3. Mathros. URL: https://www.mathros.net.ua/obhid-grafa-v-glybynu.html (дата звернення: 28.11.2023).

4. Biz. URL: http://www.ni.biz.ua/5/5_15/5_159663 (дата звернення: 28.11.2023).

УДК 004.021

Поліщук О. С., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,

Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

СУТНІСТЬ І СКЛАДНИКИ РЕКУРЕНТНИХ АЛГОРИТМІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Рекурсія є фундаментальним поняттям у комп'ютерних науках, що використовується для вирішення різноманітних завдань. Вона базується на ідеї функції, що викликає саму себе, і відіграє важливу роль у розробці алгоритмів та програм.

Рекурсія – процес повторення чого-небудь самоподібним способом. Наприклад, вкладені віддзеркалення, утворені двома точно паралельними одне одному дзеркалами, є однією з форм нескінченної рекурсії. Цей термін має більш спеціальні значення в різних галузях знань – від лінгвістики до логіки. Кількість вкладених викликів функції або процедури називається глибиною рекурсії.

Перевага рекурсивного визначення об'єкта полягає в тому, що таке вичерпне визначення теоретично здатне описувати нескінченно велику кількість об'єктів. Із допомогою рекурсивної програми можливо описати нескінченне обчислення, причому без явних повторень частин програми [1].

Рекурсивну програму завжди можна перетворити в нерекурсивну (ітеративну, з використанням циклів), яка виконує ті ж обчислення. І навпаки, використовуючи рекурсію, будь-яке обчислення, що припускає використання циклів, можна реалізувати, не застосовуючи цикли. Рекурентне співвідношення є рекурсивною функцією з цілочисельними значеннями. Значення будь-якої такої функції можна визначити, обчислюючи усі її значення, починаючи з найменшого, використовуючи на кожному кроці раніше обраховані значення для підрахунку поточного значення. Рекурентні вирази використовуються, зокрема, для визначення складності рекурсивних обчислень [2].

Властивості рекурентних алгоритмів передбачають [2]:

1. Самовиклик (Self-calling). Рекурсія полягає у виклику функції або алгоритму самим собою, що покладено в основну ідею рекурсивного підходу.

2. Базовий випадок (Base Case). Кожен рекурсивний алгоритм повинен мати базовий випадок, коли виклик функції просто повертає результат без нового рекурсивного виклику. Базовий випадок є умовою завершення рекурсії.

3. Повторюваність (Repetition). Рекурсивний алгоритм повинен робити прогрес у кожному новому виклику, зменшуючи або збільшуючи розмір задачі, щоб спростити вирішення.

4. Взаємодія зі стеком викликів (Call Stack Interaction). Кожен рекурсивний виклик додає новий елемент у стек викликів, і виклик функції вирішується у зворотному порядку, коли досягається базовий випадок або обробляється кожен рекурсивний виклик.

5. Зменшення задачі (Task Reduction). У кожному рекурсивному виклику задача повинна зменшуватися або наближатися до базового випадку, щоб рекурсія була ефективною.

6. Можливість використання для деревоподібних структур даних. Рекурсія особливо корисна під час роботи з деревоподібними або рекурсивними структурами даних, як-от бінарні дерева, графи тощо.

7. Потенційне переповнення стека. Рекурсивний підхід може призвести до переповнення стека, особливо за великої глибини рекурсії. Оптимізація, як-от хвостова рекурсія, може допомогти уникнути цього.

8. Читабельність коду. Рекурсивні алгоритми можуть бути більш читабельними та компактними, особливо коли вони добре підходять для розв’язання конкретних завдань.

9. Можливість заміни ітераційних конструкцій. Рекурсія може бути використана замість ітераційних конструкцій, але варто враховувати витрати на рекурсивні виклики та потенційне переповнення стека.

Оптимізація та нові підходи до використання рекурсії можуть привести до розробки більш ефективних та швидших алгоритмів. Застосування рекурсії в обробці та аналізі великих обсягів даних дає змогу проводити розробки нових алгоритмів для швидкого та ефективного вирішення завдань, пов’язаних з великими об’ємами інформації.

Список використаних джерел

1. Завіша В. В. Алгоритми і структури даних. URL: https://e-tk.lntu.edu.ua/pluginfile.php/20064/mod_resource/content/0/ Тема%209.%20Рекурсія.pdf

2. Завіша В. В. Алгоритмічні стратегії. URL: https://e-tk.lntu.edu.ua/pluginfile.php/20063/mod_resource/content/0/ Тема%2010.%20Алгоритмічні%20стратегії.pdf