Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Криптография»

Студент: Л. Я. Вельтман
Преподаватель: А. В. Борисов
Группа: М8О-307Б

Дата: 28.02.2020
Оценка:
Подпись:

Москва, 2020

Лабораторная работа №1

Задача: Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

Вариант 7:

n1 = 26888732002909002811721449825320409576588413648336619384236128377 6500643966781

n2 = 14177467869787507650387834432016948376930580071471350079285831921 4425694670422365904947589804271577823515302608521263525608934810569555 9658561967608516134648218041362591071855477293688831113885128127003390 5970826200499692827568755840858440733991917454028255326174744965696470 3936447130918315087871163722894672660845644433050799802860493503622897 6139386330779518797479718798595753346147608882581639592255872792033006 6823211210594296302676261707432217348305112187

Выходные данные:

Для каждого числа необходимо найти и вывести все его множители - простые числа.

1 Описание

В математике факторизация или факторинг — это декомпозиция объекта (например, числа, полинома или матрицы) в произведение других объектов или факторов, которые, будучи перемноженными, дают исходный объект. Чтобы решить задачу для первого числа наиболее быстрым и эффективным способом, я использовала общий метод решета числового поля — метод факторизации целых чисел, который реализован в msieve. Второе число является очень большим, и поэтому вышеописанный метод мне не подойдет. Преподаватель подсказал, что один из множителей второго числа определяется как наибольший общий делитель с одним из чисел другого варианта. Было решено реализовать перебор чисел других вариантов, подсчитывать их НОД с числом моего варианта и выводить полученный ответ, если он больше 1. Второй множитель определяется как частное деления числа моего варианта на найденный НОД на предыдущем шаге.

2 Исходный код

```
from math import gcd


if __name__ == '__main__':
    with open('num2', 'r') as file:
        n = int(file.read())
    Nums =
        [169512848540208376377324702550860778129688385180093459660532447790298998967239009844131423361916242087180680156861712994509728052535159091128844805658679025296716559404434664811725619186652725915987565442108608120026832525046666312840385351549793409109648246739235786392263979181344291927371250171497372227982026555999675170108947918951378367343470923483104158597216632066586300921566811159875654421086081200268325250466663128403853515497934091096482467392357863922639791813442919273712501714973722279820265559996751701089479189513783673434709234831041585972166320665863009215668111598756544210860812002683252504666631284038535154979340910964824673923578639226397918134429192737125017149737222798202655599967517010894791895137836734347092348310415859721663206658630092156681115987565442108608120026832525046666312840385351549793409109648246739235786392263979181344291927371611765569148804856242867384258680719850010286298191204635154152942043219729044752688614748313611589686907858960532293041950259807409089116075774905924811928369729308516275072914492447303882343144705635774304031878986296122750910474479908149467861238329198698492351931644628770804907791822412624855040201687310008422575815379573283264975224784050024653596488753568102802922445476180707271916242087180680156861712994509728052535159091128844805658679025296716559404434664811725619186652196034400067344801010996612379825913878831222300011028544413898468704368209191843772656487365265516884322685356525369761615442254049333529173484668807416465552360809404683693905337775669013748631669812028211114876035741593474021802212340044740884701344271270195832085856797314936725609969919141690844477193411432723606433569517503385556872451472327609090923890224945076116311617929837009715109385843025147460686876803591387120848268695317498338161525361070299566943782286650144848099931503349990631350512794289684313078245040080234793749288284388102811529318651334186314509247654009154062250949081794905352464916598198236271013859014568010836766059230094210745557212887498531331716265705923840344012310598594084552548100509114311455807738173203854456785977766950683127961452581342124472692680814864696039831657201341930170537490888948185909193404926961223536479474040666460
    multipliers = []
    for num in Nums:
```

```
11          first = gcd(n, num)
12          if (first != 1):
13              multipliers.append(first)
14              multipliers.append(n / multipliers[0])
15      print("Original number: {0}".format(n))
16      print("Multipliers: ")
17      print(*multipliers, sep = '\n')
```

3 Консоль

```
(base) MacBook-Pro-Lina:msieve-1.53 linuxoid$ ./msieve -m -q


next number: 268887320029090028117214498253204095765884136483366193842361283776500
2361283776500643966781

26888732002909002811721449825320409576588413648336619384236128377650064396678
1
p39: 414150068879409136107176764405542089303
p39: 649250936397607504492837402141095065227
(base) MacBook-Pro-Lina:crypto linuxoid$ python3 lab1.py
Original number: 14177467869787507650387834432016948376930580071471350079285831
92144256946704223659049475898042715778235153026085212635256089348105695559658
58561967608516134648218041362591071855477293688831113885128127003390597082620
04996928275687558408584407339919174540282553261747449656964703936447130918315
08787116372289467266084564443305079980286049350362289761393863307795187974797
18798595753346147608882581639592255872792033006682321121059429630267626170743
221734830511218 7

Multipliers:
13047386803258360982718544898036475818102572197915858405851096848067467531800
75825129914383794099000256342066029993350472539585976457519225723195197365490
21098387734031741968386885087469564811369549756562791721156060671611860553422
48910459683542763941795236487891843320360172073464935554329358004810613116664
9

1.0866135942445523e+154
```

4 Выводы

Выполнив данную лабораторную работу, я познакомилась с новой темой: факторизацией больших чисел. Важно применить наиболее эффективный алгоритм, иначе для решения поставленной задачи уйдет не один десяток лет.