Hunting prime numbers with TimoEratosthenes algorithm

Inventor of algorithm and author: Timo Kinnunen born 06/30/1956.

Subject: Programming, mathematics, prime number.

Date: 01/15/2022.

Algorithm was born at 12/28/2021 in Finland.

Introduction

What I know this is first time this kind of algorithm is presented by anybody to public. Traversing x-axis will give prime number candidates and decision is made whether it is a prime number or not using Erathostenes sieve ("knock out" those prime number candidates that can't be prime numbers). No division is needed. Biggest known Mersenne prime number has 24 million digits. With a database table having three columns (Id, PrimeNumber, ForwardX) where PrimeNumber column and ForwardX column have capacity of 24 million characters, it is possible to bypass biggest known prime number in acceptable time with a powerful super-computer. One known prime number took 14 years to compute. TimoEratosthenes algorithm needs to know as starting point that 3 is prime number. This is enough to reveal all of prime numbers! The numbers have it. The numbers have it. Look at x-axis! Traverse the x-axis!

Meaningful cryptography needs prime numbers with 400 digits. Eratosthenes sieve the old way can produce prime numbers up to 8 digits length on home computer. The technique is wrong, and computer's resources are limited. I recommend TimoEratosthenes algorithm instead when computer's resources are not limited.

Method

Showing Eratosthenes sieve as a program.

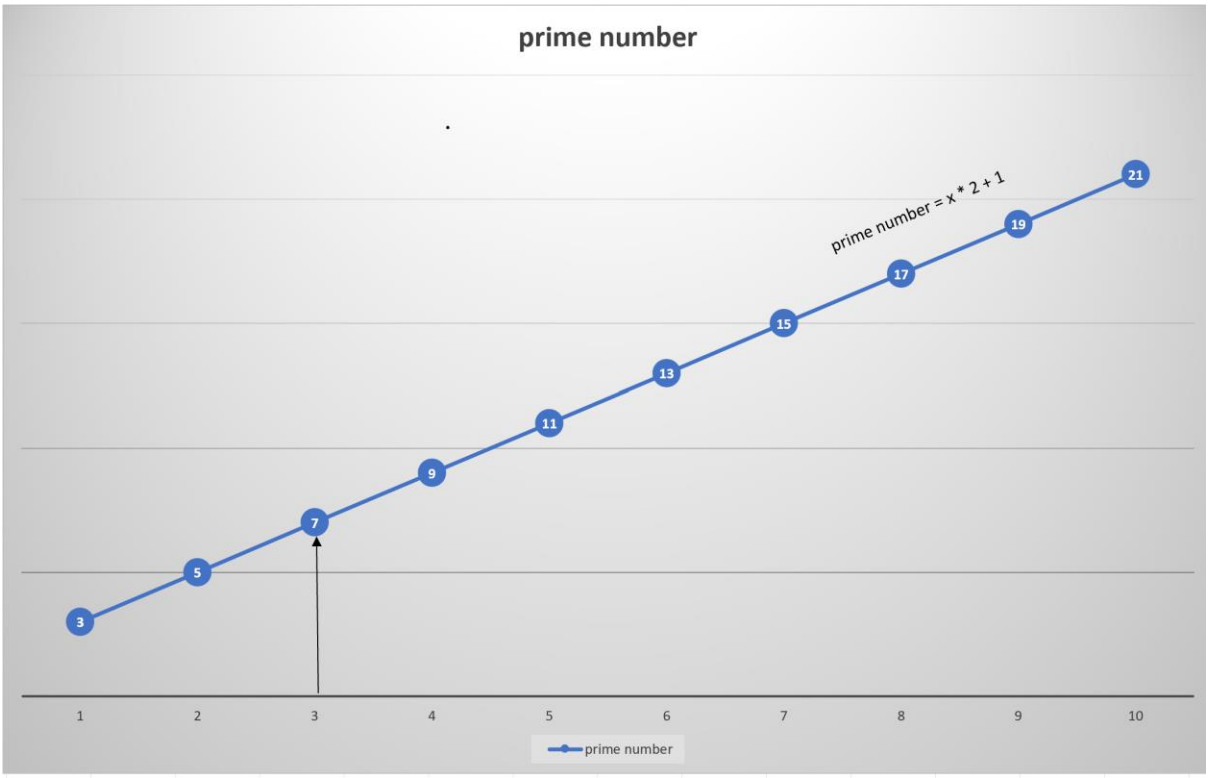
Explaining TimoEratosthes algorithm with guidelines to understanding.

Showing TimoEratosthenes algorithm as program.

Conclusion.

Programming environment

All programs are written in C# using Microsoft Visual Studio 2022 and can therefore be reproduced and studied.



Eratosthenes sieve the old way

Greek Eratosthenes lived around 240 B.C. And hunting primes has gone on for about 2460 years among mathematicians.

An example of division with modulus (%) when hunting prime numbers is below.

This is the old way. When size of `primeNumberCandidate` number grows it takes more divisions to decide if `primeNumberCandidate` is prime number.

```
using System;

namespace OldEratosthenes
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int primeNumberCandidate = 2;

            int counter = 0;

            while (true)
            {
                bool isPrimeNumber = true;
                for (int divisor = 2; divisor <= primeNumberCandidate / 2;
divisor++)
                {
                    if (primeNumberCandidate % divisor == 0)
                    {
                        isPrimeNumber = false;
                        break;
                    }
                }
```

```
                primeNumberCandidate++;

                if (!isPrimeNumber)
                {
                    continue;
                }

                counter++;
                if (counter % 1000 == 0)
                {
                    Console.Clear();
                    Console.Write($"Prime number is {primeNumberCandidate}.
Number of primenumbers is {counter}.");
                }
            }
        }
    }
}
```

Hunting prime numbers with TimoEratosthenes algorithm

There has not been a clear understanding of where prime numbers appear on the line y=2*x+1. Focus has been to find prime numbers among prime numbers. And I think that forest is not visible because of all trees. Another approach is shown here.

Think of x-axis of natural, positive numbers greater than zero and corresponding y=2*x+1 value as prime number candidate. Use Eratosthenes sieve to decide if prime number candidate is prime number.

Knowing two first prime numbers 2 and 3 is enough. The numbers have it! Or knowing 3 as prime number is enough to calculate rest of prime numbers! The numbers have it!

Considering natural, positive numbers greater than zero and a line y=2*x+1 then all prime numbers reside on the line except for two (2). Two is a special case and don't fit on line y=2*x+1.

Traversing x-axis from 1 to 2 to 3 and so on will reveal the magic prime numbers by "knocking out" prime number candidates and leaving prime numbers!

We have database table (or collection) with columns Id, PrimeNumber and ForwardX at hand.

Add one record: PrimeNumber=2, ForwardX=0. Special case.

2 is prime number.

Id=1, PrimeNumber=2, ForwardX=0.

Let x=1.

Prime number candidate that corresponds to x=1 is 2*1+1=3. And forwardx is (3*3 -1)/2=4 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=4 it is “knocked out”.

Is prime number candidate 3 prime number? Let us find out. Look through all previous records in database table for 1==ForwardX. Database table contains 1 record and no records match query 1==ForwardX.

3 is therefore prime number.

We know that when x=4 prime number candidate is 2*4+1=9 and cannot be prime number.

Add prime number candidate 3 to database table PrimeNumber=3, ForwardX=4.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=4.

Let x=2.

Prime number candidate that corresponds to x=2 is 2*2+1=5. And forwardx is (5*5 -1)/2=12 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=12 it is “knocked out”.

Is prime number candidate 5 prime number? Let us find out. Look through all previous records in database table for 2==ForwardX. Database table contains 2 records and no records match query 2==ForwardX.

5 is therefore prime number.

We know that when x=12 prime number candidate is 2*12+1=25 and cannot be prime number.

Add prime number candidate 5 to database table PrimeNumber=5, ForwardX=12.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=4.

Id=3, PrimeNumber=5, ForwardX=12.

Let x=3.

Prime number candidate that corresponds to x=3 is 2*3+1=7. And forwardx is (7*7 -1)/2=24 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=24 it is “knocked out”.

Is prime number candidate 7 prime number? Let us find out. Look through all previous records in database table for 3==ForwardX. Database table contains 3 records and no records match query 3==ForwardX.

7 is therefore prime number.

We know that when x=24 prime number candidate is 2*24+1=49 and cannot be prime number.

Add prime number candidate 7 to database table PrimeNumber=7, ForwardX=24.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=4.

Id=3, PrimeNumber=5, ForwardX=12.

Id=3, PrimeNumber=7, ForwardX=24.

Let x=4.

Prime number candidate that corresponds to x=4 is 2*4+1=9. And forwardx is (9*9 -1)/2=40 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=40 it is "knocked out".

Is prime number candidate 9 prime number? Let us find out. Look through all previous records in database table for 4==ForwardX. Database table contains 4 records and one record match query 4==ForwardX.

9 is therefore not prime number. It is "knocked out".

For future queries edit record so that ForwardX=PrimeNumber+ForwardX. And save to database.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=7.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Let x=5.

Prime number candidate that corresponds to x=5 is 2*5+1=11. And forwardx is (11*11 -1)/2=60 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=60 it is "knocked out".

Is prime number candidate 11 prime number? Let us find out. Look through all previous records in database table for 5==ForwardX. Database table contains 4 records and no records match query 5==ForwardX.

11 is therefore prime number.

We know that when x=60 prime number candidate is 2*60+1=121 and cannot be prime number.

Add prime number candidate 11 to database table PrimeNumber=11, ForwardX=60.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=7.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Let x=6.

Prime number candidate that corresponds to x=6 is 2*6+1=13. And forwardx is (13*13 -1)/2=84 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=84 it is "knocked out".

Is prime number candidate 13 prime number? Let us find out. Look through all previous records in database table for 6==ForwardX. Database table contains 5 records and no records match query 6==ForwardX.

13 is therefore prime number.

We know that when x=84 prime number candidate is 2*84+1=169 and cannot be prime number.

Add prime number candidate 13 to database table PrimeNumber=13, ForwardX=84.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=7.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Id=6, PrimeNumber=13, ForwardX=84.

Let x=7.

Prime number candidate that corresponds to x=7 is 2*7+1=15. And forwardx is (15*15 -1)/2=224 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=224 it is "knocked out".

Is prime number candidate 15 prime number? Let us find out. Look through all previous records in database table for 7==ForwardX. Database table contains 6 records and one record match query 7==ForwardX.

15 is therefore not prime number. It is "knocked out".

For future queries edit record so that ForwardX=PrimeNumber+ForwardX. And save to database.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=10.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Id=6, PrimeNumber=13, ForwardX=84.

Let x=8.

Prime number candidate that corresponds to x=8 is 2*8+1=17. And forwardx is (17*17 -1)/2=144 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=144 it is "knocked out".

Is prime number candidate 17 prime number? Let us find out. Look through all previous records in database table for 8==ForwardX. Database table contains 6 records and no records match query 8==ForwardX.

17 is therefore prime number.

We know that when x=144 prime number candidate is 2*144+1=289 and cannot be prime number.

Add prime number candidate 17 to database table PrimeNumber=17, ForwardX=144.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=10.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Id=6, PrimeNumber=13, ForwardX=84.

Id=7, PrimeNumber=17, ForwardX=144.

Let x=9.

Prime number candidate that corresponds to x=9 is 2*9+1=19. And forwardx is (19*19 -1)/2=180 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=180 it is "knocked out".

Is prime number candidate 19 prime number? Let us find out. Look through all previous records in database table for 9==ForwardX. Database table contains 7 records and no records match query 9==ForwardX.

19 is therefore prime number.

We know that when x=180 prime number candidate is 2*180+1=361 and cannot be prime number.

Add prime number candidate 19 to database table PrimeNumber=10, ForwardX=180.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=10.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Id=6, PrimeNumber=13, ForwardX=84.

Id=7, PrimeNumber=17, ForwardX=144.

Id=8, PrimeNumber=19, ForwardX=180.

Let x=10.

Prime number candidate that corresponds to x=10 is 2*10+1=21. And forwardx is (21*21 -1)/2=220 meaning that according to Eratosthenes sieve that when we traverse on x-axis to value x=220 it is "knocked out".

Is prime number candidate 21 prime number? Let us find out. Look through all previous records in database table for 10==ForwardX. Database table contains 8 records and one record match query 10==ForwardX.

21 is therefore not prime number. It is "knocked out".

For future queries edit record so that ForwardX=PrimeNumber+ForwardX. And save to database.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=13.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Id=6, PrimeNumber=13, ForwardX=84.

Id=7, PrimeNumber=17, ForwardX=144.

Id=8, PrimeNumber=19, ForwardX=180.

And so on...

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=13.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Id=6, PrimeNumber=13, ForwardX=84.

Id=7, PrimeNumber=17, ForwardX=144.

Id=8, PrimeNumber=19, ForwardX=180.

Id=9, PrimeNumber=23, ForwardX=264.

Id=1, PrimeNumber=2, ForwardX=0.

Id=2, PrimeNumber=3, ForwardX=13.

Id=3, PrimeNumber=5, ForwardX=12.

Id=4, PrimeNumber=7, ForwardX=24.

Id=5, PrimeNumber=11, ForwardX=60.

Id=6, PrimeNumber=13, ForwardX=84.

Id=7, PrimeNumber=17, ForwardX=144.

Id=8, PrimeNumber=19, ForwardX=180.

Id=9, PrimeNumber=23, ForwardX=264.

Id=10, PrimeNumber=29, ForwardX=420.

TimoEratosthenes algorithm is the new way

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace TimoEratosthenesConsoleApp
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int id = 1;

            int primeNumberCandidate = 2;

            int x = 0;

            List<Prime> primesList = new List<Prime> { new Prime { Id = id++,
PrimeNumber = primeNumberCandidate, ForwardX = x } }; //special case

            int counter = 0;

            while (true)
            {
                x++;

                bool isPrimeNumber = true;

                List<Prime> forwardXPrimesList = primesList.Where(p => p.ForwardX
== x).ToList<Prime>();
                foreach (Prime prime in forwardXPrimesList)
                {
                    Prime existingPrime = primesList.FirstOrDefault(p => p.Id ==
prime.Id);
                    if (existingPrime != null)
                    {
                        existingPrime.ForwardX = existingPrime.PrimeNumber +
existingPrime.ForwardX;
                    }
                    isPrimeNumber = false;
                }
```

```
                if (!isPrimeNumber)
                {
                    continue;
                }

                primeNumberCandidate = 2 * x + 1;

                int forwardX = (primeNumberCandidate * primeNumberCandidate - 1)
/ 2;

                primesList.Add( new Prime { Id = id++, PrimeNumber =
primeNumberCandidate, ForwardX = forwardX } );

                #region show
                Console.Clear();
                foreach (var prime in primesList)
                {
                    Console.WriteLine( prime );
                }
                Console.ReadKey();
                #endregion show

                counter++;
                if (counter % 1000 == 0)
                {
                    Console.Clear();
                    Console.Write($"Prime number is {primeNumberCandidate}.
Number of primenumbers is {counter}.");
                }
            }
        }
    }

    public class Prime
    {
        public int Id { get; set; }

        public int PrimeNumber { get; set; }

        public int ForwardX { get; set; }

        public override string ToString()
        {
            return $"id={Id} primenumber={PrimeNumber} forwardx={ForwardX}";
        }
    }
}
```

Conclusion

A new algorithm was introduced by Timo Kinnunen in this paper and explained to a degree so that reader can themselves reproduce and use it.

The numbers have it! Starting with prime number 3 and all prime numbers are revealed!

Eratosthenes sieve uses division and TimoEratosthenes algorithm not.

Saving data as strings in database table or collection makes it possible to produce big prime numbers. The limitation lies often in the chosen datatype i.e. int, uint, long, BigInteger and so on. These limitations are overridden using strings.

My hope is that next biggest prime number uses TimoEratosthenes algorithm and that mathematicians can find this algorithm useful to explain prime numbers. Today it is a jungle.

Is there hyper-computer somewhere with free time window to implement this algorithm?

That’s all folks! Thank you for reading.