РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Тема: Разложение чисел на множители

дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Койфман Кирилл Дмитриевич

Группа: НФИмд-01-25

Введение

Цель работы

Получение практических навыков реализации алгоритмов, выполняющих разложение чисел на множители.

Задачи

1. Реализовать алгоритм разложения чисел на множители, использующий p-метод Полларда.
2. Проверить работоспособность реализованного алгоритма.

Ход работы

1 задание

Для решения поставленной задачи реализуем описанный в тексте лабораторной работы алгоритм для разложения чисел на множители на языке программирования C++ (Листинг-1):

```
namespace functions
{
    long long function_with_compression_behavior_1(long long x, long long
n)
    {
        return static_cast<long long>(std::pow(x, 2) + 5) % n;
    }
}

using compressing_function_t = long long(*)(long long, long long);

long long algorithmEuclid(long long a, long long b)
{
    if (b > a)
        std::swap(a, b);

    if (a <= 0 || b <= 0)
        return 1;

    long long d = 0;
    long long gamma_prev = a;
    long long gamma_curr = b;
    long long gamma_next = 0;

    while (true)
    {
        long long remainder = gamma_prev % gamma_curr;
        gamma_next = remainder;

        if (gamma_next == 0)
        {
            d = gamma_curr;
            break;
        }
        else
        {
            gamma_prev = gamma_curr;
            gamma_curr = gamma_next;
        }
    }

    return d;
}

int algorithmPollard(long long n, long long c = 1, compressing_function_t
function = functions::function_with_compression_behavior_1)
{
    long long a = c;
    long long b = c;
```

```
    long long d = 0;

    std::cout << "|a\tb\td|\n---------------------------------------\n";

    while (true)
    {
        a = function(a, n);
        b = function(b, n);
        b = function(b, n);

        d = algorithmEuclid(std::abs(a - b), n);

        std::cout << a << '\t' << b << '\t' << d << '\n';

        if (d > 1 && d < n)
        {
            return d;
        }
        else if (d == n)
        {
            return -1;
        }
        else if (d == 1)
        {
            continue;
        }
    }
}
```

Листинг-1(реализация алгоритма, реализующего p-метод Полларда)

2 задание

Теперь проверим работу алгоритма на предоставленном в лабораторной работе числе (Листинг-2, Листинг-3):

```
int main()
{
    long long chosenValue = 1'359'331;
    long long result = algorithmPollard(chosenValue);
    std::cout << "Chosen value: " << chosenValue << '\n';
    if (result > 0)
    {
        std::cout << "\nDivider = " << result << std::endl;
        std::cout << "Result tests: \n[1]" << chosenValue << " / " <<
result << " = " << chosenValue / result
            << "\n[2]" << result << " * " << chosenValue / result << " = "
<< result * (chosenValue / result) << std::endl;
    }
    else
```

```
    {
        std::cout << "\nDivider was not found!" << std::endl;
    }
    return 0;
}
```

Листинг-2(код для тестирования алгоритма)

```
|a      b       d|
---------------------------------------
6       41      1
41      123939  1
1686    391594  1
123939  438157  1
435426  582738  1
391594  1144026 1
1090062 885749  1181
Chosen value: 1359331

Divider = 1181
Result tests:
[1]1359331 / 1181 = 1151
[2]1181 * 1151 = 1359331
```

Листинг-3(результат тестирования)

Исходя из полученных результатов (Листинг-3), можно утверждать, что реализованный алгоритм успешно работает.

Заключение

В ходе проделанной лабораторной работы мной были получены навыки по реализации алгоритмов, осуществляющих разложение чисел на множители.