МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии
и прикладная математика»
Кафедра: 806 «Вычислительная математика
и программирование»

Лабораторная работа № 2
по курсу «Криптография»

Группа: М8О-306Б-21

Студент(ка): О. А. Мезенин

Преподаватель: А. В. Борисов

Оценка:

Дата: 10.03.2024

Москва, 2024

Оглавление

1 Тема

Факторизация чисел.

2 Задание

Разложить каждое из чисел n1 и n2 на нетривиальные сомножители.

Вариант 10.

n1=284994967805859272853477327862245466978346919806585432133556769599269315271111,

n2=144705635774304031878986296122750910474479908149467861238329198698492351931644628770804907791822465652742954367322936435188718339080726275242311729821104193465515227659922543175167158889598151741902647154293244819894449690836163313270764079803935657095050060789501415065874078204207363026173352563519252477390183115045370666190418643990517658419460473214034685807819362335735214694601654947678049107321295399466077016934821144519901938606946984530618532320643996 1.

3 Теория

Факторизация числа — это его разложение на простые множители. Факторизация лежит в основе криптостойкости некоторых алгоритмов шифрования с открытым ключом, таких как RSA.

Основа алгоритма RSA заключается в следующем. Берутся два простых числа p и q, а также результат их произведения N (p*q=N). Затем идут ещё несколько вычислений, но их опустим. Смысл в том, что в открытом ключе отсутствует информация о числах p или q, но есть информация о числе N. Если суметь факторизовать число N и найти числа p и q, то шифр будет считаться взломанным.

4 Ход лабораторной работы

Идея заключалась в следующем. Найти алгоритмы факторизации и исходя из сложности алгоритмов оценить примерное время выполнения факторизации. Если время приемлемое, то факторизовать числа.

В статье описаны некоторые экспоненциальные алгоритмы. Самым эффективным из них показался метод Брента. Его сложность составляет O($N^{1/4}$). Оценим примерное время выполнения алгоритма для первого числа. Исходить будем из того, что на выполнение 10^9 простейших операций приходится одна секунда. Подставляя первое число в функцию, которая находится под оценкой сложности и деля результат на 10^9, получим количество секунд, которое требуется для факторизации числа в худшем случае.

```
>>> 284994967805859272853477327862245466978
34691980658543213355676995926931527111**(1
/4)/(10**9)
23105176898.24355
>>>
```

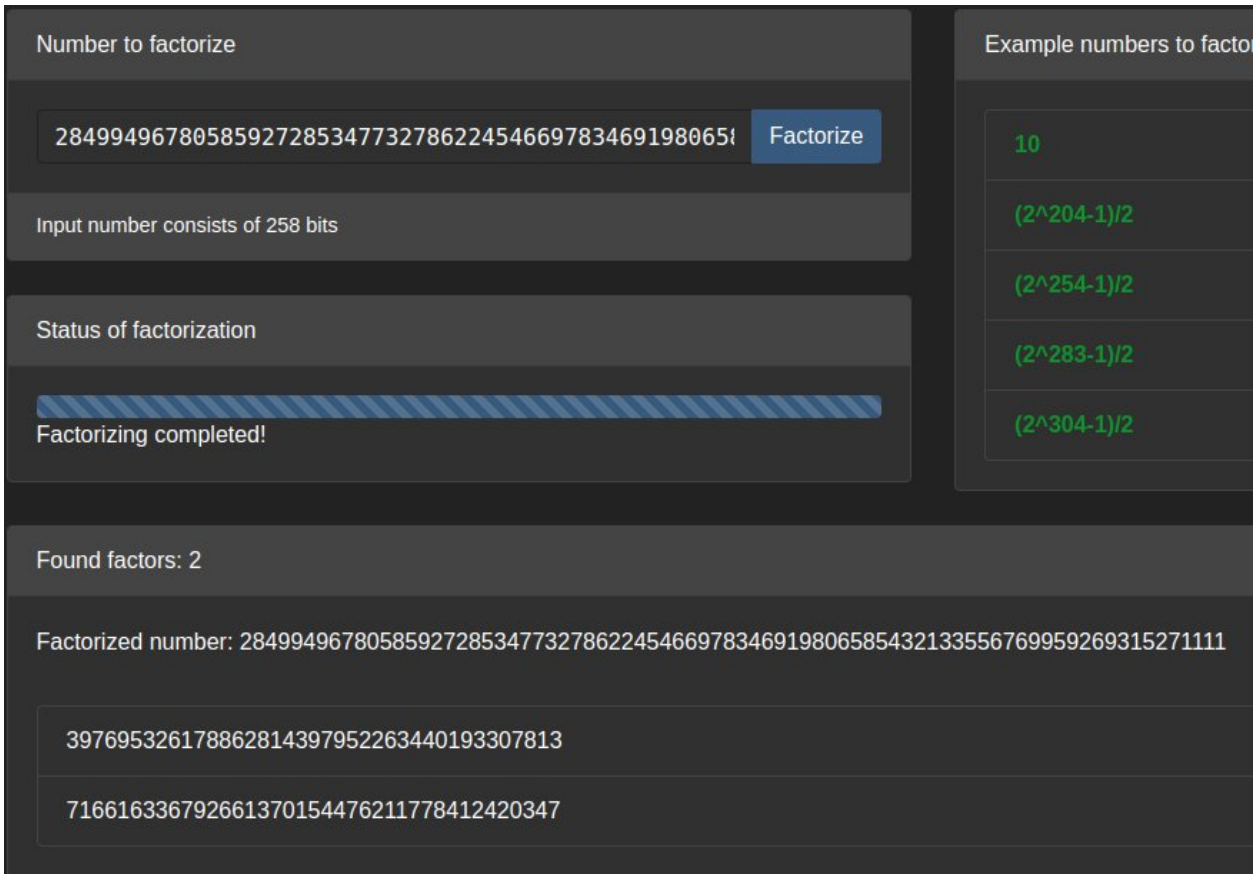
Получили 23105176898 секунд, что примерно равно 267421 день. Кажется, что если использовать этот алгоритм, то результат будет готов гораздо позже дедлайна ЛР. Поэтому данный алгоритм не подходит для поставленной задачи.

Рассматривая субэкспоненциальные алгоритмы, мною был выделен Метод квадратичного решета. Его оценка сложности составляет O(exp($\sqrt{logNloglogN}$)). Попробуем рассчитать примерное время на выполнение факторизации для первого числа.

```
>>> from math import exp, log, sqrt
>>> n1 = 284994967805859272853477327862245466
6697834691980658543213355676995926931527111
1
>>> exp(sqrt(log(n1)*log(log(n1))))/(10**9)
16031.447135671406
>>>
```

Получаем 16031 секунд, или примерно 267 минут, что уже приемлемо.

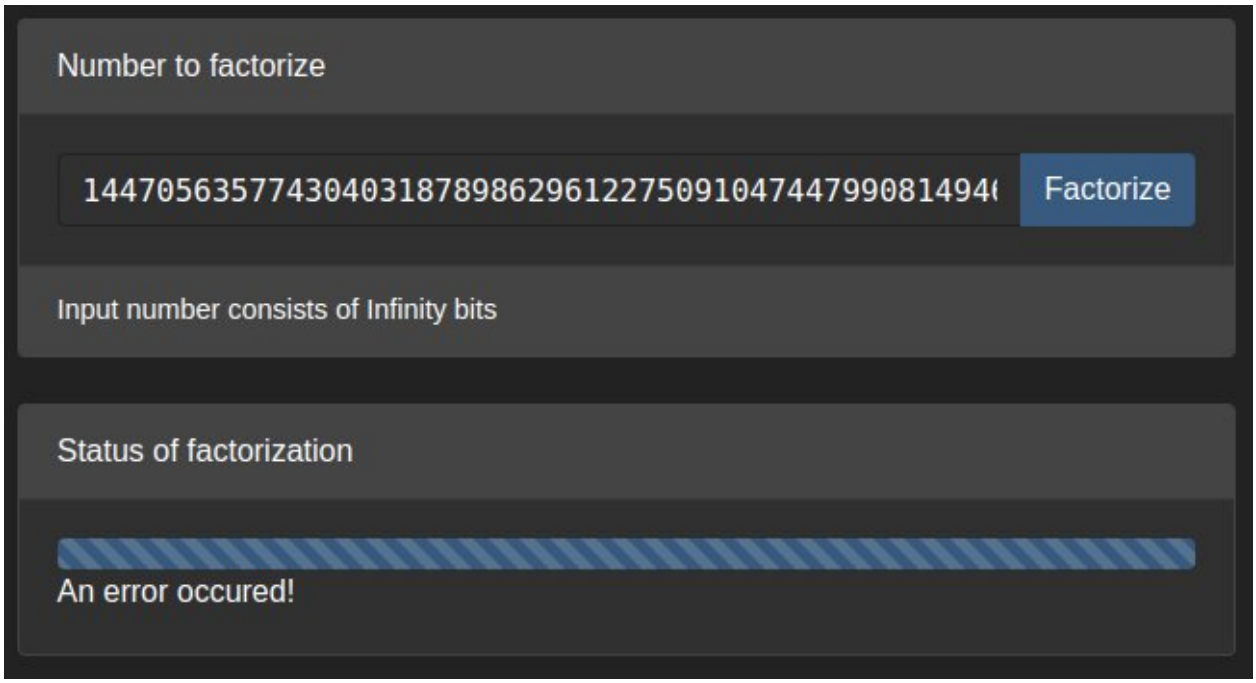
Для факторизации числа методом квадратичного был использован сайт www.cryptool.org. Процесс занял всего 3 минуты и был выдан результат.



Рассчитаем время для второго числа.

```
>>> n2 = 144705635774304031878986296122750910474479908149467861238329198698492351931644628770804907791822465652742954367322936435188718339080726275242311729821104193465515227659922543175167158889598151741902647154293244819894449690836163313270764079803935657095050060789501415065874078204207363026173352563519252477390183115045370666190418643990517658419460473214034685807819362335735214694601654947678049107321295399466077016934821144519901938606946984530618532320643996 1
>>> exp(sqrt(log(n2)*log(log(n2))))/(10**9)
2.5276875962717466e+28
>>>
```

Получаем 25276875962717466000000000000 секунд, или примерно 801524478777190100000 лет. Кажется, что Вселенная живёт гораздо меньше, и результат будет готов только в Эпоху распада Вселенной, поэтому данный алгоритм для заданного числа не подойдет. Даже сайт сломался.


Number to factorize
144705635774304031878986296122750910474479908149 4(
Factorize
Input number consists of Infinity bits
Status of factorization
An error occured!

5 Выводы

В ходе лабораторной работы ознакомился с задачей факторизации чисел и рассмотрел два алгоритма факторизации: метод Брента и метод квадратичного решета.

Ознакомился с алгоритмом RSA. Он полагается на сложность задачи факторизации двух больших простых чисел. Если числа большие, то ни один из существующих алгоритмов факторизации не сможет быть применен для взлома шифра, что и было продемонстрировано в лабораторной работе.

6 Список используемой литературы

1. Факторизация целых чисел — https://habr.com/ru/sandbox/163811/
2. RSA простыми словами и в картинках - https://habr.com/ru/articles/745820/
3. Факторизация чисел и методы решета - https://habr.com/ru/articles/521876/
4. Factorization of large numbers using a quadratic sieve - https://www.cryptool.org/en/cto/msieve