

ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ТА МЕТОДІВ СОРТУВАННЯ В СУЧАСНИХ МОВАХ ПРОГРАМУВАННЯ

Сортування даних є одним з завдань, які досить часто зустрічаються на практиці при обробці інформації [1]. Незважаючи на те, що в даний час відомо вже кілька десятків різних алгоритмів сортування, дослідження в цьому напрямку тривають.

Поява нових або вдосконалених алгоритмів сортування неодмінно приводить до зміни в реалізації методів або функцій сортування, які, як відомо, є в усіх сучасних мовах програмування. Але підходи до розробки методів сортування в різних мовах можуть істотно відрізнятися один від одного. Проведемо порівняльний аналіз методів сортування для двох найбільш поширених в даний час мов програмування - Java і C#.

Перші версії мови програмування C# використовували алгоритм швидкого сортування. Це один з найбільш ефективних алгоритмів сортування, але у нього є один істотний недолік - при невдалому виборі опорного елемента швидкість роботи алгоритму падає, і обчислювальна складність алгоритму стає квадратичною.

З розвитком фреймворка .NET і мови програмування C# робилися спроби вдосконалити процес сортування і в даний час в основу методу покладено оптимізоване інтроспективне сортування.

Алгоритм сортування Introsort (інтроспективне сортування) був розроблений в 1997 році Девідом Мюссера [2]. Цей алгоритм використовує швидке сортування і перекладається на пірамідальне сортування, коли глибина рекурсії перевищує деякий заздалегідь встановлений рівень. Цей підхід поєднує в собі переваги обох методів, навіть у гіршому випадку має обчислювальну складність $O(n \log n)$ і швидкодію, яку можна порівняти з алгоритмом швидкого сортування. Так як обидва алгоритми використовують порівняння, цей алгоритм також належить класу сортувань на основі порівнянь.

Тепер сортування в масивах є сумішшю сортувань: сортування вставками, швидкого сортування і пірамідального сортування. Використання такого підходу позитивно впливає на продуктивність методу та підвищує ефективність, оскільки в реальних задачах дані бувають частково впорядковані, а на таких даних, як відомо сортування вставками працює дуже швидко [2].

Незважаючи на деякі спільні риси в організації обробки даних, у плані сортування платформа Java досить сильно відрізняється від .NET. Однак, як і в .NET в Java алгоритм так само змінювався.

Алгоритм швидкого сортування, як відомо, є нестійким і це уповільнює його роботу при сортуванні посилальних типів. Для мови Java ця проблема посилюється, так як в цій мові всі дані представляються як об'єкти деяких класів. Тому для сортування посилальних типів використовується сортування злиттям. Дане сортування є стійким, навіть в гіршому випадку має обчислювальну складність $O(n \log n)$, але вимагає $O(n)$ додаткової пам'яті.

Оскільки проблема стійкості стосується тільки посилальних типів, для примітивів не має значення, міняються елементи з одним ключем чи ні. Тому для сортування примітивів Java використовує покращений алгоритм швидкого сортування - DualPivotQuicksort [2]. Звичайний Quicksort ділить масив на два відрізки, вибравши випадковий елемент P. Потім сортує масив так, щоб всі елементи менші P потрапили в перший відрізок, а решта - в другий. Потім алгоритм рекурсивно повторюється на першому і на другому відрізках. DualPivotQuicksort ділить масив на три відрізка, замість двох. В результаті кількість операцій переміщення елементів масиву істотно скорочується.

Починаючи з версії 7 в Java для сортування посилальних типів використовується алгоритм TimSort [2].

TimSort також є гібридним алгоритмом сортування, який поєднує сортування вставками і сортування злиттям. Тому, для посилальних типів даних TimSort істотно швидше багатьох алгоритмів сортування. У той же час, на випадкових даних, цей алгоритм поступається приблизно на 30 відсотків алгоритму швидкого сортування [2].

Таким чином можна зробити наступні висновки. Обидва методи ефективно вирішують задачу сортування, але в Java акцент зроблено на стійкість, на яку потрібно витратити пам'ять і час, а в .NET - на швидкість і економію пам'яті.

Список літератури

1. Алгоритмы: построение и анализ / [Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн]. – [3-е изд.]. – М. : Издательский дом «Вильямс», 2013. – 1328 с.
2. Сортировка в .NET [Электронный ресурс]. – Режим доступа к ресурсу: <https://habrahabr.ru/post/188012/>