Problem-3

Aarunish Sinha

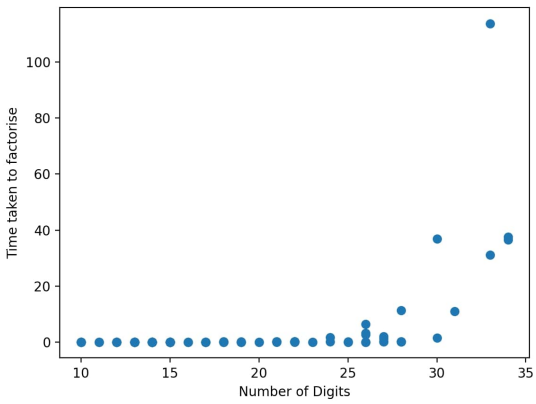
February 16, 2021

1 Configuration

CPU - 2.3GHz 8-core Intel Core i9 9880H (Turbo Boost up to 4.8GHz)
RAM - DDR4-16 GB (2666 MHz)

2 Prime Factorisation

For factorizing the numbers given in `nlist.txt`, I have implemented Pollard Rho's algorithm.
My implementation can factorize upto 1244677669356003369598194162674 97 (112th number in the list) within 5 minutes.



Pollard Rho's Algorithm is a very fast algorithm for finding one factor of a given number. In the above plot we can see that the algorithm takes < 1 second to find the factors for 25-digit numbers. It takes more than a second for factorising 26-digit numbers. It takes ~ 114 seconds to factorise 77464944213471646922209120899568 9. My implementation can factorise all the 34-digit numbers in `nlist.txt` with 40 seconds but takes more that 5 minutes thereafter.

3 Cracking RSA

My implementation of RSA can successfully encrypt and decrypt till the 94th number in the list. After that it fails while decryption due to the limitations of the modulo operator in python.
For encryption, first I converted each character in `plaintext.txt` to its ASCII decimal representation. I have stored these values in an array and the performed encryption and decryption on each element of the array.

4 README

To compile and run the program,

```
$ python crack.py n plaintext.txt
```