УДК 519.87:004.942

Андрійченко К. А., студентка 3 курсу
спеціальності 113 «Прикладна математика»
Вєтров О. С., старший викладач
кафедри прикладної математики

АНАЛІЗ АЛГОРИТМУ «ПЕРЕБІР З ПОВЕРНЕННЯМ» НА ПРИКЛАДІ ЗАДАЧІ ПОБУДОВИ ЛАТИНСЬКОГО КВАДРАТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Наш час надав математиці минулого нові та серйозні цілі. Багато задач отримали практичний сенс, а латинські квадрати знайшли найрізноманітніше застосування у вирішенні доволі різних прикладних задач, таких, як: складання розкладів, вивчення впливу добрив на рослини, планування експериментів у медицині та біології, розподіл ресурсів, черговості ініціювання подій, видачі завдань студентам тощо. Також і в суміжних областях математики: в теорії груп та напівгруп, криптографії, комбінаториці, логіці, статистиці, теорії кодів тощо. На даний час латинські квадрати активно використовуються в дослідженні проблем скінченних геометрій.

Метою даної роботи є узагальнення теорії латинських квадратів, їх властивостей та методів побудування, використання здобутих знань для написання на одній з мов програмування алгоритму перебору з поверненням, що у свою чергу є актуальним алгоритмом для відповіді на запитання задач типу «Перерахуйте всі можливі варіанти…» або «Скільки існує способів…».

1. Історична довідка

Латинський квадрат N-го порядку – це таблиця розмірністю N x N клітинок, складена з N елементів так, що жоден з них не повторюється ні в своєму стовпчику, ні в своєму рядку.

Інакше кажучи, «латинський квадрат порядку n – це квадратна таблиця Ф, задана на елементах множини Q, що містить n елементів, кожна стрічка і кожен стовпець якої містить всі n елементів множини Q. Латинський квадрат A порядку n, складений з елементів множини Q, коротко позначається $A=[\alpha_{ij}]$, де $i, j = 1, 2,\ldots n$, а α_{ij} – елемент множини Q, який знаходиться на перетині i-ї стрічки та j-го стовпчика латинського квадрату» [1, с.6-7].

Найважливішим етапом у вивченні питання латинського квадрату постає час, коли увесь зібраний досвід та необхідні матеріали починає об'єднувати у своїх працях швейцарський математик Л. Ейлер. У XVIII столітті він вводить нове математичне поняття – ортогональний латинський квадрат, або ж греко-латинський квадрат. Приставка «греко-» з'явилась у наслідку використання Ейлером у позначенні елементів другого квадрату грецьких букв.

Згідно з Вікіпедією, «два латинських квадрата $L=(l_{ij})$ и $K=(k_{ij})$ n-го порядку називаються ортогональними, якщо всі впорядковані пари (l_{ij},k_{ij}) відрізняються одна від одної» [2]. Ейлеру не вдалось побудувати такі пари для n=6 та n=10.

З’явилась гіпотеза про те, що пар ортогональних квадратів для $n=4t+2$ не існує. Для $n=6$ гіпотеза була доведена науковцем Террі. Науковці Паркер, Боуз та Шрикханде експериментально довели існування квадратів 10, 14 та 18 і т.д. порядків. Тобто гіпотеза Ейлера перестала працювати для n більше шести.

2. Алгоритм «Перебір з поверненням»

«Алгоритм перебору з поверненням (англ. backtracking), також пошук з поверненням – загальний алгоритм для знаходження всіх (або деяких) розв’язків деякої обчислювальної задачі, який поступово будує кандидатів на розв'язок, і відкидає кожного неповного кандидата c («вертається») як тільки визначає, що c не може бути доповненим до вірного розв’язку. Це важливе знаряддя для розв’язання проблеми відповідності обмеженням, таких як кросворди, судоку та багато інших головоломок. Часто це найзручніший (якщо не найефективніший) підхід для розбору, для задачі пакування рюкзака та інших задач комбінаторної оптимізації.» [3] Класичною задачею вважається «задача про 8 ферзів».

Розглянемо даний алгоритм у псевдокоді. Для застосування «перебору з поверненням» потрібно мати дані P для конкретної задачі та шість наступних підпрограм: корінь, відмова, прийняття, перший, наступний, і вихід. Ці підпрограми мають отримувати дані P як параметр та виглядають наступним чином:

- корінь(P): повертає неповний кандидат в корені дерева пошуку.
- відмова(P,c): повертає істину тільки якщо неповний кандидат c невартий завершення.
- прийняття(P,c): повертає істину якщо c є розв'язком P, і хибу якщо ні.
- перший(P,c): продукує перше розширення кандидата c.
- наступний(P,s): продукує інше наступне розширення кандидата, після розширення s.
- вихід(P,c): прийняти розв'язок c з P, як підходяще для застосування.

Пошук з вертанням зводиться до виклику bt(root(P)), де bt - наступна рекурсивна підпрограма:

« **procedure** *bt*(*c*)
  якщо *відмова*(*P*,*c*) **тоді вийти**
  якщо *прийняття*(*P*,*c*) **тоді** *вихід*(*P*,*c*)
  s ← *перший*(*P*,*c*)
  доки *s* ≠ Λ **робити**
    bt(*s*)
    s ← *наступний*(*P*,*s*)» [3]

3. Інструменти та методи досліджень

Основним методом дослідження є експериментальна перевірка коректності теоретичних побудов та оцінка можливості застосування їх на практиці. Власне, написання алгоритму відбувається на мові Python. Необхідними будуть перевірка працездатності алгоритму та спроби покращити його коректність.

4. Результати досліджень

Виконавши певну роботу по написанню та перевірці коректності алгоритму, стало помітно, що алгоритм дає збій у роботі починаючи з $n = 4$ порядку. Це робить обов’язковим подальшу оптимізацію написаного алгоритму.

5. Висновки

Питання вивчення роботи подібних алгоритмів є наразі актуальним тому, що це може значно спростити кількість використаного часу у різного роду дослідженнях. Перш за все, ми отримали перспективу дослідження цього громіздкого і не такого легкого, на перший погляд, питання. Тобто дослідження буде продовжуватись і надалі буде створено покращений алгоритм для побудови латинскього квадрату. Наступною важливою частиною є те, що виконуючи таку роботу можна навчитись краще оптимізовувати використаний на неї ресурс часу, що безперечно є плюсом у подальшій діяльності та виконанні аналогічних робіт.

Список літератури

1. В. Д. Белоусов Латинские квадраты, квазигруппы и их приложения/ В.Д. Белоусов, Г.Б. Белявская – Кишинев, «Штиинца», 1989. – 76 с.

2. Латинский квадрат, URL:
https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82

3.Перебір з вертанням, URL:
https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D1%83%D0%BA_%D0%B7_%D0%B2%D0%B5%D1%80%D1%82%D0%B0%D0%BD%D0%BD%D1%8F%D0%BC

УДК 004.01

Бойчук І. А., студент 3 курсу
спеціальності 113 «Прикладна математика»
Вєтров О. С., старший викладач
кафедри прикладної математики

МЕТОД СПРАВЕДЛИВОГО ПОДІЛУ РЕСУРСІВ

Донецький національний університет імені Василя Стуса, м. Вінниця

Справедливим вважається такий спосіб поділу ресурсу, що всі учасники вважають, що в результаті вони отримали справедливу частку ресурсу. Важливо відзначити, у загальному випадку вважається, що власне критерії справедливості у кожного учасника можуть були своїми.

Задачі справедливого поділу ресурсу у різних формулюваннях знаходять широке застосування у економіці, організації перемовин, політичній науці тощо. Для наочної ілюстрації справедливого поділу використовують приклад «справедливого поділу торта» (або як окремий випадок – «справедливий поділ пирога») [3].

Класичним алгоритмом справедливого поділу є метод "діли-та-обирай": два учасники з різними вподобаннями (критеріями корисності) можуть поділити ресурс так, що кожен буде задоволений, тобто вважати, що власне і відбувся