УДК 004.2

Попова К. О., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ РЕКУРСИВНИХ ВІДНОШЕНЬ В АЛГОРИТМІЗАЦІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

Рекурсивною називається програма, яка звертається сама до себе. Особливістю рекурсивної програми є наявність умови завершення, оскільки вона не може викликати себе до нескінченності. З огляду на це, програма, яка містить рекурсію, повинна мати як мінімум два шляхи виконання, один з яких передбачає рекурсивний виклик, а другий – виконання програми без рекурсивного виклику.

Для реалізації цієї умови будь-яка рекурсивна процедура повинна містити базис і крок рекурсії.

Базис рекурсії – це пропозиція, що визначає якусь початкову ситуацію або ситуацію, яка відбувається у момент припинення. Зазвичай в цьому реченні записується якийсь найпростіший випадок, під час якого відповідь виходить відразу навіть без використання рекурсії.

Крок рекурсії – це правило, в тілі якого обов'язково міститься в якості підцілі, виклик обумовленого предиката. Параметри повинні змінюватися на кожному кроці так, щоб внаслідок цього спрацював або базис рекурсії, або умова виходу з рекурсії, розміщена в самому правилі.

У загальному вигляді правило, що реалізує крок рекурсії, буде виглядати так:

<ім'я правила рекурсії>:-
<список предикатів>, <предикат умови виходу>,
<список предикатів>, <ім'я правила рекурсії>,
<список предикатів>.

Реукурсивне відношення для обрахунку чисел факторіалу має вигляд:

Fact (1,1):- !. / * Умова припинення рекурсії * /

Fact (N, F):- N1 = N-1, fact (N1, F1), / * F1 дорівнює факторіалу числа, на одиницю меншого вихідного числа * /

F = F1 * N. / * Факторіал числа дорівнює добутку F1 на саме число *

Використання рекурсії пов'язане з обчисленням чисел Фібоначчі, які можна визначити так: перше і друге число дорівнюють одиниці, а кожне наступне число є сумою двох попередніх. Першим базисом є твердження, що перше число Фібоначчі дорівнює одиниці. Другий базис – аналогічне твердження про друге число Фібоначчі. Крок рекурсії: для обчислення числа Фібоначчі з номером N спочатку потрібно обчислити і додати числа Фібоначчі з номерами N-1 і N-2. Записати ці міркування можна так:

Fib (1,1): - !. / * Перше число Фібоначчі дорівнює одиниці * /
Fib (2,1): - !. / * Друге число Фібоначчі дорівнює одиниці * /
Fib (N, F): -
N1 = N-1, fib (N1, F1), / * F1 це N-1-е число Фібоначчі * /
N2 = N-2, fib (N2, F2), / * F2 це N-2-е число Фібоначчі * /
F = F1 + F2. / * N-е число Фібоначчі дорівнює сумі N-1-го і N-2-го чисел Фібоначчі *

Рекурентне співвідношення – це рівняння або нерівність, яка описує функцію із використанням самої себе, але тільки з меншими аргументами.

Під час розгляду рекурентних співвідношень зазвичай вводяться два спрощення. По-перше, приймається, що час роботи алгоритму $T(n)$ визначається лише для цілочислових n, оскільки в більшості алгоритмів кількість вхідних елементів виражається цілим числом. По-друге, оскільки час роботи алгоритму із вхідними даними фіксованого розміру виражається константою, то в рекурентних співвідношеннях для достатньо малих n зазвичай справедлива тотожність $T(n) = \Theta(1)$. Тому для зручності граничні умови рекурентних співвідношень зазвичай відкидаються, і приймається, що для малих n час роботи алгоритму є $T(n)$ константою. Методи вирішення рекурентних співвідношень:

1. Метод підстановки, який застосовується для вирішення рекурентних співвідношень, складається з двох етапів:

- робиться припущення про вигляд розв'язку;
- за допомогою методу математичної індукції визначаються константи та доводиться, що рішення правильне.

Метод підстановки можна використовувати для визначення або верхньої, або нижньої межі рекурентного співвідношення. В якості прикладу розглянемо верхню границю рекурентного співвідношення:

$$T(n) = 2T([n/2]) + n.$$

Розв'язок має вигляд $T(n) = O(n \times lgn)$. Метод полягає в доведенні того, що за придатного вибору константи $c>0$ виконується нерівність $T(n) \le c \times n\ lgn$.

2. Метод дерев рекурсії – прямий шлях до вдалого припущення. В дереві рекурсії кожний вузол представляє час, необхідний для виконання окремо взятої підзадачі, яка розв'язується за одного з багатьох рекурсивних викликів функції. Далі значення часу роботи окремих етапів підсумовується в межах кожного рівня, а потім – за всіма рівнями дерева, внаслідок чого отримуємо повний час роботи алгоритму.

Дерева рекурсії краще за все підходять для того, щоб допомогти зробити припущення про вигляд розв'язку, який потім перевіряється методом підстановок. Водночас у припущенні часто допускається наявність невеликих неточностей, оскільки воно потім все одно перевіряється. Якщо ж побудова дерева рекурсії та підсумування час роботи за всіма складниками відбувається ретельно, то саме дерево рекурсії може стати джерелом доведення коректності розв'язку. (рис. 1).

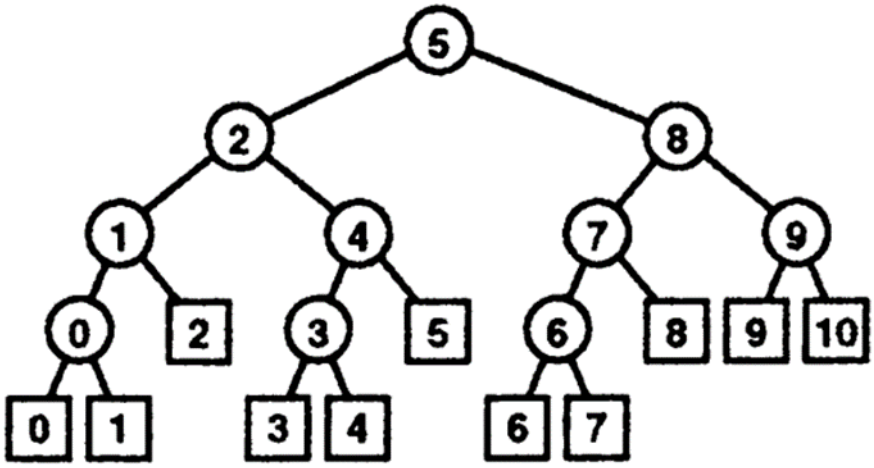

Рисунок 1. Дерево рекурсії

Отже, розглянуто поняття рекурсії та рекурсивних відношень. В алгоритмізації рекурсія є одною з найпотужніших технологій реалізації алгоритмів за принципом «розділяй та володарюй».

Список використаних джерел

1. Introduction to Algorithms / H. C. Thomas, E. L. Charles, L. R. Ronald, Clifford S. 3rd Edition. MIT PRESS, 2009. 1292 p.

2. Креневич А. Алгоритми та структури даних. Київ: ВПЦ Київський Університет, 2018. 172 с.

3. Мелешко Є. В., Якименко М. С., Поліщук Л. І. Алгоритми та структури даних. Кропивницький: Видавець – Лисенко В. Ф., 2019. 156 с.