

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет
«Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт
з дисципліни
“Алгоритми та структури даних”
для студентів спеціальностей
121 “Інженерія програмного забезпечення” та
122 “Комп’ютерні науки”
(всіх форм навчання)

2023

Методичні вказівки до виконання лабораторних робіт з дисципліни “Алгоритми та структури даних” для студентів спеціальностей 121 “Інженерія програмного забезпечення” та 122 “Комп’ютерні науки” (всіх форм навчання) / В.М. Льовкін, О.О. Олійник. – Запоріжжя : НУ «Запорізька політехніка», 2023. – 49 с.

Автори: В.М. Льовкін, к.т.н., доцент
О.О. Олійник, к.т.н., доцент

Рецензент: С.О. Субботін, д.т.н., професор

Відповідальний
за випуск: С.О. Субботін, д.т.н., професор

Затверджено
на засіданні кафедри
програмних засобів

Протокол № 8
від “30” березня 2023 р.

ЗМІСТ

Вступ	5
1 Лабораторна робота № 1 Лінійні структури даних.....	6
1.1 Мета роботи	6
1.2 Основні теоретичні відомості.....	6
1.3 Завдання на лабораторну роботу.....	7
1.4 Зміст звіту.....	14
1.5 Контрольні запитання	14
2 Лабораторна робота № 2 Древа та геш-таблиці.....	16
2.1 Мета роботи	16
2.2 Основні теоретичні відомості.....	16
2.3 Завдання на лабораторну роботу.....	17
2.4 Зміст звіту.....	20
2.5 Контрольні запитання	20
3 Лабораторна робота № 3 Жадібні алгоритми	21
3.1 Мета роботи	21
3.2 Основні теоретичні відомості.....	21
3.3 Завдання на лабораторну роботу.....	22
3.4 Зміст звіту.....	24
3.5 Контрольні запитання	24
4 Лабораторна робота № 4 Динамічне програмування.....	26
4.1 Мета роботи	26
4.2 Основні теоретичні відомості.....	26
4.3 Завдання на лабораторну роботу.....	26
4.4 Зміст звіту.....	30
4.5 Контрольні запитання	31
5 Лабораторна робота № 5 Алгоритми для роботи з графами.	
Алгоритми обходу графів.....	32
5.1 Мета роботи	32
5.2 Основні теоретичні відомості.....	32
5.3 Завдання на лабораторну роботу.....	34
5.4 Зміст звіту.....	37
5.5 Контрольні запитання	37
6 Лабораторна робота № 6 Алгоритми для роботи з графами.	
Алгоритми пошуку найкоротших шляхів	39
6.1 Мета роботи	39
6.2 Основні теоретичні відомості.....	39

6.3 Завдання на лабораторну роботу.....	40
6.4 Зміст звіту.....	44
6.5 Контрольні запитання	44
7 Лабораторна робота № 7 Моделювання транспортної мережі за допомогою графів.....	45
7.1 Мета роботи	45
7.2 Основні теоретичні відомості.....	45
7.3 Завдання на лабораторну роботу.....	45
7.4 Зміст звіту.....	47
7.5 Контрольні запитання	47
Література.....	49

ВСТУП

Дане видання призначене для вивчення студентами всіх форм навчання теорії алгоритмів і структур даних та практичного засвоєння вміння використовувати дані алгоритми в процесі розв'язання практичних завдань шляхом розроблення програмного забезпечення.

Відповідно до графіка студенти перед виконанням лабораторної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою. Дані методичні вказівки містять тільки основні, базові теоретичні відомості, необхідні для виконання лабораторних робіт, тому для виконання лабораторної роботи та при підготовці до її захисту необхідно ознайомитись з конспектом лекцій та опрацювати весь необхідний матеріал, наведений в переліку рекомендованої літератури, використовуючи також статті в інтернет-виданнях та актуальних наукових журналах з комп'ютерних наук.

Для одержання заліку з кожної роботи студент повинен у відповідності зі всіма наведеними вимогами розробити програмне забезпечення та оформити звіт, після чого продемонструвати на комп'ютері розроблене програмне забезпечення з виконанням всіх запропонованих викладачем тестів.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші вміщують у канцелярський файл.

Усі завдання повинні виконуватись студентами індивідуально і не містити ознак плагіату як в оформленому звіті так і в розробленому програмному забезпеченні.

Під час співбесіди при захисті лабораторної роботи студент повинен виявити знання щодо мети роботи, теоретичного матеріалу, методів виконання кожного етапу роботи, змісту основних розділів звіту з демонстрацією результатів на конкретних прикладах, практичних прийомів використання теоретичного матеріалу в розробленому програмному забезпеченні. Студент повинен вміти обґрунтувати всі прийняті ним рішення та правильно аналізувати і використовувати на практиці отримані результати. Для базової самоперевірки при підготовці до виконання і захисту роботи студент повинен відповісти на контрольні запитання, наведені наприкінці опису відповідної роботи.

1 ЛАБОРАТОРНА РОБОТА № 1 ЛІНІЙНІ СТРУКТУРИ ДАНИХ

1.1 Мета роботи

1.1.1 Вивчити основні концепції побудови лінійних структур даних: зв'язних списків, стеків, куп та черг з пріоритетами.

1.1.2 Навчитися обирати та реалізовувати структури даних для сортування, вставки, видалення та пошуку елементів.

1.1.3 Навчитися реалізовувати та застосовувати алгоритм пірамідального сортування на практиці.

1.2 Основні теоретичні відомості

Пірамідальне сортування ґрунтується на використанні спеціалізованих структур даних, що дозволяють керувати інформацією в процесі виконання алгоритму.

Існує багато різних структур даних, які краще використовувати в тих чи інших випадках. Для одного з найпоширеніших випадків – сортування – зручно використовувати такі структури даних, як черги з пріоритетами та купи.

Черга з пріоритетами – структура даних, яка зберігає множину елементів, у якій кожен елемент v має певний пріоритет із значенням $key(v)$.

Двійкова купа (binary heap) або *піраміда* – легка для реалізації структура даних, що дозволяє швидко вставляти та вилучати елемент з максимальним пріоритетом (наприклад, максимальний за значенням).

Двійкова купа представляє собою майже повне бінарне дерево, для якого виконується основна властивість купи: пріоритет кожної вершини більше пріоритетів її нащадків. В найпростішому випадку пріоритет кожної вершини можна вважати рівним її значенню. В такому випадку структура називається *max-heap*, оскільки корінь піддерева є максимумом зі значень елементів піддерева.

Алгоритм пірамідального сортування ґрунтується на використанні структури даних піраміда та складається з двох етапів:

– побудова піраміди;

– видалення найбільших елементів.

При підготовці матеріалів даного розділу використано джерела [1]-[4], [6]-[8], [12].

1.3 Завдання на лабораторну роботу

1.3.1 Ознайомитися з літературою та основними теоретичними відомостями, необхідними для виконання роботи.

1.3.2 Розробити програмне забезпечення, що виконує базові операції з лінійними структурами даних.

1.3.2.1 Розроблюваний програмний проєкт має складатися з окремих класів, що реалізують структури даних двозв'язний список та купа (черга з пріоритетами). На найвищий рівень може бути передбачено графічну інтерфейсну взаємодію з користувачем для роботи зі створеними класами.

1.3.2.2 Клас, що реалізує двозв'язний список, має дозволяти виконувати наступні операції на основі окремих методів: додавання вузла в початок списку, додавання вузла після заданого, пошук вузла в списку, видалення вузла, виведення вузлів на екран з початку та з кінця.

1.3.2.3 Клас, що реалізує купу (чергу з пріоритетами), має дозволяти виконувати наступні операції на основі окремих методів: вставлення елемента, сортування елементів, побудова купи з неупорядкованого масиву, видалення елемента, сортування елементів із використанням купи, виведення елементів на екран.

1.3.3 Виконати тестування розробленого програмного забезпечення.

1.3.4 Розробити окремий модуль програмного забезпечення для реалізації пірамідального сортування на основі розробленого класу.

1.3.5 Розв'язати індивідуальне завдання за допомогою розробленої реалізації пірамідального сортування.

Вважати, що масиви даних зберігаються в файлах.

Варіант № 1

Інформація про речі, вилучені на митниці, включає:

- назву;
- людину, у якої було вилучено даний об'єкт;
- кількість одиниць;
- вартість одиниці;

– дату вилучення.

Відсортувати речі за загальною вартістю, у випадку рівної загальної вартості – за вартістю одиниці, інакше – за датою вилучення.

Варіант № 2

Фінансова діяльність підприємства представлена записами, що містять:

- назву підприємства;
- адресу;
- місяць;
- рік;
- прибуток за даний місяць.

Визначити перелік підприємств, які за вказаний рік входять у число 25 % найприбутковіших за даний рік. У випадку неоднозначності віддавати пріоритет підприємству, яке має більший максимальний прибуток протягом всіх місяців даного року.

Варіант № 3

Інформація про студентів групи включає наступну інформацію:

- прізвище, ім'я, по батькові;
- адреса;
- середній бал;
- стать;
- дата народження.

Вивести перелік місяців, відсортований за кількістю студентів групи, які народилися в даний місяць. У випадку однакової кількості за абеткою.

Варіант № 4

Інформація про підрозділи підприємства містить наступні дані:

- назва підрозділу;
- контактні дані;
- директор;
- кількість співробітників;
- сума, виплачена заробітна платня за місяць.

Вивести підрозділи підприємства в порядку зменшення середньої заробітної платні. У випадку неоднозначностей віддавати перевагу за кількістю співробітників.

Варіант № 5

Інформація про академічні заборгованості студентів зберігається в наступному вигляді:

- прізвище, ім'я, по батькові студента;
- факультет;
- дисципліна;
- дата іспиту або заліку;
- стан.

Вивести інформацію про факультети в порядку зменшення кількості заборгованостей студентів. У випадку рівної кількості порівнювати за кількістю студентів на факультеті, що мають заборгованості.

Варіант № 6

Відомості інтернет-провайдера про абонентів включають наступну інформацію:

- логін;
- прізвище, ім'я, по батькові;
- адреса;
- сума витрачених коштів;
- поточний тариф.

Вивести інформацію про всіх абонентів у алфавітному порядку за їх прізвищами та іменами, а у випадку збігу – за логіном.

Варіант № 7

Інформація про товари в магазинах містить наступні параметри:

- назва магазину;
- адреса;
- назва товару;
- кількість одиниць;
- вартість одиниці товару.

Вивести перелік всіх магазинів, у яких наявний заданий товар у кількості, не меншій за задану, та їхні адреси у порядку збільшення вартості одиниці товару.

Варіант № 8

Книжки в бібліотеці характеризуються наступними даними:

- автор;
- назва;
- жанр;
- видавництво;
- рік публікації;

- кількість сторінок;
- загальна кількість екземплярів;
- кількість екземплярів у читачів.

Визначити книжки, кількість наявних екземплярів яких у бібліотеці в поточний момент входить у перші 50 %. Обчислити сумарну кількість наявних екземплярів таких книжок.

Варіант № 9

Дані про товари, продані в магазині, містять наступні параметри:

- назва товару;
- дата і час продажу;
- кількість одиниць товару;
- вартість одиниці товару;
- дисконт;
- продавець.

Вивести інформацію про продавців у відповідності з доходом, який отримав магазин у результаті їхньої роботи. У випадку рівного доходу – за датою останнього проданого товару.

Варіант № 10

Щоденник погоди містить наступну інформацію:

- дата;
- температура;
- напрям вітру;
- сила вітру;
- атмосферний тиск;
- наявність опадів.

Вивести місяці заданого користувачем року в порядку зменшення кількості днів, у які траплялися опади.

Варіант № 11

Записи в блозі характеризуються наступними даними:

- порядковий номер;
- назва;
- текст;
- теги;
- дата публікації;
- кількість відвідувачів.

Вивести інформацію про записи (назва, теги) в порядку зменшення зацікавленості з боку користувачів. За однакового значення виводити раніше записи, які було опубліковано пізніше.

Варіант № 12

Результати іспитів учнів містять наступну інформацію:

- прізвище, ім'я, по батькові;
- школа;
- вік;
- предмет, з якого складався іспит;
- оцінка за іспит.

Вивести перелік шкіл, учні яких отримали за іспит за заданим користувачем предметом бали, що входять у 15 % найкращих.

Варіант № 13

Інформація про потяги містить наступні дані:

- номер потяга;
- початкова станція;
- кінцева станція;
- пункт призначення;
- час від'їзду;
- час прибуття.

Вивести інформацію про потяги у алфавітному порядку за початковою станцією. Для потягів з однаковою початковою станцією – за часом від'їзду.

Варіант № 14

Інформація про клієнтів деякої мережі магазинів містить наступні дані:

- прізвище, ім'я;
- телефон;
- e-mail;
- витрачена сума;
- кількість придбаних речей.

Вивести інформацію про клієнтів, які входять у перші 30 % таких, що купували речі в магазинах даної мережі на найбільшу суму. У випадку неоднозначності віддавати перевагу клієнтам з більшою кількістю придбаних речей.

Варіант № 15

Провайдер бажає розрахувати суму коштів, яку йому потенційно вдасться заробити до кінця року. Дані про абонентів включають наступні параметри:

- логін;
- прізвище, ім'я, по батькові;
- дата підключення;
- сплачені суми за кожен з місяців року;
- поточний тариф.

Для кожного тарифу відома сума, яку за нього сплачує абонент за місяць. Вивести інформацію про кожного абонента в порядку зменшення суми, яку він сплатив і потенційно сплатить за своїм поточним тарифом до кінця року. У разі неоднозначності виводити дані за алфавітом. Обчислити загальну суму доходу провайдера.

Варіант № 16

На підприємстві дані про кожного співробітника в одному з відділів включають наступні параметри:

- прізвище, ім'я, по батькові;
- посада;
- стаж;
- домашня адреса;
- дата народження.

Вивести перелік співробітників у порядку зменшення їх віку, для кожного вказуючи кількість повних років. У випадку однакової дати народження виводити спочатку співробітника з меншим стажем.

Варіант № 17

Факультет одного з університетів зберігає інформацію про заборгованості студентів у наступному вигляді:

- прізвище, ім'я, по батькові студента;
- спеціальність;
- дисципліна;
- дата іспиту або заліку;
- кількість балів.

Вивести перелік студентів, які мають більше 2 заборгованостей, у порядку зменшення їх кількості заборгованостей. У випадку рівної кількості порівнювати за середнім балом, отриманим студентом за даними заборгованостями у відомостях.

Варіант № 18

Інформація про деякий товар на складах включає наступні дані:

- назва складу;
- адреса;
- відстань до складу;
- вартість постачання;
- вартість одиниці;
- кількість одиниць.

Вивести інформацію про наявність товарів на складах у порядку збільшення вартості одиниці даного товару для замовника (тобто з урахуванням вартості постачання). У випадку неоднозначності використовувати дані про кількість одиниць товару на складі.

Варіант № 19

Інформація про кожне підприємство визначена наступними даними:

- ідентифікаційний код суб'єкта господарської діяльності;
- назва підприємства;
- адреса;
- розмір доходів за рік;
- розмір витрат за рік.

Вивести перелік підприємств за зменшенням сплаченого податку на прибуток. У випадку неоднозначності – за отриманими доходами. Податок на прибуток розраховується в розмірі 18 % від отриманого підприємством прибутку.

Варіант № 20

У відділі кадрів міститься інформація про захворювання співробітників, що включає:

- прізвище, ім'я, по батькові співробітника;
- відділ;
- посаду;
- вік;
- дату початку лікарняного;
- дату завершення лікарняного;
- хвороба.

Вивести інформацію про всі хвороби, якими хворіли співробітники, за зменшенням кількості випадків.

1.3.6 Порівняти одержані результати виконаних тестів, провести аналіз вірності, коректності та адекватності роботи розробленого програмного забезпечення.

1.3.7 Виконати порівняння пірамідального сортування з іншими відомими студенту алгоритмами сортування.

1.3.8 Оформити звіт з роботи.

1.3.9 Відповісти на контрольні запитання.

1.4 Зміст звіту

1.4.1 Мета роботи.

1.4.2 Завдання до роботи.

1.4.3 Текст розробленого програмного забезпечення з коментарями.

1.4.4 Результати роботи програмного забезпечення: таблиця тестових запусків, копії екранних форм.

1.4.5 Таблиця порівняння результатів застосування алгоритмів сортування.

1.4.6 Висновки, що відображають особисто отримані результати виконання роботи, їх критичний аналіз.

1.5 Контрольні запитання

1.5.1 Що представляє собою піраміда і яким властивостям така структура даних повинна задовольняти?

1.5.2 У яких випадках краще використовувати структуру даних типу купа, а у яких – чергу з пріоритетами?

1.5.3 Яким чином можна побудувати піраміду?

1.5.4 У чому полягає принцип додавання елементу у чергу з пріоритетами?

1.5.5 У чому полягає принцип сортування списку елементів із використанням купи?

1.5.6 Які способи реалізації черги з пріоритетами?

1.5.7 Які операції підтримуються в незростальній, а які в неспадній черзі з пріоритетами?

1.5.8 Які піраміди називаються незростальними, а які неспадними?

1.5.9 Чи є масив з відсортованими елементами неспадною пірамідою?

1.5.10 У чому полягає алгоритм пірамідального сортування?

1.5.11 Які операції можна виконувати з однозв'язним переліком?

1.5.12 Які операції можна виконувати з двозв'язним переліком?

1.5.13 Які операції можна виконувати зі стеком?

1.5.14 Що таке дек?

1.5.15 Який принцип роботи стеку?

2 ЛАБОРАТОРНА РОБОТА № 2 ДЕРЕВА ТА ГЕШ-ТАБЛИЦІ

2.1 Мета роботи

2.1.1 Засвоїти основні концепції геш-таблиць та бінарних дерев пошуку і В-дерев зокрема.

2.1.2 Навчитися використовувати геш-таблиці та В-дерева на практиці.

2.2 Основні теоретичні відомості

Гешування ґрунтується на ідеї розподілу ключів у одновірному масиві H $[0..m-1]$, який називається геш-таблицею. Розподіл виконується шляхом обчислення для кожного ключа значення деякої наперед визначеної функції h , яка називається геш-функцією.

Геш-функція призначає кожному з ключів геш-адресу, яка представляє собою ціле число від 0 до $m-1$. У загальному випадку геш-функція має задовольняти двом суперечливим вимогам:

- геш-функція має розподіляти ключі за комірками геш-таблиці як можна більш рівномірно;

- геш-функція повинна легко обчислюватися.

Методи побудови геш-функцій:

- метод ділення;

- метод множення;

- універсальне гешування.

Колізія – ситуація, коли два або більше ключів гешуються в одну й ту саму комірку геш-таблиці. Будь-яка схема гешування повинна мати механізм розв'язання колізій. Існує два основні підходи до оброблення колізій: з застосуванням ланцюжків і з застосуванням адресації.

Бінарне дерево – скінченна множина вузлів, яка може бути або порожньою, або складатися з кореня та двох бінарних дерев, які не перетинаються та називаються лівим та правим піддеревами кореня.

Існують три алгоритми обходу бінарних дерев:

- у прямому порядку;

- симетричний;

– центрований.

В-дерева є узагальненням бінарних дерев пошуку. Вони представляють собою збалансовані дерева пошуку, призначені для ефективної роботи з дисковою пам'яттю. Вузли В-дерев можуть мати багато дочірніх вузлів.

У В-деревих всі записи даних або ключі зберігаються в листі у зростаючому порядку ключів, а батьківські вузли використовують для індексування.

У процесі пошуку в В-дереві, починаючи з кореня слідують ланцюжком покажчиків до листа, який може містити шуканий ключ. Після цього пошук відбувається серед ключів даного листа.

При підготовці матеріалів даного розділу використано джерела [1], [3]-[8], [11], [12].

2.3 Завдання на лабораторну роботу

2.3.1 Ознайомитися з літературою та основними теоретичними відомостями, необхідними для виконання роботи.

2.3.2 Розробити програмне забезпечення, що виконує базові операції з геш-таблицями та В-деревими.

2.3.2.1 Розроблюваний програмний проєкт має складатися з окремих класів, що реалізують структури даних геш-таблиця та бінарне дерево пошуку, а також має містити окремий модуль, що забезпечує інтерфейсну взаємодію з користувачем для роботи зі створеними класами.

2.3.2.2 Клас, що реалізує геш-таблицю, має дозволяти виконувати наступні операції на основі окремих методів: вставлення елементу, видалення елементу, пошук елементу, відображення структури геш-таблиці на основі використання параметрів, обраних у відповідності з варіантом індивідуального завдання з п. 2.3.4.

2.3.2.3 Клас, що реалізує В-дерево, має дозволяти виконувати наступні операції на основі окремих методів: створення порожнього дерева, відображення структури дерева, пошук у дереві, вставлення ключа, видалення ключа.

2.3.3 Виконати тестування розробленого програмного забезпечення.

2.3.4 Розв'язати індивідуальне завдання з таблиці 2.1, що складається з 2 задач, наведених нижче, за допомогою розроблених

модулів програмного забезпечення.

Таблиця 2.1 – Індивідуальні завдання

№ варіанта	Задача 1	Задача 2	№ варіанта	Задача 1	Задача 2
1	А	А	11	В	Г
2	Б	Б	12	Б	А
3	В	В	13	Д	В
4	Г	Г	14	Г	А
5	Д	Д	15	А	Б
6	А	Д	16	Б	Д
7	Б	Г	17	А	В
8	В	Б	18	Д	Г
9	Г	В	19	Г	Б
10	Д	А	20	В	Д

Задача 1:

А. Створити геш-таблицю, що використовує метод ланцюжків для розв'язання колізій та геш-функцію ділення з остачею. Геш-таблицю заповнити на основі виділення слів з текстового файлу, заповненого текстом художнього твору, обчислюючи кількість входжень кожного слова. Визначити задану кількість слів, що зустрічаються найчастіше.

Б. Створити геш-таблицю, що використовує метод відкритої адресації для розв'язання колізій, геш-функцію ділення з остачею та метод лінійного дослідження для обчислення послідовностей досліджень. Геш-таблицю заповнити на основі виділення інформації з текстового файлу, в якому містяться підказки для кожного зарезервованого слова. Виконати пошук інформації.

В. Створити геш-таблицю, що використовує метод ланцюжків для розв'язання колізій та геш-функцію множення. Геш-таблицю заповнити на основі виділення інформації з текстового файлу, в якому містяться прізвища, ім'я і по батькові співробітників фірми та займані ними посади. Визначити посаду заданого співробітника.

Г. Створити геш-таблицю, що використовує метод відкритої адресації для розв'язання колізій, геш-функцію множення та метод квадратичного дослідження для обчислення послідовностей

досліджень. Геш-таблицю заповнити на основі виділення інформації з текстового файлу, в якому містяться ідентифікаційні номери платників податків та їх прізвища, імена і по батькові. Визначити платника податку за заданим ідентифікаційним номером.

Д. Створити геш-таблицю, що використовує метод відкритої адресації для розв'язання колізій, геш-функцію ділення з остачею та метод подвійного гешування. Геш-таблицю заповнити на основі виділення інформації з текстового файлу, де містяться електронні адреси та відповідні їм назви підприємств. Визначити, якому підприємству належить задана електронна адреса.

Задача 2:

А. У процесі планування роботи на заводі виникло завдання проаналізувати ефективність використання різних засобів. Дані про верстати складаються з інвентарного номеру, найменування, типу, часу перестою та часу роботи за місяць. Сформувати дерево з даних про верстати на даному заводі. Забезпечити пошук інформації про верстат за інвентарним номером з визначенням відсотку перестою за останній місяць (відносно загального робочого часу за місяць), а також визначення типу верстатів, який використовується найбільше.

Б. Дані про власників автомобілів включають ідентифікаційний номер транспортного засобу, дату реєстрації та власника (прізвище, ім'я, по батькові). Сформувати дерево з інформації про власників автомобілів. Реалізувати пошук інформації про автомобіль за заданим ідентифікаційним номером транспортного засобу, визначення осіб, які володіють більше ніж одним автомобілем.

В. Інформація про студентів представлена номером залікової книжки, прізвищем, ініціалами, групою, результатами навчання. Результати навчання представлені середнім балом. Сформувати дерево з інформації про студентів. Забезпечити визначення студентів, що навчаються найкраще та найгірше, а також визначення прізвища та ініціалів студента за номером залікової книжки.

Г. Щоденник погоди містить інформацію, що складається з дати, температури, вологості повітря, напрямку вітру, атмосферних явищ, атмосферного тиску, рівня опадів. На основі даних щоденника погоди сформувати дерево. Визначити інформацію про погоду за вказану користувачем дату. Визначити день заданого місяця, коли рівень опадів був найвищий.

Д. Мобільний оператор повинен мати інформацію про абонентів

для забезпечення послуг. Кожний абонент характеризується номером, прізвищем, ім'ям, по батькові, тарифним планом. Сформувати дерево з відповідної інформації про абонентів, забезпечити пошук інформації про абонента за його телефонним номером та визначення кількості підключень за кожним з тарифів.

2.3.5 Порівняти одержані результати виконаних тестів, провести аналіз вірності, коректності та адекватності роботи розробленого програмного забезпечення.

2.3.6 Оформити звіт з роботи.

2.3.7 Відповісти на контрольні запитання.

2.4 Зміст звіту

2.4.1 Мета роботи.

2.4.2 Завдання до роботи.

2.4.3 Текст розробленого програмного забезпечення з коментарями.

2.4.4 Результати роботи програмного забезпечення, що включають результати тестування та копії екранних форм.

2.4.5 Висновки, що відображають особисто отримані результати виконання роботи, їх критичний аналіз.

2.5 Контрольні запитання

2.5.1 Що розуміють під гешуванням?

2.5.2 За яких умов слід використовувати геш-таблиці?

2.5.3 Що таке геш-функція та які висуваються вимоги до геш-функцій?

2.5.4 Які існують методи побудови геш-функцій?

2.5.5 Що таке колізія та які існують підходи для її оброблення?

2.5.6 Що таке бінарне дерево?

2.5.7 Які існують алгоритми обходу бінарних дерев і в чому вони полягають?

2.5.8 Які операції можна виконувати над бінарними деревами пошуку?

2.5.9 Які дерева називають В-деревами? Якими властивостями вони характеризуються?

2.5.10 Які основні операції можна виконати над В-деревами?

3 ЛАБОРАТОРНА РОБОТА № 3

ЖАДІБНІ АЛГОРИТМИ

3.1 Мета роботи

3.1.1 Вивчити основні принципи та особливості жадібних алгоритмів.

3.1.2 Навчитися використовувати жадібні алгоритми для розв'язання практичних завдань та обґрунтовувати прийняті рішення.

3.2 Основні теоретичні відомості

Жадібний алгоритм – метод розв'язання оптимізаційних задач, який ґрунтується на тому, що процес прийняття рішень можна розбити на елементарні кроки, на кожному з яких приймається окреме рішення. Рішення, що приймається на кожному кроці, повинно бути оптимальним тільки на поточному кроці і повинно прийматися без урахування попередніх або наступних рішень. Тобто на кожному кроці обирається найкращий варіант, вважаючи при цьому, що підсумкове рішення буде оптимальним.

Існують задачі, для яких послідовність таких жадібних виборів призведе до оптимального рішення для будь-якого екземпляра задачі, що розглядається. Однак для інших задач дане твердження не виконується: для розв'язання таких задач жадібний алгоритм може використовуватися у випадку, якщо прийнятним є приблизне рішення.

Коди Хаффмана широко розповсюджені для стиснення даних. Це доволі ефективний алгоритм, який дозволяє зазвичай зекономити від 20 до 90 % даних.

Алгоритм Хаффмана належить до жадібних алгоритмів і дозволяє закодувати текст, який побудований на основі n -символьного алфавіту. Кодування ґрунтується на принципі, за яким більш короткі коди призначаються символам, що зустрічаються частіше, а більш довгі – символам, що зустрічаються рідше. Тобто коди символів мають змінну довжину. Такий підхід не дозволяє визначити, скільки бітів кодованого тексту представляють кожний символ. Тому в кодах Хаффмана для розв'язання даної проблеми використовуються префіксні коди.

При підготовці матеріалів даного розділу використано джерела [1]-[4], [9], [11], [12].

3.3 Завдання на лабораторну роботу

3.3.1 Ознайомитися з літературою та основними теоретичними відомостями за темою роботи.

3.3.2 Розробити програмне забезпечення, що розв'язує задачу у відповідності з індивідуальним завданням з пункту 3.3.2.3 із використанням жадібного алгоритму.

3.3.2.1 Розроблюваний програмний проєкт має складатися з класу, що описує задачу, сформульовану в індивідуальному завданні, а також має містити окремий модуль, що забезпечує інтерфейсну взаємодію з користувачем.

3.3.2.2 Клас вхідних даних задачі має дозволяти:

- задавати початкові дані;
- вводити нові параметри;
- коригувати та видаляти існуючі;
- розв'язувати задачу з використанням жадібного алгоритму.

3.3.2.3 Індивідуальне завдання обрати у відповідності з номером варіанта на основі наступної формули: $((N - 1) \% 5) + 1$, де N – номер варіанта, що використовувався в попередніх лабораторних роботах.

Варіант № 1.

Інвестор оголосив конкурс на фінансування проєктів протягом заданого періоду часу. Після подачі заявок і попереднього відбору було сформовано відповідний пул заявок, кожному з яких визначено датою початку, датою завершення та прибутком, який даний проєкт принесе після виконання. Оцінивши надані заявки та власні ресурси, інвестор зрозумів, що зможе фінансувати не більше ніж один проєкт одночасно. Під час конкурсного відбору інвестор поставив задачу вибрати для виконання набір проєктів, які принесуть йому максимальну вигоду.

Варіант № 2.

Власник крамниці, справи якого йдуть не дуже добре, вирішив зайнятися іншою справою і домовився з власником крамниці з віддаленого містечка про те, щоб достатньо вдало перепродати йому товари, що залишились в магазині. Маючи обмежені кошти, він вирішив найняти тільки один фургон для перевезення товарів. Фургон

зможе завантажитись, відвести продукти, розвантажитись і повернутись назад за повний робочий день, перевозячи при цьому тільки один набір товарів. Для кожного набору товарів обумовлено, за яку кількість діб до завершення строку придатності його може бути придбано за домовленою ціною, в іншому випадку ціна має бути зменшена на 50 %. Визначте стратегію перевезення товарів, яка дозволить мінімізувати втрати власника крамниці.

Варіант № 3.

На дачі стоїть велика діжка, яка вміщує задану кількість рідини. Хазяїн використовує її для поливу рослин, але не маючи централізованого водопостачання, має принести воду з річки. У його розпорядженні є відра заданого обсягу. Визначити, яку мінімальну кількість відер води хазяїну потрібно принести з річки, щоб заповнити діжку. Вважати, що кожне відро може бути принесене тільки повністю заповненим, адже хазяїн не хоче носити зайвий вантаж. Результати виводити, демонструючи кількість відер кожного обсягу.

Варіант № 4.

Капітан корабля, що зупинився на закинутому острові, вирішив придбати у туземців спеції та заповнити ними всі вільні трюми, щоб заробити якомога більше грошей. Однак всі спеції вмістити у трюми було неможливо, а туземці всі їх продавали за одну й ту саму ціну. Знайшовши у одного з матросів записи про те, скільки коштує кожний вид спецій на континенті, приблизно оцінивши наявну кількість кожного виду спецій та знаючи, яку кількість вантажу вмістять вільні трюми, капітан пішов на переговори з туземцями. Визначте оптимальне рішення для капітана.

Варіант № 5.

Продавець у маленькій крамниці має звичку тримати під рукою найпопулярніші товари для того, щоб пришвидшити роботу з чергою покупців. Кількість таких товарів визначається додатково. Кожен покупець замовляє по черзі товари, а продавець перевіряє, чи є такий товар під рукою. Якщо товару немає, то він дістає його з полиць, що займає більше часу. При цьому для кожного такого товару продавець має вирішити, чи не треба його помістити до найпопулярніших замість якогось з існуючих. Допоможіть продавцю приймати дане рішення протягом просування всієї черги замовлень покупців, якщо він прагне зменшити час, за який вся черга залишить крамницю.

Припустіть, що продавець знає наперед про те, які товари бажає купити кожен з покупців у черзі.

3.3.3 Довести, що жадібний вибір для розглядаємої задачі є оптимальним рішенням або принаймні є частиною деякого оптимального рішення.

3.3.4 Розробити програмне забезпечення, яке реалізує використання алгоритму Хаффмана для стискання даних текстового файлу у вигляді класу. Клас повинен мати методи, які дозволяють задати файл з даними, виконати стискання даних, визначити параметри виконаного стискання та зворотнє перетворення, записати результати кодування/декодування в файл.

3.3.5 Виконати тестування розробленого у п. 3.3.2 та 3.3.4 програмного забезпечення.

3.3.6 Порівняти одержані результати виконаних тестів, провести аналіз вірності, коректності та адекватності роботи розробленого програмного забезпечення.

3.3.7 Оформити звіт з роботи.

3.3.8 Відповісти на контрольні запитання.

3.4 Зміст звіту

3.4.1 Мета роботи.

3.4.2 Завдання до роботи.

3.4.3 Опис алгоритмів, за допомогою яких розв'язано індивідуальне завдання.

3.4.4 Обґрунтування можливості використання жадібних алгоритмів для розглядаємого індивідуального завдання.

3.4.5 Текст розробленого програмного забезпечення з коментарями.

3.4.6 Результати роботи програмного забезпечення, що включають результати тестування та копії екранних форм.

3.4.7 Висновки, що відображають особисто отримані результати виконання роботи та їх критичний аналіз.

3.5 Контрольні запитання

3.5.1 У чому полягає жадібний вибір?

3.5.2 В яких випадках можна використовувати жадібні алгоритми?

3.5.3 Що таке матроїд?

3.5.4 Наведіть приклади задач, які можна розв'язувати з використанням жадібних алгоритмів.

3.5.5 Які задачі непридатні для розв'язання за допомогою жадібних алгоритмів?

3.5.6 Які алгоритмічні методи є жадібними алгоритмами?

3.5.7 У чому полягає алгоритм Хаффмана?

3.5.8 Чи можна вважати алгоритм Хаффмана жадібним і чому?

3.5.9 Що таке префіксні коди?

3.5.10 Яким чином використовуються дерева під час створення кодів Хаффмана?

4 ЛАБОРАТОРНА РОБОТА № 4 ДИНАМІЧНЕ ПРОГРАМУВАННЯ

4.1 Мета роботи

4.1.1 Вивчити основні принципи динамічного програмування.

4.1.2 Навчитися використовувати динамічне програмування для розв'язання практичних завдань.

4.2 Основні теоретичні відомості

За своєю суттю *динамічне програмування* є методом проектування алгоритмів, які дозволяють розв'язувати задачі з підзадачами, які перекриваються. Такі підзадачі зазвичай виникають з рекурентних співвідношень, які зв'язують рішення даної задачі з рішеннями менших підзадач того ж виду.

Замість того, щоб розв'язувати підзадачі, що перекриваються, знову і знову, динамічне програмування дозволяє розв'язати кожну з менших підзадач тільки один раз, записуючи при цьому результат розв'язання в таблицю, з якої потім можна отримати розв'язання початкової задачі.

При підготовці матеріалів даного підрозділу використано джерела [1], [3], [4], [9], [11], [12].

4.3 Завдання на лабораторну роботу

4.3.1 Ознайомитися з літературою та основними теоретичними відомостями за темою роботи.

4.3.2 Розробити програмне забезпечення, що розв'язує задачу у відповідності з індивідуальним завданням з пункту 4.3.2.3 із використанням принципів динамічного програмування.

4.3.2.1 Розроблюваний програмний проєкт має складатися з класу, що описує задачу, сформульовану в індивідуальному завданні, а також має містити окремий модуль, що забезпечує інтерфейсну взаємодію з користувачем.

4.3.2.2 Клас вхідних даних задачі має дозволяти:

– задавати початкові дані;

- вводити нові параметри;
- коригувати та видаляти існуючі;
- розв’язувати відповідну задачу.

4.3.2.3 Індивідуальне завдання, що складається з 2 задач, наведених нижче, обрати з таблиці 4.1 у відповідності з варіантом.

Таблиця 4.1 – Індивідуальні завдання

№ ва- ріанта	Задача 1	Задача 2	№ ва- ріанта	Задача 1	Задача 2
1	А	Д	11	Г	Д
2	Б	В	12	А	В
3	В	Г	13	Д	Б
4	Г	Б	14	Б	Г
5	Д	А	15	В	А
6	Д	Г	16	А	А
7	Г	В	17	Б	Б
8	В	Д	18	В	В
9	Б	А	19	Г	Г
10	А	Б	20	Д	Д

Задача 1:

А. Для заданого рядка символів визначити послідовність дій з розбиття рядка на задані частини, що характеризується мінімальними вимогами до часу виконання дій. Частини, на які необхідно розбити рядок, задаються порядковими номерами символів, які є межами частин рядка. Тобто якщо рядок довжиною 10 символів необхідно розділити на частини після символів 3, 5, то в результаті має бути сформовано три нові рядка: з першого до третього символу, з четвертого до п'ятого та з шостого до десятого. Дані розбиття можна виконати в порядку зліва направо або справа наліво (у випадку більшої кількості частин таких варіацій стає ще більше). Розбиття зліва направо має наступну вартість: 10 (розбиває рядок на 3 та 7 символів) + 7 (розбиває рядок на 2 та 5 символів). Таким чином, в результаті необхідно визначити послідовність розбиття рядка, що характеризується мінімальною загальною вартістю за часом, та вартість такого розбиття.

Б. Гра реалізується на прямокутному полі $N_1 \times N_2$, де N_1 – кількість клітинок за вертикаллю, а N_2 – кількість клітинок за горизонталлю. Клітинки визначаються координатами, перша з яких – номер за вертикаллю (зліва направо), друга – номер за горизонталлю (знизу вгору). Гра полягає в тому, щоб дістатися з однієї заданої клітинки з координатами (x_1, y_1) в іншу клітинку з координатами (x_2, y_2) . При цьому $x_1 < x_2$, $y_1 > y_2$. За умовами гри кожна клітинка має деяку вартість, тобто суму грошей, які доведеться сплатити за потрапляння на неї. Гравець може рухатися на одну клітинку вниз, вправо або за діагоналлю вправо вниз (тобто за один крок на одну клітинку вправо і вниз). Визначити ходи гравця з (x_1, y_1) у (x_2, y_2) , які мінімізують витрати, та значення витрат для даного набору ходів.

В. Послідовність складається з деякого набору цілих чисел. Елементи послідовності можуть бути, як від'ємними, так і невід'ємними цілими числами. Визначити найдовшу спадну підпослідовність даної послідовності. Вивести на екран довжину такої послідовності та всі її члени.

Г. Рядок складається з символів української абетки. У загальному випадку такий рядок представляє собою слово українською мовою. Визначити довжину найдовшого паліндрома (непорожній рядок, який однаково читається як зліва направо, так і справа наліво), який можна утворити шляхом вилучення деяких літер даного рядка, та вивести безпосередньо сам паліндром.

Д. Під час експериментального тренування роботів їх було посаджено на велосипеди. Уся група роботів відправляється на велосипедах вузькою велосипедною доріжкою в однаковому напрямку з деякими проміжками. Кожен робот керує велосипедом на деякій заданій швидкості, яка змінюється тільки у тому випадку, якщо робот наздожене велосипедиста з меншою швидкістю: не маючи змоги обігнати більш повільного велосипедиста, він знизить швидкість до швидкості велосипедиста, який їде попереду. Таким чином, через деякий час роботи будуть розбиті на групи, кожна з яких буде рухатись зі своєю сталою швидкістю. Завдання програміста полягає в тому, щоб розбити роботів на деяку задану кількість груп. Визначити, скільки існує способів старту роботів (тобто порядку, в якому кожен робот почне рух велосипедною доріжкою), які в результаті сформулюють задану кількість груп.

Задача 2:

А. У процесі формування тексту в текстовому редакторі виникає завдання переносу слів за рядками. Для розв'язання даного завдання задається текст, який складається з деякого набору слів, та кількість символів, яка може міститися в одному рядку. Визначити, яким чином розташувати слова за рядками тексту (тобто визначити слова, які будуть розташовані в першому рядку, другому тощо), залишаючи між ними по одному пробілу, таким чином, щоб мінімізувати кількість пробілів, яка при цьому сумарно залишається в кінці рядків (в якості критерію використати суму кубів кількості пробілів у кінці кожного рядка окрім останнього).

Б. Компанія здає автомобіль престижного класу в оренду. В один робочий день було отримано n заявок для оренди. У кожній заявці $1 \leq i \leq n$ визначено проміжок часу (s_i, f_i) , протягом якого клієнт бажає використовувати автомобіль. При цьому кожній такій заявці та відповідному інтервалу часу відповідає сума коштів p_i , яку отримає компанія за оренду. Визначити таке рішення з надання даного автомобіля в оренду за заявками, яке дозволить отримати максимальний прибуток.

В. У службу розшуку надійшла заява про загублення людини. На схематичній мапі все місто, яке контролюється даною службою, представлено у вигляді кола. Усе коло поділено на деяку кількість сегментів (наприклад, 12). Для пошуку дана служба виділяє деяку кількість команд, меншу ніж кількість секторів. За проміжними результатами розслідування встановлено ймовірність знаходження людини у кожному з сегментів. Окрім того служба пошуку спирається в своїх розслідуваннях на те, що якщо об'єкт знаходиться на ділянці розміру S , то ймовірність того, що m команд, що проводять пошук на даній ділянці, знайдуть даний об'єкт, дорівнює

$$P = e^{-k_1 S} (1 - e^{-k_2 m}),$$

де k_1, k_2 – додатні наперед відомі константи.

Визначити кількість команд, яку службі розшуку слід направити на кожний з секторів.

Г. Проблема інвестора полягає в тому, щоб розмістити деяку наявну суму коштів у вигляді капіталовкладень у наявні проєкти. Відомо, який дохід буде отримано в результаті вкладання коштів у кожний проєкт. Однак, для кожного рівня інвестування (дискретного значення виділених коштів) встановлено свій рівень доходу. Тобто

наприклад, вкладення у деякий проєкт 10 тис. грн. призведе до отримання 12 тис. грн. у результаті, а вкладення в цей же проєкт 20 тис. грн. – принесе 27 тис. грн. Визначити стратегію інвестування, яка дозволить отримати найбільший прибуток.

Д. На одній з вулиць містечка будинки класифіковано за трьома типами: перший – звичайні житлові споруди, другий – промислові споруди, а третій – міські заклади (лікарні, школи тощо). У результаті вулиця схематично зображена набором літер, кожна з яких визначає тип будинку. У процесі збору інформації про місто була створена матриця – таблиця, в якій кожен стовпчик і рядок відповідають одному з типів будівель. Відповідно клітинка такої таблиці визначає, чи розташовані на даній вулиці міста поруч будівлі заданого типу. Матриця симетрична. Визначити, скільки існує способів взаємного розташування будинків даних типів між собою за заданою матрицею для заданої кількості будинків на вулиці, тобто кількість можливих наборів літер заданої довжини, що відповідають заданій матриці.

4.3.3 Виконати аналіз розроблених алгоритмів для розв’язання індивідуального завдання щодо часу їх роботи та кількості використаної пам’яті.

4.3.4 Виконати тестування розробленого програмного забезпечення.

4.3.5 Порівняти одержані результати виконаних тестів, провести аналіз вірності, коректності та адекватності роботи розробленого програмного забезпечення і використаних методів.

4.3.6 Оформити звіт з роботи.

4.3.7 Відповісти на контрольні запитання.

4.4 Зміст звіту

4.4.1 Мета роботи.

4.4.2 Завдання до роботи.

4.4.3 Текст розробленого програмного забезпечення з коментарями.

4.4.4 Опис та аналіз часу роботи і необхідної кількості пам’яті для роботи розроблених алгоритмів.

4.4.5 Результати роботи програмного забезпечення, що включають результати тестування та копії екранних форм.

4.4.6 Висновки, що відображають особисто отримані результати виконання роботи та їх критичний аналіз,.

4.5 Контрольні запитання

4.5.1 У чому полягає динамічне програмування?

4.5.2 Для розв'язання яких задач можна використовувати динамічне програмування?

4.5.3 З яких етапів складається процес розроблення алгоритмів динамічного програмування?

4.5.4 Які існують види динамічного програмування і чим вони відрізняються?

4.5.5 У чому полягає перекриття підзадач? Наведіть приклади.

4.5.6 Яким чином можна реалізувати запам'ятовування в процесі динамічного програмування?

4.5.7 Яким чином можна поліпшити час роботи розробленого Вами для розв'язання однієї із задач алгоритму динамічного програмування?

4.5.8 Чи можна розв'язати завдання попередньої лабораторної роботи за допомогою динамічного програмування?

4.5.9 Яким чином побудувати граф підзадач для розробленого Вами алгоритму?

4.5.10 Порівняйте жадібні алгоритми і алгоритми динамічного програмування.

5 ЛАБОРАТОРНА РОБОТА № 5

АЛГОРИТМИ ДЛЯ РОБОТИ З ГРАФАМИ.

АЛГОРИТМИ ОБХОДУ ГРАФІВ

5.1 Мета роботи

5.1.1 Вивчити алгоритми обходу графів на основі пошуку в ширину та на основі пошуку в глибину.

5.1.2 Навчитися застосовувати алгоритми обходу графів для розв'язання практичних задач.

5.2 Основні теоретичні відомості

Граф – це сукупність двох скінченних множин: множини точок та множини ліній, що попарно з'єднують деякі з цих точок.

Множина точок формує вершини (вузли) графа. Множина ліній, що з'єднують вершини графа, формує ребра (дуги) графа.

Приклад графічного зображення графа наведено на рис. 5.1.

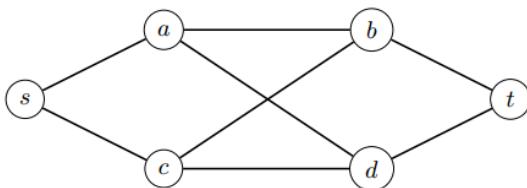


Рисунок 5.1 – Приклад графа

Найбільш розповсюдженими способами зображення графів є:

- списки суміжних вершин;
- матриця суміжності.

Для обходу графів існують два основні алгоритми:

- пошук в глибину;
- пошук в ширину.

Алгоритм пошуку в ширину в процесі обходу графа виконує обхід всіх вершин на відстані k перед тим, які перейти до пошуку вершин на відстані $k+1$.

Для відслідковування роботи алгоритму пошук в ширину розфарбовує вершини графа в кольори: білий, сірий та чорний. Спочатку вершини мають білий колір. Після того, як під час пошуку вершина досягається в перший раз, вона фарбується у сірий колір. Далі в процесі пошуку вершина фарбується в чорний колір.

Алгоритм пошуку в глибину орієнтований на те, щоб під час пошуку заглиблюватися в граф на стільки, на скільки це можливо. Пошук в глибину досліджує всі ребра, що виходять з вершини, яка була відкрита останньою, і залишає вершину тільки тоді, коли не залишається недосліджених ребер – в такому випадку пошук повертається у вершину, з якої була досягнута дана вершина.

Аналогічно пошуку в ширину пошук в глибину розфарбовує вершини графа в білий, сірий та чорний кольори.

Окрім безпосередньо обходу ребер графа та відвідування всіх вершин алгоритми пошуку в глибину та ширину корисні під час дослідження ряду важливих властивостей графів.

Сильно зв'язним компонентом орієнтованого графа називають максимальну множину вершин даного графа, для кожної пари яких справедливо, що вони досяжні одна з одної.

Для виконання пошуку сильно зв'язних компонент необхідно транспонувати заданий граф, тобто перетворити напрям ребер графа на протилежний.

Сам алгоритм пошуку сильно зв'язних компонент складається з двох пошуків у глибину:

- у початковому графі для обчислення часу завершення роботи з кожною вершиною;

- у транспонованому, де вершини розглядаються в порядку зменшення отриманих під час першого пошуку значень часу завершення роботи з кожною вершиною.

У такому випадку кожне дерево лісу пошуку в глибину, отримане під час другого пошуку, буде окремим сильно зв'язним компонентом.

При підготовці матеріалів даного розділу використано джерела [1], [3]-[6], [8], [9], [11], [12].

5.3 Завдання на лабораторну роботу

5.3.1 Ознайомитися з літературою та основними теоретичними відомостями, необхідними для виконання роботи.

5.3.2 Розробити програмне забезпечення, в якому реалізується алгоритм обходу графу на основі пошуку в глибину. Передбачити, що граф може бути як орієнтований, так і неорієнтований. В процесі пошуку має бути сформовано ліс пошуку в глибину. Для реалізації має використовуватися стек. Програмне забезпечення має бути побудовано на основі відповідного класу, який повинен дозволяти визначати граф, виконувати пошук в глибину, виводити побудований ліс пошуку в глибину, виводити результат обходу тощо.

5.3.3 Розробити програмне забезпечення, в якому реалізується алгоритм обходу графу на основі пошуку в ширину. Передбачити, що граф може бути як орієнтований, так і неорієнтований. В процесі пошуку має бути сформовано дерево пошуку в ширину. Для реалізації має використовуватися черга. Програмне забезпечення має бути побудовано на основі відповідного класу, який повинен дозволяти визначати граф, виконувати пошук в ширину, виводити побудоване дерево пошуку в ширину, виводити результат обходу тощо.

5.3.4 Виконати тестування розробленого програмного забезпечення.

5.3.5 Використовуючи розроблене програмне забезпечення, розробити окремі модулі, які використовуються для розв'язання задач у відповідності з індивідуальним завданням з таблиці 5.1, що складається з трьох задач, наведених нижче, і реалізують інтерфейсну взаємодію користувача з програмою, та виконати відповідне тестування. Обґрунтувати вибір алгоритму обходу графа для кожної задачі, де це не задано явно.

Таблиця 5.1 – Індивідуальні завдання

№ ва- ріанта	Задача 1	Задача 2	Задача 3	№ ва- ріанта	Задача 1	Задача 2	Задача 3
1	А	А	А	11	Д	А	Б
2	Б	Б	Б	12	Г	Б	А
3	В	В	В	13	В	Д	Г
4	Г	Г	Г	14	Б	В	Д

Продовження таблиці 5.1

№ ва- ріанта	Задача 1	Задача 2	Задача 3	№ ва- ріанта	Задача 1	Задача 2	Задача 3
5	Д	Д	Д	15	А	Г	В
6	А	В	Б	16	В	А	Б
7	Б	Д	А	17	Г	Б	Д
8	В	Г	Д	18	А	Д	Г
9	Г	А	В	19	Д	Г	В
10	Д	Б	Г	20	Б	В	А

Задача 1:

А. Перевірити, чи є заданий граф зв'язним чи ні. Зв'язним вважають граф, у якому між будь-якою парою вершин існує щонайменше один шлях.

Б. У заданому користувачем графі поставлено у відповідність кожній вершині деяке ціле число (може бути як від'ємним, так і додатним). Визначити такі шляхи між парами вершин, які в результаті додавання всіх чисел з кожної вершини дозволяють отримати задане користувачем значення.

В. Користувач визначає граф, задає вершину даного графа та деяку відстань. Визначити перелік всіх вершин графа, які знаходяться на заданій відстані від заданої вершини.

Г. Знайти мінімальний шлях між двома заданими вершинами у графі. Визначити довжину такого шляху та всі вершини, через які він проходить.

Д. Неорієнтований граф, у кожній вершині якого зберігаються натуральні числа, задано користувачем. Визначити, скільки в даному графі є пар взаємно простих чисел (тобто чисел, які не мають ніяких спільних дільників крім одиниці), розташованих у вершинах графа, що з'єднані ребрами.

Задача 2:

А. Селянин привів на берег річки козла, двох вовків, собаку та приніс капусту. На березі річки знаходиться човен, який вміщує тільки три об'єкти: селянина і двох тварин, або селянина, тварину і капусту. При цьому вовка не можна лишати разом з козлом та собакою, собаку – з козлом, а козла – з капустою. Визначити, яким

чином селянину перевезти всіх тварин і капусту на інший берег, щоб продовжити свій шлях.

Б. У заданому графі пронумерувати вершини у відповідності з порядком обходу в глибину, визначити кількість ребер даного графа та кількість його вершин. Обчислити середню щільність графа у вигляді частки від ділення кількості ребер графа на кількість його вершин.

В. Задано деякий набір арифметичних операцій (наприклад, додати 3, помножити на 2), які можуть бути виконані над операндом. Визначити мінімальний набір операцій, за допомогою якого можна отримати з одного заданого числа a число b . Якщо таке перетворення за допомогою заданого користувачем набору операцій виконати неможливо, то вивести відповідне повідомлення.

Г. Виконати класифікацію ребер графа на ребра дерева, зворотні ребра, прямі ребра та перехресні ребра. Визначити, чи є заданий орієнтований граф ациклічним.

Д. Два коня різного кольору розташовані на шаховій дошці: координати задаються користувачем. Всі інші чарунки шахової дошки вільні. Визначити кількість кроків кожного гравця, в результаті виконання яких коні виявляться в одній чарунці.

Задача 3:

А. Гемптон-Кортський лабіринт площею у 60 акрів привертає увагу багатьох туристів. Ваш товариш перед тим, як потрапити до одного з таких лабіринтів і продемонструвати свої здібності, вирішив вивчити план лабіринту та запитав Вас про допомогу, яким чином знайти шлях у лабіринті. Змодельуйте лабіринт за допомогою вершин, що відповідають входу в лабіринт, виходу, глухим кутам, всім точкам лабіринту, в яких є можливість вибору шляху, та з'єднань даних вершин ребрами, що відповідають шляхам у лабіринті.

Б. У квадратній коробці, що складається з чотирьох чарунок за горизонталлю і чотирьох чарунок за вертикаллю, лежать 15 пронумерованих кубиків. Кубики на початку перемішані деяким чином, тобто їх позиції та вільна чарунка визначені випадковим чином. Пересуваючи тільки один кубик за раз, необхідно отримати розташування, коли незайнятою у коробці буде задана користувачем позиція (наприклад, крайній нижній кут).

В. Виконати розкладання заданого орієнтованого графа на сильно зв'язні компоненти.

Г. Визначити порядок, за яким мають бути прочитані дисципліни під час навчання однієї з академічних груп. Моделювання виконати за допомогою графа, вершини якого представляють дисципліни, а орієнтовані ребра позначають вимоги до дисциплін (наприклад, курс С має бути прочитаний перед курсами А і В, а курс D – після курсу А тощо).

Д. Перетворити задане користувачем слово *A* у слово *B*, створюючи при цьому ланцюжок перетворень. У кожному такому перетворенні змінюється тільки одна буква слова на іншу. При цьому всі слова мають існувати у відповідній мові. Використати перелік слів з орфографічного словника для побудови графа, у якому програмно визначити ребра, які мають з'єднувати вершини тільки у тому випадку, якщо з одного слова можна отримати інше заміною однією літери.

5.3.6 Порівняти одержані результати виконаних тестів, провести аналіз вірності, коректності та адекватності роботи розробленого програмного забезпечення.

5.3.7 Оформити звіт з роботи.

5.3.8 Відповісти на контрольні запитання.

5.4 Зміст звіту

5.4.1 Мета роботи.

5.4.2 Завдання до роботи.

5.4.3 Опис та обґрунтування використання алгоритмів, за допомогою яких розв'язано індивідуальне завдання.

5.4.4 Текст розробленого програмного забезпечення з коментарями.

5.4.5 Результати роботи програмного забезпечення.

5.4.6 Висновки, що відображають особисто отримані результати виконання роботи, їх критичний аналіз.

5.5 Контрольні запитання

5.5.1 Які існують способи зображення графів та яким чином вони пов'язані між собою?

5.5.2 У чому полягає алгоритм обходу графа на основі пошуку в ширину?

5.5.3 У чому полягає алгоритм обходу графа на основі пошуку в глибину?

5.5.4 Що таке дерево пошуку в ширину та яким чином його сформувати?

5.5.5 Що таке ліс пошуку в глибину та яким чином його сформувати?

5.5.6 Яким чином розфарбовуються вершини графа під час обходу графа?

5.5.7 Яким чином від виду або способу зображення графа залежить ефективність алгоритмів пошуку в глибину і в ширину?

5.5.8 Чи є вимогливим до ресурсів алгоритм пошуку в ширину?

5.5.9 Яким чином під час реалізації в коді програми виконується повернення з тупикових вершин під час обходу графа?

5.5.10 Які структури даних використовуються для реалізації кожного алгоритму обходу графів?

6 ЛАБОРАТОРНА РОБОТА № 6

АЛГОРИТМИ ДЛЯ РОБОТИ З ГРАФАМИ.

АЛГОРИТМИ ПОШУКУ НАЙКОРОТШИХ ШЛЯХІВ

6.1 Мета роботи

6.1.1 Вивчити основні алгоритми пошуку найкоротшого шляху в графах: алгоритми Дейкстри, Флойда-Воршелла та Беллмана-Форда.

6.1.2 Навчитися застосовувати алгоритми пошуку найкоротшого шляху в графі для розв'язання практичних задач.

6.2 Основні теоретичні відомості

Алгоритм пошуку в ширину є алгоритмом пошуку найкоротшого шляху в незваженому графі, тобто в графі, кожному ребру якого відповідає одинична вага. Алгоритми Дейкстри, Беллмана-Форда та Флойда-Воршелла дозволяють працювати з графами, ребра яких мають не одиничну вагу.

Алгоритми пошуку найкоротших шляхів зазвичай ґрунтуються на твердженні, що найкоротший шлях між двома вершинами містить в собі інші найкоротші шляхи.

Алгоритм Дейкстри дозволяє виконувати пошук найкоротших шляхів з однієї вершини до всіх інших вершин у зваженому орієнтованому графі. Даний алгоритм знаходить найкоротші шляхи до вершин графа в порядку їх віддаленості від вихідної вершини, тобто спочатку знаходиться найкоротший шлях від вихідної вершини до найближчої, потім до другої найближчої тощо. Таким чином, перед початком i -ої ітерації алгоритм визначає найкоротші шляхи до $(i - 1)$ -ої вершин, найближчих до вихідної. Дані вершини, вихідна вершина та ребра найкоротших шляхів утворюють піддерево даного графа. Чергова найближча до вихідної вершина може бути знайдена серед вершин, суміжних з отриманим піддеревом: для кожної суміжної вершини обчислюється сума відстаней до найближчої вершини дерева, після чого обирається вершина з найменшою сумою.

Алгоритм Беллмана-Форда дозволяє виконувати пошук найкоротших шляхів з однієї вершини до всіх інших вершин у

випадку, коли вага кожного з ребер може бути від'ємною. Алгоритм повертає логічне значення, яке вказує, чи міститься в графі цикл з від'ємною вагою, що досягається з джерела. Якщо такий цикл існує, то алгоритм вказує на те, що рішення не існує. Якщо таких циклів не існує, то алгоритм повертає найкоротші шляхи та їх ваги.

Алгоритм Флойда-Воршелла призначений для пошуку найкоротшого шляху між всіма парами вершин. Даний алгоритм може бути реалізовано за допомогою обчислення матриці відстаней зваженого графа шляхом послідовних обчислень. Кожна матриця на відповідному кроці містить довжини найкоротших шляхів між будь-якими двома вершинами, при чому кількість проміжних вершин у них не може перевищувати відповідний номер поточного кроку.

При підготовці матеріалів даного розділу використано джерела [1]-[3], [5], [6], [8], [9], [11], [12].

6.3 Завдання на лабораторну роботу

6.3.1 Ознайомитися з літературою та основними теоретичними відомостями, необхідними для виконання роботи.

6.3.2 Розробити програмне забезпечення, в якому реалізується алгоритм Дейкстри на основі виділення відповідного класу для виконання всіх необхідних обчислень, визначення параметрів (в тому числі безпосередньо визначення графа) та отримання результатів.

6.3.3 Розширити функціональність розробленого програмного забезпечення за допомогою реалізації алгоритму Флойда-Уоршелла на основі відповідного класу, який повинен забезпечувати виконання операцій аналогічних п. 6.3.2.

6.3.4 Розширити функціональність розробленого програмного забезпечення за допомогою реалізації алгоритму Беллмана-Форда на основі відповідного класу, який повинен забезпечувати виконання операцій аналогічних п. 6.3.2.

6.3.5 Виконати тестування розробленого програмного забезпечення.

6.3.6 Використовуючи розроблене програмне забезпечення, розробити окремі модулі, які використовуються для розв'язання трьох задач, наведених нижче, у відповідності з індивідуальним завданням з таблиці 6.1 і реалізують інтерфейсну взаємодію користувача з програмою, та виконати відповідне тестування. Вибір методів

розв'язання задач обґрунтувати (якщо можливо застосувати декілька методів, то застосувати їх).

Таблиця 6.1 – Індивідуальні завдання

№ ва- ріанта	Задача 1	Задача 2	Задача 3	№ ва- ріанта	Задача 1	Задача 2	Задача 3
1	В	А	А	11	А	В	В
2	Г	Б	Б	12	Б	Д	Д
3	А	Д	Д	13	В	Г	Г
4	Д	Г	Г	14	Г	А	А
5	Б	В	В	15	Д	Б	Б
6	Д	А	А	16	А	А	А
7	Г	Б	Б	17	Б	Б	Б
8	В	Д	Д	18	В	В	В
9	Б	В	В	19	Г	Г	Г
10	А	Г	Г	20	Д	Д	Д

Задача 1:

А. Арбітражні операції дозволяють використовувати різницю в поточному курсі валюти для перетворення одиниці валюти у більшу кількість одиниць тієї ж валюти. Вхідні дані задаються таблицею обмінних курсів, яка визначає, скільки за одну одиницю однієї валюти можна купити одиниць іншої валюти. Знайти для заданої користувачем валюти найвигіднішу можливість виконання таких арбітражних операцій. Необхідні перетворення можна виконати за допомогою логарифмування.

Б. Розв'язати задану систему різницевих обмежень з n невідомими та m нерівностями вигляду

$$\begin{cases} x_i - x_j \leq q, \\ \dots \\ x_k - x_l \leq h. \end{cases}$$

В. Мапа визначає авіасполучення між містами Північної Америки. Кожний переліт з однієї точки на мапі в іншу має деяку мінімальну вартість, при чому зворотній рейс може коштувати іншу суму. Мандрівник хоче визначити авіапереліт між заданими містами,

який має мінімальну вартість, розглядаючи зокрема і варіанти з пересадками. При цьому мандрівник має дисконтну програму з деякими авіаперевізниками, за якою ціна на деякі рейси може бути для нього знижена на деяку постійну суму (тобто вартість деяких рейсів може виявитися для нього прибутковою, в такому разі сума накопичується на окремому його рахунку).

Г. Комп'ютерна мережа складається з серверів, маршрутизаторів, окремих комп'ютерів і пристроїв та може бути представлена у вигляді графа. Середній час пересилання одного пакету між вузлами для кожного фізичного з'єднання в мережі визначається за допомогою швидкості передачі даних та завантаженості даної ділянки, які задаються користувачем. Визначити найкращий шлях, яким можна передати пакети в даній мережі від заданого джерела до заданого приймача. В якості основного використати алгоритм Беллмана-Форда.

Д. Перелік завдань, які мають бути виконані в процесі роботи над деяким проектом групою спеціалістів, та їх планована тривалість визначаються користувачем. Після цього користувач визначає зв'язок між даними завданнями, визначаючи завдання, які мають завершитися до початку кожного завдання. Визначити мінімальний період часу, який знадобиться на виконання проекту.

Задача 2:

А. Компанія забезпечує інтернет-підключення користувачів за допомогою оптичного кабелю. Для підключення нових абонентів було сформовано граф, який представляє можливі траси оптичного з'єднання. На графі ребрами представлені кабельні канали, а вершинами – місця, де є можливість вибору подальшого напрямку прокладення кабелю. Визначити трасу, яка з'єднає дві задані точки на мапі і буде мати при цьому найменшу довжину.

Б. Комп'ютерна мережа складається з серверів, маршрутизаторів, окремих комп'ютерів і пристроїв та може бути представлена у вигляді графа. Час пересилання одного пакету між вузлами для кожного фізичного з'єднання в мережі визначається користувачем. Передбачити, що деякі ділянки можуть бути тимчасово недосяжними. Визначити найкращий шлях, яким можна передати пакети в даній мережі від заданого джерела до заданого приймача. В якості основного використати алгоритм Дейкстри.

В. Визначити найкоротший шлях, яким можна дістатися з Запоріжжя до Львова, використовуючи дані про перетин доріг на мапі України, представлені у вигляді графа.

Г. На мапі, представлений у вигляді графа, визначені аеропорти Європи. Для кожного існуючого перельоту між аеропортами визначено його вартість, при цьому вартість перельоту з аеропорту А в аеропорт В може не дорівнювати вартості перельоту з аеропорту В у аеропорт А. Визначити маршрут мінімальної вартості з Києва до Стокгольма, розглядаючи як прямі авіарейси, так і з пересадками.

Д. Мапа визначає автомобільні шляхи деякої частини міста Запоріжжя. Деякі вулиці мають односторонній рух, а на деяких можуть зустрічатися корки. Використовуючи дану інформацію та враховуючи обмеження швидкості на вулицях, визначити найкоротший шлях, яким можна дістатися з однієї заданої точки у Запоріжжі до іншої в даний момент часу.

Задача 3:

А. Враховуючи, що граф, описаний у попередньому завданні, представляє можливості з'єднання району з існуючими магістральними лініями, визначити всі найкоротші шляхи всередині графу.

Б. Визначити оптимальні маршрути для всіх вузлів у мережі, визначеній попереднім завданням.

В. Визначити найкоротші шляхи між будь-якими парами міст, селищ, транспортних вузлів України, використовуючи мапу, побудовану в попередньому завданні.

Г. Розглядаючи попереднє завдання, знайти найдешевші перельоти між будь-якими парами міст із заданого графа аеропортів Європи.

Д. Визначити найкоротші шляхи між всіма точками на мапі міста Запоріжжя, використовуючи обмеження попереднього завдання.

6.3.7 Порівняти одержані результати виконаних тестів, провести аналіз вірності, коректності та адекватності роботи розробленого програмного забезпечення.

6.3.8 Оформити звіт з роботи.

6.3.9 Відповісти на контрольні запитання.

6.4 Зміст звіту

6.4.1 Мета роботи.

6.4.2 Завдання до роботи.

6.4.3 Опис та обґрунтування використання алгоритмів, за допомогою яких розв'язано індивідуальне завдання.

6.4.4 Текст розробленого програмного забезпечення з коментарями.

6.4.5 Результати роботи програмного забезпечення.

6.4.6 Висновки, що відображають особисто отримані результати виконання роботи, їх критичний аналіз.

6.5 Контрольні запитання

6.5.1 Для роботи з якими видами графів призначені алгоритми пошуку найкоротших шляхів?

6.5.2 За яких умов може використовуватися алгоритм Дейкстри?

6.5.3 У чому полягає алгоритм Дейкстри?

6.5.4 На основі яких структур даних побудований алгоритм Дейкстри?

6.5.5 Для розв'язання яких задач призначений алгоритм Дейкстри?

6.5.6 Для якого виду графів може застосовуватися алгоритм Флойда-Воршелла?

6.5.7 У чому полягає алгоритм Флойда-Воршелла?

6.5.8 У яких випадках можна застосовувати алгоритм Беллмана-Форда?

6.5.9 У чому полягає алгоритм Беллмана-Форда?

6.5.10 Що таке метод послаблення та для чого він використовується?

7 ЛАБОРАТОРНА РОБОТА № 7 МОДЕЛЮВАННЯ ТРАНСПОРТНОЇ МЕРЕЖІ ЗА ДОПОМОГОЮ ГРАФІВ

7.1 Мета роботи

7.1.1 Вивчити метод Форда-Фалкерсона для розв'язання задачі про максимальний потік у транспортній мережі.

7.1.2 Навчитися застосовувати метод Форда-Фалкерсона для розв'язання практичних завдань.

7.2 Основні теоретичні відомості

Під час *моделювання транспортної мережі* за допомогою графів кожне орієнтоване ребро можна розглядати як деякий канал, яким пересувається деякий продукт. Кожний канал має задану пропускну здатність, що характеризує максимальну швидкість пересування продуктів каналом. Вершини є точками перетину каналів. Через вершини, відмінні від джерела та стоку, продукт просувається, не накопичуючись.

Задача про максимальний потік полягає у знаходженні такого потоку за транспортною мережею, що сума потоків з витоку (до стоку) є максимальною.

Метод Форда-Фалкерсона дозволяє розв'язати задачу про максимальний потік шляхом ітеративного збільшення значення потоку. Спочатку потік обнуляється, а далі на кожній ітерації величина потоку збільшується шляхом пошуку збільшуючого шляху.

При підготовці матеріалів даного розділу використано джерела [1], [3], [8], [9].

7.3 Завдання на лабораторну роботу

7.3.1 Ознайомитися з літературою та основними теоретичними відомостями, необхідними для виконання роботи.

7.3.2 Розробити програмне забезпечення, в якому реалізується алгоритм Форда-Фалкерсона на основі виділення відповідного класу для виконання всіх необхідних обчислень, визначення параметрів (в

тому числі безпосередньо визначення графа) та отримання результатів.

7.3.3 Виконати тестування розробленого програмного забезпечення.

7.3.4 Використовуючи розроблене програмне забезпечення, розв'язати наступну задачу, реалізуючи інтерфейсну взаємодію користувача з програмою, та виконати відповідне тестування. Задачу обрати у відповідності з номером варіанта з лабораторної роботи № 3.

Варіант № 1

Компанія, що виробляє сірники, має задану кількість фабрик, розташованих у містах $1, \dots, n$, а сховища для зберігання продукції знаходяться у містах $n + 1, \dots, n + m$. Для постачання продукції на склад з фабрики орендуються вантажівки іншої фірми, які окрім даної продукції перевозять також і іншу. Вантажівки пересуваються за деякими маршрутами між містами та мають обмежену вантажність, тобто для кожної пари міст задано максимальну кількість продукції, яку можна перевезти. Транспортна мережа визначається фірмою, якій належать вантажівки. Визначити максимальну кількість продукції, яку можна виробити та доставити на склад за день, користуючись існуючою транспортною мережею, та відповідний мінімальний розріз.

Варіант № 2

Для виконання заданої кількості завдань виділено задану кількість однакових робітників. Для кожного завдання визначено тривалість його виконання, час, коли завдання поступає на виконання, та максимальний строк завершення завдання. Визначити розклад виконання завдань, який має мінімальну затримку виконання.

Варіант № 3

Компанія розмістила оголошення при прийом на роботу спеціалістів у результаті розширення. На оголошення було отримано n резюме від потенційних працівників. Перед прийомом на роботу з ними має бути проведене інтерв'ю. Для кожного з m інтерв'юерів виділено окремий кабінет. Бажаючи отримати якісних працівників, компанія обмежена в часі. У результаті попереднього аналізу визначено, які кандидати можуть бути направлені до яких інтерв'юерів. Визначити максимальну кількість інтерв'ю, які можна провести одночасно, та учасників відповідних інтерв'ю.

Варіант № 4

Комп'ютери з'єднані між собою за допомогою деякої мережі. Мережа реалізується через деякі вузли (маршрутизатори) та лінії з'єднання. Кожна така лінія з'єднання характеризується максимальною швидкістю передачі даних. Яка максимальна швидкість передачі даних між двома заданими комп'ютерами даної мережі? Визначити відповідний мінімальний розріз.

Варіант № 5

Підприємство з виробництва іграшок в процесі постачання їх клієнтам реалізує двоетапну логістику: спочатку продукція доставляється до сховищ, а потім перевозиться в магазини, де продається кінцевим споживачам. Перевезення між підприємством та сховищами реалізується за допомогою літаків, при чому кількість продукції, яку можна перевезти між підприємством та кожним сховищем, обмежена. Кількість продукції, яку вміщає кожне сховище, обмежено. Постачання продукції зі сховищ до магазинів виконується за допомогою вантажівок, які такою мають обмежену вантажність. Визначити максимальну кількість продукції, яку можна за один раз доставити з виробництва в магазини, та відповідні такому постачанню маршрути.

7.3.5 Порівняти одержані результати виконаних тестів, провести аналіз вірності, коректності та адекватності роботи розробленого програмного забезпечення.

7.3.6 Оформити звіт з роботи.

7.3.7 Відповісти на контрольні запитання.

7.4 Зміст звіту

7.4.1 Мета роботи.

7.4.2 Завдання до роботи.

7.4.3 Текст розробленого програмного забезпечення з коментарями.

7.4.4 Результати роботи програмного забезпечення.

7.4.5 Висновки, що відображають особисто отримані результати виконання роботи, їх критичний аналіз.

7.5 Контрольні запитання

7.5.1 Що таке транспортна мережа?

7.5.2 Що таке потік у транспортній мережі?

7.5.3 Яким властивостям повинен задовольняти потік у мережі?

7.5.4 Які шляхи називаються збільшувачими у транспортній мережі?

7.5.5 У чому полягає теорема Форда-Фалкерсона?

7.5.6 Що таке залишкова мережа?

7.5.7 Що таке розріз транспортної мережі?

7.5.8 Який розріз транспортної мережі називають мінімальним?

7.5.9 Що таке паросполучення?

7.5.10 З яких кроків складається алгоритм Форда-Фалкенсона?

ЛІТЕРАТУРА

1. Кормен, Т. Вступ до алгоритмів [Текст] / Томас Кормен, Чарльз Лейзерсон, Рональд Рівест. – К. : К. І. С., 2019. – 1288 с.
2. Кормен, Т. Г. Алгоритми доступно [Текст] / Т. Г. Кормен. – Київ : К. І. С., 2021. – 194 с.
3. Introduction to Algorithms, fourth edition [Текст] / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. – The MIT Press, 2022. – 1312 p.
4. Ришковець, Ю. В. Алгоритмізація та програмування. Частина 2 [Текст] : навчальний посібник / Ю. В. Ришковець, В. А. Висоцька. – Львів: Видавництво «Новий Світ-2000», 2020. – 320 с.
5. Мелешко, Є. В. Алгоритми та структури даних [Текст] : Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання / Є. В. Мелешко, М. С. Якименко, Л. І. Поліщук. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
6. Крєневич, А.П. Алгоритми і структури даних. Підручник [Текст] / А.П. Крєневич. – К. : ВПЦ "Київський Університет", 2021. – 200 с.
7. Knuth, D. The Art of Computer Programming, Vol. 1 [Текст] : Fundamental Algorithms / D. Knuth. – Addison-Wesley Professional, 1997. – 672 p.
8. Sedgewick, R. Algorithms (4th Edition) [Текст] / Robert Sedgewick, Kevin Wayne. – Addison-Wesley Professional, 2011. – 976 p.
9. Erickson, J. Algorithms [Текст] / J. Erickson. – 2019. – 472 p. – ISBN : 978-1-792-64483-2.
10. Louridas, P. Algorithms [Текст] : The MIT Press Essential Knowledge series / P. Louridas. – Cambridge, Massachusetts : The MIT Press, 2020. – 312 p.
11. Bhargava, A. Y. Grokking Algorithms [Текст] : An illustrated guide for programmers and other curious people / A. Y. Bhargava. – New York : Manning Publications, 2016. – 256 p.
12. Mehlhorn, K. Algorithms and Data Structures [Текст] : The Basic Toolbox / Kurt Mehlhorn, Peter Sanders. – Berlin : Springer, 2008. – 300 p.