

Факторизация чисел Ро-методом Полларда

Кузнецова Арина 6373

Рo-алгоритм Полларда

В 1975 году Поллард опубликовал статью, в которой он, основываясь на алгоритме обнаружения циклов Флойда, изложил идею алгоритма факторизации чисел, работающего за время, пропорциональное $N^{1/4}$.

С его помощью было разложено на множители число Ферма $F_8 = 2^{256} + 1$.

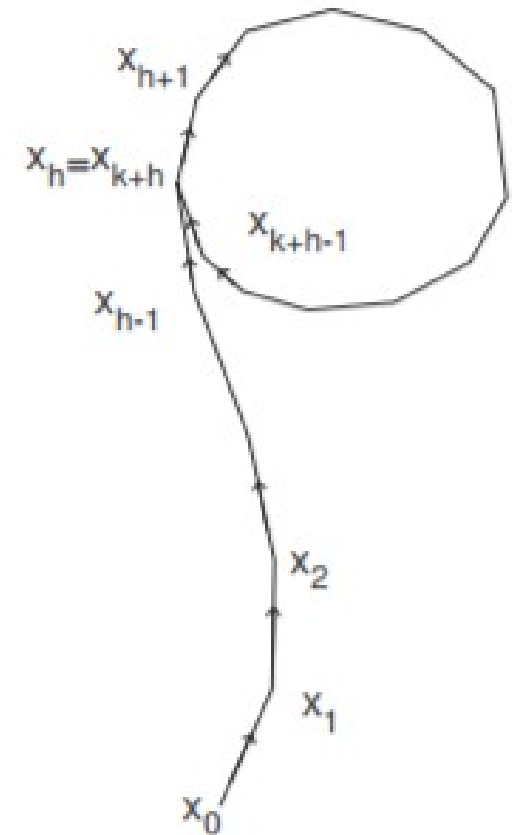
Оригинальная версия алгоритма

Возьмем некоторое случайное отображение

$f : Z_n \rightarrow Z_n$, которое сгенерирует некоторую случайную последовательность x_0, x_1, x_2, \dots где $x_i = f(x_{i-1})$. Обычно берётся многочлен $f(x) = x^*x + 1$.

Функция f имеет не более, чем n значений, поэтому последовательность зациклится.

$$x_{k+h} = x_h$$



Оригинальная версия алгоритма

Рассмотрим теперь алгоритм подробнее.

При выбранном многочлене $F(x) = x^2 + 1 \pmod{n}$ рассматриваем

$x_i = f(x_{i-1})$. Причём i - индекс элемента перед заикливанием последовательности.

И для всех $j < i$ вычисляем НОД $d = \gcd(|x_i - x_j|, n)$.

Если $n > d > 1$, то d – нетривиальный делитель n .

Если n/d - составное число, то применяем данный алгоритм ещё раз уже к числу n/d . И так продолжаем до тех пор, пока не получим разложение только из простых чисел.

Особенности использования алгоритма

Следует отметить, что рассматриваемый алгоритм в значительной степени случаен, его эффективность сильно и непредсказуемо зависит от выбора многочлена и начального элемента в последовательности.

Метод эффективен для нахождения небольших простых делителей числа n . Делители большего размера тоже могут быть обнаружены, однако лишь с некоторой вероятностью.

Тест на простоту чисел Миллера-Рабина

Для реализации факторизации чисел любого метода нужен тест на простоту чисел.

Я выбрала вероятностный тест Миллера-Рабина, так как при проверке k чисел на условия простоты вероятность того, что составное число будет принято за простое, будет меньше $(\frac{1}{4})^k$. В своей программе я проверяла 3 случайных числа, так в этом случае достигается достаточно удовлетворительная вероятность 0.015625.

Тест на простоту чисел Миллера-Рабина

Для проверки числа n (причём $n > 2$) на простоту, во-первых, оно представляется в виде $n-1 = t * 2^s$, где t - нечётно.

Дальше проверяются следующие утверждения.

Если n - простое, то для любого a из Z_n ($a < n$) выполняются:

- 1) $a^t \equiv 1 \pmod{n}$
- 2) Существует такое r , что при $0 \leq r < s$: $a^{t \cdot (2^r)} \equiv -1 \pmod{n}$

Если хотя бы одно из этих условий соблюдается, то число a является свидетелем простоты числа n .

Делая проверку на 3 случайных числах, мы повышаем вероятность того, что число n не псевдопростое.

Использование модуля факторизации числа Ро-методом Полларда

В репозитории с программой лежит отдельный файл `factRhoPollard.cpp` с примером использования разработанного мной модуля. Также я выделила в него те функции, которые я дополнительного прописывала для реализации поставленного метода. На гитхабе не отображается кириллица, но при копировании репозитория всё должно быть нормально. Но, на всякий случай, размещаю тут, что делают конкретные функции:

1) возведение в степень

```
LNum power(LNum const&, LNum const&);
```

2) проверка на простоту числа с помощью теста Миллера-Рабина

```
bool isPrimeNum(LNum const&);
```


Использование модуля факторизации числа Ро-методом Полларда (как это всё выглядит)

The image shows a C++ IDE with a source file `LNum.cpp` and three command prompt windows. The source code implements the Rho-Pollard factorization algorithm. The command prompt windows show the results of running the program for inputs 144, 154, and 231.

```
1  #pragma once
2
3  #include "LNum.h"
4  #include "ILNum.h"
5  #include "RNum.h"
6  #include "Ordinal.h"
7  #include "Signum.h"
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char** argv) {
13
14     LNum N;
15     cin >> N;
16     vector<LNum> mas = factRhoPollard(N);
17     cout << "Factorization of RhoPollard: ";
18     for (size_t i = 0; i < mas.size(); i++)
19     {
20         cout << mas.at(i) << ' ';
21     }
22     return EXIT_SUCCESS;
23 }
```

Command Prompt 1 (Input: 144):

```
C:\WINDOWS\system32\cmd.exe
144
Factorization of RhoPollard: 3 3 2 2 2 Для продолжения нажмите любую клавишу . . .
```

Command Prompt 2 (Input: 154):

```
C:\WINDOWS\system32\cmd.exe
154
Factorization of RhoPollard: 7 2 11 Для продолжения нажмите любую клавишу . . .
```

Command Prompt 3 (Input: 231):

```
C:\WINDOWS\system32\cmd.exe
231
Factorization of RhoPollard: 3 7 11 Для продолжения нажмите любую клавишу . . .
```