**Звіт**

**З дисципліни «Системи паралельного програмування»**

**На тему**

**Quadratic residue mod pq**

**ПАРКС Python**

Виконав

Ковальчук Максим МІ-4

**Поставка задачі і теоретичні відомості**

Нехай задано деякі числа , є великими простими. Необхідно перевірити, чи є a квадратичним лишком по модулю n і, якщо є, знайти корені рівняння .

Для розв’язку даного рівняння спочатку необхідно факторизувати число n. Нехай дане число уже факторизовано на два прості множники . Тоді розглянемо наступні два рівняння:

Оскільки обидва модулі прості, критерію Ейлера достатньо щоб перевірити існування розв’язку для кожного з рівнянь, а власне корінь, можна знайти використовуючи алгоритм Чіполи або для деяких випадків просто за формулою:

Якщо хоча б 1 з рівнянь не має розв’язок, то і початкове рівняння (по складеному модулю) також не має розв’язок.

Якщо ж обидва рівняння мають розв’язок (а насправді два розв’язки, враховуючи властивості рівнянь), то початкове рівняння має 4 розв’язки.

Нехай розв’язано рівняння (завдяки розширеному алгоритму Евкліда)

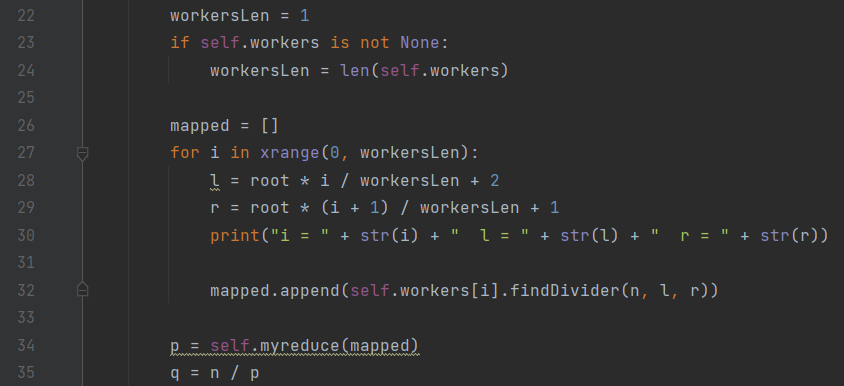
Можна показати (що і було продемонстровано в минулому семестрі Анісімовим. А. В.), що тоді розв’язки матимуть вигляд:

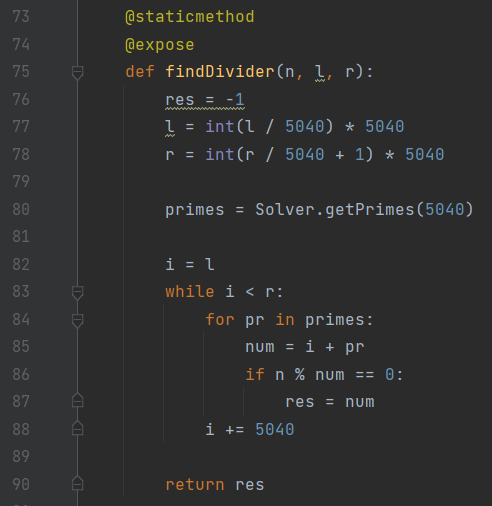
**Особливості реалізації**

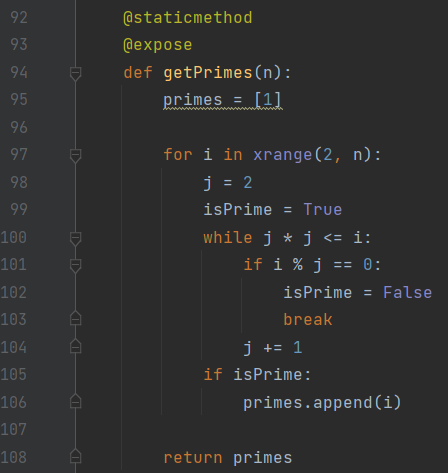
Розроблене програмне забезпечення було реалізовано використовуючи мову програмування Python, і відповідно технологію PARKS-Python.

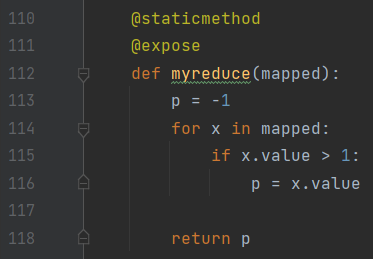
Найскладнішою складовою розв’язку поставленої задачі є NP-hard задача факторизації числа. Для факторизації числа використано простий алгоритм перебору дільників до кореня з заданого числа з деякими модифікаціями для пришвидшення часу виконання. Особливістю алгоритму є те, що він перебирає лише частину потенційних дільників. Нехай це прості числа серед перших 5040 чисел. Тоді неважно показати, що будь яке просте число можна подати в вигляді . Таким чином значно зменшується константа алгоритму перебору дільників.

Для розпаралелення алгоритму достатньо просто розділити всі числа на декілька відрізків, по одному відрізку для кожного потоку (сервера). Нижче наведено відповідні фрагменти коду.









**Виконання програми**

Розроблена програма була протестована в середовищі Google Cloud. Алгоритм був запущений для 1, 2, 3, 4 обчислювальних серверів, всі у одному регіоні.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Кількість серверів** | **4** | **3** | **2** | **1** |
| **Тест1, час (секунди)** | 3s | 4s | 6s | 12s |
| **Тест2, час (секунди)** | 9s | 13s | 18s | 35s |

Повний код програми можна подивитись за посиланням:

<https://github.com/reeWorlds/CloudComputingLabs>

Скріни виконання тестів:

