МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №1а
з дисципліни «Методи оптимізації планування експерименту»
на тему: «Реалізація задачі розкладання числа на прості множники»

Виконала:
студентка групи ІО-91
Кійченко А. К.

Перевірив:
Регіда П. Г.

Київ 2021

Мета: Ознайомитись з основними принципами розкладання числа на прості множники з використанням різних алгоритмів факторизації.

Основні теоретичні відомості

Факторизації лежить в основі стійкості деяких криптоалгоритмів, еліптичних кривих, алгебраїчній теорії чисел та кванових обчислень, саме тому дана задача дуже гостро досліджується, й шукаються шляхи її оптимізації.

На вхід задачі подається число $n \in N$, яке необхідно факторизувати. Перед виконанням алгоритму слід переконатись в тому, що число не просте. Далі алгоритм шукає перший простий дільник, після чого можна запустити алгоритм заново, для повторної факторизації.

В залежності від складності алгоритми факторизації можна розбити на дві групи:

- Експоненціальні алгоритми (складність залежить експоненційно від довжини вхідного параметру);
- Субекспоненціальні алгоритми.

Існування алгоритму з поліноміальною складністю – одна з найважливіших проблем в сучасній теорії чисел. Проте, факторизація з даною складністю можлива на квантовому комп'ютері за допомогою алгоритма Шора.

Розглянемо принципи роботи найпростіших алгоритмів факторизації.

Метод перебору можливих дільників.

Один з найпростіших і найочевидніших алгоритмів заключається в тому, щоб послідовно ділити задане число n на натуральні числа від 1 до $|\sqrt{n}|$. Формально, достатньо ділити лише на прості числа в цьому інтервалі, але для цього необхідно знати їх множину. На практиці складається таблиця простих чисел і на вхід подаються невеликі числа (до 2^{16}), оскільки даний алгоритм має низьку швидкість роботи.

Приклад алгоритму:

1. Початкова установка: $t = 0$, $k = 0$, $n = N$ (t,k,n такі, що $n = N / p_1 \dots p_n$ і n не мають простих множників, менших за d_k).
2. Якщо $n = 1$, закінчуємо алгоритм.
3. Присвоюємо $q = [n / d_k]$, $r = n \bmod d_k$.
4. Якщо $r \neq 0$, переходимо на крок 6.
5. Присвоюємо t++, $p_t = d_k$, $n = q$ і повертаємось на крок 2.
6. Якщо $q > d_k \rightarrow$ k++ і повертаємось на крок 3.
7. Присвоїти t++, $p_t = n$ і закінчити виконання алгоритму.

Модофікований метод факторизації Ферма.

Ідея алгоритму заключається в пошуку таких чисел A і B, щоб факторизоване число n мало вигляд: $n = A^2 – B^2$. Даний метод гарний тим, що реалізується без використання операцій ділення, а лише з операціями додавання й віднімання.

Приклад алгоритму:

1. Початкова установка: $x = 2[\sqrt{n}] + 1$, $y = 1$, $r = [\sqrt{n}]^2 – n$.
2. Якщо $r = 0$, то алгоритм закінчено: $n = \frac{x-y}{2} * \frac{x+y-2}{2}$
3. Присвоюємо $r = r + x$, $x = x + 2$.
4. Присвоюємо $r = r - y$, $y = y + 2$.
5. Якщо $r > 0$, повертаємось до кроку 4, інакше повертаємось до кроку 2.

Метод факторизації Ферма.

Ідея алгоритму заключається в пошуку таких чисел A і B, щоб факторизоване число n мало вигляд: $n = A^2 – B^2$. Даний метод гарний тим, що реалізується без використання операцій ділення, а лише з операціями додавання й віднімання.

Приклад алгоритму:

Початкова установка: x = [$\sqrt{n}$] – найменше число, при якому різниця x^2-n невід'ємна.
Для кожного значення k Є ℕ, починаючи з k = 1, обчислюємо $(\lceil\sqrt{n}\rceil + k)^2 - n$ і перевіряємо чи не є це число точним квадратом.

- Якщо не є, то k++ і переходимо на наступну ітерацію.
- Якщо є точним квадратом, тобто $x^2 - n = \left(\lceil\sqrt{n}\rceil + k\right)^2 - n = y^2$, то ми отримуємо розкладання: $n = x^2 - y^2 = (x+y)(x-y) = A * B$, в яких
$$x = (\lceil\sqrt{n}\rceil + k)$$

Якщо воно є тривіальним і єдиним, то n - просте

Завдання на лабораторну роботу

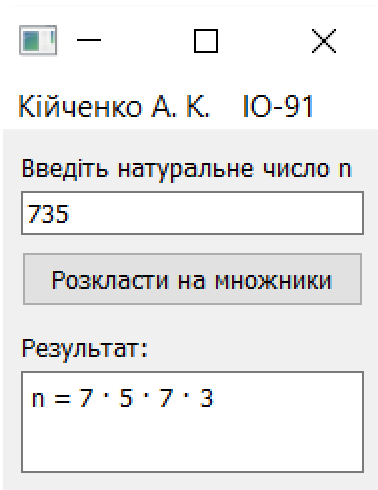
Розробити програму для факторизації заданого числа методом Ферма. Реалізувати користувацький інтерфейс з можливістю вводу даних.

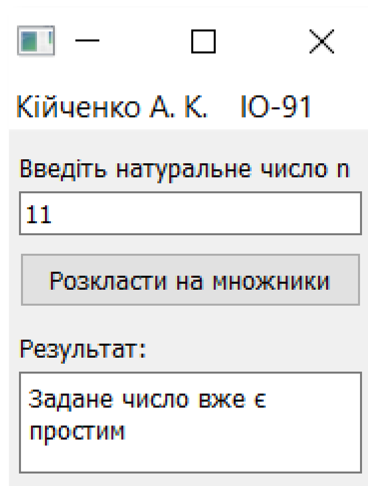
Лістинг програми

```
def factorization(n):
    def isqrt(n):
        x = n
        y = (x + n // x) // 2
        while y < x:
            x = y
            y = (x + n // x) // 2
        return x

    def fermat(n):
        a = isqrt(n)
        b2 = a * a - n
        b = isqrt(n)
        count = 0
        while b * b != b2:
            a = a + 1
            b2 = a * a - n
            b = isqrt(b2)
            count += 1
        p = a + b
        q = a - b
        assert n == p * q
```

```
        return [p, q]

    result = fermat(n)
    if 1 in result:
        result.remove(1)
        return result
    else:
        new_result = []
        for i in result:
            res = factorization(i)
            for j in res:
                new_result.append(j)
        return new_result
```

Результат виконання роботи





Висновки:

В результаті виконання лабораторної роботи було досягнуто поставленої мети: ознайомлено з основними принципами розкладання числа на прості множники з використанням різних алгоритмів факторизації. А також розроблено програму, яка реалізує розкладання числа на прості множники методом Ферма.