

*Киселевич Володимир,
здобувач другого (магістерського) рівня вищої освіти
фізико-математичного факультету
Іванов Дмитро,
доктор технічних наук, доцент,
професор кафедри комп'ютерних наук та інформаційних технологій,
Житомирський державний університет імені Івана Франка,
м. Житомир, Україна*

ВИКОРИСТАННЯ НОТАЦІЇ О-ВЕЛИКЕ У РОЗРОБЦІ ІНФОРМАЦІЙНИХ СИСТЕМ

У сучасній розробці програмного забезпечення нотація О-велике є невід'ємним інструментом для оцінки ефективності алгоритмів котрий дозволяє зрозуміти, як змінюється ефективність програмного коду зі збільшенням обсягів даних. Втім, більшість розробників, навіть не усвідомлюючи, використовують О-велике у повсякденній роботі при проектуванні усіх рівнів системи, фактично оперуючи поняттями, які базуються на нотації О-велике.

Традиційно О-велике застосовують для аналізу простих алгоритмів – сортування, пошуку, роботи зі структурами даних тощо. Однак потенціал даної нотації виходить далеко за межі пересічних завдань. О-велике може стати ефективним інструментом для моделювання, аналізу та оптимізації складних

Секція 4. Технології розробки інформаційних систем

інформаційних систем. Наприклад, його можна використовувати для оцінки швидкодії різних підсистем, таких як: бази даних, мережева комунікація чи навіть взаємодію користувачів із системою.

Мета статті – дослідити новий погляд на проектування та розробку інформаційних систем через призму нотації *O*-велике. Проаналізуємо переваги та недоліки даного підходу, а також розглянемо застосування на різних рівнях системи.

Для початку розглянемо основні положення нотації *O*-велике. Нотація *O*-велике (Big O) – це спеціальна нотація, яка описує, наскільки швидкий алгоритм [1, с.33]. Дана нотація дозволяє розробникам описати, як змінюється час виконання або обсяг споживаної пам'яті в залежності від розміру вхідних даних

Основою нотації є опис зростання складності алгоритму через його найзначущіший компонент. Наприклад, алгоритм із часовою складністю $T(n)=3n^2+5n+2$ оцінюється як $O(n^2)$, оскільки із збільшенням n саме компонент n^2 має найбільшим вплив та є домінуючим [2].

Розглянемо основні види алгоритмічної складності [1, с.38], а також їх час виконання в залежності від кількості вхідних параметрів на Рис. 1:

- $O(1)$ — константна: час виконання не залежить від розміру вхідних даних;
- $O(\log n)$ — логарифмічна: кількість операцій зростає повільно зі збільшенням вхідних даних;
- $O(n)$ — лінійна: час виконання прямо пропорційний розміру вхідних даних;
- $O(n \log n)$ — лінійно-логарифмічна: поєднання лінійної та логарифмічної складності, часто зустрічається в сортуванні;
- $O(n^2)$ — квадратична: час виконання пропорційний квадрату розміру вхідних даних;
- $O(2^n)$ — експоненціальна: час виконання зростає експоненційно з розміром вхідних даних;
- $O(n!)$ — факторіальна: час виконання зростає факторіально зі збільшенням розміру вхідних даних.

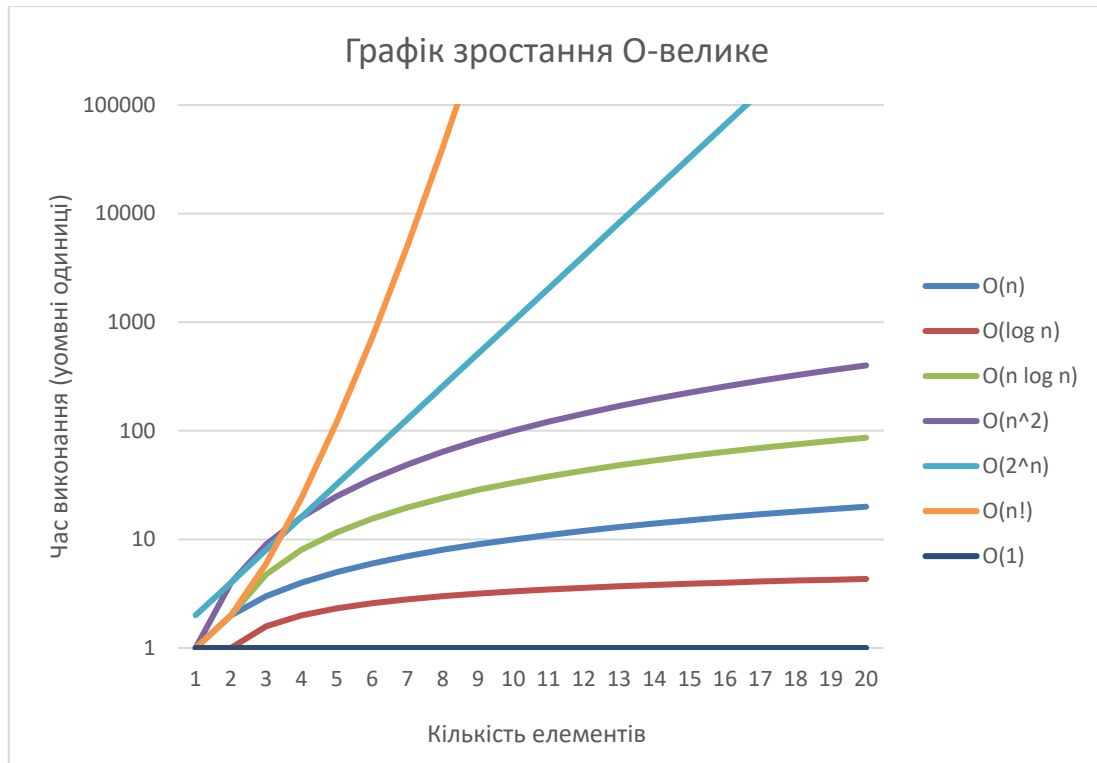


Рис. 1 Графік зростання O-велике

Отже, використовуючи попередньо зазначені положення, розглянемо як визначити O-велике для кожного рівня системи.

Визначення O-великого базових процесів. На найнижчому рівні системи нотація O-велике використовується для оцінки базових операцій і процесів, що виконуються в системі. Цей рівень охоплює ключові алгоритми, взаємодію з базами даних, мережеву комунікацію, операції вводу/виводу тощо, які є основою для роботи системи. Наприклад, у базах даних запит на отримання інформації з індексованої таблиці визначається як $O(\log n)$, де n — кількість записів, тоді як відсутність індексації матиме складність $O(n)$. Алгоритми, які використовуються для обробки даних, такі як сортування $O(n \log n)$ або лінійний пошук $O(n)$, є ключовими для розуміння швидкодії програмного коду на даному рівні. Аналогічно, операції, пов'язані з мережею чи файловою системою, мають свої показники складності, наприклад, $O(d)$ для мережевого запиту, де d — час затримки, або $O(f)$ для читання файлу розміром f .

Визначення O-великого на рівні модулів. Рівень модулів передбачає агрегацію O-великого базових процесів у загальне O-велике кожного модуля, що в свою чергу, дозволяє оцінити швидкодію модуля як ізольованої частини системи, враховуючи всі внутрішні операції та процеси, які він виконує.

Наприклад, уявімо модуль, котрий аналізує дані, обробляючи масиви інформації, комунікуючи зі сторонніми API та виконуючи обчислення. Складність цього модуля може складатися з кількох компонентів: читання вхідних даних із джерела $O(\log n)$, де n — кількість записів, виконання фільтрації через базові алгоритми $O(f)$. У результаті загальна складність модуля за правилом агрегування визначається як лінійна $O(n)$.

Визначення O-великого взаємодії ланцюжка модулів. Розглянемо визначення O-великого ланцюжка взаємодії модулів один з одним для визначення слабкої

Секція 4. Технології розробки інформаційних систем

ланки. Професійні інструменти моніторингу далеко не завжди дозволяють виявити джерело проблем, оскільки вони переважно фокусуються на низькорівневих аспектах, таких як: робота баз даних чи мережевої комунікації, в той час, коли саме процес взаємодії між модулями залишається поза полем.

Розглянемо приклад визначення O -великого умовного ланцюжка взаємодії трьох модулів, які взаємодіють один з одним та для котрих попередньо було визначено їх складність на основі їх внутрішніх процесів та алгоритмів: компонент A із агрегованою складністю $O(n)$, компонент B із $O(\log n)$ та компонент C із $O(n^2)$. Повний цикл взаємодії даних можна визначити як суму їхніх O -велике: $O(n) + O(\log n) + O(n^2)$. Аналіз цієї сумарної складності дозволяє зрозуміти, як кожен компонент впливає на загальну швидкодію системи.

Таким чином, використання нотації O -велике на рівні ланцюжка взаємодії модулів надає можливість виявити вразливі ланки, які уповільнюють увесь ланцюжок взаємодії.

Якщо розглядати переваги та недоліки, то насамперед, даний підхід дозволяє ще на етапі проектування оцінити ефективність ключових компонентів, що в свою чергу, сприяє прийняттю ефективних архітектурних рішень, розподілу навантаження та раціонального використання ресурсів. Погляд через призму O -велике також дозволяє виявити вузли, котрі потребуватимуть масштабування в майбутньому та заздалегідь закласти для них необхідний фундамент.

Проте є й певні недоліки: у разі відсутності фізичного або семантичного поділу на модулі (абстракції), даний підхід буде неефективним. Окрім цього, визначення O -великого потребує від розробників ґрунтовного розуміння алгоритмів і процесів, що ускладнює використання цього методу у проєктах із обмеженими ресурсами чи недостатнім досвідом команди.

Таким чином, у даній статті було досліджено використання нотації O -велике в контексті розробки інформаційних систем. Оцінка складності операцій та алгоритмів за допомогою цієї нотації є важливим інструментом для покращення швидкодії. Однак, застосування O -великого не обмежується лише аналізом окремих алгоритмів, а може бути ефективно використано для оцінки складності систем та її модулів.

Даний підхід дозволяє оцінити складність ключових модулів ще на етапі проектування, що сприяє прийняттю оптимальних архітектурних рішень, раціонального використання ресурсів та підготовки до масштабування. Завдяки цьому підходу закладаються основи для створення систем, які залишатимуться ефективними навіть за умов зростання навантаження.

Таким чином, використання нотації O -велике в розробці інформаційних систем відкриває новий погляд на процес проектування, дозволяючи розробникам оцінювати складність і ефективність систем не лише з точки зору окремих алгоритмів, а й загальної архітектури та взаємодії її компонентів.

Список використаних джерел та літератури

1. Адітья Бхаргава. Грокаємо алгоритми: ілюстративний посібник для програмістів та допитливих. Київ : ArtHuss, 2023. 285 с.

Секція 4. Технології розробки інформаційних систем

2. Оцінка ефективності алгоритму. Національний технічний університет України «Київський політехнічний інститут». Київ, 2016. URL: <https://studfile.net/preview/5994725/page:2/> (Дата звернення: 18.11.2024)