3. What are the advantages and disadvantages of binary search? *Quora*. URL: https://www.quora.com/What-are-the-advantages-and-disadvantages-of-binary-search (дата звернення: 28.11.2025).

4. Algorithms – How come the time complexity of Binary Search is log *n*. URL: https://math.stackexchange.com/questions/3224473/how-come-the-time-complexity-of-binary-search-is-log-n (дата звернення: 28.11.2025).

5. Binary search algorithm. URL: https://upload.wikimedia.org/wikiversity/en/e/ea/Binary_search_algorithm.pdf (дата звернення: 28.11.2025).

**УДК 004.422.5**

*Коваль В. С., здобувач 2 курсу спеціальності*
*122 Комп'ютерні науки,*
*Фриз І. В., канд. фіз.-мат. наук,*
*старший викладач кафедри інформаційних технологій*

## РОЛЬ АЛГОРИТМІВ ТА СТРУКТУР ДАНИХ У СУЧАСНИХ ІТ-СИСТЕМАХ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Алгоритми та структури даних становлять основу будь-якого програмного забезпечення і визначають ефективність роботи інформаційних систем. Алгоритм є формальним описом послідовності дій, необхідних для розв'язання певної задачі, тоді як структура даних визначає спосіб організації, зберігання та обробки інформації. Саме від поєднання цих двох компонентів залежить продуктивність, масштабованість і надійність програмних рішень.

Сучасні структури даних поділяють на базові та абстрактні. До базових належать числа, логічні значення, символи, а до абстрактних – масиви, списки, стеки, черги, дерева, хеш-таблиці, графи. Масиви забезпечують швидкий доступ за індексом та часто використовуються у задачах сортування. Зв'язні списки ефективні під час частих вставок та видалень. Стек застосовується для зберігання історії викликів, реалізації рекурсії та механізмів відміни дій, тоді як черги актуальні у системах планування процесів та потокової обробки даних. Дерева та графи є основою алгоритмів пошуку, оптимізації маршрутів, роботи компіляторів та навігаційних систем.

Алгоритми, що працюють із різними структурами даних, оцінюються за допомогою нотації Big O, яка дає змогу визначити час виконання та обсяг використаної пам'яті залежно від розміру вхідних даних. Наприклад, алгоритм сортування злиттям має складність $O(n \log n)$, тоді як наївний пошук у масиві – $O(n)$. Порівняння таких характеристик дає змогу обирати найбільш ефективні методи для вирішення конкретних задач.

Важливою частиною аналізу алгоритмів є визначення компромісу між швидкодією та ресурсами пам'яті. Деякі структури даних забезпечують високу швид-

кість доступу (наприклад, хеш-таблиці), але потребують додаткової пам'яті. Інші, навпаки, є компактними, але повільнішими. Тому програміст повинен обирати структуру даних не лише за формальними характеристиками, а й з урахуванням особливостей прикладної задачі.

У сучасних мовах програмування реалізовано широкий спектр готових структур даних:

– Python – списки, словники, множини, deque, heapq;
– Java – колекції List, Set, Map, дерева, черги;
– C++ STL – вектори, списки, черги, множини, пріоритетні черги;
– C# – List, Dictionary, Stack, Queue, LinkedList.

Попри наявність готових інструментів, розробник повинен розуміти внутрішню архітектуру цих структур, щоб уникати неефективних рішень, наприклад, частого розширення масивів чи неправильного використання списків.

Алгоритми та структури даних залишаються незамінними у критичних галузях, зокрема у системах реального часу, штучному інтелекті, комп'ютерній графіці, криптографії, машинному навчанні. Від коректності їх проєктування залежить надійність та швидкодія всієї системи. Тому вивчення цих понять має велике значення для майбутніх фахівців у сфері інформаційних технологій.

### Список використаних джерел

1. Introduction to algorithms / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. 4th ed. Cambridge, MA: The MIT Press, 2022. 1312 p.
2. Skiena S. The Algorithm Design Manual. Springer. 2020. 810 p.
3. Sedgewick R., Wayne K. Algorithms. 4th ed. Boston: Addison-Wesley. 2011. 955 p.

**УДК 004.422.5**

*Головня С. О., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,*
*Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій*

## СТРАТЕГІЯ «РОЗДІЛЯЙ І ВОЛОДАРЮЙ»: КОНЦЕПТУАЛЬНА ОСНОВА ШВИДКИХ АЛГОРИТМІВ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Однією з найважливіших парадигм у сучасній комп'ютерній науці є стратегія «розділяй і володарюй». Цей підхід забезпечує концептуальну основу для створення ефективних алгоритмів, наприклад, сортування злиттям та швидке сортування, даючи змогу вирішувати складні обчислювальні задачі шляхом їх декомпозиції. Розуміння цієї стратегії вимагає аналізу її теоретичної структури, формальних методів синтезу та застосування у багатовимірних просторах.

У фундаментальній теорії алгоритмів метод «розділяй і володарюй» визначається як процес, що складається з трьох ключових етапів: розподілу, підко-