

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І. МЕЧНИКОВА
Кафедра математичного забезпечення комп'ютерних систем



“ЗАТВЕРДЖУЮ”

Проректор з науково-педагогічної роботи

_____ 20__ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК21 «Структури даних та алгоритми»

(назва навчальної дисципліни)

Рівень вищої освіти _____

Перший (бакалаврський)

Галузь знань _____

12 – Інформаційні технології

Спеціальність _____

123 – Комп'ютерна інженерія

(код і назва спеціальності)

Освітньо-професійна програма _____

Комп'ютерна інженерія

(назва ОПП/ОНП)

Робоча програма навчальної дисципліни «Структури даних та алгоритми». –
Одеса: ОНУ, 2023. – 14 с.

Розробник:

Антоненко О.С., к.ф.-м.н., доцент кафедри МЗКС

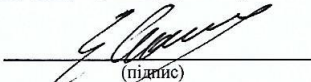
Шпінарева І.М., к.ф.-м.н., доцент кафедри МЗКС

Лісіцина І.М., старший викладач кафедри МЗКС

Робоча програма затверджена на засіданні кафедри математичного забезпечення комп'ютерних систем

Протокол № 1 від. “ 29 ” серпня 2023 р.

Завідувач кафедри


(підпис)

(Святослав МАЛАХОВ)
(Ім'я ПРІЗВИЩЕ)

Погоджено із гарантом ОПП «Комп'ютерна інженерія»


(підпис)

(Людмила ВОЛОШУК)
(Ім'я ПРІЗВИЩЕ)

Схвалено навчально-методичною комісією (НМК) з ІТ спеціальностей факультету МФІТ

Протокол № 1 від. “ 31 ” серпня 2023 р.

Голова НМК


(підпис)

(Алла РАЧИНСЬКА)
(Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри МЗКС

Протокол № 1 від. “ 28 ” 08 2024 р.

Завідувач кафедри


(підпис)

(Святослав МАЛАХОВ)
(Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № _____ від. “ _____ ” _____ 20__ р.

Завідувач кафедри

(підпис)

(_____)
(Ім'я ПРІЗВИЩЕ)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, спеціалізація, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Загальна кількість: кредитів – 4 годин – 120 змстових модулів – 3	Галузь знань <u>12 – Інформаційні технології</u> (шифр і назва) Спеціальність <u>123 – Комп'ютерна інженерія</u> (шифр і назва) Рівень вищої освіти: <u>перший</u> (бакалаврський)	<i>Обов'язкова</i>	
		<i>Рік підготовки:</i>	
		2	2
		<i>Семестр</i>	
		1 (3)	1 (3)
		<i>Лекції</i>	
		36 год.	6 год.
		<i>Практичні, семінарські</i>	
		<i>Лабораторні</i>	
		18 год.	6 год.
		<i>Самостійна робота</i>	
		66 год.	108 год.
		Форма підсумкового контролю: іспит	

2. Мета дисципліни

Мета курсу полягає ознайомлення студентів з основними, класичними структурами даних та алгоритмами, які використовуються при програмуванні, а саме різним методам сортування, рекурсивним алгоритмам, рекурсивним структурам даних, такі як списки, дерева, а також придбання практичних навичок щодо проектування складних програм з використанням сучасних технологій структурного програмування, формування професійного відношення до створення програмного продукту, розуміння того факту, що тільки ретельно продуманий вибір необхідних структур даних та алгоритмів для конкретної задачі може бути гарантією написання програми, що відповідає сучасним вимогам.

Предметом дисципліни є алгоритми використання та обробки структур даних у різних класах завдань.

Основними задачами дисципліни:

- вивчення різних форм організації даних в програмах і методів їх обробки і застосування в різних класах завдань,
- освоєння технології програмування мовами C/C++/C# структур даних і алгоритмів їх обробки,
- розбір найважливіших алгоритмів пошуку та сортування, їх порівняльна характеристика, оцінка складності;

— ознайомлення студентів з базовими структурами даних, такими як стеки, черги, лінійні списки, дерева пошуку, з найбільш простій і доступній їх реалізацією;

— розгляд рекурсії як одного з найважливіших засобів алгоритмічного вирішення завдань.

Курс припускає лекційний і практичний блоки. Лекційний блок містить докладний теоретичний матеріал, велика кількість прикладів по кожній з даних структур даних. В рамках лекційного курсу проводиться також порівняльний аналіз деяких алгоритмів, реалізованих на різних структурах даних. Лабораторний блок припускає закріплення теоретичного матеріалу у формі рішення завдань по кожній з розглянутих тем.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей** (згідно ОПП «Комп'ютерна інженерія» від 2022 р.):

а) загальних:

- *Z12. Здатність застосовувати базові знання з фундаментальної та прикладної математики в професійній діяльності.*

б) фахових:

- *P15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення;*

- *P17. Здатність застосовувати закономірності випадкових явищ, ймовірнісно-статистичні методи, основи теорії чисельних методів та сучасні методи дискретної математики для аналізу і синтезу складних систем, методи кількісної оцінки інформації і створення коригуючих кодів при розв'язанні прикладних і наукових завдань в області комп'ютерної інженерії.*

Програмні результати навчання:

- *N1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж;*

- *N2. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах;*

- *N6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей;*

- *N7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності;*

- *N8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.*

Очікувані результати навчання. У результаті вивчення навчальної дисципліни студент повинен

знати:

- *різноманітні типи та структури даних;*

- *про методи пошуку та сортування, рекурсію;*

- *про методи аналізу алгоритмів і критерії оцінки їх складності;*

- *про прийоми роботи з базовими структурами даних;*

- *про класифікацію структур даних, їх особливості, розміщення в пам'яті, дисципліни доступу;*

- про формальне завдання абстрактних типів даних.

ВМІТИ:

- використовувати у програмах механізм адресних покажчиків;
- використовувати динамічну область пам'яті програми для розміщення змінних, масивів, структур даних;
- програмувати ітераційні і рекурсивні функції доступу до структур даних;
- вибирати і використовувати структури даних для організації складних керуючих та інформаційних структур;
- використовувати лінійні послідовні структури даних, а також деревовидні структури даних для вирішення практичних задач програмування;
- використовувати технологію структурного та об'єктно-орієнтованого програмування при створенні програм обробки складних структур даних.

3. Зміст навчальної дисципліни

ЗМІСТОВИЙ МОДУЛЬ 1. ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ СТРУКТУР ДАНИХ.

ТЕМА 1. Складність алгоритмів.

Поняття складності алгоритму. Аналіз алгоритмів сортування і порівняння їх ефективності. Взаємозв'язок алгоритму і структур даних. Застосування структурної і об'єктно-орієнтованої технології проектування програм і структур даних. [1, 2, 3, 6].

ТЕМА 2. Алгоритми внутрішнього сортування.

Прості алгоритми внутрішнього сортування: обмінне сортування, сортування вставками, сортування вибором. Вдосконалені алгоритми внутрішнього сортування: швидке сортування, сортування злиттям, коміркове сортування [1, с. 163-228], [2, 3, 5, 6].

ТЕМА 3. Абстрактні типи даних.

Абстрактні типи даних (АТД). Визначення абстрактного типу даних: тип і структура даних, оператори. АТД і базові типи даних. АТД і програмні функції. АТД і класи. Класифікація АТД: лінійні і нелінійні АТД. Лінійні АТД з прямим доступом: масиви, записи, файли. Лінійні АТД з доступом по ключу: словник, хеш-таблиця [1, с. 247-300], [2, 3].

ТЕМА 4. Лінійні структури даних.

Список. Математичне визначення списку. Термінологія списків. АТД "Список". Оператори АТД "Список". Реалізація АТД у вигляді масиву, зв'язного списку. Однонаправлений і двонаправлений списки. Стек, як різновид списку. Метод доступу до стека LIFO. АТД "Стек". Оператори АТД "Стек". Реалізація АТД за допомогою масиву, однонаправленого списку. Черга, як різновид списку. Метод доступу до черги FIFO. АТД "Черга". Оператори АТД "Черга". Реалізація АТД за

допомогою масиву, однонаправленого списку. АТД "Дек". Оператори АТД "Дек". Реалізація АТД за допомогою масиву, двонаправленого списку. АТД "Словник". Оператори АТД "Словник". Реалізація АТД за допомогою масиву, за допомогою хеш - таблиці. Методи хешування. Вирішення колізій хешування. [1, с. 247-300], [2, 6, 7, 8].

ЗМІСТОВИЙ МОДУЛЬ 2. НЕЛІНІЙНІ СТРУКТУРИ ДАНИХ.

ТЕМА 1. ДЕРЕВА.

Рекурсивне визначення структури дерева. Термінологія дерев. Впорядковані і невлпорядковані дерева. Обхід дерева: прямий, зворотний, симетричний. Позначене дерево. Дерево виразів, обхід дерева виразів, префіксна і постфіксна форма виразів. АТД "Дерево". Оператори АТД "Дерево". Дерево двійкового пошуку. АТД "Дерево пошуку". Оператори АТД "Дерево пошуку". Збалансовані дерево пошуку. АВЛ - дерево. 2-3 - дерево. В-дерево. АТД "В-дерево" і його оператори. Червоно-Чорне дерево. АТД "Червоно-Чорне - дерево" і його оператори [1, с. 301-370], [2, 6, 8].

ТЕМА 2. МНОЖИНА "ЧЕРГА З ПРІОРИТЕТАМИ".

АТД "Черга з пріоритетами". Оператори АТД "Черга з пріоритетами". Реалізація АТД за допомогою зв'язного списку, купи (heap). Реалізація купи за допомогою масиву, пірамідальне сортування [1, с. 168-185], [2, 6, 8].

ТЕМА 3. Алгоритми на рядках.

Основні поняття и термінологія. Пошук підрядка. Алгоритм Боєра-Мура. Алгоритм Кнутта-Морріса-Пратта. Префіксне дерево. Суфіксне дерево. Алгоритм Укконена побудови суфіксного дерева за лінійний час [1, 2, 6].

ЗМІСТОВИЙ МОДУЛЬ 3. ГРАФИ

ТЕМА 1. Орієнтований граф.

Основні поняття и термінологія. АТД "Орграф" та оператори. Реалізація АТД за допомогою матриці суміжності і списків суміжності. Основні алгоритми обробки орграфа: алгоритм знаходження найкоротшого шляху (алгоритм Дейкстри) [1, с. 559-776], [2, 3].

ТЕМА 2. Неорієнтований граф.

Основні поняття і термінологія. АТД "Неорієнтований граф" і його оператори. Реалізація АТД за допомогою матриці суміжності, списків суміжності. Обхід неорієнтованого графа. Пошук в глибину. Пошук завширшки [1, с. 559-776], [2, 3].

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин									
	Денна форма					Заочна форма				
	Усього	у тому числі				Усього	у тому числі			
		л	п	лаб	ср		л	п	лаб	ср
1	2	3	4	5	6	7	8	9	10	11
Змістовний модуль 1. ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ СТРУКТУР ДАНИХ										
Тема 1.	10	4			6	11,5	0,5			11
Тема 2.	13	4		3	6	13,5	0,5		2	11
Тема 3.	9	3			6	11,5	0,5			11
Тема 4.	13	4		3	6	12,5	0,5		1	11
Змістовний модуль 2. НЕЛІНІЙНІ СТРУКТУРИ ДАНИХ										
Тема 1.	17	5		4	8	16	1		1	14
Тема 2.	17	5		4	8	16	1		1	14
Тема 3.	17	5		4	8	16	1		1	14
Змістовний модуль 3. ГРАФИ										
Тема 1.	11	3			8	11,5	0,5			11
Тема 2.	13	3			10	11,5	0,5			11
Всього годин	120	36		18	66	120	6		6	108

5. Теми семінарських занять

Семінарські заняття не передбачені

6. Теми практичних занять

Практичні заняття не передбачені

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	СОРТУВАННЯ ПРОЕКТ «Порівняння методів сортувань за часом виконання»	3/2
2	СТЕК(Обернений Польський запис) ПРОЕКТ	3/1
3	Обчислення виразу методом рекурсивного спуску з використанням двох стеків або дерева ПРОЕКТ	4/1
4	Черга з пріоритетами. Задача моделювання ПРОЕКТ	4/1
5	МНОЖИНА Операції над множинами ПРОЕКТ	4/1
	Разом	18/6

Методичне забезпечення лабораторних робіт – методичні матеріали у електронному вигляді.

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	АТД "Словник". Оператори АТД "Словник". Реалізація АТД за допомогою масиву, за допомогою хеш - таблиці.	8/14
2	АТД "В-дерево" і оператори.	8/15
3	Червоно-Чорне дерево. АТД "Червоно-Чорне - дерево" і його оператори	8/15
4	Методи хешування	8/14
5	Пошук підрядка	8/14
6	АТД "Орграф" та оператори.	8/14
7	Основні алгоритми обробки орграфа	8/11
8	Обхід неорієнтованого графа. Пошук в глибину. Пошук завширшки	10/11
	Разом	66/108

До самостійної роботи відноситься:

[1] – підготовка до лекцій та лабораторних занять

8.1. Курсовий проект

Курсовий проект не передбачений

9. Методи навчання

Лекції з використанням мультимедійного презентаційного матеріалу.

10. Методи контролю

Протягом семестру студенти виконують 1 теоретичну контрольну роботу.

При виконанні **теоретичних контрольних робіт** студент повинен пройти тест, який складається з 20 теоретичних запитань відкритого та закритого типів, направлених на оцінювання знання студентом визначень понять, принципів функціонування алгоритмів та структур даних, а також завдань, направлених на застосування логічного мислення.

Під час **підсумкового контролю** студент повинен відповісти на 2 теоретичні запитання з переліку, наведеному у п. 11.1, та виконати 3 практичних завдання, приклади яких наведені у п. 11.2.

10.1. Критерії оцінювання відповідей на підсумковому контролі

Результат підсумкового контролю оцінюється за 100-бальною шкалою, яка потім перераховується на 30-бальну шляхом множення на коефіцієнт 0,3 та округлення. Мінімальна кількість балів, що зараховується як позитивний результат, дорівнює 60. Бали розподіляються наступним чином: 40 балів – теоретична частина та 60 балів – практична.

На підсумковому контролі студент отримує 2 теоретичні запитання (кожне максимально оцінюється у 20 балів) та 3 практичних завдання, (кожне максимально оцінюється у 20 балів). При оцінюванні якості відповідей беруться до уваги наступні положення.

1. Відповідь повинна бути повною і короткою. Вона не повинна містити матеріал, що не відноситься до суті запитання.
2. Твердження повинні бути сформульовані чітко.
3. Відповіді, що мають помилкові твердження, оцінюються виходячи з близькості відповіді до правильної.
4. Пропуски в обґрунтуванні тверджень призводять до зменшення кількості балів.
5. Недоліки та неточності при викладенні матеріалу зменшують кількість балів.
6. Незнання і нерозуміння основної ідеї теоретичного запитання або практичного завдання призводить до зняття до 90% балів.
7. Якщо відповідь на запитання відсутня, то виставляється нуль балів.

11. Питання до контролю

11.1. Питання для підсумкового контролю

- 1) " Якісність " програми - критерії оцінки. Функції для оцінки часу : точність : секунди , мілісекунди. Поняття про автоматизацію тестування.
- 2) Функція " O " , призначення, властивості , тут же буде завдання : дається нескладна програмка кшталт підрахунку суми елементів масиву , необхідно буде визначити значення функції " O " для цього прикладу .
- 3) Критерії порівняння та класифікацію алгоритмів сортування. Сортування за ключем.
- 4) Сортування методом прямого обміну . Приклад коду програми . Взяти набір чисел X_1, X_2, \dots, X_n і на папері відобразити , як буде змінюватися даний масив після кожного проходу зовнішнього циклу.
- 5) Сортування методом прямої вставки. Приклад коду програми . Взяти набір чисел X_1, X_2, \dots, X_n і на папері відобразити , як буде змінюватися даний масив після кожного проходу зовнішнього циклу.
- 6) Сортування методом прямого вибору. Приклад коду програми . Взяти набір чисел X_1, X_2, \dots, X_n і на папері відобразити , як буде змінюватися даний масив після кожного проходу зовнішнього циклу.
- 7) Сортування методом злиття в разі , коли вихідний масив не впорядковані і має розмір кратний 2^n і у випадку , коли розмір не кратний цій величині . Також взяти набір чисел X_1, X_2, \dots, X_n і на папері відобразити , як буде змінюватися даний масив після кожного проходу зовнішнього циклу .
- 8) Пошук елемента в масиві : простий перебір , метод половинного поділу. Генерація випадкових величин з рівномірним і випадковим законом розподілу.
- 9) Алгоритм пошуку " k - ого " за величиною елемента масиву. Приклад коду. На папері показати , як будуть переміщатися елементи , в ході "частковою сортування" , набір чисел : X_1, X_2, \dots, X_n ...
- 10) Сортування методом прямого вибору . Приклад коду програми. Взяти набір чисел X_1, X_2, \dots, X_n і на папері відобразити , як буде змінюватися даний масив

після кожного проходу першого зовнішнього циклу і другого зовнішнього .
Способи розрахунку " відстаней " для даного алгоритму .

11) " Швидке сортування " . Приклад коду програми . Взяти набір чисел X_1 , X_2 , ... , X_n і на папері відобразити , як буде змінюватися даний масив на кожному з рівнів спуску рекурсії .

12) " Пірамідальне сортування " . Приклад коду програми . Взяти набір чисел X_1 , X_2 , ... , X_n і на папері відобразити , як буде змінюватися даний масив на кожному кроці побудови піраміди на стадіях прямого і зворотного ходу .

13) Рекурсія. Основні характеристики. Переваги та недоліки + написати нескладну задачу рекурсивним шляхом розрахувати , наприклад суму елементів масиву або щось схоже.

14) Рекурсія і спосіб імітації рекурсії за допомогою стека. Переваги . Написати нескладну задачу за допомогою " імітації рекурсії - стека " розрахувати , наприклад , суму елементів масиву або щось схоже.

15) Динамічні структури даних, визначення. Модель представлення оперативної пам'яті. Стек . Види реалізацій стеків (з практичної роботи) , недоліки і плюси кожної з трьох моделей.

16) Динамічні структури даних. Список односпрямований. Черга. Операції додавання, видалення елементів, пошук у черзі елемента - приклади коду.

17) Динамічні структури даних. Список з двома зв'язками. Дек. Операції додавання, видалення елементів, пошук у черзі елемента - приклади коду.

18) Древа . Способи подання дерев. Операція додавання в дерево нового вузла.

19) Операції проходу по дереву.

20) Асоціативні масиви. Узагальнення основних операцій.

21) Двійкові кореневі дерева. Спосіб їх зберігання в пам'яті.

22) Черги з пріоритетами. Визначення куп (heap) і їх інваріант.

23) Зберігання куп в масиві. Ідентифікація елементів.

24) Допоміжні операції на купах. Відновлення мінімальності і зменшення ключа.

25) Основні операції на купах та їх складність.

26) Швидка побудова купи.

27) Древа пошуку. Принцип реалізації основних операцій.

28) Необхідність балансування. Привести приклад.

29) Повороти. Збереження впорядкованості.

30) Вставка елемента в AVL-дерево.

31) Видалення елемента з AVL-дерева.

32) Червоно-чорні дерева.

33) Префіксні дерева. Структура і призначення.

34) Принципи реалізації основних операцій для префіксних дерев.

35) Вимоги за обсягом пам'яті для префіксних дерев.

36) Суффіксні дерева.

37) Принцип хешування списками. Основні операції.

38) Алгоритмічна складність операцій по роботі з хеш. Порівняння зі збалансованими деревами.

39) Детерміновані методи вибору хеш-функцій.

40) Універсальне хешування. Середня амортизована складність.

41) Принцип хешування з відкритою адресацією. Пристрій основних операцій.

42) Порівняння хешування списками і відкритою адресацією.

43) Вибір послідовностей перегляду для вирішення колізій при відкритій адресації.

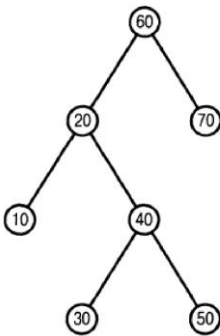
11.2. Приклад практичних завдань для підсумкового контролю

1. Проаналізуйте дерево, зображене на рисунку. Вкажіть такі вузли дерева.

- Корінь дерева.
- Батьківські вузли.
- Дочірні вузли.
- Брати.
- Батьки вузла 50.
- Нащадки вузла 50.
- Листя.

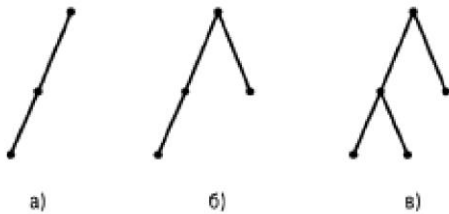
1. Відповіді.
 1.1. 60;
 1.2. 60, 20, 40;
 1.3. 20, 70; 10, 40; 30, 50;
 1.4. 20 и 70, 10 и 40; 30 и 50.
 1.5. 40, 20, 60;

2. Чому дорівнює висота дерева, зображеного на рисунку?

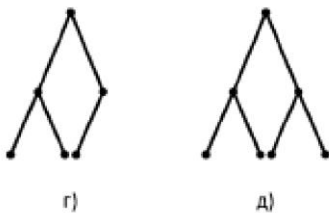


2. Відповідь. 4

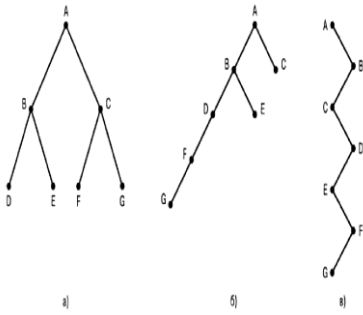
3. Проаналізуйте бінарні дерева, зображені на рисунку. Яке з них є досконалим? Повним? Збалансованим?



3. Відповідь
 досконалі: б, г, д; повні: е;
 сбалансированное: б, в, г, д.

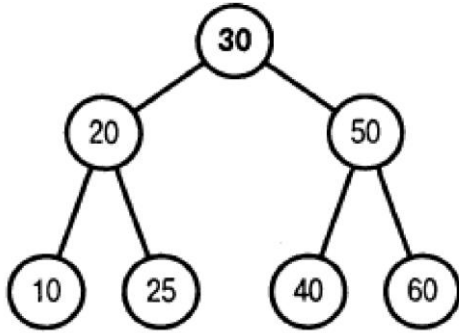


4. Вкажіть порядок прямого, симетричного та зворотного обходу бінарного дерева, представленого на рисунку.



4. Відповіді
 Обхід у прямому порядку: A, B, D, E, C, F, G;
 у симетричному порядку: D, B, E, A, F, C, G;
 в зворотньому порядку: D, E, B, F, G, C, A.

5. Подайте у вигляді масиву повне бінарне дерево пошуку, показане на рисунку.



6. Яке досконале бінарне дерево є масивом, зображеним на рисунку?

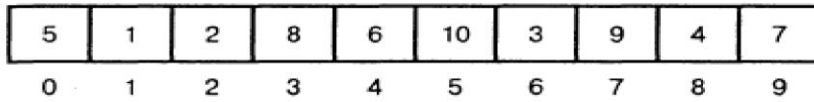


Рис. 10.43. Масив, згадуваний в питанні 9

7. Чи є дерево, зображене на рисунку бінарним?

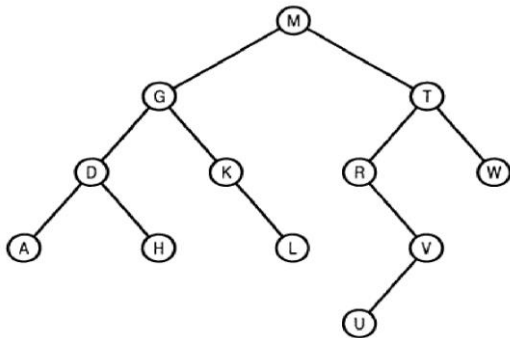


Рис. 10.44. Дерево, згадуване в питанні 10 і вправі 2.1

12. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота			Сума
Змістовий модуль №1			
T1	T2	T3	30
10	10	10	

Поточне тестування та самостійна робота					Сума	Підсумковий контроль	Загалом
Змістовий модуль №2		Змістовий модуль №3					
T1	T2	T3	T1	T2	40	30	100
10	10	10	5	5			

T1, T2 ... – теми змістових модулів.

Шкала оцінювання: національна та ECTS

Загальна сума балів	Оцінка ECTS	Національна шкала	
90 — 100	A – «відмінно»	5 «відмінно»	« за лік »
85 — 89	B – «дуже добре»	4 «добре»	
75 — 84	C – «добре»		
70 — 74	D – «задовільно»	3 «задовільно»	
60 — 69	E – «допустимо»		
35 — 59	F – «незадовільно з можливістю повторного складання»	2 «незадовільно»	« н ез а лі к »
0 — 34	FХ – «незадовільно з обов’язковим повторним курсом»		

13. Методичне забезпечення

Комплекс навчально-методичного забезпечення дисципліни; нормативні документи; презентаційні матеріали.

14. Рекомендована література

14.1. Основна література

1. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліффорд Стайн Вступ до алгоритмів. — К. : К. І. С., 2019. — 1288 с. ISBN 978-617-684-239-2.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest; Clifford Stein, Introduction to Algorithms (4th ed.). MIT Press and McGraw-Hill., 2022. – 1312 pp. ISBN 0-262-04630-X
3. A. V. Aho, J. E. Hopcroft, J. D. Ullman, Data Structures and Algorithms. Addison-Wesley, 1983. ISBN 0-201-00023-7.

4. Donald E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms. Third Edition* / Reading, MA: Addison-Wesley, 1997. – 650pp. ISBN 0-201-89683-4
5. Donald E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching. Second Edition* / Reading, MA: Addison-Wesley, 1998. – 780pp. ISBN 0-201-89685-0
6. William H. Ford, William R. Topp. *Data Structures with C++ Using STL (2nd. ed.)*. Prentice Hall PTR, USA, 2002.
7. Robert Sedgewick, *Algorithms, 3rd Edition, in C++, Parts 1–4: Fundamentals, Data Structures, Sorting, and Searching*. Reading, MA: Addison-Wesley. 1998. – ISBN 978-0201350883.
8. Robert Sedgewick, Kevin Wayne. *Algorithms, 4th Edition*, Addison-Wesley Professional, ISBN 978-0321573513, 2011. – 976 p.

14.2. Допоміжна література

9. Adam Drozdek, *Data Structures and Algorithms in C++, 4th Edition* / Cengage Learning, 2012. – 784 p.
10. Крєневич А.П. *Алгоритми і структури даних. Підручник*. – К.: ВПЦ "Київський Університет", 2021. – 200 с.
11. В.М. Ільман. *Алгоритми, дані і структури. Навч. посіб.* / О.П. Іванов, Л.О. Панік. Дніпропет. нац. ун-т залізн. трансп.ім. акад. В. Лазаряна. – Дніпро, 2019. – 134 с.
12. *Алгоритми і структури даних: навч. посіб.* / Т. О. Коротєєва ; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». — Львів: Вид-во Львів. політехніки, 2014. — 280 с.
13. E. Ukkonen. On-line construction of suffix trees. *Algorithmica* 14, 3 (September 1995), 249–260. <https://doi.org/10.1007/BF01206331>
14. Jon Bentley, *Programming Pearls, Second Edition* / Addison-Wesley, Inc., 2000. — 239 p.
15. N. Wirth, *Algorithms and Data Structures* / Prentice Hall, 1985. — 288 с.
16. Кубенский А. А. *Структуры и Алгоритмы обработки данных. Объектно – ориентированный подход и реализация на C++*. М., Издат-во "Вильямс", 1998 г.
17. Jeffrey McConnel, *Analysis of Algorithms 2nd Edition* / Jones & Bartlett Learning, 2007. — 451 p.
18. Bertrand Meyer, *Object-Oriented Software Construction, second edition*, Prentice Hall, 1997. — 1296 p.
19. Шень А. *Программирование: теоремы и задачи. 2-е изд.*, М.: МЦНМО, 2004.

15. Електронні інформаційні ресурси

1. *Data Structure and Algorithms Tutorial* – Tutorialspoint – Режим доступу: https://www.tutorialspoint.com/data_structures_algorithms/index.htm
2. *Data Structure Visualization - Computer Science* – Режим доступу: <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
3. *Розробка та аналіз алгоритмів, Частина 1* – Prometheus: https://courses.prometheus.org.ua/courses/KPI/Algorithms101/2015_Spring/course/
4. Презентації лекцій та методичні вказівки до виконання лабораторних робіт в електронному вигляді