

А. В. Снігур
М. М. Підгорний
Д. В. Куклій
С. А. Велянський

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОЦІНКИ ПРИДАТНОСТІ КОМП'ЮТЕРА ДЛЯ ПЕВНОГО ВИДУ СОРТУВАННЯ НА ОСНОВІ АНАЛІЗУ ШВИДКОСТІ СОРТУВАННЯ

Вінницький національний технічний університет

***Анотація.** Комп'ютерна програма «Методи сортування, їх аналіз та порівняння» - це інструмент, призначений для порівняння та аналізу різних алгоритмів сортування. Аналіз відбувається на основі трьох масивів, які мають однаковий набір даних: згенерований випадковими числами, відсортований масив повністю та відсортований на половину. Відсортований масив на половину означає, що у сортуванні приймали участь усі числа і коли 50 відсотків чисел знаходяться на своєму місці сортування припиняється. Сортування відбувається від меншого до більшого числа. Програма рахує кількість порівнянь елементів масиву та кількість перестановок елементів..*

Ключові слова: сортування, порівняння швидкодії, масив.

***Abstract.** The computer program "Sorting methods, their analysis and comparison" is a tool designed to compare and analyze different sorting algorithms. The analysis is based on three arrays that have the same data set: a random number generated array, a fully sorted array, and a half-sorted array. A half-sorted array means that all numbers participated in the sorting and when 50 percent of the numbers are in place, the sorting stops. Sorting is from the smallest to the largest number. The program counts the number of comparisons of array elements and the number of permutations of elements.*

Keywords: sorting, speed comparison, array.

Вступ

Швидкість сортування масивів є важливим фактором при роботі з великими обсягами даних. Існує багато алгоритмів сортування, кожен з яких має свої переваги та недоліки, а також різну складність в обчислювальній та часовій площинах. Наприклад, алгоритм сортування бульбашкою є простим у реалізації, але має складність $O(n^2)$, що робить його неефективним для великих масивів даних. З іншого боку, алгоритм QuickSort має складність $O(n \cdot \log n)$, що робить його більш ефективним для великих масивів.

Важливо враховувати не лише складність алгоритмів, але й розмір масиву, кількість його елементів та їх тип даних. Наприклад, алгоритми сортування, що використовують порівняння елементів, можуть бути повільними для сортування масивів з великою кількістю даних, тоді як алгоритми сортування, які використовують не порівняння, а знаходження місця розташування елементів, можуть бути швидшими для великих масивів.

Отже, при виборі алгоритму сортування масивів важливо розглянути кількість даних, тип даних та час, який доступний для сортування. Тільки враховуючи ці фактори, можна вибрати найефективніший алгоритм сортування, який забезпечить оптимальну швидкість та ефективність роботи з масивами даних.

Результати дослідження

Постановка задачі

Постановка задачі "швидкість сортування масивів" включає в себе дослідження різних алгоритмів сортування та їхньої ефективності у вирішенні завдань сортування масивів. Метою дослідження є з'ясування, які алгоритми сортування є найбільш швидкими та ефективними у різних умовах, таких як розмір масиву, тип даних та їх розподіл, а також архітектура комп'ютерної системи.

Для досягнення цієї мети необхідно виконати наступні завдання:

1. Обрати набір алгоритмів сортування, які будуть порівнюватись. Цей набір буде включати алгоритми, такі як: quicksort, mergesort, heapsort, insertion sort, selection sort, Shell sort, shaker sort, combination sort.
2. Розробити тести для вимірювання часу виконання кожного алгоритму на різних розмірах масивів з однаковим типом даних та їх розподілом. Тести можуть бути засновані на випадкових даних, відсортованих даних та відсортованих на половину даних.
3. Виконати тести для кожного алгоритму та зібрати статистичні дані щодо часу виконання. Статистичні дані повинні включати час виконання, кількість порівнянь елементів та кількість переміщень елементів.
4. Проаналізувати статистичні дані та порівняти ефективність кожного алгоритму. Оцінити, які алгоритми є найбільш ефективними на різних розмірах масивів та типах даних, і чи існує залежність між швидкістю сортування та характеристиками даних.
5. Зробити висновки та рекомендації

Структура комп'ютерної програми

Перед розробкою програми, необхідно представити задачу у вигляді необхідних структурних блоків (див. рис. 1):



Рис. 1 Структура програми для порівняння методів сортування

Ці структурні блоки відповідають узагальненому алгоритму роботи програми. Розробка цього алгоритму входить до відповідного етапу повної побудови алгоритму.

Опис основних функцій програмного засобу, та його тестування

При запуску програми відкривається інтерфейс взаємодії з програмою. Є такі елементи управління:

1. Поле “Кількість елементів в масиві” - заповнюється числом, означає розмір масиву, на якому буде проходити дослідження;
2. Два поля “Границі генерируємих чисел” - перше поле - мінімальна границя, друге - максимальна границя;
3. Чек-бокс “Використовувати усі методи” - якщо активовано, то програма запустить усі методи сортування;
4. Кнопка “Старт” - запускає процес дослідження
5. Кнопка “Стоп” - примусово закінчує процес дослідження та вимикає програму
6. Вісім блоків з вихідними даними: час виконання сортування, кількість порівнянь, елементів, кількість переміщень елементів

Кількість символів в масиві		5000		Використовувати усі методи		<input checked="" type="checkbox"/>	
Границі генеруємих чисел		1 332					
Старт		Стоп					
<input checked="" type="checkbox"/> Sort by selection	00:00:00.0883233	Порівняння	Переміщення	<input checked="" type="checkbox"/> Shaker sorting	00:00:00.0797556	Порівняння	Переміщення
Невідсортований масив	12497500	12497500	4979	Невідсортований масив	8441958	8441958	6327576
Відсортований масив	12497500	12497500	0	Відсортований масив	9998	9998	0
Відсортований на 1/2	12497500	12497500	2480	Відсортований на 1/2	2095475	2095475	1553893
<input checked="" type="checkbox"/> Sort by combination	00:00:00.0026948			<input checked="" type="checkbox"/> Sort by inserts	00:00:00.0276887		
Невідсортований масив	169813	169813	18428	Невідсортований масив	4999	4999	6327576
Відсортований масив	169813	169813	0	Відсортований масив	4999	4999	0
Відсортований на 1/2	169813	169813	7908	Відсортований на 1/2	4999	4999	1553893
<input checked="" type="checkbox"/> Shell sorting	00:00:00.0026370			<input checked="" type="checkbox"/> Merge sort	00:00:00.0029360		
Невідсортований масив	107376	107376	52359	Невідсортований масив	56778	56778	131319
Відсортований масив	55017	55017	0	Відсортований масив	37215	37215	131319
Відсортований на 1/2	74734	74734	19717	Відсортований на 1/2	46466	46466	131319
<input checked="" type="checkbox"/> Quick sorting	00:00:00.0018307			<input checked="" type="checkbox"/> Heapsort	00:00:00.3651985		
Невідсортований масив	21068	21068	20848	Невідсортований масив	10944543	10944543	2113224
Відсортований масив	11952	11952	11952	Відсортований масив	6249999	6249999	0
Відсортований на 1/2	16000	16000	15907	Відсортований на 1/2	10861599	10861599	1553022

Висновки

В ході проведеної роботи було створено комп'ютерну програму для дослідження швидкодії методів сортування і при 5000 елементів та діапазоном від 1 до 332 найефективніший метод сортування - швидке сортування, найгірший - сортування кучами. Але цей інструмент необхідно використовувати для більш глибоко та детального дослідження на різних розмірах, діапазонах та комп'ютерах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
2. "Sorting and Searching" by Donald E. Knuth
3. "Data Structures and Algorithms in Python" by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser
4. "The Art of Computer Programming, Volume 3: Sorting and Searching" by Donald E. Knuth
5. Колмогоров А. Н. Три подхода к определению понятия «Количество информации»

Снігур Анатолій Васильович - к.т.н., доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: algorithms530@gmail.com

Підгорний Максим Максимович - студент групи ІСП-196, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: maksonpatiphone@gmail.com

Куклій Данило Вячеславович - студент групи ІСП-196, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: danylokukliy@gmail.com

Велянський Сергій Андрійович - студент групи ІСП-196, факультет

інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: sergiy.velyanskiy48@gmail.com

Snigur Anatoliy Vasyliovych - Ph.D., Associate Professor of the Computer Engineering Department, Vinnytsia National Technical University, Vinnytsia

Pidhornyi Maksym Maksymovych - student of group 1SP-19b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: maksonpatiphone@gmail.com

Kuklii Danylo Viacheslavovych - student of group 1SP-19b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: danylokukliy@gmail.com

Velianskiy Serhii Andriiovych - student of group 1SP-19b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: sergiy.velyanskiy48@gmail.com