

**Павлюковець А.В., магістр спеціальності 122 Комп'ютерні науки
Науковий керівник: Журба А.О., к.т.н., доцент кафедри інформаційних технологій
і систем**

(Український державний університет науки і технологій, м. Дніпро, Україна)

ОЦІНКА СКЛАДНОСТІ АЛГОРИТМІВ ТА ВИБІР ЕФЕКТИВНОГО АЛГОРИТМУ ДЛЯ ВИРІШЕННЯ ЗАДАЧ РІЗНИХ ТИПІВ

Для вирішення завдань у різних галузях застосовують різні алгоритми, і оцінка їхньої складності грає ключову роль. Оцінюючи складність алгоритмів, можна визначити, наскільки вони ефективні з точки зору витрат ресурсів, часу та обсягу необхідних даних. Після оцінки складності алгоритмів, важливо вибрати найбільш ефективний алгоритм для конкретної задачі. Це сприятиме оптимізації роботи програм та забезпечувати швидше вирішення завдань. Завдяки використанню ефективних алгоритмів, можна покращити продуктивність систем, зменшити час виконання окремих операцій та знизити витрати ресурсів [1].

В роботі проаналізовано алгоритми сортування вибором і вставками, написано їх програмну реалізацію та проведено їх емпіричний аналіз та порівняння ефективності. Алгоритм сортування - це алгоритм для впорядкування елементів в списку. Зі всіх задач програмування, сортування має найбагатший вибір алгоритмів розв'язку. Алгоритми сортування оцінюються за швидкістю роботи та за ефективністю використання пам'яті [2]. Час роботи (або обчислювальна складність алгоритму) – це основний параметр, що характеризує швидкодію алгоритму. При сортуванні важливо розуміти наявність найгіршого, найкращого та середнього випадків в обробці конкретних даних. Це дає розуміння того, що не лише об'єм даних впливає на швидкість роботи алгоритму, але й їх зміст.

Пам'ять - це виділення додаткової пам'яті під тимчасове зберігання даних для ряду алгоритмів сортування. При оцінці не враховується місце, яке займає вихідний масив і незалежні від вхідної послідовності витрати, наприклад, на зберігання коду програми. Алгоритми сортування, що не споживають додаткової пам'яті, відносять до сортувань на місці.

Ефективність методів сортування визначається двома параметрами: кількістю порівнянь та кількістю переміщень елементів.

Різноманітність алгоритмів сортування вимагає деякої їхньої класифікації. Розрізняють наступні стратегії [2]:

- Стратегія вибірки. З вхідної множини вибирається наступний за критерієм впорядкованості елемент і включається в вихідну множину на місце, наступне за номером.
- Стратегія включення. З вхідної множини вибирається наступний за номером елемент і включається в вихідну множину на те місце, яке він повинен займати відповідно до критерію впорядкованості.
- Стратегія розподілу. Вхідна множина розбивається на ряд підмножин і сортування ведеться у середині кожної такої підмножини.

Стратегія злиття. Вихідна множина отримується шляхом злиття маленьких впорядкованих підмножин.

На рисунку 1 наведено залежність часу виконання алгоритмів сортування вибором та вставками від розміру вхідних даних. Виходячи з графіку залежності часу виконання алгоритму сортування вибором від розміру вхідних даних алгоритм відноситься до

квадратичного класу ефективності $O(N^2)$, а для сортування вставками – до лінійного класу ефективності $O(N)$.

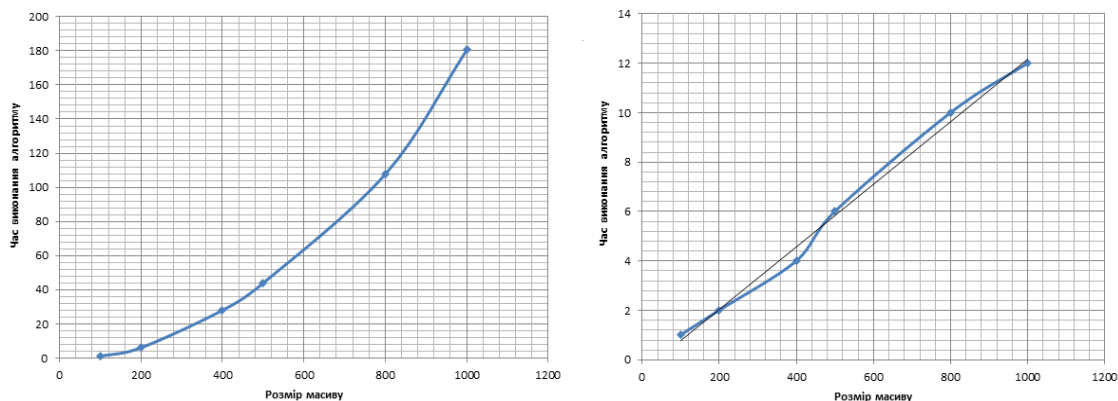


Рисунок 1 - Залежність часу виконання алгоритмів сортування вибором (ліворуч) та вставками (праворуч) від розміру вхідних даних

Проаналізувавши різні алгоритми сортування, можна назвати деякі чинники, які впливають на вибір алгоритму [2].

1. Наявний ресурс пам'яті. Чи повинні вхідна й вихідна множини розташовуватися в різних ділянках пам'яті чи вихідна множина може бути сформована на місці вхідної? В останньому випадку наявна ділянка пам'яті повинна в ході сортування динамічно перерозподілятися між вхідною і вихідною множинами; для одних алгоритмів це пов'язане з великими витратами, для інших – з меншими.

2. Початкова впорядкованість вхідної множини. У вхідній множині можуть попадатися впорядковані ділянки. У деяких випадках вхідна множина може виявитися вже впорядкованою. Одні алгоритми не враховують початкової впорядкованості і вимагають одного і того ж часу для сортування будь-якої множини даного обсягу, інші виконуються тим швидше, чим краще впорядкованість на вході.

3. Часові характеристики операцій. При визначенні порядку алгоритму час виконання вважається звичайно пропорційним кількості порівнянь ключів та їх переміщень. При цьому порівняння числових ключів виконується швидше, ніж символьних; операції переміщення, характерні для деяких алгоритмів, виконуються тим швидше, ніж менший об'єм записів. Залежно від характеристик запису таблиці може бути вибраний алгоритм, що забезпечує мінімізацію числа тих чи інших операцій.

4. Складність алгоритму. Ця характеристика також має важливе значення при виборі алгоритму. Простий алгоритм вимагає меншого часу для його реалізації і вірогідність помилки в його реалізації менше. При промисловому виготовленні програмного продукту вимоги дотримання термінів розробки і надійності продукту можуть навіть превалювати над вимогами ефективності функціонування.

Список використаних джерел:

1. Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Алгоритми та структури даних» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньою програмою «Інженерія програмного забезпечення у промисловості і бізнесі» спеціальності 121 Інженерія програмного забезпечення денної та заочної форм навчання [Електронне видання] / Журба А.О. – Дніпро: УДУНТ, ННІ «ІПБТ», 2021. – 62 с.

2. Седжвік Роберт. Фундаментальні алгоритми на C++. Аналіз / Структури даних / Сортування / Пошук: Пер. з англ./Роберт Седжвік. – К.: Видавництво «ДіаСофт», 2001. 688 с.