

УДК 004.422.5.

*Ліхоткіна В. І., здобувачка вищої освіти,
Потапова Н. А., канд. екон. наук, доцент,
доцент кафедри інформаційних технологій*

СУТНІСТЬ ТА СКЛАДНИКИ РЕКУРСИВНИХ АЛГОРИТМІВ У ПРОГРАМУВАННІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Термін «рекурсія» зустрічається у багатьох галузях науки і в більшості випадків має значення способу визначення множини об'єктів або функції через себе, з використанням раніше заданих значень. Рекурсія є потужним інструментом для розв'язку численних задач, опису природних явищ та станів. У галузі комп'ютерних наук рекурсія є важливою частиною реалізації програмного продукту. У програмуванні під рекурсією розуміють таку реалізацію, в якій метод використовує у своєму тілі виклик самого себе. Такі виклики називають рекурсивними. Рекурсію іноді важче реалізувати у вигляді програмного коду, але це додає більшої гнучкості для розв'язання задач [1].

За допомогою рекурсії реалізованого багато алгоритмів, які базуються на принципі повторення однієї і тієї самої дії певну кількість разів, поки завдання не буде виконане. За допомогою рекурсії можна реалізувати метод «розділяй та володарюй». Складну проблему ми розбиваємо на підзадачі, які є меншими копіями самої себе. Кожну підзадачу вирішуємо таким самим способом, поки не дійдемо до найпростішого випадку. Потім об'єднуємо результати всіх підзадач і отримуємо розв'язання початкової задачі. Є певний тип, задач які краще розв'язувати рекурсивним методом через те, що це значно спрощує завдання. До такого типу належать такі задачі:

1. Використання рекурсивних методів у розрахунку чисел Фібоначчі. Числа Фібоначчі – це послідовність чисел, у якій кожне наступне число є сумою двох попередніх. Починається вона з чисел 0 і 1. Для швидкого знаходження наступних чисел цієї послідовності зручно використовувати рекурсивний метод. Рекурсія в цьому випадку полягає в тому, що повторюється дія додавання двох попередніх членів цієї послідовності для знаходження наступного. Рекурсія в програмуванні реалізовується за допомогою циклів, на рисунку 1 це показано на прикладі задачі про числа Фібоначчі. За допомогою циклу задача виконується, доки не виконується умова.

Програма, що показана на рис. 1, за допомогою рекурсивного методу мовою програмування C# розв'язує задачу, і результат показаний на рис. 1.

Суть роботи коду в тому, що під час кожного повтору використовуються два числа – (first) та (second). Після проходження циклу і виконання дій у тілі циклу ці числа змінюють свої значення для проходження наступного значення – число (first) стає (second), число (second) стає (next).

```

using System;

0 references
class Fibonacci
{
    0 references
    public static void Main()
    {
        // Введення перших двох значень ряду Фібоначчі
        Console.WriteLine("Введіть перше число ряду Фібоначчі: ");
        int first = int.Parse(Console.ReadLine());

        Console.WriteLine("Введіть друге число ряду Фібоначчі: ");
        int second = int.Parse(Console.ReadLine());

        Console.WriteLine("\nПерші 10 чисел ряду Фібоначчі :");

        // Виведення перших 10 чисел ряду Фібоначчі
        Console.WriteLine(first); // Перше число
        Console.WriteLine(second); // Друге число

        for (int i = 3; i <= 10; i++)
        {
            int next = first + second; // Обчислення наступного числа
            Console.WriteLine(next);

            // Оновлення значень для наступної ітерації
            first = second;
            second = next;
        }
    }
}

```

Рисунок 1 – Програмний код для реалізації задачі з числами Фібоначчі

2. Використання рекурсивних методів у задачах на графах. Рекурсивний метод для обходу бінарного дерева в прямому порядку (NLR – вузол, лівий піддерево, правий піддерево) працює на основі принципу «розділяй і володарюй», обробляючи кожну вершину дерева та її дочірні елементи по черзі [2].

3. Використання рекурсивних методів в алгоритмах сортування. Рекурсивні методи є основою роботи швидкого сортування та сортування злиттям, забезпечуючи ефективний розподіл задачі на підзадачі. У швидкому сортуванні рекурсія використовується для поділу масиву на дві частини відносно опорного елемента: всі менші елементи розташовуються зліва, а більші – справа. У сортуванні злиттям рекурсія використовується для поділу масиву на дві половини, які сортуються окремо. Після цього дві відсортовані половини зливаються у єдиний відсортований масив шляхом попарного порівняння елементів. Базовий випадок у сортуванні злиттям – це також масиви розміром один або порожні. Рекурсивна структура обох алгоритмів дає змогу ефективно розділяти та впорядковувати масиви, забезпечуючи логарифмічну глибину викликів. Рекурсивні методи часто використовують для реалізації завдань, розуміння рекурсії допомагає писати більш гнучкий та візуально кращий код, а також сприяє глибшому розумінню принципів програмування.

Список використаних джерел

1. Кублій Л. І. Алгоритми та структури даних. Основи алгоритмізації. Київ: Київський політехнічний інститут ім. Ігоря Сікорського, 2022. 528 с.
2. Грудзинський Ю. Є. Алгоритми та структури даних: навч. посіб. Київ: Київський політехнічний інститут ім. Ігоря Сікорського, 2022. 215 с.