

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІРПІНСЬКИЙ ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА ПРАВА**

ЗАТВЕРДЖУЮ

Заступник директора з навчальної роботи

 Олена ПУСТОВА

«9» 09 2023 р.

РОБОЧА ПРОГРАМА

**навчальної дисципліни «Теорія алгоритмів»
для підготовки фахового молодшого бакалавра
за галуззю знань 12 Інформаційні технології
спеціальності 122 «Комп'ютерні науки»**

Ірпінь – 2023

Робоча програма навчальної дисципліни «Теорія алгоритмів» складена на основі програми навчальної дисципліни, затвердженої у 2023 році.

Розробник:

Анастасія ІЩУК


Розглянуто і схвалено на засіданні циклової комісії інформаційних технологій, математичних та природничих дисциплін, протокол №02 від «31» серпня 2023 р.

Голова циклової комісії



Анастасія ІЩУК

Завідувач навчально-методичного
кабінету коледжу



Людмила РОМАНЕНКО

ЗМІСТ

ПЕРЕДМОВА.....	4
КРИТЕРІЇ ОЦІНЮВАННЯ	5
ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	7
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ	20

ПЕРЕДМОВА

У сучасному світі, де обчислювальна техніка переплетена з усіма аспектами нашого життя, знання про алгоритми є надзвичайно важливими. Вони визначають те, як ми обробляємо дані, вирішуємо проблеми та створюємо інновації. Навчальна дисципліна «Теорія алгоритмів» допоможе зрозуміти, як працюють алгоритми та як їх створювати, щоб вони були ефективними.

Алгоритми використовують у всіх сферах, пов'язаних з інформаційними технологіями. Вони є основою програмування. Знання існуючих алгоритмів і їхнє застосування дає програмістам можливість швидше створювати надійні й ефективні програми для розв'язування практичних задач. Невід'ємною складовою роботи кожного спеціаліста в даній галузі також є розробка нових алгоритмів. Підготовка конкурентноспроможного фахівця спрямована не тільки на надання їм певних знань і вироблення відповідних навичок застосовувати ці знання, а й на формування в них алгоритмічного мислення. Для цього, насамперед, треба проаналізувати досить велику кількість різноманітних алгоритмів, зрозуміти принципи їхньої роботи, з'ясувати умови і обмеження щодо їхнього використання, застосовувати в практичних навчальних завданнях існуючі алгоритми, навчитися адаптувати їх, модифікувати і розробляти свої.

Засвоєння принципів розробки алгоритмів, відомих стандартних алгоритмів, вивчення їх внутрішньої будови є важливою складовою базової комп'ютерної підготовки фахової передвищої освіти. Рівень комп'ютерної підготовки визначається вмінням розробляти нові ефективні алгоритми.

Мета: 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.

Вивчення навчальної дисципліни «Теорія алгоритмів» передбачає набуття здобувачами освіти програмних компетентностей, а саме:

Загальні компетентності (ЗК):

ЗК 3. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 4. Здатність застосовувати знання у практичних ситуаціях.

ЗК 8. Здатність вчитися і оволодівати сучасними знаннями.

Спеціальні компетентності (СК):

СК 2. Здатність використовувати теоретичні та фундаментальні знання в галузі комп'ютерних наук та інформаційних технологій для вирішення різноманітних проблем.

СК 3. Здатність розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання конкретних професійних задач залежно від предметного середовища

Очікувані результати навчання

Результати навчання (РН):

РН 3. Використовувати професійно-профільовані знання і практичні навички методів фундаментальної та прикладної математики під час розв'язання стандартних задач і задач прикладного характеру в галузі комп'ютерних наук.

РН 4. Застосовувати сучасні методи математичного та комп'ютерного моделювання і будувати ефективні алгоритми для чисельного дослідження та розв'язання прикладних задач.

Передумовами вивчення даної дисципліни є вивчення предмету «Інформатика», навчальних дисциплін «Алгоритмізація та програмування», «Вища математика», «Дискретна математика». В свою чергу є базою для вивчення подальших навчальних дисциплін, а саме: «Об'єктно-орієнтоване програмування», «Крос-платформне програмування», «Технологія створення програмних продуктів», «Технології захисту інформації» та ін.

В процесі викладання дисципліни використовується **5-бальна система оцінювання**.

У навчальному процесі використовуються такі методи навчання:

- лекція;
- лабораторна робота;
- самостійна робота.

Оцінювання знань студентів здійснюється у формі усного опитування, тестів, оцінювання лабораторних робіт. Після вивчення навчальної дисципліни «Теорія алгоритмів» студенти складають екзамен.

Форми та засоби поточного і підсумкового контролю

Об'єктивність оцінки знань студентів залежить від форм та засобів поточного й підсумкового контролю. Процес вивчення курсу передбачає здійснення поточного та підсумкового контролю. Форми проведення поточного контролю, їх періоди визначаються робочим планом викладача. Поточний контроль проводиться викладачем на лекційних, контрольних, лабораторних заняттях, під час перевірки самостійних робіт. Формою підсумкового семестрового контролю є екзамен, що визначається навчальним планом. Підсумковий контроль здійснюється після вивчення курсу під час екзамену. До екзамену допускаються студенти, які засвоїли всі теми програми, виконали лабораторні, рубіжну контрольну роботу.

КРИТЕРІЇ ОЦІНЮВАННЯ

(5-бальна шкала)

Критерії оцінки знань і вмінь

"**відмінно**" – «5» (високий рівень) – якщо студент вільно, глибоко й у повному обсязі засвоїв програмний матеріал; вичерпно, логічно викладає теоретичний матеріал в усній і письмовій формі; швидко і впевнено приймає правильні рішення при виконанні практичних завдань; має стійкі навички рішення різних задач;

"**добре**" – «4» (достатній рівень) – якщо студент вільно володіє матеріалом; у повному обсязі засвоїв програмний матеріал; здатний самостійно приймати

правильні рішення при виконанні практичних завдань; має добрі навички рішення практичних задач;

"**задовільно**" – «**3**» (середній рівень) якщо студент може самостійно виконати більшу частину навчального матеріалу, дати основні поняття та визначення; допускає неточності при прийнятті рішень; при вирішенні практичних завдань потребує додаткових вказівок.

"**незадовільно**" – «**2**» (початковий рівень) – якщо студент засвоїв і може відтворити лише окремі питання програми; допускає суттєві помилки при прийнятті рішень; не має стійких навичок при вирішенні практичних завдань.

ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема №1. ВВЕДЕННЯ В ТЕОРІЮ АЛГОРИТМІВ.

План лекційного заняття № 1. Введення в теорію алгоритмів. Базові алгоритмічні структури.

1. Мета та завдання дисципліни.
2. Значення та місце дисципліни в системі підготовки фахівців у галузі «Інформаційні технології».
3. Загальні відомості про дисципліну, її зв'язок з іншими дисциплінами.
4. Визначення поняття "Алгоритм", порівняння визначень.
5. Основні властивості алгоритма.
6. Способи запису алгоритмів.
7. Блок-схема алгоритма, основні елементи блок-схеми.
8. Алгоритми та програми.
9. Приклади типових алгоритмів.

Самостійна робота:

Історичний огляд теорії алгоритмів.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 7, 9.

Додатковий: 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №1.

Тема: Складання алгоритмів. Виконання алгоритмів.

Мета: сформувати практичні навички складання алгоритмів у різних формах; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

Тема №2. РЕКУРСИВНІ АЛГОРИТМИ.

План лекційного заняття № 2. Рекурсивні алгоритми.

1. Поняття про рекурсію. Основні принципи рекурсії та її важливість в обчисленнях.
2. Рекурсивні алгоритми. Приклади.
3. Основні компоненти рекурсивних функцій.
4. Базовий випадок та рекурсивний випадок.
5. Приклади простих рекурсивних функцій.
6. Переваги використання рекурсії в програмуванні.
7. Недоліки та обмеження рекурсії.
8. Вплив рекурсії на продуктивність програм.
9. Порівняння рекурсивних та ітеративних алгоритмів.
10. Визначення випадків, коли краще використовувати рекурсію та випадків, коли варто уникати її використання.
11. Порівняння ефективності рекурсивних та ітеративних реалізацій одного алгоритму.

Самостійна робота:

Складна рекурсія.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6, 7, 9.

Додатковий: 2, 4.

ЛАБОРАТОРНА РОБОТА №2. .

Тема: Рекурсивні алгоритми.

Мета: вміти розробляти алгоритми, які використовують рекурсію; обирати оптимальний тип для представлення вхідних та вихідних даних; реалізувати розроблений алгоритм на мові програмування; виконувати тестування програми; вміти пояснювати хід виконання алгоритму та принцип роботи програми.

ЛАБОРАТОРНА РОБОТА №3.

Тема: Рекурентні співвідношення.

Мета: вміти розробляти алгоритми, які використовують рекурентні співвідношення; вміти виділяти рекурентну формулу для піднесення до степеня та обчислення факторіалу; вміти реалізувати рекурсивну функцію, задану рекурентним співвідношенням; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

Тема №3. МАШИНА ТЮРІНГА.

План лекційного заняття № 3. Машина Тюрінга.

1. Машина Тюрінга.
2. Властивості машини Тюрінга як алгоритму
3. Формальний опис машини Тюрінга.
4. Функція переходів в машині Тюрінга.
5. Поняття про алгоритмічно нерозв'язні проблеми.
6. Проблема позиціонування в машині Тюрінга.
7. Проблеми, які не розв'язуються алгоритмічно.

ЛАБОРАТОРНА РОБОТА №4.

Тема: Машини Тюрінга.

Мета: реалізувати алгоритми для вирішення простих задач з використанням машини Тюрінга; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

План лекційного заняття № 4. Технології розробки програм для машини Тюрінга.

1. Переміщення автомата, заміна символів.

2. Аналіз символів.
3. Порівняння символів, стирання символів.
4. Видалення символу із слова.
5. Стиснення слова.
6. Вставка символу у слово.
7. Розсунення слова.
8. Фіксація місця на стрічці.

ЛАБОРАТОРНА РОБОТА №5.

Тема: Розробка програми для машини Тьюринга.

Мета: реалізувати алгоритми для вирішення простих задач з використанням машини Тьюринга; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

Тема №4. МАШИНА ПОСТА.

План лекційного заняття № 5. Формальний опис машини Поста.

1. Основні поняття та операції.
2. Склад машини Поста.
3. Команди машини Поста.
4. Прийоми розробки програми для машини Поста.
5. Арифметичні задачі
6. Орієнтація на стрічці
7. Дії над заданою на стрічці множиною міток.

Самостійна робота:

Алгоритмічна нерозв'язність проблеми Поста.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 5, 6, 10.

Додатковий: 1, 2.

ЛАБОРАТОРНА РОБОТА №6.

Тема: Машини Поста.

Мета: реалізувати алгоритми для вирішення простих задач з використанням машини Поста; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

ТЕМА №5. ФУНДАМЕНТАЛЬНІ АЛГОРИТМИ ТА ЇХ ПОБУДОВА

План лекційного заняття № 6-7. Алгоритми сортування

1. Значення сортувань при реалізації алгоритмів.
2. Класифікація сортувань. Характеристики сортувань.
3. Прості сортування як спосіб швидкої реалізації алгоритму.
4. Приклади простих сортувань – метод простого включення, метод простого обміну (бульбашкове сортування), сортування вставками, сортування підрахунком.

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

Самостійна робота:

Шейкерне сортування.

Цифрове сортування.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 5, 7, 8.

Додатковий: 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №7.

Тема: Програмування з використання алгоритмів сортування.

Мета: реалізувати алгоритми різних методів сортування масивів до розв’язування поставлених задач; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

План лекційного заняття № 8. Алгоритми пошуку.

1. Алгоритми пошуку в рядках: бінарний пошук, алгоритм Бойера – Мура, алгоритм Кнута – Морріса – Пратта, алгоритм Карпа – Рабіна, наближений пошук.
2. Прості алгоритми побудови дерева суфіксів.
3. Алгоритм Укконена. Масиви суфіксів.
4. Задача про найбільший спільний підрядок двох рядків.
5. Основні алгоритми обробки рядків – розбиття рядків, об’єднання рядків, алгоритми вставки, видалення, заміни.

Самостійна робота:

Алгоритм Укконена. Масиви суфіксів.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6, 8.

Додатковий: 1, 2,.

ЛАБОРАТОРНА РОБОТА №8.

Тема: Програмування з використання алгоритмів пошуку.

Мета: реалізувати алгоритми різних методів пошуку до розв’язування поставлених задач; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

ЛАБОРАТОРНА РОБОТА №9.

Тема: Алгоритми пошуку підрядків в рядках.

Мета: реалізувати алгоритми різних методів пошуку підрядків в рядках; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

Ознайомитись з різними методами пошуку підрядків в рядках.

ТЕМА №6. ЕФЕКТИВНІСТЬ АЛГОРИТМІВ.

План лекційного заняття № 9. Вступ до аналізу алгоритмів.

1. Порівняльні оцінки алгоритмів.
2. Система позначень в аналізі алгоритмів.
3. Класифікація алгоритмів по виду функції трудомісткості.
4. Асимптотичний аналіз функцій.

План лекційного заняття № 10. Теорії складності обчислень і класи складності задач.

1. Теоретична межа трудомісткості завдання.
2. Класи складності задач.
3. Проблема $P = NP$.
4. Клас NPC (NP – повні задачі).
5. Приклади NP – повних задач.
6. Задача про виконуваність схеми.
7. Задача про суму.
8. Задача про клік.

Самостійна робота:

Задача про незалежну множину.

Перелік рекомендованих джерел інформації

Основний: 3, 5, 6, 7, 8, 10.

Додатковий: 1, 2, 3.

ЛАБОРАТОРНА РОБОТА №10

Тема: Часова ефективність алгоритмів.

Мета: дослідити залежність часу виконання алгоритму сортування одновимірного масиву методом включень від кількості елементів масиву.

ТЕМА №7. АЛГОРИТМИ НА ГРАФАХ

План лекційного заняття № 11. Алгоритми на графах.

1. Основні поняття теорії графів. Предмет теорії графів.
2. Способи представлення графів.
3. Основні види графів.
4. Маршрути у графі.

5. Метричні характеристики графів.
6. Зв'язність. Компоненти зв'язності.
7. Класифікація ребер та вершин графів з точки зору зв'язності.

ЛАБОРАТОРНА РОБОТА №11.

Тема: Розв'язування задач за допомогою графів.

Мета: навчитися застосовувати основи графів до розв'язування задач за; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

ЛАБОРАТОРНА РОБОТА №12.

Тема: Способи подання графів.

Мета: навчитися подавати графи різними способами для розв'язування задач; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

План лекційного заняття № 12. Алгоритми пошуку остовних дерев.

1. Алгоритми пошуку остовного дерева в глибину.
2. Алгоритми пошуку остовного дерева в ширину.
3. Алгоритми пошуку максимальної кількості остовних дерев.

ЛАБОРАТОРНА РОБОТА №13.

Тема: Пошук у ширину та глибину. Алгоритми на графах.

Мета: навчитися виконувати алгоритми на графах; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

План лекційного заняття № 13. Алгоритми пошуку найкоротших шляхів.

1. Алгоритм Дейкстри.
2. Кроки алгоритму Дейкстри для пошуку найкоротшого шляху в орфографі.
3. Алгоритм Флойда для знаходження найкоротших шляхів між усіма парами вершин з визначенням їхніх маршрутів.
4. Кроки алгоритму Флойда для знаходження найкоротших шляхів між усіма парами вершин з визначенням між ними маршруту.

ЛАБОРАТОРНА РОБОТА №14.

Тема: Пошук найкоротшого шляху.

Мета: Навчитися реалізовувати алгоритм Дейкстри для пошуку найкоротшого

шляху від однієї вершини графа до інших; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

***План лекційного заняття № 14. Фундаментальні алгоритми на деревах.
Бінарні дерева пошуку.***

1. Обхід бінарного дерева.
2. Обхід в прямому порядку.
3. Обхід в зворотньому порядку.
4. Обхід в ширину.
5. Впорядковані дерева.
6. Додавання вершин. Пошук вершин. Вилучення вершин.

Самостійна робота:

Алгоритм перевірки на ізоморфність бінарних дерев.

Перелік рекомендованих джерел інформації

Основний: 1, 4, 5, 6,7, 8, 10.

Додатковий: 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №15.

Тема: Бінарні дерева.

Мета: Вивчити способи ефективного зберігання та обробки інформації на прикладі бінарних дерев.

***План лекційного заняття № 15. Алгоритми пошуку мінімальних
остовних дерев.***

1. Алгоритм Крускала. Кроки алгоритму Крускала. Реалізація алгоритму на прикладі графа.
2. Алгоритм Прима: звичайний та матричний. Реалізація алгоритму на прикладі графа.
3. Порівняння алгоритмів Прима та Крускала.
4. Опис та основні кроки алгоритму Крускала.
5. Реалізація алгоритму на прикладі графа.
6. Врахування вагових коефіцієнтів в ребрах графа.
7. Задачі з додатковими обмеженнями та умовами.
8. Реалізація алгоритмів Крускала, Прима програмному коді.
9. Вирішення реальних завдань зі створення та оптимізації мереж.

Самостійна робота:

Алгоритм Борувки. Кроки алгоритму Борувки. Реалізація алгоритму на прикладі графа. Переваги та обмеження алгоритму Борувки.

Перелік рекомендованих джерел інформації

Основний: 1, 5, 6, 8, 9.

Додатковий: 3, 4.

ЛАБОРАТОРНА РОБОТА №16.

Тема: Алгоритми знаходження остовного дерева мінімальної ваги.

Мета: Вивчення алгоритмів побудови остовного дерева мінімальної ваги і реалізація алгоритмів на мові високого рівня; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

ТЕМА№8. КРИПТОГРАФІЧНІ АЛГОРИТМИ.

План лекційного заняття № 16. Криптографічні алгоритми.

1. Класифікація криптографічних алгоритмів.
2. Криптосистеми із закритим ключем. Поняття закритого ключа.
3. Шифр Цезаря. Шифр Віжинера. Зламування шифрів Цезаря та Віжинера.
4. Криптосистеми з відкритим ключем. Поняття відкритого ключа.
5. Необхідність криптографічного шифрування.

Самостійна робота:

Китайська теорема про остачі. Функція Ейлера. Криптографічні хеш-функції. Поняття хеш-функції.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 5, 6, 7, 8, 10.

Додатковий: 1, 2, 3.

ЛАБОРАТОРНА РОБОТА №17.

Тема: Класичний шрифт простої заміни та його криптоаналіз. Біграмний шифр.

Мета: набути вміння із зашифрування та дешифрування повідомлень за допомогою шифру простої заміни, зокрема шифру Цезаря; використовуючи частотний криптоаналіз, навчитися зламувати шифротекст, зашифрований методом простої заміни; навчитися шифруванню біграмним шифром Плейфера.

ТЕМА№9. ЖАДІБНЕ ТА ДИНАМІЧНЕ ПРОГРАМУВАННЯ.

План лекційного заняття № 17. Динамічне програмування.

1. Поняття про динамічне програмування. Основні підходи до розв'язання задач методом динамічного програмування.
2. Переваги та недоліки використання динамічного програмування
3. Типові задачі, що вирішуються за допомогою динамічного програмування. Задачі на знаходження найкоротшого шляху. Задачі на пошук найбільшої підпоследовності. Задачі на рюкзак. Задачі на розбиття последовності.
4. Реалізація динамічного програмування в практиці. Порівняння з іншими методами розв'язання задач.

План лекційного заняття № 18. Жадібні алгоритми.

1. Поняття "жадібного" алгоритму. Теоретичні основи "жадібних" алгоритмів.
2. Основні принципи жадібних алгоритмів. Принцип оптимальності. Принцип жадібної вибірковості. Принцип локальної оптимізації.
3. Методи використання жадібних алгоритмів. Задачі на знаходження максимуму або мінімуму. Задачі на оптимізацію. Задачі на побудову оптимального розкладу.
4. Переваги та недоліки "жадібних" алгоритмів. Класичні приклади "жадібних" алгоритмів.
5. Типові задачі, що вирішуються за допомогою жадібних алгоритмів: задача про вкладання рюкзака, задачі на покриття інтервалів Розв'язання задач із застосуванням "жадібних" алгоритмів. Геометричні, транспортні, економічні задачі.

ЛАБОРАТОРНА РОБОТА №18.

Тема: Динамічне програмування. Жадібні алгоритми.

Мета: використання набутих знань алгоритму динамічного програмування для розв'язування задач; використання набутих знань для розв'язування задач жадібними алгоритмами; розвивати алгоритмічне мислення, інтерес до предмета; формувати вміння діяти за алгоритмом, застосовувати алгоритми в повсякденному житті.

ПЕРЕЛІК ПИТАНЬ ДЛЯ РУБІЖНОГО КОНТРОЛЮ

1. Визначення поняття "Алгоритм", порівняння визначень.
2. Основні властивості алгоритма.
3. Способи запису алгоритмів.
4. Блок-схема алгоритма, основні елементи блок-схеми.
5. Алгоритми та програми.
6. Приклади типових алгоритмів.
7. Поняття про рекурсію. Основні принципи рекурсії та її важливість в обчисленнях.
8. Рекурсивні алгоритми. Приклади.
9. Основні компоненти рекурсивних функцій.
10. Базовий випадок та рекурсивний випадок.
11. Приклади простих рекурсивних функцій.
12. Переваги використання рекурсії в програмуванні.
13. Недоліки та обмеження рекурсії.
14. Вплив рекурсії на продуктивність програм.
15. Порівняння рекурсивних та ітеративних алгоритмів.
16. Визначення випадків, коли краще використовувати рекурсію та випадків, коли варто уникати її використання.
17. Порівняння ефективності рекурсивних та ітеративних реалізацій одного алгоритму.
18. Основна теорема про рекурентних співвідношеннях.
19. Приклади розв'язання рекурентних співвідношень на основі теореми Дж. Бенглі, Д. Хакена та Дж. Сакса.
20. Машина Тюринга.
21. Властивості машини Тюринга як алгоритму
22. Формальний опис машини Тюринга.
23. Функція переходів в машині Тюринга.
24. Поняття про алгоритмічно нерозв'язні проблеми.
25. Проблема позиціонування в машині Тюринга.
26. Проблеми, які не розв'язуються алгоритмічно.
27. Переміщення автомата, заміна символів.
28. Аналіз символів.
29. Порівняння символів, стирання символів.
30. Видалення символу із слова.
31. Стиснення слова.
32. Вставка символу у слово.
33. Розсунення слова.
34. Фіксація місця на стрічці.
35. Склад машини Поста.
36. Команди машини Поста.
37. Прийоми розробки програми для машини Поста.
38. Арифметичні задачі для машини Поста. Орієнтація на стрічці.
39. Дії над заданою на стрічці множиною міток для машини Поста.

40. Класифікація сортувань. Характеристики сортувань.
41. Прості сортування як спосіб швидкої реалізації алгоритму.
42. Приклади простих сортувань – метод простого включення, метод простого обміну (бульбашкове сортування), шейкерне сортування, сортування вставками, сортування підрахунком, цифрове сортування.
43. Переваги і недоліки простих сортувань.
44. Складні сортування як спосіб створення ефективних алгоритмів. Приклади складних сортувань – сортування Шелла, сортування Хоара (швидке сортування), сортування злиттям.
45. Переваги і недоліки складних сортувань.
46. Порівняння простих та складних сортувань.
47. Алгоритми пошуку в рядках: бінарний пошук, алгоритм Бойера – Мура, алгоритм Кнута – Морріса – Пратта, алгоритм Карпа – Рабіна, наближений пошук.
48. Прості алгоритми побудови дерева суфіксів.
49. Алгоритм Укконена. Масиви суфіксів.
50. Задача про найбільший спільний підрядок двох рядків.
51. Основні алгоритми обробки рядків – розбиття рядків, об'єднання рядків, алгоритми вставки, видалення, заміни.

ПЕРЕЛІК ПИТАНЬ ДО ЕКЗАМЕНУ

52. Визначення поняття "Алгоритм", порівняння визначень.
53. Основні властивості алгоритма.
54. Способи запису алгоритмів.
55. Блок-схема алгоритма, основні елементи блок-схеми.
56. Алгоритми та програми.
57. Приклади типових алгоритмів.
58. Поняття про рекурсію. Основні принципи рекурсії та її важливість в обчисленнях.
59. Рекурсивні алгоритми. Приклади.
60. Основні компоненти рекурсивних функцій.
61. Базовий випадок та рекурсивний випадок.
62. Приклади простих рекурсивних функцій.
63. Переваги використання рекурсії в програмуванні.
64. Недоліки та обмеження рекурсії.
65. Вплив рекурсії на продуктивність програм.
66. Порівняння рекурсивних та ітеративних алгоритмів.
67. Визначення випадків, коли краще використовувати рекурсію та випадків, коли варто уникати її використання.
68. Порівняння ефективності рекурсивних та ітеративних реалізацій одного алгоритму.
69. Основна теорема про рекурентних співвідношеннях.
70. Приклади розв'язання рекурентних співвідношень на основі теореми Дж. Бентлі, Д. Хакена та Дж. Сакса.

71. Машина Тюринга.
72. Властивості машини Тюринга як алгоритму
73. Формальний опис машини Тюринга.
74. Функція переходів в машині Тюринга.
75. Поняття про алгоритмічно нерозв'язні проблеми.
76. Проблема позиціонування в машині Тюринга.
77. Проблеми, які не розв'язуються алгоритмічно.
78. Переміщення автомата, заміна символів.
79. Аналіз символів.
80. Порівняння символів, стирання символів.
81. Видалення символу із слова.
82. Стиснення слова.
83. Вставка символу у слово.
84. Розсунення слова.
85. Фіксація місця на стрічці.
86. Склад машини Поста.
87. Команди машини Поста.
88. Прийоми розробки програми для машини Поста.
89. Арифметичні задачі для машини Поста. Орієнтація на стрічці.
90. Дії над заданою на стрічці множиною міток для машини Поста.
91. Класифікація сортувань. Характеристики сортувань.
92. Прості сортування як спосіб швидкої реалізації алгоритму.
93. Приклади простих сортувань – метод простого включення, метод простого обміну (бульбашкове сортування), шейкерне сортування, сортування вставками, сортування підрахунком, цифрове сортування.
94. Переваги і недоліки простих сортувань.
95. Складні сортування як спосіб створення ефективних алгоритмів. Приклади складних сортувань – сортування Шелла, сортування Хоара (швидке сортування), сортування злиттям.
96. Переваги і недоліки складних сортувань.
97. Порівняння простих та складних сортувань.
98. Алгоритми пошуку в рядках: бінарний пошук, алгоритм Бойера – Мура, алгоритм Кнута – Морріса – Пратта, алгоритм Карпа – Рабіна, наближений пошук.
99. Прості алгоритми побудови дерева суфіксів.
100. Алгоритм Укконена. Масиви суфіксів.
101. Задача про найбільший спільний підрядок двох рядків.
102. Основні алгоритми обробки рядків – розбиття рядків, об'єднання рядків, алгоритми вставки, видалення, заміни.
103. Порівняльні оцінки алгоритмів.
104. Система позначень в аналізі алгоритмів.
105. Класифікація алгоритмів по виду функції трудомісткості.
106. Асимптотичний аналіз функцій.
107. Теоретична межа трудомісткості завдання.
108. Класи складності задач.

109. Проблема $P = NP$. Клас NPC (NP – повні задачі). Приклади NP – повних задач.
110. Задача про виконуваність схеми.
111. Задача про суму. Задача про клік.
112. Способи представлення графів.
113. Основні види графів.
114. Маршрути у графі.
115. Метричні характеристики графів.
116. Зв'язність. Компоненти зв'язності.
117. Класифікація ребер та вершин графів з точки зору зв'язності.
118. Алгоритми пошуку остовного дерева в глибину.
119. Алгоритми пошуку остовного дерева в ширину.
120. Алгоритми пошуку максимальної кількості остовних дерев.
121. Алгоритм Дейкстри.
122. Кроки алгоритму Дейкстри для пошуку найкоротшого шляху в орфографі.
123. Алгоритм Флойда для знаходження найкоротших шляхів між усіма парами вершин з визначенням їхніх маршрутів.
124. Кроки алгоритму Флойда для знаходження найкоротших шляхів між усіма парами вершин з визначенням між ними маршруту.
 - a. Обхід бінарного дерева.
 - b. Обхід в прямому порядку.
 - c. Обхід в зворотньому порядку.
 - d. Обхід в ширину.
 - e. Впорядковані дерева.
 - f. Додавання вершин. Пошук вершин. Вилучення вершин.
125. Алгоритм Крускала. Кроки алгоритму Крускала. Реалізація алгоритму на прикладі графа.
126. Алгоритм Борувки. Кроки алгоритму Борувки. Реалізація алгоритму на прикладі графа. Переваги та обмеження алгоритму Борувки.
127. Алгоритм Прима: звичайний та матричний. Реалізація алгоритму на прикладі графа.
128. Порівняння алгоритмів Прима та Крускала.
129. Опис та основні кроки алгоритму Крускала.
130. Реалізація алгоритму на прикладі графа.
131. Врахування вагових коефіцієнтів в ребрах графа.
132. Задачі з додатковими обмеженнями та умовами.
133. Реалізація алгоритмів Крускала, Прима та Борувки у програмному коді.
134. Вирішення реальних завдань зі створення та оптимізації мереж.
135. Класифікація криптографічних алгоритмів.
136. Криптосистеми із закритим ключем. Поняття закритого ключа.
137. Шифр Цезаря. Шифр Віжинера. Зламування шифрів Цезаря та Віжинера.
138. Криптосистеми з відкритим ключем. Поняття відкритого ключа.

139. Китайська теорема про остачі. Функція Ейлера. Криптографічні хеш-функції. Поняття хеш-функції. Необхідність криптографічного шифрування.
140. Поняття про динамічне програмування. Основні підходи до розв'язання задач методом динамічного програмування.
141. Переваги та недоліки використання динамічного програмування
142. Типові задачі, що вирішуються за допомогою динамічного програмування. Задачі на знаходження найкоротшого шляху. Задачі на пошук найбільшої підпоследовності. Задачі на рюкзак. Задачі на розбиття последовності.
143. Реалізація динамічного програмування в практиці. Порівняння з іншими методами розв'язання задач.
144. Поняття "жадібного" алгоритму. Теоретичні основи "жадібних" алгоритмів.
145. Основні принципи жадібних алгоритмів. Принцип оптимальності. Принцип жадібної вибірковості. Принцип локальної оптимізації.
146. Методи використання жадібних алгоритмів. Задачі на знаходження максимуму або мінімуму. Задачі на оптимізацію. Задачі на побудову оптимального розкладу.
147. Переваги та недоліки "жадібних" алгоритмів. Класичні приклади "жадібних" алгоритмів.
148. Типові задачі, що вирішуються за допомогою жадібних алгоритмів: задача про вкладання рюкзака, задачі на покриття інтервалів Розв'язання задач із застосуванням "жадібних" алгоритмів. Геометричні, транспортні, економічні задачі.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ

Основний:

1. Бородкіна І.Л., Бородкін Г.О. Теорія алгоритмів: Навчальний посібник. – Київ : Центр учбової літератури, 2018. - 184 с.
2. Караванова Т.П. Теорія алгоритмів. Частина 2. Обчислювальні алгоритми : навч. посіб. / Т.П. Караванова. Чернівці : Чернівець. нац. ун-т ім. Ю. Федьковича. 2022. 288 с.
3. Клакович Л.М. Теорія алгоритмів /Л.М. Клакович, С.М. Левицька, О. М. Костів. – Львів: Вид-во Львів ун-ту, 2008. –154с.
4. Клакович Л. М. Теорія алгоритмів: навч. посіб. / Л. М. Клакович, С.М.Левицька ; Львів. нац. ун-т ім. Івана Франка. - 2-ге вид., допов. - Львів : ЛНУ ім. Івана Франка, 2015. - 161 с.
5. 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, Кліфорд Стайн, —К. : К. І. С., 2019. — 1288 с
6. Матвієнко М. П. Теорія алгоритмів. Навчальний посібник. — К.: Видавництво Ліра-К, 2022. — 340 с.
7. Нікітченко М.С. Теорія алгоритмів: навч. посіб. / М. С. Нікітченко, О.С. Шкільняк, С.С. Шкільняк ; Київ. нац. ун-т ім. Тараса Шевченка. - Київ : Київський

університет, 2015. - 239 с.

8. Прийма С.М. Теорія алгоритмів: навчальний посібник. Мелітополь: ФОП Однорог Т. В., 2018. 116 с

9. Теорія алгоритмів і обчислювальних процесів.: навч. посіб. / І. В. Шостак [та ін.] ; Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харк. авіац. ін-т". - Х. : ХАІ, 2013. - 81 с.

10. Шкільняк С.С. Теорія алгоритмів. Приклади й задачі: навч. посіб. / С. С. Шкільняк ; Київ. нац. ун-т ім. Тараса Шевченка. - К. : Київський університет, 2012. - 151 с.

Додатковий:

1. Богданов В. Основи алгоритмізації та програмування. Посібник./ В. Богданов –К.: , 2010. – 136 с.

2. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 2 : навчальний посібник. Львів : Видавництво «Новий Світ-2000», 2020. 320 с.

3. Саволук А. П. Основи алгоритмізації та програмування. Збірник завдань /А. П. Саволук . –К.: «Основа», 2011. – 208 с.

4. Ткачук В. М. Алгоритми та структура даних : навчальний посібник. ІваноФранківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. 286 с.

Інформаційні ресурси Інтернет

1. Sorting Algorithm Animations [Electronic resource]. – Access mode : <http://www.sortingalgorithms.com/>.

2. <https://math.hws.edu/eck/js/turing-machine/TM.html>

3. <https://post.dmtry.me/>

4. <https://disted.edu.vn.ua/media/graph/>

5. <https://www.cryptool.org/en/cto/>