ECE 3790 Lab #5

Submitted By: Richard Constantine
#7686561

1) What is the distribution of prime numbers among all numbers? This is a function of n.

$$\pi(n) \to \frac{n}{\ln(n)}\Big|_{n\to\infty} \; or \; \int_2^n \frac{1}{\ln(n)}$$

2) How many numbers on average are selected/tested for primality using the BigInteger constructor before one is likely found? This is a function of n and s.

Using distribution of Primes:

$$Probability\ of\ Selecting\ Prime\ = \frac{\pi(n)}{n} = \frac{1}{\ln(n)}$$

$$\therefore Expected\ Number\ of\ Primes\ to\ be\ Tested = \frac{1}{\frac{1}{\ln(n)}} = \ln(n)$$

where ln(*n)* is the approximate number primes less than *n*

However, if we consider the total number of primality tests occurring:

$$Probability\ of\ Miller\ Rabin\ Proving\ Composite\ Witness = 1 - \frac{1}{2^s}$$

where s is the number of iterations of the test

Since each iteration is independent:

$$P(x) = \frac{1}{2}$$

$$\therefore\ E(x) = \frac{1}{P(x)} = 2\ Trials$$

$$\therefore Expected\ No.of\ Trials\ Until\ Number\ is\ Proven\ Composite\ = 2$$

Connecting both ideas, the total number of primality tests can be expressed as:

$$Total\ Expected\ Number\ of\ Primality\ Tests\ = \ 2\,(\ln(n) - 1) + s$$

because all primes tested (except the last) have 2 expected trials – then last trial (ie the prime number) undergoes/passes all *s* iterations of the test

3) What are the basic number theory algorithms used in RSA?

Some basic number theory algorithms used in RSA are:

- Euclid's Recursive Greatest Common Divisor Algorithm
- Prime Factorization
- Modular/Clock Arithmetic
- Euler Phi Function (calculating distributing of primes)
- Euler Theorem (relating Euler Phi Function to Modular Arithmetic)
- Fermat's Little Theorem & Miller Rabin Primality Testing
- Chinese Remainder Theorem

4) How can RSA be used within private key crypto systems?

A public key can be used to encrypt/decrypt the common/shared key that is required of private key crypto systems (like AES). This provides a safe means to exchange keys without "meeting" or compromising security. It also allows for RSA (which is relatively inefficient for transmitting large pieces of data quickly) to only encrypt the key, while letting more efficient algorithms (like AES) to handle the actual data transfer.

Since RSA has been in use for over 30 years without being cracked (due to the complex nature of factorizing large numbers) – RSA is especially useful in securing these private crypto keys, connections to web servers (e.g. via HTTPS), and keeping online transactions encrypted.

5) Does RSA require prime numbers or numbers that are very likely prime?

Not necessary, however, it makes calculating the Euler Phi Function, and generating a large prime number much easier – i.e. given two primes: $\varphi(n) = (p_1 - 1)(p_2 - 1)$.

6) Late-ish breaking news: When was the problem of generating a larger prime number shown to be in P.

The AKS (Agrawal-Kayal-Saxena) primality test was the first deterministic, primality-proving algorithm to operate within polynomial time. It was developed in August, 2002 in a paper titled, "PRIMES is in P".

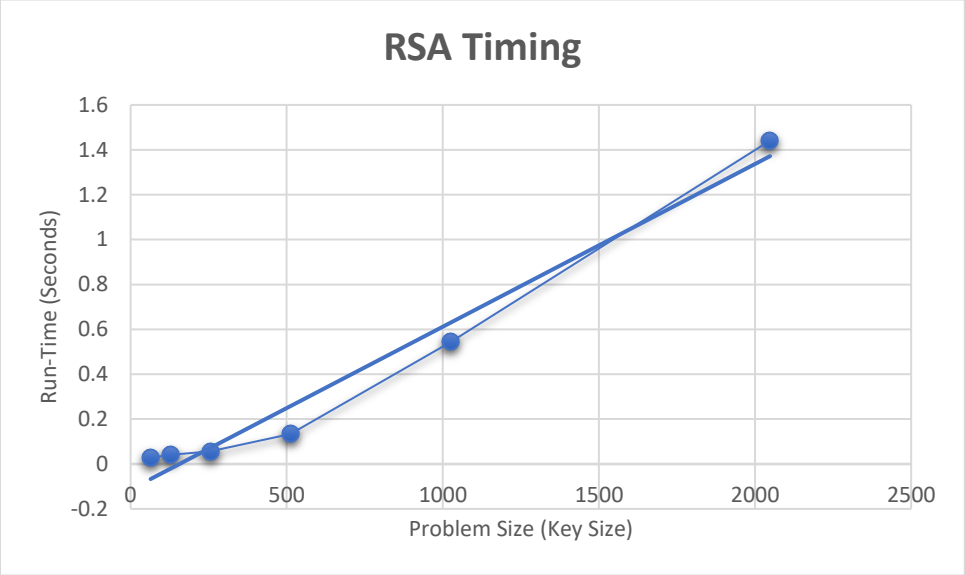
"The algorithm was the first to determine whether any given number is prime or composite within polynomial time."
(Source: https://en.wikipedia.org/wiki/AKS_primality_test)

Note: All relevant code has been attached, and the results/sample output are provided below (in the appendix).

Appendix

Timing Results (Using System.nanotime() – Test2.txt)

Key Size	Time (seconds)
64	0.027812503
128	0.041270627
256	0.05703045
512	0.134234289
1024	0.542330011
2048	1.441214982



Sample Output – Test2.txt (without printing intermediate results)

run:
The messageString is:

The Pan Galactic Gargle Blaster

The Hitchhiker's Guide to the Galaxy states that the effect of drinking a Pan Galactic Gargle Blaster is like having your brains smashed out by a slice of lemon wrapped round a large gold brick.

How to make one

- Take the juice from one bottle of Ol' Janx Spirit.

- Pour into it one measure of water from the seas of Santraginus V (Oh, that Santraginus seawater. Oh, those Santrginus fish!)

- Allow three cubes of Artutan Mega-gin to melt into the mixture (it must be properly iced or the benzine is lost)

- Allow four liters of Fallian marsh gas to bubble through it, in memory of all those happy hikers who have died of pleasure in the Marshes of Fallia.

- Over the back of a silver spoon float a measure of Qualactin Hypermint extract, redolent of all the heady odors of the dark Qualactin Zones,

subtle, sweet, and mystic.

- Drop in the tooth of an Algolian Suntiger. Watch it dissolve, spreading the fires of the Algolian Suns deep into the heart of the drink.

- Sprinkle Zamphuor.

- Add an olive.

- Drink ... but ... very carefully ...

****

Decyphering Message as Blocks and Returning Decyphered Message

****

The messageStringBack is:

The Pan Galactic Gargle Blaster

The Hitchhiker's Guide to the Galaxy states that the effect of drinking a Pan Galactic Gargle Blaster is like having your brains smashed out by a slice of lemon wrapped round a large gold brick.

How to make one

- Take the juice from one bottle of Ol' Janx Spirit.

- Pour into it one measure of water from the seas of Santraginus V (Oh, that Santraginus seawater. Oh, those Santrginus fish!)

- Allow three cubes of Artutan Mega-gin to melt into the mixture (it must be properly iced or the benzine is lost)

- Allow four liters of Fallian marsh gas to bubble through it, in memory of all those happy hikers who have died of pleasure in the Marshes of Fallia.

- Over the back of a silver spoon float a measure of Qualactin Hypermint extract, redolent of all the heady odors of the dark Qualactin Zones, subtle, sweet, and mystic.

- Drop in the tooth of an Algolian Suntiger. Watch it dissolve, spreading the fires of the Algolian Suns deep into the heart of the drink.

- Sprinkle Zamphuor.

- Add an olive.

- Drink ... but ... very carefully ...

Elapsed Time: 0.039071124 seconds
BUILD SUCCESSFUL (total time: 0 seconds)