

*Чулюк В. В., здобувач вищої освіти,
Якубич К. А., асистент кафедри
інформаційних технологій*

СУТНІСТЬ ТА ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МЕТОДУ СОРТУВАННЯ ВСТАВКОЮ

Донецький національний університет імені Василя Стуса, м. Вінниця

Для структуризації та впорядкування різноманітних масивів даних і для їх подання у певному вигляді використовуються алгоритми сортування. Для сортування даних було розроблено безліч алгоритмів, що мають як свої переваги, так і недоліки. З розвитком розподілених систем та паралельних обчислень розвиваються і алгоритми з використанням цих технологій. Такий підхід вдосконалює та підвищує ефективність самих алгоритмів з використанням паралельних технологій. Виникає необхідність порівняння ефективності алгоритмів, що використовують технології розподілених систем та паралельних обчислень. Також можливість порівняння ефективності алгоритмів є цікавою та пізнавальною в навчальному процесі підготовки ІТ-спеціалістів [1].

Відомо багато методів сортування масиву, що відрізняються швидкістю й обсягом оперативної пам'яті, яка під час цього використовується. Серед цих методів можна виділити методи внутрішнього та зовнішнього сортування.

Методи внутрішнього сортування не передбачають використання допоміжних масивів. Ці методи застосовують до масивів, що повністю розташовані в оперативній пам'яті.

Методи зовнішнього сортування застосовують до великих масивів даних, які зберігаються на зовнішніх носіях. Методи внутрішнього сортування прийнято поділяти на дві групи: елементарні (прямі) та удосконалені методи [2]. Найбільш відомий елементарний метод сортування масиву – сортування вставкою (включенням).

Цей метод полягає у тому, що кожен елемент послідовно вставляється у вже відсортований список. Розмір уже відсортованого списку спочатку дорівнює одиниці. Алгоритм сортування вставкою забезпечує сортування перших k елементів після k -ї ітерації. У результаті відсортована частина збільшується на один елемент, а невідсортована – на один елемент зменшується [3].

Отже, на кожному кроці алгоритму сортування методом вставки треба виконати дві операції:

- пошук позиції для вставки елемента в лівій (відсортованій) частині;
- власне його вставку з подальшим зсувом на одну позицію вправо від елементів відсортованої частини.

Програма нижче моделює роботу сортування масиву методом вставки та виводить задіяну кількість операцій, використовуючи мову програмування C#, код демонструє структуру цього методу, реалізованого за допомогою циклів «for» та «while».

```

1 {} using System;
2
3 class Program
4 {
5     // Алгоритм сортування вставками (Insertion Sort)
6     static void InsertionSort(int[] arr)
7     {
8         int n = arr.Length;
9         int operations = 0; // Лічильник операцій
10
11         for (int i = 1; i < n; i++) // Починаємо з другого елемента
12         {
13             int key = arr[i]; // Поточний елемент, який потрібно вставити
14             int j = i - 1;
15
16             // Зсуваємо елементи вправо, щоб знайти правильне місце для вставки
17             while (j >= 0 && arr[j] > key)
18             {
19                 operations++; // Кожне порівняння додається до кількості операцій
20                 arr[j + 1] = arr[j];
21                 j--;
22             }
23
24             arr[j + 1] = key; // Вставляємо елемент у правильну позицію
25             operations++;
26         }
27
28         // Виводимо результат
29         Console.WriteLine($"Метод сортування вставкою: {string.Join(", ", arr)}");
30         Console.WriteLine($"Кількість операцій: {operations}\n");
31     }

```

Рисунок 1 – Код роботи сортування масиву методом вставки

```

34 // Головна функція
35 static void Main()
36 {
37     int[] array = { 50, 80, 90, 40, 20, 70, 55, 10, 85, 30 };
38
39     Console.WriteLine("Початковий масив: " + string.Join(", ", array) + "\n");
40
41     // Метод сортування вставкою
42     int[] insertionArray = (int[])array.Clone();
43     InsertionSort(insertionArray);
44 }
45 }
46

```

Рисунок 2 – Продовження коду

```

Початковий масив: 50, 80, 90, 40, 20, 70, 55, 10, 85, 30

Вдсортований масив методом вставки: 10, 20, 30, 40, 50, 55, 70, 80, 85, 90
К-ксть операцій: 36

```

Рисунок 3 – Результат коду

Цей код демонструє алгоритм сортування вставкою у мові програмування C#. Він ілюструє, як елементи по черзі вставляються у правильне місце від-

сортованої частини масиву. Алгоритм підходить для невеликих масивів через свою простоту, але для великих масивів він може працювати повільно через високу кількість операцій у випадку невпорядкованих даних. Сортування методом вставки відіграє фундаментальну роль у багатьох сферах програмування та інформаційних технологій. Це базовий, надійний і простий у реалізації алгоритм, який знаходить застосування в широкому спектрі задач – від навчальних прикладів до складних систем обробки даних.

Список використаних джерел

1. Програмна реалізація та дослідження алгоритмів паралельного швидкого сортування / В. О. Денисюк, Н. А. Потапова, О. В. Зелінська, М. Б. Тарасюк. *Вісник Хмельницького національного університету. Технічні науки*. 2023. № 4. С. 95–105. URL: http://journals.khnu.km.ua/vestnik/?page_id=41 (дата звернення: 05.12.2024).
2. Ковалюк Т. В. Основи програмування: підручник. Київ: Вид-во BHV, 2005. 384 с.
3. GURU99. URL: <https://www.guru99.com/uk/insertion-sort-algorithm.html> (дата звернення 05.12.2024).

УДК 004.42

*Щербина Д. С., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук., доцент,
доцент кафедри інформаційних технологій*

ЗАСТОСУВАННЯ АЛГОРИТМУ АЛЬФА-БЕТА ВІДСІКАННЯ ДЛЯ ОПТИМІЗАЦІЇ ШАХОВИХ РІШЕНЬ

Донецький національний університет імені Василя Стуса, м. Вінниця

З кожним роком шахи стають більш популярними. Вони виростили з простої гри у великий спорт, і сприяють розвитку мислення, оскільки поєднують стратегічне і тактичне планування з логічним аналізом. Хоча успіх у спорті прийшов не відразу, проте за шахами вже багато років слідкують галузі алгоритмічного програмування та штучного інтелекту. Цей інтерес сприяє розвитку шахових програм, які постійно вдосконалюються для того, щоб вказувати як на помилки досвідчених шахістів, так і для навчання новачків.

Однією з ключових задач, які насамперед мають виконувати різні шахові програми, є забезпечення швидкої і точної оцінки позиції, щоб обрати оптимальний хід або варіант. Одним з перших алгоритмів, який застосовувався для знаходження оптимального рішення, був мінімакс. Але він не є ефективним, оскільки потрібно перевірити всі варіанти ходів, що, особливо в шахах, є дуже ресурсомістким процесом. Тому щоб оптимізувати цей алгоритм, було розроблено алгоритм альфа-бета відсікання. Цей метод, обрізаючи непотрібні гілки дерева рішень, дає змогу скоротити час та ресурси на пошук оптимального рішення [1].

Алгоритм альфа-бета відсікання відрізняється від мінімаксу тим, що він використовує не повний перебір усіх варіантів, а додає обмеження, які називаються альфа-бета обмеженнями.