксів.
 11. Сортуння даних.
 12. Задача про найбільший спільний підрядок двох рядків.
 13. Алгоритми наближеного пошуку підрядків.
 14. Алгоритми обробки рядків.
 15. Алгоритми розбиття рядків та об'єднання рядків.
 16. Алгоритми вставки, видалення та заміни підрядків в рядках.
- Література:** основна [1 – 3]; додаткова [4; 7; 8].

Тема 7. Геометричні алгоритми

1. Основні поняття обчислювальної геометрії – точка, відрізок, пряма.
 2. Алгоритми визначення взаємного розташування точок, прямих та відрізків на площині.
 3. Алгоритми обробки багатокутників.
 4. Алгоритми визначення взаємного розташування кіл, прямих та точок.
 5. Алгоритм побудови опуклої оболонки.
 6. Порядок обходу вершин опуклого багатокутника.
 7. Задачі мінімізації в геометричній інтерпретації.
- Література:** основна [1 – 3]; додаткова [4; 7; 8].

Тема 8. Евристичні та криптографічні алгоритми

1. Поняття евристичного алгоритму.
2. Види евристичних алгоритмів.
3. "Жадібні" алгоритми.
4. Теоретичні основи "жадібних" алгоритмів.
5. Переваги та недоліки "жадібних" алгоритмів.
6. Приклади "жадібних" алгоритмів.
7. Поняття про динамічне програмування.
8. Основні підходи до розв'язання задач методом динамічного програмування.
9. Прикладні задачі динамічного програмування.
10. Задачі лінійного програмування.

11. Симплекс-метод розв'язання задач економічного планування.
12. Поняття фіксованого документа.
13. Поняття про криптографію.
14. Основні криптографічні алгоритми.
15. Шифр Цезаря.
16. Використання криптографічних алгоритмів для захисту інформації.

Література: основна [1; 2]; додаткова [4; 7; 8].

7. Контрольні запитання для самодіагностики

Модуль 1. Аналіз алгоритмів та алгоритмічні стратегії

Тема 1. Математичні основи аналізу алгоритмів.

Алгоритмічні стратегії

1. Що вивчає дисципліна "Теорія алгоритмів"?
2. Що таке алгоритм?
3. Як оцінити ефективність алгоритму?
4. Що таке складність алгоритму?
5. Чим характеризується будь-який алгоритм?
6. Що таке обчислювальна складність, якою вона буває?
7. Що таке верхня оцінка складності?
8. Як використовується O-нотація для визначення складності алгоритму?
9. Чому алгоритм з поліноміальною оцінкою обчислювальної складності вважається кращим, ніж з експоненціальною?
10. Які алгоритмічні стратегії існують?
11. Що таке просторова складність алгоритму?
12. Що таке асимптотична складність алгоритму?
13. Що називають рекурсією?
14. Що називають ітерацією?
15. Що таке рекурентне співвідношення?
16. Алгоритм, який реалізований в рекурсивній формі, може бути переписаний в ітераційній формі?
17. У чому полягає алгоритм "Решето Ератосфена" для знаходження всіх простих чисел?

18. Які існують алгоритми визначення найбільшого загального дільника?
19. Привести класифікацію алгоритмів залежно від асимптотичної складності?
20. Привести приклад обчислення факторіалу цілого позитивного числа за допомогою рекурсії.

Тема 2. Основи теорії обчислюваності. Класи складності P і NP

1. Які функції називають обчислюваними?
2. Які множини називають вирішуваними? Якими властивостями вони володіють?
3. Що таке характеристична функція для вирішуваної множини?
4. Які множини називають такими, що перераховують? Якими властивостями вони володіють?
5. Що таке напівхарактеристична функція для множини, що перераховує?
6. Які зв'язані множини, що перераховують, і обчислювані функції?
7. Які зв'язані множини, що перераховують і вирішувані множини?
8. Які властивості графіка обчислюваної функції?
9. Які властивості образу і прообразу множини, що перераховує, при обчислюваній функції?
10. У чому полягає теорема Поста?
11. Що розуміють під складністю обчислення?
12. У чому полягають аксіоми Блюма?
13. У чому полягає теорема про прискорення?
14. Що розуміють під класом складності?
15. У чому полягає опис класів P і NP?
16. Навести приклади завдань, що належать P і NP класам.
17. Що розуміють під поліноміальним зведенням NP-повної задачі?
18. У чому полягає теорема Кука?
19. Навести приклади NP-повних задач.
20. Як використовується теорія NP-повноти для аналізу складності завдань?

Модуль 2. Фундаментальні алгоритми та їх побудова

Тема 3. Алгоритми сортування, злиття та пошуку

1. Що розуміють під сортуванням масивів?
2. Яка класифікація алгоритмів сортування?

3. Які існують прості методи сортування?
4. Яка обчислювальна складність простих методів сортування?
5. У чому полягає алгоритм сортування простим обміном?
6. У чому полягає алгоритм сортування простим вибором?
7. У чому полягає алгоритм сортування простими вставками?
8. У чому полягає алгоритм шейкерного сортування?
9. Які існують способи підвищення ефективності простих алгоритмів сортування?
10. Які існують складні (покращені) методи сортування?
11. У чому полягає алгоритм сортування Шелла?
12. У чому полягає алгоритм швидкого сортування?
13. У чому полягає алгоритм сортування злиттям?
14. У чому полягає алгоритм пірамідального сортування?
15. Чи залежить складність алгоритмів сортування від кількості елементів масиву?
16. Чи залежить складність алгоритмів сортування від типу елементів масиву?
17. Як працюють алгоритми пошуку у масивах?
18. Яким чином здійснюється лінійний пошук у масиві?
19. Яким чином здійснюється бінарний пошук у масиві?
20. Яка обчислювальна складність алгоритмів пошуку у масивах?
21. За допомогою яких засобів шаблон може вибрати значення з елемента керування, до якого він застосовується?
22. Яка обчислювальна складність алгоритму сортування Шелла?
23. Яка обчислювальна складність алгоритму швидкого сортування?
24. Яка обчислювальна складність алгоритму сортування злиттям?
25. Яка обчислювальна складність алгоритму пірамідального сортування?
26. Основні параметри, за якими здійснюється оцінка алгоритмів сортування?
27. Охарактеризуйте сферу застосування алгоритмів сортування?
28. Охарактеризуйте стійкість алгоритму сортування простим вибором?
29. Охарактеризуйте стійкість алгоритму сортування методом бульбашки?
30. Які способи покращення алгоритму швидкого сортування?

Тема 4. Комбінаторні та рекурсивні алгоритми

1. Що розуміють під комбінаторною задачею?
2. Що розуміють під розміщенням та сполученням?
3. У чому полягає алгоритм перебору з поверненням?
4. У чому полягає алгоритм "гілок та границь"?
5. У чому полягає алгоритм повного перебору?
6. Як можна обмежити варіанти повного перебору?
7. Які алгоритми використовуються для генерації псевдовипадкових чисел?
8. Оцініть обчислювальну складність рекурсивного алгоритму.
9. Як можна порівняти рекурсивний та не рекурсивний алгоритми?

Тема 5. Фундаментальні алгоритми на графах і деревах

1. Що розуміють під ступенем вершини графа?
2. Що розуміють під петлею?
3. У чому полягає матричне подання графів?
4. Що розуміють під матрицею зв'язності графу?
5. Що розуміють під матрицею відстаней на графі?
6. У чому полягає представлення графа у вигляді списку суміжних вершин?
7. Що розуміють під простим шляхом на графі?
8. У чому полягають алгоритми пошуку найкоротших шляхів та оптимальних маршрутів у графах?
9. Що розуміють під деревом найкоротших шляхів у графах?
10. Що розуміють під Гамільтоновим шляхом на графі?
11. Що розуміють під Ейлеровим шляхом на графі?
12. Які існують способи подання дерев?
13. У чому полягає пошук у ширину на графах та деревах?
14. У чому полягає пошук в глибину на графах та деревах?
15. Що розуміють під перевіркою зв'язності графа?
16. У чому полягає алгоритм Тар'яна знаходження найменшого спільного предка?
17. Що таке орієнтований граф?
18. Що розуміють під орієнтованим шляхом?
19. Що розуміють під задачею створення розкладів?
20. У чому полягає алгоритм Дейкстри?

21. Що розуміють під мінімальним остовним деревом графа?
22. Що розуміють під перетином графа?
23. У чому полягає знаходження мінімального остовного дерева графа за алгоритмом Прима – Краскала?
24. У чому полягає задача комівояжера?
25. У чому полягає алгоритм Крістофідеса?
26. У чому полягає задача про найменше вершинне покриття?

Тема 6. Алгоритми пошуку в рядках та їх обробки

1. Приведіть класифікацію основних алгоритмів пошуку в рядках.
2. Що таке префікс рядка?
3. Що таке суфікс рядка?
4. У чому полягає алгоритм послідовного пошуку підрядка?
5. У чому полягає алгоритм пошуку підрядка Кнута – Морріса – Пратта?
6. У чому полягає алгоритм пошуку підрядка Бойера – Мура?
7. У чому полягає алгоритм бінарного пошуку рядків?
8. У чому полягає алгоритм Карпа – Рабіна?
9. У чому полягає алгоритм наближеного пошуку рядка?
10. Що таке дерево суфіксів?
11. Що таке зворотне префіксне дерево?
12. Що таке вкладений суфікс рядка?
13. У чому полягають прості алгоритми побудови дерева суфіксів?
14. Що таке масиви суфіксів?
15. У чому полягає алгоритм Укконена?
16. Які основні алгоритми обробки рядків?
17. У чому полягають алгоритми розбиття рядків?
18. У чому полягають об'єднання рядків?
19. У чому полягають алгоритми вставки рядків?
20. У чому полягають алгоритми видалення рядків?
21. У чому полягають алгоритми заміни рядків?
22. Які основні алгоритми пошуку найбільш спільного підрядка двох рядків?
23. Що таке відстань між рядками?
24. Які основні алгоритми обчислення відстані між рядками?
25. У чому полягає алгоритм Вагнера – Фішера?
26. У чому полягає алгоритм Хешберга?

27. Що таке дерево для рядкової послідовності?
28. У чому полягає наближене порівняння рядків з патерном?
29. Які основні алгоритми наближеного порівняння рядків з патерном?
30. У чому полягає алгоритм Майерса?

Тема 7. Геометричні алгоритми

1. Що розуміють під обмежувачим прямокутником геометричної фігури?
2. Як визначити довжину відрізка в n -вимірному просторі?
3. Які умови належності точки прямій?
4. Які умови перетину відрізків?
5. Які умови перетину відрізків на прямій?
6. Як визначити відстань від точки до відрізка?
7. Що розуміють під параметричним рівнянням відрізка?
8. Що розуміють під довжиною спільної частини відрізків?
9. У чому полягає алгоритм визначення взаємного розташування точок, прямих та відрізків на площині?
10. У чому полягають алгоритми обробки багатокутників?
11. У чому полягають алгоритми визначення взаємного розташування кіл, прямих та точок.
12. У чому полягає алгоритм побудови опуклої оболонки?
13. Який порядок обходу вершин опуклого багатокутника?
14. Що розуміють під задачами мінімізації в геометричній інтерпретації?

Тема 8. Евристичні та криптографічні алгоритми

1. Які алгоритми називають евристичними?
2. Коли виникає потреба використовувати евристичні алгоритми?
3. Викласти "жадібний алгоритм" для "задачі про комівояжера".
4. Викласти алгоритми розв'язку задачі про процесори.
5. Як оцінюється помилка при використанні евристичних алгоритмів? Привести приклади.
6. Що таке програмування з відходом назад? Яка структура даних при цьому звичайно використовується?
7. У чому полягає метод динамічного програмування?
8. Які задачі розв'язують методом динамічного програмування?

9. Як оцінити робочу функцію алгоритму з відходом назад? У яких випадках доцільно цей метод використовувати?

8. Індивідуально-консультативна робота

Індивідуально-консультативна робота виконується за графіком у таких формах: індивідуальні заняття, консультації, перевірка виконання індивідуальних завдань і захист результатів їхнього виконання тощо.

Формами організації індивідуально-консультативної роботи є:

а) консультації з теоретичного матеріалу:

інтерактивне спілкування (питання – відповідь);

групові (розгляд типових завдань);

диспути;

б) індивідуальні та групові консультації з освоєння практичного матеріалу;

в) індивідуальна задача та захист виконаних робіт для комплексної оцінки ступеня освоєння програмного матеріалу.

9. Методики активізації процесу навчання

Викладання навчальної дисципліни "Теорія алгоритмів" передбачає активізацію пізнавальної діяльності студентів за рахунок використання таких навчальних технологій, як: проблемні лекції, робота в малих групах, мозкові атаки, кейс-методи, презентації тощо (табл. 5).

Таблиця 5

Використання навчальних технологій для активізації процесу навчання

Методики активізації процесу навчання	Практичне використання навчальних технологій
1	2
<p>Проблемні лекції спрямовані на розвиток логічного мислення студентів, коло питань теми обмежується двома – трьома ключовими моментами; при читанні лекції перед студентами формулюється проблема для самостійного осмислення того, що далі розкривається викладачем; у ході лекції студентам видається надрукований роздатковий матеріал (там, де це доречно) та здійснюється показ слайдів презентацій</p>	Проблемна лекція з питання "Математичні основи аналізу алгоритмів. Алгоритмічні стратегії" (з теми 1)
	Проблемна лекція з питання "Основи теорії обчислюваності. Класи складності P і NP" (з теми 2)
	Проблемна лекція з питання "Алгоритми сортування, злиття та пошуку" (з теми 3)
	Проблемна лекція з питання "Фундаментальні алгоритми на графах і деревах" (з теми 5)

1	2
	Проблемна лекція "Геометричні алгоритми" (з теми 7)
	Проблемна лекція з питання "Евристичні та криптографічні алгоритми" (з теми 8)
<p>Робота в малих групах дає можливість для кожного студента на лабораторних заняттях індивідуально використовувати комп'ютери, що створює сприятливі умови активізації його роботи при виконанні дослідницьких дій та при пошуку оптимальних технічних рішень</p>	Робота в малих групах під час визначення динамічних структур даних (лабораторна робота 2)
	Робота в малих групах при розробленні алгоритмів на графах та деревах (лабораторна робота 5)
	Робота в малих групах під час оцінки ефективності криптографічних алгоритмів (лабораторна робота 8)
<p>Мозкові атаки – метод вирішення суперечливих технологічних ситуацій, сутність якого полягає в тому, щоб визначити більшу кількість ідей за короткий проміжок часу, обговорити й виконати селекцію оптимальних технічних рішень</p>	Мозкова атака з аналізом можливих варіантів використання алгоритмів сортування масивів залежно від початкового розташування елементів (лабораторне заняття з теми 3)
	Мозкова атака, пов'язана з аналізом вибору оптимального алгоритму для пошуку найкоротшого шляху в графі (лабораторне заняття з теми 5)
	Мозкова атака, пов'язана зі способами обчислення кількості комбінаторних елементів при використанні алгоритму повного перебору (лабораторне заняття з теми 4)
<p>Кейс-метод – метод аналізу конкретних ситуацій, що дає можливість наблизити процес навчання до реальної практичної діяльності фахівців і передбачає розгляд виробничих, управлінських і інших ситуацій</p>	Проблемна ситуація, пов'язана з вибором між простими та складними алгоритмами сортування (лабораторне заняття з теми 3)
	Управлінська ситуація пов'язана з застосуванням засобів автоматизованого тестування програм для оформлення звітів з лабораторних робіт (лабораторні заняття з тем 1 – 8)
	Проблемна ситуація, пов'язана з реалізацією оптимального алгоритму побудови опуклої оболонки (лабораторне заняття з теми 7)
Управлінська ситуація під час вибору та реалізації потрібного динамічного типу даних (лабораторне заняття з теми 2)	

1	2
	Конфліктна ситуація, пов'язана з різким зростанням обчислювальної складності алгоритму повного перебору (лабораторне заняття з теми 4)
	Управлінська ситуація під час додавання електронного підпису при реалізації криптографічного алгоритму (лабораторне заняття з теми 8)
Презентації – виступи перед аудиторією з поданням результатів роботи, з поданням звітів про виконання індивідуальних завдань, з демонстрацією технічних рішень на основі сучасних комп'ютерних технологій	Презентація результатів аналізу різних алгоритмів визначення простих чисел
	Презентація результатів розробки власного динамічного типу даних
	Презентація результатів дослідження ефективності різних алгоритмів сортування масивів
	Презентація результатів аналізу обчислювальної складності комбінаторних алгоритмів
	Презентація результатів застосування алгоритмів пошуку в ширину та в глибину
	Презентація результатів реалізації криптографічних алгоритмів для захисту інформації

10. Система поточного та підсумкового контролю знань

10.1. Форми перевірки й оцінювання знань

Система оцінювання знань, умінь і навичок студентів урахує види занять, які передбачені програмою навчальної дисципліни "Теорія алгоритмів": лекції, лабораторні заняття, а також самостійна робота і виконання індивідуальних завдань.

Контрольні заходи передбачають поточний і підсумковий контроль.

Тестування як засіб перевірки знань має такі переваги.

1. Тести є значно якіснішими і об'єктивним способом оцінювання завдяки стандартизованій процедурі проведення. На всіх етапах тестування неможливо внести суб'єктивну складову в оцінку, вона не залежить від настрою викладача, його стосунків з конкретним студентом, враженням від відповідей на попередні запитання.

2. Тести мають велику ємність. Їхні показники орієнтовані на вимірювання ступеня, визначення рівня засвоєння ключових понять, тем і розділів навчальної програми, умінь, навичок, а не на констатацію наявності у студентів певної сукупності засвоєних знань. Стандартизована форма оцінки досягнень дозволяє співставити рівень досягнень студента з дисципліни в цілому та окремих його розділів із середнім рівнем досягнень студентів у групі й рівнями досягнень кожного з них. Виконуючи підсумкову тестову роботу, кожний студент використовує знання з усіх тем програми.

3. Тести є м'якішим інструментом. Тестування ставить усіх студентів у рівні умови, використовуючи єдину процедуру і єдині критерії оцінки, що призводить до зниження передекзаменаційних нервових напружень.

4. Тести мають ширшу шкалу оцінювання, її можна розширити як угору, так і донизу.

5. Гуманізм тестування полягає в тому, що всім надаються рівні можливості, широта тесту дозволяє студенту показати свої досягнення на широкому полі матеріалу. Студент має деяке право на помилку, яке при традиційному способі оцінювання відсутнє.

З урахуванням місця в навчальній дисципліні "Теорія алгоритмів" передбачаються такі різновиди тестів:

1. Превентивні тести. Містять анонс матеріалів, з якими має познайомитися студент у новому розділі або дисципліні. Ілюструють, які саме знання й навички отримає студент у результаті його освоєння. Тут матеріали подаються в привабливій, емоційно яскравій формі, шляхом включення мультимедійних об'єктів (графіка, відео, аудіо) у текст запитання. Ці тести доступні для незареєстрованих (потенційних) студентів. Широко використовують мультимедійні форми подання матеріалу при повідомленні про помилки. Такі повідомлення провокуються відсутністю правильних відповідей з метою підказати, які методики має вивчити студент, щоб відповідати правильно.

2. Вхідний контроль. Передує вивченню дисципліни. Заснований на загальному банку завдань (він компілюється розробником, а не створюється). При неправильній відповіді повинен містити посилання до інших дисциплін. Його можна ініціювати повторно, але обов'язковий позитивний результат.

3. Самоконтроль. Ініціюється (у т. ч. і повторно) і перевіряється студентом. Позитивний результат не обов'язковий. Передбачається, що він передує тестам проміжного або підсумкового контролю, сприяючи адаптації студента до програми тестування й для попередньої самоперевірки. При неправильній відповіді повинен містити докладні коментарі.

4. Проміжний і підсумковий контроль. Проміжний контроль завершує частину дисципліни, а підсумковий – всю дисципліну. Вимагає авторизації, найчастіше накладає обмеження на того, хто тестується за часом і місцем проведення тесту. Ураховується не тільки точність відповіді, але і його час. Передбачає індивідуальний добір списку питань. Результати подаються студенту в узагальненому вигляді (оцінка), але бажано структурувати оцінку за тематичними блоками.

5. Контроль залишкових знань. Проводиться за рамками дисципліни. Перевірка й оцінювання знань під час вивчення дисципліни "Теорія алгоритмів" проводяться в таких формах:

1. Оцінювання знань під час проведення контролю підготовленості студентів до виконання лабораторних робіт.

2. Оцінювання знань студентів під час захисту звітів з лабораторних робіт.

3. Оцінювання результатів виконання індивідуального завдання.

4. Проведення проміжного тестового контролю.

5. Проведення поточного модульного контролю з кожного модуля навчальної дисципліни.

6. Проведення підсумкового письмового іспиту.

10.2. Оцінювання знань та вмінь під час проведення лабораторних занять

Оцінювання знань студентів під час проведення контролю підготовленості студентів до виконання лабораторних робіт має на меті перевірку рівня освоєння теоретичних положень з теми лабораторної

роботи, ступеня підготовленості студентів до виконання індивідуальних завдань (варіантів), перевірку якості самостійної розробки ескізів документів, які повинні бути одержані програмно, і т. д.

Оцінювання проводиться за 12-бальною системою за такими критеріями:

а) розуміння, ступінь освоєння теорії й практики досліджуваної теми;

б) ступінь вивчення матеріалів рекомендованої літератури, а також сучасної літератури з досліджуваних питань;

в) уміння застосовувати теорію при розв'язанні практичних задач на основі сучасних комп'ютерних технологій, уміння обґрунтовувати прийняті технічні рішення;

г) логіка, структура, стиль викладення матеріалу усно або письмово, уміння робити обґрунтовані висновки з питань, що викладаються.

Оцінювання знань студентів під час прийому звітів з лабораторних робіт виконується із застосуванням таких критеріїв:

а) якість виконання індивідуального завдання до лабораторної роботи;

б) ступінь самостійності виконання завдання;

в) обґрунтованість прийнятих у роботі технічних рішень;

г) повнота й глибина аналізу отриманих результатів;

д) наявність і повнота ілюстрацій (використовувані діалогові вікна, таблиці, графіки, формули й т. д.);

е) якість оформлення звіту.

Добір завдань має відбуватися так, щоб він був посильним для розуміння всіма студентами, з врахуванням їхніх навчально-пізнавальних можливостей і когнітивних стилів. З цією метою кожний студент самостійно вибирає рівень складності з поміж таких:

1) фронтальний,

2) індивідуальний,

3) компетентнісний.

Якщо вибрано фронтальний рівень, то студент виконує завдання базового рівня, що детально описано в інструкції. За його виконання студент отримує чотири бали за дванадцятибальною системою оцінювання.

З метою випробування своїх сил і підвищення оцінки студент може самостійно розв'язати ще кілька задач, частина з яких репродуктивного, а інші – креативного типу. За правильне їхнє вирішення додається ще до двох балів. До отриманої суми балів студент може додати ще два бали, якщо самостійно запропонує і розв'яже оригінальну задачу за темою, що вивчається. Ця задача має бути з предметної галузі навчання чи майбутньої професії студента. Загальна оцінка за цим рівнем не перевищує восьми балів.

У разі вибору індивідуального рівня студент ознайомлюється з інструкцією щодо виконання завдання базового рівня і розв'язує аналогічну задачу з множини варіантів, поданих в інструкції. За виконання такого індивідуального завдання студент отримує шість балів. Ще два бали він може отримати, якщо адаптує до предметної галузі обраного варіанту задачі, які подані в інструкції і розв'яже їх. Подібно до фронтального рівня студент може додати ще два бали до отриманої суми балів, якщо сформулює і розв'яже оригінальну задачу за темою, що вивчається. Загальна оцінка за цим рівнем не перевищує десяти балів.

На компетентністному рівні студент демонструє можливість самостійно ставити і розв'язувати задачі за темою, що вивчається з предметної галузі навчання чи майбутньої професії. Спочатку він формулює і розв'язує задачу аналогічну базовій (сім балів), потім – аналогічні додатковим задачам (ще два бали) і на сам кінець – оригінальну задачу (до трьох балів). Загальна оцінка за цим рівнем може досягати дванадцяти балів.

У табл. 6 зведено критерії оцінювання за рівнями складності завдань, що виконуються на лабораторній роботі. З таблиці видно, що найбільш слабкий, але добросовісний студент не може отримати менше чотирьох балів, а найуспішнішим студентам є можливість отримати дванадцять балів, продемонструвавши високий рівень компетентності і креативності.

Критерії оцінювання лабораторних робіт за рівнями складності

Фронтальний		Індивідуальний		Компетентністний	
Задача	Максимальна оцінка	Задача	Максимальна оцінка	Задача	Максимальна оцінка
Базова	4	Варіант по базовій	6	Оригінальна аналогічна базовій	7
Додаткові	+2	За варіантом аналогічні додатковим фронтального рівня	+2	Оригінальні аналогічні додатковим фронтального рівня	+2
Оригінальна	+2	Оригінальна за варіантом	+2	Оригінальна підвищеної складності	+3
Загальна оцінка	8		10		12

10.3. Тестовий контроль

Проміжний тестовий контроль проводиться із застосуванням персонального комп'ютера або у вигляді експрес-контрольної двічі протягом поточного модуля. При проведенні поточного тестового контролю визначається рівень знань студентів з теоретичних питань навчальної дисципліни.

Бази тестових завдань охоплюють основні теми навчальної дисципліни. Кількість запитань у експрес-контрольній не менше 10, контрольний час – 30 сек. на запитання.

З наявного множини тестових питань (80 – 100 залежно від теми) програма тестування випадковим чином вибирає зазначене число (зазвичай їх 15). Тест проводиться протягом обмеженого часу. Практика показала, що коли тести здаються після виконання лабораторної роботи та часткового вирішення креативних завдань, студент з першого разу в середньому отримує 9 балів, а якщо до виконання лабораторної роботи, то 3 – 4 бали.

При застосуванні персонального комп'ютера тестова програма містить тестові запитання і набір можливих відповідей. Кількість варіантів відповідей з кожного запитання може досягати 20. При цьому обмеження на кількість правильних і неправильних відповідей не накладаються. Кількість запитань при кожному тестуванні та час, що відводиться на нього, визначається викладачем програмно.

Приклад питань експрес-контрольної:

1. Виникнення терміну "алгоритм" пов'язують з ім'ям:

- a) Архімеда;
- b) Евкліда;
- c) Аль-Хорезмі;
- d) Ньютона.

2. Алгоритм обчислення суми елементів двовимірного масиву має обчислювальну складність:

- a) експоненціальну;
- b) лінійну;
- c) квадратичну;
- d) кубічну;
- e) логарифмічну.

3. Алгоритм Шелла використовують для:

- a) пошуку циклів на графі;
- b) перевірки числа на простоту;
- c) знаходження відстані між точками на площині;
- d) сортування масиву;
- e) шифрування інформації.

4. Алгоритм швидкого сортування використовує:

- a) будь-який елемент;
- b) опорний елемент;
- c) максимальний елемент;
- d) мінімальний елемент;
- e) випадковий елемент.

5. До динамічних структур даних належать:

- a) стек;
- b) змінні цілого типу;
- c) дек;
- d) черга.

6. Алгоритм Дейкстри на графі знаходить:

- a) всі шляхи;
- b) всі вершини;
- c) найкоротшу відстань;
- d) мінімальне остовне дерево.

7. До алгоритмів пошуку підрядка в рядку належать:

- a) алгоритм Кнута – Морріса – Пратта;
- b) алгоритм Уоршелла;
- c) алгоритм Бойера – Мура;
- d) алгоритм Карпа – Рабіна;
- e) алгоритм Евкліда.

8. Комбінаторні алгоритми можуть обчислювати:

- a) кількість розміщень;
- b) кількість порівнянь;
- c) кількість сполучень;
- d) кількість перестановок.

9. До комбінаторних алгоритмів відносяться:

- a) алгоритм Флойда;
- b) алгоритм Беллмана – Форда;

- c) алгоритм Дейкстри;
- d) алгоритм Кнута – Морриса – Пратта.

10. До алгоритмів обчислення найкоротший шляху в графі належить:

- a) алгоритм Флойда;
- b) алгоритм Беллмана – Форда;
- c) алгоритм Дейкстри;
- d) алгоритм Кнута – Морриса – Пратта.

11. Алгоритми формування псевдовипадкових чисел належать:

- a) алгоритмів на графах;
- b) комбінаторних алгоритмів;
- c) алгоритмів пошуку;
- d) алгоритмів визначення максимального потоку.

12. Розширений алгоритм Евкліда використовується для визначення:

- a) ряду простих чисел;
- b) найбільшого загального дільника;
- c) розкладання числа на прості множники;
- d) результату множення великих чисел.

13. Алгоритм Фібоначчі використовується для визначення:

- a) ряду чисел Фібоначчі;
- b) результату порівняння рядків з патерном;
- c) коду, який кодує позитивні цілі числа в двійкові кодові слова;
- d) відстані між рядками.

14. Алгоритм двійкового пошуку базується на використанні:

- a) бінарного дерева для зберігання елементів;
- b) визначення елемента у відсортованому списку;
- c) визначення елемента у не відсортованому списку;
- d) визначення k -го по величині елемента в списку.

15. Алгоритм Беллмана – Форда використовується для визначення:

- a) найкоротшого шляху у зваженому графі;
- b) мінімального остовного дерева в графі;

- с) найкоротшого шляху в графі з ненегативними вагами ребер;
 d) всіх найкоротших шляхів у зваженому графі.

Для оцінювання рівня підготовленості студентів результати тестування визначаються в 12-бальній системі відповідно до табл.7.

Таблиця 7

Оцінювання експрес-контрольних

Оцінка	Відсоток правильних відповідей на питання тесту
1	2
Дванадцять	від 96 до 100
Одинадцять	від 91 до 95
Десять	від 84 до 90
Дев'ять	від 76 до 83
Вісім	від 71 до 75
Сім	від 64 до 70
Шість	від 61 до 65
П'ять	від 56 до 60
Чотири	від 51 до 55
Три	від 36 до 50
Два	від 16 до 35
Один	від 0 до 15

10.4. Підсумковий письмовий іспит

Підсумковий письмовий іспит за весь термін вивчення навчальної дисципліни проводиться та оцінюється за двома складовими: практичний контроль та теоретичний контроль.

Оцінка за практичну складову письмового іспиту виставляється за результатами оцінювання якості та оригінальності виконання алгоритму практичного завдання, результатів комп'ютерного тестування цього завдання.

Теоретичний модульний контроль здійснюється в письмовій формі за відповідними завданнями, зміст яких охоплює основні теми, які вивчалися протягом семестру.

Кожна складова підсумкового письмового іспиту оцінюється за 12-бальною системою, а підсумкова оцінка визначається шляхом усереднення сукупності оцінок.

Далі наведений зразок завдання до підсумкового письмового іспиту.

Критерії оцінки підсумкового письмового іспиту з навчальної дисципліни "Теорія алгоритмів" за напрямом "Комп'ютерні науки" такі:

Зразок завдання до підсумкового письмового іспиту

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
Напрямок підготовки "Комп'ютерні науки"
Навчальна дисципліна
"Теорія алгоритмів"

Теоретична частина (2 бали)

1. Навести основні властивості алгоритму і дати їх коротку характеристику (1 бал).

2. Перерахувати основні алгоритми сортування масивів і навести оцінку обчислювальної складності кожного алгоритму (1 бал).

Практична частина (10 балів)

За отриманим варіантом виконати такі завдання:

Завдання 1. Скласти схему алгоритму розв'язання задачі (3 бали).

Завдання 2. Реалізувати алгоритм на алгоритмічній мові високого рівня (5 балів).

2.1. Реалізувати введення та перевірку вихідних даних (1 бал).

2.2. Реалізувати обчислення (2 бали).

2.3. Реалізувати виведення результатів у текстовий файл (1 бал).

2.4. Розробити тестовий приклад і перевірити правильність отриманих результатів (1 бал).

Завдання 3. Обґрунтувати обчислювальну складність розробленого алгоритму (2 бали).

3. 1. Обґрунтувати просторову складність (1 бал).

3.2. Обґрунтувати часову складність (1 бал).

Викладач _____

Оцінка 12 балів. Студент дає абсолютно правильні відповіді на 100 % запитань. Теоретичне запитання розкрито повністю, з викладенням оригінальних висновків, отриманих на основі програмного, додаткового матеріалу, законодавчих актів та нормативних документів. При виконанні практичної частини студент застосовує системні знання навчального матеріалу, передбачені навчальною програмою. Практичні за-

вдання виконані як з використанням типового алгоритму, так і за самостійно розробленим алгоритмом, висновки до завдань аргументовані та обґрунтовані.

Оцінка 11 балів. Студент дає правильні відповіді на запитання. Теоретичне запитання розкрито повністю, на основі програмного та додаткового матеріалу зроблено висновки й узагальнення. При виконанні практичної частини завдання студент застосовує узагальнені знання навчального матеріалу, що передбачені навчальною програмою. Практичні завдання виконуються як з використанням типового алгоритму, так і за самостійно розробленим алгоритмом, робляться аргументовані висновки.

При виконанні практичної частини завдання припускається декілька незначних неточностей.

Оцінка 10 балів. Студент дає правильні відповіді на запитання. Теоретичне запитання розкрито повністю, програмний матеріал викладено згідно з програмним матеріалом дисципліни. При виконанні практичної частини завдання студент застосовує глибокі знання навчального матеріалу, що передбачені навчальною програмою. Практичні завдання виконуються в цілому правильно, в повному обсязі як з використанням типового алгоритму, так і в дещо змінених умовах. При виконанні практичних завдань припускається декілька окремих неточностей.

Оцінка 9 балів. Студент дає правильні відповіді на запитання. Теоретичне запитання розкрито повністю, програмний матеріал викладено згідно з матеріалом дисципліни. При виконанні практичної частини завдання студент ефективно застосовує основні знання навчального матеріалу, що передбачені навчальною програмою. Практичні завдання виконуються в цілому правильно з використанням типового алгоритму, при їх виконанні припускається декілька окремих несуттєвих помилок.

Оцінка 8 балів. Теоретичне запитання розкрито повністю, програмний матеріал викладено з незначними помилками або без узагальнень. При виконанні практичної частини завдання студент застосовує основні знання навчального матеріалу, що передбачені навчальною програмою. Практичні завдання виконуються в цілому правильно з використанням типового алгоритму, при їх виконанні студент допустив несуттєві помилки.

Оцінка 7 балів. Теоретичне запитання студентом розкрито повністю, проте при викладенні програмного матеріалу допущені незначні помилки. При виконанні практичної частини завдання студент застосовує

основні знання навчального матеріалу, що передбачені навчальною програмою. Практичні завдання виконуються в цілому правильно з використанням типового алгоритму. При їх виконанні студент допустив декілька суттєвих помилок.

Оцінка 6 балів. Теоретичне запитання студентом розкрито неповно, допущено суттєві або помітні помилки. При виконанні практичної частини завдання без достатнього розуміння студент застосовує навчальний матеріал, припускається декількох суттєвих помилок.

Оцінка 5 балів. Теоретичне запитання студентом розкрито неповно, допущено суттєві помилки, які впливають на зміст відповіді. При виконанні практичної частини завдання без достатнього розуміння студент застосовує навчальний матеріал, припускається декількох значних помилок.

Оцінка 4 бали. Теоретичне запитання студентом розкрито неповно, з суттєвими помилками. При виконанні практичної частини завдання без достатнього розуміння студент застосовує навчальний матеріал, припускається значної кількості суттєвих помилок, стикається зі значними труднощами при аналізі та порівнянні аналітичних аспектів використання навчальних методів.

Оцінка 3 бали. Теоретичне запитання студентом розкрито неповно, або зовсім не розкрито. При виконанні практичної частини завдання припускається досить великої кількості грубих помилок, стикається зі значними труднощами при використанні інструментальних засобів обробки інформації, виявляє нездатність до викладення думки на елементарному рівні.

Оцінка 2 бали. Теоретичне запитання студентом зовсім не розкрито. Виконати практичні завдання не може, стикається зі значними труднощами при виконанні, виявляє нездатність до викладення думки на елементарному рівні.

Оцінка 1 бал. Теоретичне запитання студентом зовсім не розкрито. Виконати практичні завдання не може, стикається зі значними труднощами при використанні програмних продуктів.

Оцінка 0 балів. Оцінка 0 балів виставляється у випадку, коли студент не виконував завдання у зв'язку з великою кількістю пропусків занять.

10.5. Підсумкова оцінка з навчальної дисципліни

Для підведення підсумків роботи студентів з дисципліни ставиться загальна оцінка (середня оцінка округлюється на користь студентові), яка

враховує оцінки кожного виду контролю: оцінювання знань студентів під час лабораторних занять та захист робіт; оцінювання знань самостійного опрацювання теоретичного матеріалу (опитування студентів під час лекцій); проведення підсумкового письмового іспиту.

Підсумкова оцінка з навчальної дисципліни "Теорія алгоритмів" розраховується за формулою:

$$P_o = 0.6 \times P_{\text{ПІ}} + 0.4 \times (M_1 + M_2)/2,$$

де P_o – підсумкова оцінка з навчальної дисципліни;

$P_{\text{ПІ}}$ – оцінка підсумкового письмового іспиту;

M_1, M_2 – оцінки за першим і другим модулями.

Для отримання остаточної оцінки, підсумкова оцінка округлюється до найближчого цілого.

11. Рекомендована література

11.1. Основна

1. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест и др. – М. : ИД "Вильямс", 2011. – 1296 с. : ил.

2. Ахо А. Структуры данных и алгоритмы : учебн. пособ. / А. Ахо, Д. Хопкрофт, Д. Ульман ; пер. с англ. – М. : ИД "Вильямс", 2000. – 384 с.

3. Седжвик Р. Алгоритмы на C++. Фундаментальные алгоритмы и структуры данных / Р. Седжвик. – М. : ИД "Вильямс", 2011. – 1056 с. : ил.

11.2. Додаткова

4. Ахо А. Построение и анализ вычислительных алгоритмов / А. Ахо, Д. Хопкрофт, Д. Ульман. – М. : Мир, 1979. – 536 с.

5. Вирт Н. Алгоритмы и структуры данных / Н. Вирт ; пер с англ.. – М. : Мир, 1989. – 360 с.

6. Кнут Д. Э. Сортировка и поиск : учебн. пособ. / Д. Э. Кнут ; пер. с англ. 2-е изд. – М. : ИД "Вильямс", 2000. – 832 с.

7. Марков А. А. Теория алгоритмов / А. А. Марков, Н. М. Нагорный. – М. : Наука, 1984. – 432 с.

8. Техніка обчислень і алгоритмізація / І. Ф. Следзінський, А. М. Ломакович, Ю. С. Рамський та ін. – К. : Вища шк., 1991. – 199 с.

11.3. Ресурси мережі Інтернет

9. Теорія алгоритмів. Вікіпедія. [Електронний ресурс]. – Режим доступу : <http://uk.wikipedia.org/wiki/>.
10. Розвиток теорії алгоритмів [Електронний ресурс]. – Режим доступу : <http://wiki.kspu.kr.ua>.
11. Теорія алгоритмів [Електронний ресурс]. – Режим доступу : <http://cybportal.univ.kiev.ua/wiki/>.
12. Дискретная математика: алгоритмы. Теория [Электронный ресурс]. – Режим доступа : <http://rain.ifmo.ru/cat/view.php/theory>.
13. Введение в теорию алгоритмов [Электронный ресурс]. – Режим доступа : <http://th-algoritmov.narod.ru/1.htm>.
14. Список алгоритмов и структур данных на C++ [Электронный ресурс]. – Режим доступа : http://sites.google.com/site/indy256/algo_cpp.
15. Литература по дискретной математике [Электронный ресурс]. – Режим доступа : <http://eek.diary.ru/p49631731.htm>.
16. Sorting Algorithm Animations [Electronic resource]. – Access mode : <http://www.sorting-algorithms.com/>.
17. Алгоритмы сортировки на Си [Электронный ресурс]. – Режим доступа : http://www.adrutsa.ru/content/codes/sort_array_c.html.
18. Алгоритмы и структуры данных. Видеолекции в свободном доступе. [Электронный ресурс]. – Режим доступа : <http://www.lektorium.tv/course/?id=22823>.

Зміст

Вступ	3
1. Кваліфікаційні вимоги до студентів	5
2. Тематичний план початкової дисципліни	6
3. Зміст навчальної дисципліни за модулями та темами	8
4. Плани лекцій	11
5. Плани лабораторних занять	13
6. Самостійна робота студентів	15
6.1 Загальні положення	15
6.2. Питання для самостійного опрацювання	15
7. Контрольні запитання для самодіагностики	19
8. Індивідуально-консультативна робота	25
9. Методики активізації процесу навчання	25
10. Система поточного та підсумкового контролю знань	27
10.1. Форми перевірки й оцінювання знань	27
10.2. Оцінювання знань та вмінь під час проведення лабораторних занять	29
10.3. Тестовий контроль	33
10.4. Підсумковий письмовий іспит	36
10.5. Підсумкова оцінка з навчальної дисципліни	39
11. Рекомендована література	40
11.1. Основна	40
11.2. Додаткова	40
11.3. Ресурси мережі Інтернет	41

НАВЧАЛЬНЕ ВИДАННЯ

**Робоча програма
навчальної дисципліни
"ТЕОРІЯ АЛГОРИТМІВ"
для студентів напряму підготовки 6.050101
"Комп'ютерні науки"
всіх форм навчання**

Укладачі: **Щербаков** Олександр Всеволодович
Лосєв Михайло Юрійович

Відповідальний за випуск **Пономаренко В. С.**

Редактор **Пушкар І. П.**

Коректор **Мартовицька-Максимова В. А.**

План 2012 р. Поз. № 273.

Підп. до друку Формат 60 x 90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. 2,75. Обл.-вид. арк. 3,44. Тираж прим. Зам. №

Видавець і виготівник — видавництво ХНЕУ, 61166, м. Харків, пр. Леніна, 9а

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № 481 від 13.06.2001 р.*