УДК 004.422.8

Яковенко Т.П., старший викладач,
Генералов С. В., студент,
Полтавський національний технічний університет
імені Юрія Кондратюка

ПОРІВНЯННЯ ЧАСОВОЇ СКЛАДНОСТІ АЛГОРИТМІВ СОРТУВАННЯ ВСТАВКАМИ, ЗЛИТТЯМ, ШВИДКОГО СОРТУВАННЯ.

Анотація *.У статті викладено результати роботи програми, створеної для порівняння часової складності алгоритмів сортування.*

Ключові слова: *алгоритм, часова складність, сортування, сортування вставками, сортування злиттям, швидке сортування.*

1 Постановка проблеми.

Метою порівняння складності алгоритмів сортування вставками, злиттям, швидкого сортування є визначення часу, який використовують дані алгоритми для розв’язання певної задачі, так як при використанні їх у великих проектах невдалий вибір алгоритму може погіршити швидкодію програми.

2 Виклад основного матеріалу.

Часова складність алгоритму - це функція розміру вхідних і вихідних даних, що дорівнює максимальній кількості елементарних операцій, які проробляються алгоритмом для розв’язання екземпляру задачі зазначеного розміру. Також розглядають поняття середнього часу роботи алгоритму, тобто математичне очікування часу роботи алгоритму. Іноді говорять просто: «часова складність алгоритму» або «час роботи алгоритму», маючи на увазі часову складність алгоритму в гіршому, найкращому або середньому випадку (в залежності від контексту).

Сортування вставками – досить простий алгоритм. Як в і будь-якому іншому алгоритмі сортування, зі збільшенням розміру сортованого масиву збільшується і час сортування. Основною перевагою алгоритму сортування вставками є можливість сортувати масив у міру його отримання, тобто маючи частина масиву, можна починати його сортувати. У паралельному програмування така особливість відіграє не останню роль.

Сортований масив можна розділити на дві частини – відсортована частина і не відсортована. На початку сортування перший елемент масиву вважається відсортованим, всі інші – не відсортовані. Починаючи з другого елемента масиву і закінчуючи останнім, алгоритм вставляє невідсортований елемент масиву в потрібну позицію у відсортованій частині масиву. Таким чином, за один крок сортування відсортована частина масиву збільшується на один елемент, а не відсортована частина масиву зменшується на один елемент.

Сортування злиттям — алгоритм сортування, в основі якого лежить принцип «розділяй і володарюй». В основі цього способу сортування лежить злиття двох упорядкованих ділянок масиву в одну впорядковану ділянку іншого масиву. Злиття двох упорядкованих послідовностей можна порівняти з перебудовою двох колон солдатів, вишикуваних за зростом, в одну, де вони також розташовуються за зростом.

Швидке сортування (англ. QuickSort) — алгоритм сортування, добре відомий, як алгоритм розроблений Чарльзом Гоаром, який не потребує додаткової пам'яті і виконує у середньому O(nlogn) операцій. Однак у найгіршому випадку алгоритм робить $O(n^2)$ порівнянь. Оскільки алгоритм швидкого сортування використовує дуже прості цикли і операції, він працює швидше інших алгоритмів, що мають таку ж асимптотичну оцінку складності. Наприклад, зазвичай він більш ніж удвічі швидший порівняно з сортуванням злиттям.

Алгоритм швидкого сортування може бути реалізований як у масиві, так і в двозв'язному списку. Швидке сортування є алгоритмом на основі порівнянь, і не є стабільним.

В даній роботі було створено програму яка дозволяє порівняти алгоритми сортування за часом їх виконання. Для більшої наглядності програма будує для кожного алгоритму свій графік.

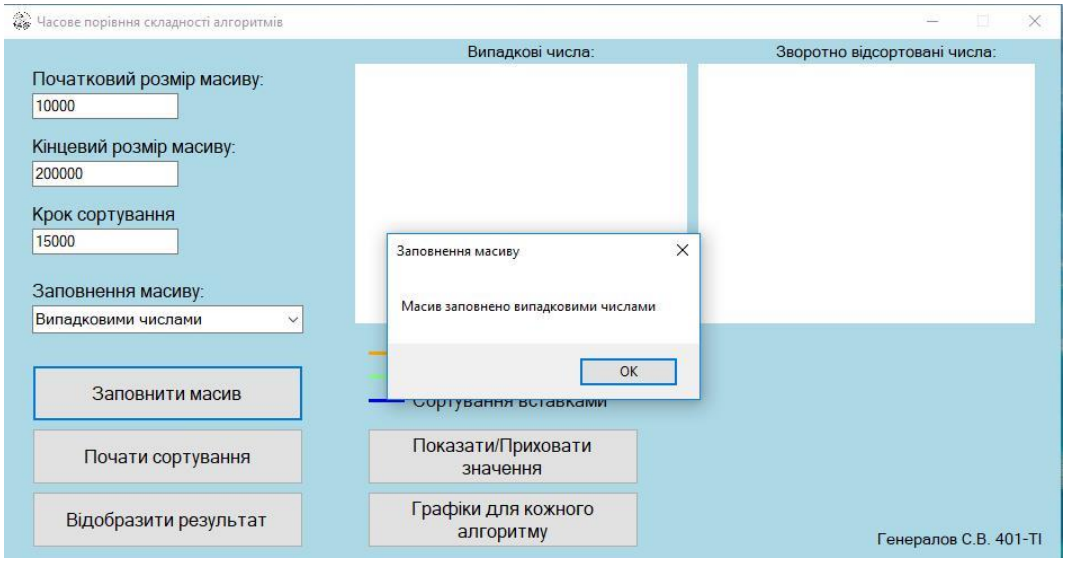

Рис.1 Заповнення масиву даними

Спочатку, як ми можемо бачити на рис.1., вибирається розмір вхідного масиву, крок його виконання, а також чим буде заповнено масив - випадковими чи зворотно відсортованими числами. Після цього масив заповнюється і програма може використовувати його для виконання сортування.

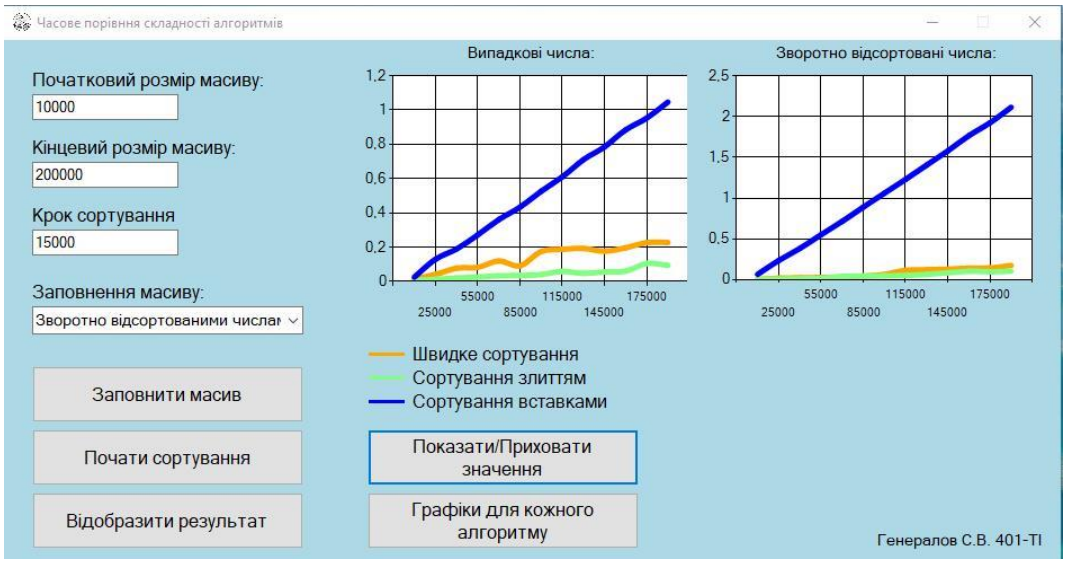

Рис.2 Побудовані графіки

Для сортування програма запускає процеси, кожен з яких відповідає певному алгоритму сортування. Всі процеси зчитують дані з масиву та виконують сортування над ним. Після того, як програма виконала сортування, ми можемо побачити графіки часу виконання алгоритмів сортування, які зображені на рис. 2. Тут ми можемо порівняти ці алгоритми між собою за часом виконання.

При апробації програми виконано сортування для масивів від 10000 до 200000. При сортуванні випадкових чисел найкращий результат показує алгоритм сортування злиттям, а алгоритм сортування вставками є найповільнішим, що відповідає попереднім розрахункам. При сортуванні зворотно відсортованих чисел алгоритми сортування злиттям та швидкого сортування працюють майже однаково та істотно не вповільнюються. Проте алгоритм сортування вставкою істотно вповільнився, так як для нього це є найгіршим випадком.

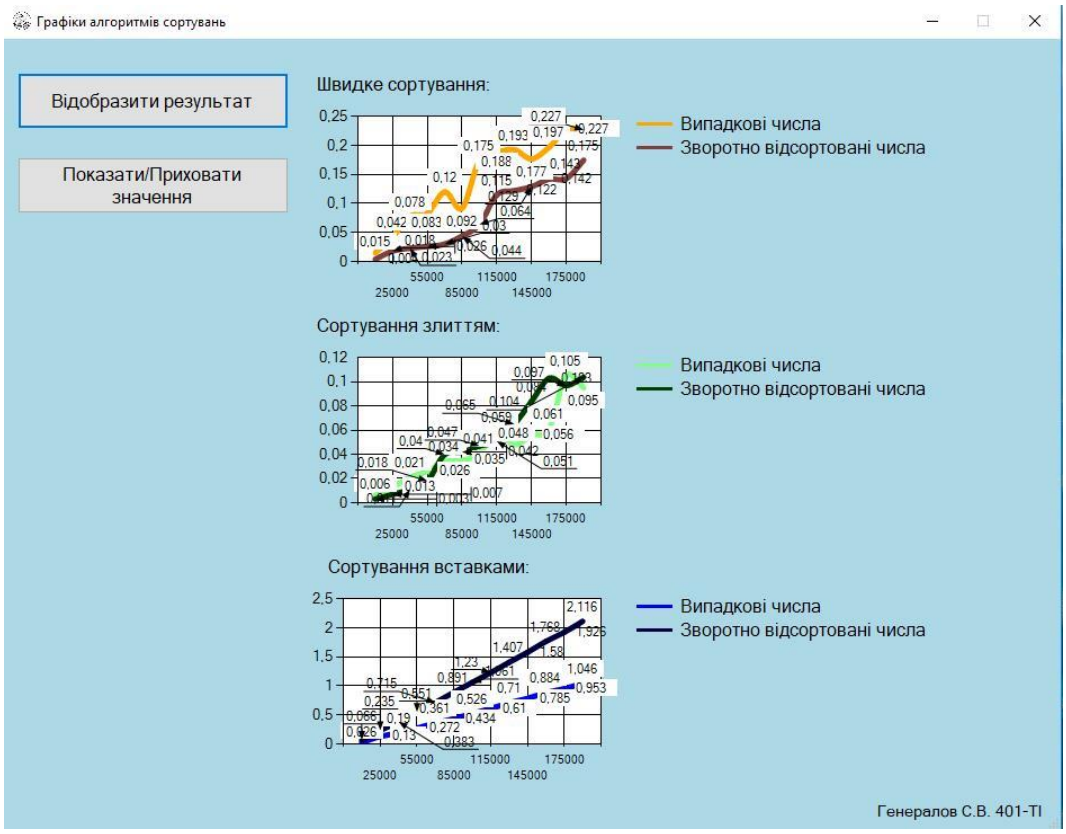

Рис.3 Графіки для порівняння алгоритмів сортування, побудовані програмою

Якщо є необхідність порівняти, як зміниться швидкість виконання певного алгоритму при сортуванні випадкових чисел або зворотно відсортованих, можна скористатися графіками для кожного алгоритму, зображеними на рис. 3.

Всі алгоритм вповільнюються при збільшенні вхідного розміру. Тому алгоритм сортування вставками є більш ефективним від інших при вхідних даних невеликого розміру. Алгоритми швидкого сортування та сортування злиттям є більш швидкими при сортуванні великих вхідних даних і згідно з нашими розрахунками час їх виконання приблизно однаковий і дорівнює $T(n) = (n \log n)$. Однак цей час для швидкого сортування може змінитися до $T(n) = O(n^2)$ при невірному виборі опорного елемента, а для цього необхідно знати які данні ми отримуємо. При цьому алгоритм сортування злиттям є найефективнішим серед представлених алгоритмів. Він не залежить від вхідних даних, на відміну від інших, і час його виконання навіть при «найгіршому випадку» буде $T(n) = (n \log n)$.

Отже, можна зробити висновок, що алгоритм сортування вставками ефективний при вхідних даних малої розмірності, алгоритм швидкого сортування слід використовувати, коли відомі вхідні дані та можна вибрати вірний опорний елемент. Якщо невідома розмірність вхідних даних та самі вхідні дані, слід використовувати алгоритм сортування злиттям.

Посилання

1. *Сергиенко И. В., Каспшицкая М. Ф. Модели и методы решения на ЭВМ комбинаторных задач оптимизации. – К.: Наук. думка, 1981. – 288 с.*

2. *Сергієнко І. В. Інформатика в Україні: становлення, розвиток, проблеми. – К.: Наук. думка , 1999. – 354 с.*

3. *Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн,Клиффорд. Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ. — М. :Издательский дом "Вильяме", 2005. — 1296 с.*

4. *Оценка сложности алгоритмов - Режим доступу: https://habrahabr.ru/post/104219/ 16.05.2017*

5. *Введение в анализ сложности алгоритмов - Режим доступу: https://habrahabr.ru/post/196560/ 16.05.2017*

6. *Швидке сортування – Режим доступу: http://www.kytok.org.ua/?p=347 16.05.2017*

7. *Лекция №6 Сортировка слиянием – Режим доступу http://iproc.ru/parallel-programming/lection-6/ 16.05.2017*

8. *Быстрая сортировка – Режим доступу https://ru.wikipedia.org/wiki/Быстрая_сортировка 16.05.2017*

Authors:
Iakovenko T.P., Generalov S.V.

Comparison of the time complexity of insertion, merge and quicksort sorting algorithms

Abstract. In this article we present an output of a program created to compare the time complexity of sorting algorithms.

Keywords: algorithm, time complexity, sorting, insertion sorting, merge sorting, quick sorting.

Авторы:
Яковенко Т.П., Генералов С. В.

Сравнение временной сложности алгоритмов сортировки вставками, слиянием, быстрой сортировки

Аннотация. В статье изложены результаты работы программы, созданной для сравнения временной сложности алгоритмов сортировки.

Ключевые слова: алгоритм, временная сложность, сортировка, сортировка вставками, сортировка слиянием, быстрая сортировка.