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-

рення та об’єднання. На етапі розподілу задача розбивається на кілька підзадач меншого розміру, які є подібними до початкової. Етап підкорення передбачає рекурсивне розв’язання цих підзадач; якщо ж розмір підзадачі доволі малий, вона розв’язується безпосередньо. Зрештою етап об’єднання формує підсумковий розв’язок на основі рішень усіх підзадач. Ефективність цього підходу часто оцінюється за допомогою рекурентних співвідношень, які описують час виконання алгоритму через час виконання його частин [1].

Для суворої побудови таких алгоритмів застосовуються формальні методи синтезу. Д. Р. Сміт пропонує розглядати структуру алгоритмів класу «розділяй і володарюй» через програмні схеми, де функціональність алгоритму виводиться з його структури та функціональності компонентів (операторів декомпозиції та композиції) [2]. Ключовим елементом тут є «сильний принцип редукції задачі», який стверджує: якщо вхідні дані коректно декомпозуються, підзадачі розв’язуються правильно, і результати коректно об’єднуються, то отриманий результат буде правильним розв’язком вихідної задачі [2]. Такий підхід дає змогу формально виводити специфікації для підалгоритмів, гарантуючи коректність усієї програми.

Потужність парадигми особливо яскраво проявляється під час розв’язання задач у багатовимірному просторі. Дж. Бентлі демонструє, що багатовимірний метод «розділяй і володарюй» дає змогу знизити часову складність багатьох геометричних задач (як-от пошук найближчої пари точок або побудова емпіричної функції розподілу) з наївного $O(N^2)$ до $O(N \log N)$ [3]. Алгоритм для k-вимірного простору будується рекурсивно: задача для N точок у k-вимірному просторі зводиться до двох задач для $N/2$ точок у k-вимірному просторі та однієї задачі для N точок у $(k-1)$-вимірному просторі [3]. Ця рекурсія за розмірністю дає змогу ефективно уникати комбінаторного вибуху складності.

Отже, стратегія «розділяй і володарюй» є не просто набором евристик, а строгою математичною концепцією. Вона поєднує в собі чітку алгоритмічну структуру [1], можливості формального синтезу програм [2] та високу ефективність під час обробки багатовимірних даних [3], що робить її незамінним інструментом у сучасному алгоритмічному проєктуванні.

Список використаних джерел

1. Introduction to algorithms / C. Stein et al. Cambridge, MA: The MIT Press. 2009. 1292 p.

2. Smith D. R. The design of divide and conquer algorithms. Science of computer programming. 1985. Vol. 5. P. 37–58. URL: https://doi.org/10.1016/0167-6423(85)90003-6 (дата звернення: 30.11.2025).

3. Bentley J. L. Multidimensional divide-and-conquer. Communications of the ACM. 1980. Vol. 23, № 4. P. 214–229. URL: https://doi.org/10.1145/358841.358850 (дата звернення: 30.11.2025).