ОТЧЕТ К ЛАБОРАТОРНОЙ РАБОТЕ № 6

Лабораторная работа № 6

Методы факторизации числа

Вариант №5

Ф. И. О. студента: Гюнтер Тимофей Вячеславович

Группа: ФИТ-221

Проверил: Дата:

Результаты

$m = 467983$

Метод квадратичного решета

$a = 2, b = 3, c = 5$

$x = 728, y = 249$

$p = 977, q = 479$

$\rho -$ метод факторизации

$n = 17$

$a_n = 17244, d_n = 479$

$p = 479, q = 977$

Код программы

```
from lab5 import is_prime

from lab2 import euclid_extended


def integer_sqrt(m: int):
    '''
```

```
    Метод поиска целочисленного корня от числа m
    '''
    assert m > 0
    x = m
    y = (x + int(m/x)) >> 1
    while y < x:
        x = y
        y = (x + int(m/x)) >> 1
    return x


# x^2 ≡ q (mod n)
def is_quad_res(q, modulus):
    '''
    Является ли данное число q кв.вычетом по заданному модулю
    modulus
    '''
    for x in range(10_000):
        if x**2 % modulus == q:
            return True
    else:
        return False


# print(is_quad_res(0, 7))
```

```
def quadratic_sieve_fact(m, a=3, b=5, c=7):
    '''
    Факторизация числа методом квадратичного решета:\n
    m - факторизумое число;\n
    a, b, c - модульные решёта\n
    returns: p, q - два делителя числа m
    '''
    # step 1
    # within list: x^2, x^2 - m, S_a
    a_dict = {key:[] for key in range(a)}
    b_dict = {key:[] for key in range(b)}
    c_dict = {key:[] for key in range(c)}
    for dict in [a_dict, b_dict, c_dict]:
        mod = len(dict.keys())
        for key in dict:
            dict[key].append(key**2 % mod)
            z = (key**2 - m) % mod
            dict[key].append(z)
            dict[key].append(is_quad_res(z, mod))

    # step 2
    interval = range(integer_sqrt(m)+1, (m+1)//2+1)
```

```
    bool_values_dict = {x:[] for x in interval}
    for x in interval:
        local_bool = []
        for dict in [a_dict, b_dict, c_dict]:
            mod = len(dict.keys())
            # for key in dict
            rem = x % mod
            local_bool.append(dict[rem][-1])
        bool_values_dict[x] = local_bool

    # step 3, 4
    for x in bool_values_dict:
        if all(value for value in bool_values_dict[x]):
            z = x**2 - m
            sqrt = integer_sqrt(z)
            if sqrt**2 == z:
                print(f'x: {x}, y: {sqrt}')
                y = sqrt
                return x+y, x-y

# p, q = quadratic_sieve_fact(m=445051)

# print(is_prime(p), is_prime(q))
```

```
def quadratic_sieve_fact_interface():
    list_input = input('Введите последовательно через пробел фактуризуемое число m и три решета a, b, c соответственно:\n')
    m, a, b, c = map(int, list_input.split())
    p, q = quadratic_sieve_fact(m, a, b, c)
    print(f'p: {p}, q: {q}')
# quadratic_sieve_fact_interface()


def rho_fact(m, initial_1=2, initial_2=2, f = None):
    if not f:
        f = lambda x: (x**2 + 1) % m

    iter = 0
    x_0_1, x_0_2 = f(initial_1), f(f(initial_2))
    x_1 = f(x_0_1)
    x_2 = f(f(x_0_2))

    a = abs(x_1 - x_2)
    gcd = euclid_extended(a, m)[0]
    while not (gcd > 1 and gcd < m):
        iter += 1
        x_1 = f(x_1)
```

```
        x_2 = f(f(x_2))


        a = abs(x_1 - x_2)
        gcd = euclid_extended(a, m)[0]
    print(f'a: {a}, d: {gcd}')
    return gcd, m//gcd, iter


def induvid_var():
    m = 445051
    print(f'm = {m}')
    p, q = quadratic_sieve_fact(m)
    p1, q1, iters = rho_fact(m)
    print(f'Результат кв. решета: p={p}, q={q}\
          \nРезультат ро-факторизации: p={p}, q={q}, итерации: {iters}')
# induvid_var()
# print(rho_fact(533, 2, 2))


def rho_fact_interface():
     list_input = input('Введите последовательно через пробел фактуризуемое число m, два начальных члена послед-тей:\n')
    m, initial_1, initial_2 = map(int, list_input.split())
    p, q, iters = rho_fact(m, initial_1, initial_2)
    print(f'p: {p}, q: {q}, кол-во итераций: {iters}')
```

rho_fact_interface()