



## Algorithms of Three Prime Generating Sieves Improvised Through Nonprimality of Even Numbers (Except 2)

Neeraj Anant Pande

Department of Mathematics & Statistics, Yeshwant Mahavidyalaya (College), Nanded -431605, INDIA.

**Abstract:** Three classical prime generating sieves obtained through successive refinements are well known. A divisibility property has helped third amongst them to be far better than the former two. In this paper, use of nonprimality of all positive even numbers greater than 2 has been employed in all the three sieves to obtain their improved versions. The corresponding algorithms are implemented in Java programming language to evaluate the performance enhancement.

**Keywords:** Algorithm, Sieve, Prime Number

**Mathematics Subject Classification (MSC) 2010 :** 11Y11, 11Y16, 65Y04, 65Y20, 68Q25

### I. Introduction

The Age of Computers has automated almost all tasks except possibly indigenous inventions. The computers have been immensely useful in calculation oriented tasks. Particularly, heavy number crunch demanding procedures have been the most benefitted domains due to computer applications.

One such domain is that of prime numbers from the branch Number Theory of Mathematics. The quest for primes is unending [3].

**Definition (Prime Number or Prime) :** A positive integer  $p > 1$  is said to be prime number or prime if, and only if, it has only two positive divisors, viz., 1 and itself.

A quotation by mathematician Leopold Kronecker goes to the extent of proclaiming “God made the integers; all else is the work of man” [1]. The deep analysis of integers has led to development of many new branches of mathematics. Amongst the integers, prime numbers are very dear to number theorists. And why they should not be; these primes equally deserve, being infinite in number and without any hitherto recognized regular pattern of occurrence.

There are two ways that deal with determination of a prime. In first of them, a given particular number is tested for being prime or not; while in the second one all the positive integers upto that number are tested and primes in the corresponding range are determined. The first ones are called primality tests while second types are known as sieves. In this paper, we concentrate on sieves that generate all primes in range 1 to  $n$ .

Algorithms define systematic process of accomplishing a task through computer programming [2]. This paper is concerned about development of better algorithms improvising the three classical sieves given by author himself in [4].

### II. Sieve 1 Improvised to Sieve 1.2.1

The author [4] has given the following most simple, yet most inefficient(!), sieve

```
Take an integer  $n$  larger than 1
For all values of  $k$  from 2 to  $n$ 
    For values of integer  $d$  from 2 to  $k-1$ 
        If  $d$  divides  $k$  perfectly,
            Stop checking as  $k$  is not prime
        Else
            Continue checking
    If checks do not stop for any value of  $d$  till  $k-1$ ,  $k$  is prime
```

and properly called it as the dumb simplest Sieve 1. For checking all numbers  $k$  for primality from 1 to  $n$ , the algorithm of this sieve checks all positive integers from 2 to  $k-1$  for perfect divisors of concerned  $k$ . On finding a perfect divisor, it terminates its loop after declaring that  $k$  is not prime. On the other hand, on not finding any perfect divisor till  $k-1$ , it declares that  $k$  is prime.

Here we now make use of a property that any even integer except 2 is never prime and so doesn't need be checked for primality. So the algorithm can skip all even integers with the gap of 2 as they are known to be nonprime. The improved sieve is enumerated as 1.2.1 for reference purpose and will be like the following :

```

Take an integer  $n$  larger than 1
For all values of  $k$  from 2 to  $n$  and only odd values after 2
    For values of integer  $d$  from 2 to  $k-1$ 
        If  $d$  divides  $k$  perfectly,
            Stop checking as  $k$  is not prime
        Else
            Continue checking
    If checks do not stop for any value of  $d$  till  $k-1$ ,  $k$  is prime
    
```

The underlined part in the algorithm for Sieve 1.2.1 is the change, i.e., added extra part, to Sieve 1. This reduces the number of integers to be checked in the range directly to almost half as it never tests even numbers after 2.

When a number being tested for primality in both algorithms is composite, the process of determination will terminate at the instance first divisor is encountered. For a composite number, in algorithm for Sieve 1, the first divisor may occur as early as 2, while in improved algorithm for Sieve 1.2.1, it may occur at the earliest at 3.

But if the number being tested is prime, then the conclusion will be drawn only after all  $n - 2$  testing's in both the algorithms. Naturally as the value of  $n$  being tested for primality increases, the number of testing's significantly increase equally in both algorithms.

The author has implemented algorithms for both sieves in Java programming language (see [5]) and has determined all primes in the range of 1 to 100000 ( $10^5$ ) and recorded the testing times taken by an electronic computer on 64 bit platform. The summary of the time requirements of Sieve 1.2.1 for testing all numbers in the ranges of increasing powers of 10 are given in the following table.

**TABLE 1. RUNTIMES FOR SIEVE 1.2.1**

| All numbers in the Range | The Number of Primes Found | Time Taken in Seconds |
|--------------------------|----------------------------|-----------------------|
| 1 – 10                   | 4                          | 0.000000406           |
| 1 – 100                  | 25                         | 0.000018234           |
| 1 – 1000                 | 168                        | 0.001292511           |
| 1 – 10000                | 1229                       | 0.092504415           |
| 1 – 100000               | 9592                       | 7.315765452           |

The data has been generated in all multiples of 1000 and detail table will follow later. These values will be used in graphs being plotted later during comparison with performance of other improvised algorithms.

It has already been determined [4] that for algorithm of Sieve 1, the time requirement grows rapidly with increasing range. It is comparatively reduced by introduction of the property of nonprimality of all even numbers after 2.

As in case of Sieve 1 [4], in the last range of Table 1 for Sieve 1.2.1, the number of primes increases by about 8 times than those in the previous range but the time required shoots up by about 80 times.

The comparative summary of the steps requirements in determination all numbers in the ranges of increasing powers of 10 are given in the following table.

**TABLE 2. STEPS TAKEN BY ALGORITHMS OF SIEVE 1 AND SIEVE 1.2.1**

| All numbers in the Range | The Number of Primes Found | Steps Taken |             | Steps Difference |
|--------------------------|----------------------------|-------------|-------------|------------------|
|                          |                            | Sieve 1     | Sieve 1.2.1 |                  |
| 1 – 10                   | 4                          | 15          | 11          | 4                |
| 1 – 100                  | 25                         | 1133        | 1084        | 49               |
| 1 – 1000                 | 168                        | 78022       | 77523       | 499              |
| 1 – 10000                | 1229                       | 5775223     | 5770224     | 4999             |
| 1 – 100000               | 9592                       | 455189150   | 455139151   | 49999            |

It is clear that the in the improved version, Sieve 1.2.1 doesn't even check even numbers greater than 2 for primality and hence in each range, the number of steps required reduces to 1 less than half the range as half numbers (even of those) just don't need be tested. (1 less for first pair of numbers 1 and 2 which are not tested by any sieve)

### III. Sieve 2 Improvised to Sieve 1.2.2

Making use of the fact that any divisor of any positive integer cannot exceed half of it, the author [4] has given the following Sieve 2 which is an improved version of Sieve 1 :

```

Take an integer  $n$  larger than 1
    
```

```

For all values of  $k$  from 2 to  $n$ 
  For values of integer  $d$  from 2 to  $k/2$ 
    If  $d$  divides  $k$  perfectly,
      Stop checking as  $k$  is not prime
    Else
      Continue checking
  If checks do not stop for any value of  $d$  till  $k/2$ ,  $k$  is prime

```

Here also, similar improvisation can be induced to yield Sieve 1.2.2 as follows :

```

Take an integer  $n$  larger than 1
For all values of  $k$  from 2 to  $n$  and only odd values after 2
  For values of integer  $d$  from 2 to  $k/2$ 
    If  $d$  divides  $k$  perfectly,
      Stop checking as  $k$  is not prime
    Else
      Continue checking
  If checks do not stop for any value of  $d$  till  $k/2$ ,  $k$  is prime

```

The runtimes of this version of algorithm for all numbers in the ranges of increasing powers of 10 are given in the following table.

**TABLE 3. RUNTIMES FOR SIEVE 1.2.2**

| All numbers in the Range | The Number of Primes Found | Time Taken in Seconds |
|--------------------------|----------------------------|-----------------------|
| 1 – 10                   | 4                          | 0.000000405           |
| 1 – 100                  | 25                         | 0.000010129           |
| 1 – 1000                 | 168                        | 0.000634505           |
| 1 – 10000                | 1229                       | 0.046327369           |
| 1 – 100000               | 9592                       | 3.671328082           |

The detail table for the data in all multiples of 1000 will follow later. These values will be used for plotting graphs and comparing performance of improvised algorithms.

As going from Sieve 1 to Sieve 1.2.1, in this refinement too, even numbers after 2 are omitted from being tested for primes; as it is well known that they are never so.

Next follows the table containing the comparative analysis of number of steps required by Sieve 2 and its improved version Sieve 1.2.2.

**TABLE 4. STEPS TAKEN BY ALGORITHMS OF SIEVE 2 AND SIEVE 1.2.2**

| All numbers in the Range | The Number of Primes Found | Steps Taken |             | Steps Difference |
|--------------------------|----------------------------|-------------|-------------|------------------|
|                          |                            | Sieve 2     | Sieve 1.2.2 |                  |
| 1 – 10                   | 4                          | 9           | 5           | 4                |
| 1 – 100                  | 25                         | 616         | 567         | 49               |
| 1 – 1000                 | 168                        | 40043       | 39544       | 499              |
| 1 – 10000                | 1229                       | 2907640     | 2902641     | 4999             |
| 1 – 100000               | 9592                       | 227995678   | 227945679   | 49999            |

Same difference as in the previous case is seen in the step requirements of Sieve 2 and Sieve 1.2.2.

#### IV. Sieve 3 Improvised to Sieve 1.2.3

Employing the further property of positive integers that their positive divisors occur in pairs in which at least one is less than or equal to their square root, Sieve 2 was refined by the author [4] to following Sieve 3 :

```

Take an integer  $n$  larger than 1
For all values of  $k$  from 2 to  $n$ 
  For values of integer  $d$  from 2 to  $\sqrt{k}$ 
    If  $d$  divides  $k$  perfectly,
      Stop checking as  $k$  is not prime
    Else
      Continue checking
  If checks do not stop for any value of  $d$  till  $\sqrt{k}$ ,  $k$  is prime

```

On the same lines of bettering Sieve 1 to Sieve 1.2.1 and Sieve 2 to Sieve 1.2.2, here there is way for Sieve 1.2.3 :

```

Take an integer  $n$  larger than 1

```

For all values of  $k$  from 2 to  $n$  and only odd values after 2

For values of integer  $d$  from 2 to  $\sqrt{k}$

If  $d$  divides  $k$  perfectly,

Stop checking as  $k$  is not prime

Else

Continue checking

If checks do not stop for any value of  $d$  till  $\sqrt{k}$ ,  $k$  is prime

Now for this algorithm, we get the best runtimes given in the following table.

**TABLE 5. RUNTIMES FOR SIEVE 1.2.3**

| All numbers in the Range | The Number of Primes Found | Time Taken in Seconds |
|--------------------------|----------------------------|-----------------------|
| 1 – 10                   | 4                          | 0.000000405           |
| 1 – 100                  | 25                         | 0.000004052           |
| 1 – 1000                 | 168                        | 0.000084682           |
| 1 – 10000                | 1229                       | 0.002088680           |
| 1 – 100000               | 9592                       | 0.044884540           |

Due to discard of all even numbers after 2 in primality test, the reduction in the number of steps as compared to Sieve 3 is similar to that in previous two improvisations.

**TABLE 6. STEPS TAKEN BY ALGORITHMS OF SIEVE 3 AND SIEVE 1.2.3**

| All numbers in the Range | The Number of Primes Found | Steps Taken |             | Steps Difference |
|--------------------------|----------------------------|-------------|-------------|------------------|
|                          |                            | Sieve 2     | Sieve 1.2.2 |                  |
| 1 – 10                   | 4                          | 8           | 4           | 4                |
| 1 – 100                  | 25                         | 236         | 187         | 49               |
| 1 – 1000                 | 168                        | 5288        | 4789        | 499              |
| 1 – 10000                | 1229                       | 117527      | 112528      | 4999             |
| 1 – 100000               | 9592                       | 2745694     | 2695695     | 49999            |

## V. Comparative Analysis

In all, there are six sieves and their corresponding algorithms which are the candidates for one-to-one comparison. Of these, the Sieves 1, 2 and 3 have already been analyzed for performance [4]. So, we give exhaustive data of runtimes of all improved version algorithms for ranges of numbers 1 – 10, 1 – 100 and then 1 to all successive multiples of 1000 till 100000.

**TABLE 7. COMPARATIVE RUNTIMES IN SECONDS FOR THE THREE IMPROVED SIEVES**

| Numbers Range | Number of Primes | Time Taken by Sieve 1.2.1 | Time Taken by Sieve 1.2.2 | Time Taken by Sieve 1.2.3 |
|---------------|------------------|---------------------------|---------------------------|---------------------------|
| 1-10          | 4                | 0.000000406               | 0.000000405               | 0.000000405               |
| 1-100         | 25               | 0.000018234               | 0.000010129               | 0.000004052               |
| 1-1000        | 168              | 0.001292511               | 0.000634505               | 0.000084682               |
| 1-2000        | 303              | 0.004536751               | 0.002334217               | 0.000247157               |
| 1-3000        | 430              | 0.009750145               | 0.004898978               | 0.000448529               |
| 1-4000        | 550              | 0.016466335               | 0.008286246               | 0.000672186               |
| 1-5000        | 669              | 0.025078341               | 0.012618788               | 0.000895843               |
| 1-6000        | 783              | 0.035153436               | 0.017657551               | 0.001108155               |
| 1-7000        | 900              | 0.047422557               | 0.023795960               | 0.001339510               |
| 1-8000        | 1007             | 0.060301062               | 0.030256890               | 0.001576538               |
| 1-9000        | 1117             | 0.075425860               | 0.037800854               | 0.001824910               |
| 1-10000       | 1229             | 0.092504415               | 0.046327369               | 0.002088680               |
| 1-11000       | 1335             | 0.110509606               | 0.055545518               | 0.002359743               |
| 1-12000       | 1438             | 0.129591754               | 0.065159118               | 0.002631616               |
| 1-13000       | 1547             | 0.151732572               | 0.080935033               | 0.002919291               |
| 1-14000       | 1652             | 0.174557327               | 0.092903514               | 0.003213854               |
| 1-15000       | 1754             | 0.198237812               | 0.105369550               | 0.003514495               |
| 1-16000       | 1862             | 0.225764223               | 0.119241950               | 0.003827291               |
| 1-17000       | 1960             | 0.252065384               | 0.132367612               | 0.004138061               |
| 1-18000       | 2064             | 0.281229920               | 0.147767525               | 0.004463822               |
| 1-19000       | 2158             | 0.308988497               | 0.162353035               | 0.004779048               |
| 1-20000       | 2262             | 0.341589732               | 0.178647980               | 0.005123852               |
| 1-21000       | 2360             | 0.373647626               | 0.194757354               | 0.005459743               |
| 1-22000       | 2464             | 0.409811569               | 0.212751605               | 0.005807789               |
| 1-23000       | 2564             | 0.446106789               | 0.230928591               | 0.006162317               |
| 1-24000       | 2668             | 0.485232971               | 0.250740865               | 0.006523329               |
| 1-25000       | 2762             | 0.522181740               | 0.270091644               | 0.006881910               |
| 1-26000       | 2860             | 0.562421749               | 0.290229679               | 0.007248999               |

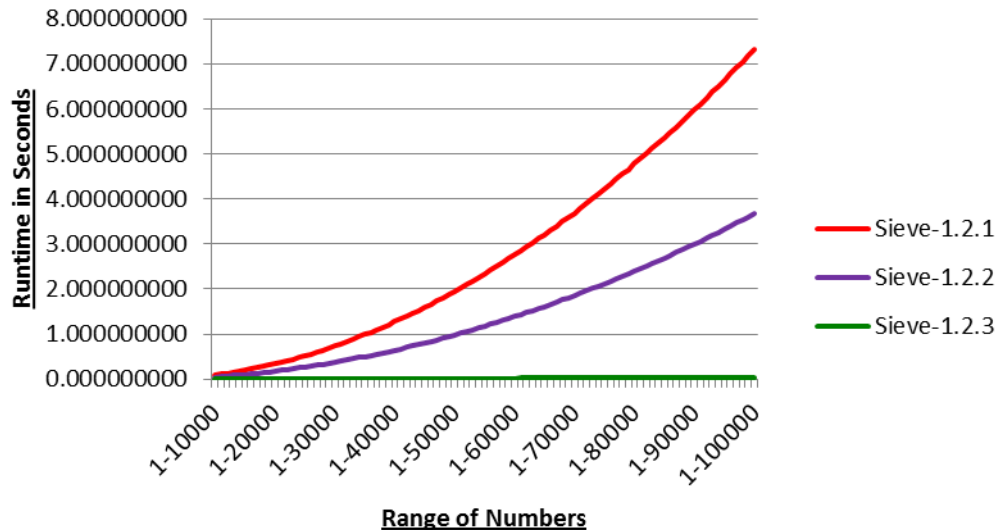
| <b>Numbers<br/>Range</b> | <b>Number of<br/>Primes</b> | <b>Time Taken by<br/>Sieve 1.2.1</b> | <b>Time Taken by<br/>Sieve 1.2.2</b> | <b>Time Taken by<br/>Sieve 1.2.3</b> |
|--------------------------|-----------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| 1-27000                  | 2961                        | 0.605867508                          | 0.312386704                          | 0.007632700                          |
| 1-28000                  | 3055                        | 0.647703505                          | 0.333329823                          | 0.008005462                          |
| 1-29000                  | 3153                        | 0.692814940                          | 0.355753048                          | 0.008391189                          |
| 1-30000                  | 3245                        | 0.736291492                          | 0.377940056                          | 0.008772865                          |
| 1-31000                  | 3340                        | 0.783163415                          | 0.401240487                          | 0.009159807                          |
| 1-32000                  | 3432                        | 0.829651637                          | 0.424819679                          | 0.009554043                          |
| 1-33000                  | 3538                        | 0.884980384                          | 0.452619584                          | 0.009974210                          |
| 1-34000                  | 3638                        | 0.938900741                          | 0.479545930                          | 0.010388299                          |
| 1-35000                  | 3732                        | 0.991188646                          | 0.505987687                          | 0.010803199                          |
| 1-36000                  | 3824                        | 1.043310429                          | 0.532193226                          | 0.011215262                          |
| 1-37000                  | 3923                        | 1.102147592                          | 0.561363029                          | 0.011638265                          |
| 1-38000                  | 4017                        | 1.159735193                          | 0.589708705                          | 0.012062484                          |
| 1-39000                  | 4107                        | 1.215269364                          | 0.617542240                          | 0.012483462                          |
| 1-40000                  | 4203                        | 1.276086214                          | 0.648026433                          | 0.012919431                          |
| 1-41000                  | 4291                        | 1.333687996                          | 0.676746896                          | 0.013345270                          |
| 1-42000                  | 4392                        | 1.400898517                          | 0.710527597                          | 0.013803523                          |
| 1-43000                  | 4494                        | 1.470836679                          | 0.745387685                          | 0.014306751                          |
| 1-44000                  | 4579                        | 1.530207457                          | 0.775077329                          | 0.014724892                          |
| 1-45000                  | 4675                        | 1.599680883                          | 0.809855167                          | 0.015176663                          |
| 1-46000                  | 4761                        | 1.662645568                          | 0.841379446                          | 0.015594399                          |
| 1-47000                  | 4851                        | 1.730525843                          | 0.875090457                          | 0.016042118                          |
| 1-48000                  | 4946                        | 1.803343182                          | 0.911206590                          | 0.016505234                          |
| 1-49000                  | 5035                        | 1.873218947                          | 0.946263189                          | 0.016947281                          |
| 1-50000                  | 5133                        | 1.950941341                          | 0.985181516                          | 0.017420931                          |
| 1-51000                  | 5222                        | 2.023321496                          | 1.021334925                          | 0.017880400                          |
| 1-52000                  | 5319                        | 2.103731826                          | 1.061603297                          | 0.018362559                          |
| 1-53000                  | 5408                        | 2.179073409                          | 1.099083251                          | 0.018824459                          |
| 1-54000                  | 5500                        | 2.259048580                          | 1.138607721                          | 0.019296488                          |
| 1-55000                  | 5590                        | 2.337883181                          | 1.178244830                          | 0.019774595                          |
| 1-56000                  | 5683                        | 2.420660546                          | 1.219955627                          | 0.020258780                          |
| 1-57000                  | 5782                        | 2.510444611                          | 1.265155796                          | 0.020760387                          |
| 1-58000                  | 5873                        | 2.594357278                          | 1.307651013                          | 0.021247003                          |
| 1-59000                  | 5963                        | 2.678940915                          | 1.349989022                          | 0.021739696                          |
| 1-60000                  | 6057                        | 2.769671066                          | 1.395036845                          | 0.022255890                          |
| 1-61000                  | 6145                        | 2.854905416                          | 1.437834728                          | 0.022750204                          |
| 1-62000                  | 6232                        | 2.940698502                          | 1.481098563                          | 0.023257484                          |
| 1-63000                  | 6320                        | 3.031404747                          | 1.525209620                          | 0.023767195                          |
| 1-64000                  | 6413                        | 3.126248645                          | 1.572790196                          | 0.024297569                          |
| 1-65000                  | 6493                        | 3.208965234                          | 1.614446295                          | 0.024789047                          |
| 1-66000                  | 6591                        | 3.312772020                          | 1.667754432                          | 0.025328336                          |
| 1-67000                  | 6675                        | 3.402443042                          | 1.713153137                          | 0.025840883                          |
| 1-68000                  | 6774                        | 3.509783769                          | 1.766826337                          | 0.026394353                          |
| 1-69000                  | 6854                        | 3.598528965                          | 1.811308940                          | 0.026890288                          |
| 1-70000                  | 6935                        | 3.688749405                          | 1.857063389                          | 0.027401215                          |
| 1-71000                  | 7033                        | 3.799862722                          | 1.912574871                          | 0.028147548                          |
| 1-72000                  | 7128                        | 3.908996351                          | 1.967044241                          | 0.028695751                          |
| 1-73000                  | 7218                        | 4.014002052                          | 2.019355646                          | 0.029232203                          |
| 1-74000                  | 7301                        | 4.112257122                          | 2.068500604                          | 0.029749612                          |
| 1-75000                  | 7393                        | 4.222278086                          | 2.123802205                          | 0.030290926                          |
| 1-76000                  | 7484                        | 4.33266609                           | 2.178984684                          | 0.030844396                          |
| 1-77000                  | 7567                        | 4.435250915                          | 2.229929431                          | 0.031359374                          |
| 1-78000                  | 7662                        | 4.553377415                          | 2.289264959                          | 0.031925404                          |
| 1-79000                  | 7746                        | 4.660110784                          | 2.342310947                          | 0.032464287                          |
| 1-80000                  | 7837                        | 4.776150225                          | 2.400471872                          | 0.033025050                          |
| 1-81000                  | 7925                        | 4.890521153                          | 2.457363786                          | 0.033574468                          |
| 1-82000                  | 8017                        | 5.010846541                          | 2.517426604                          | 0.034135637                          |
| 1-83000                  | 8106                        | 5.128938197                          | 2.576758486                          | 0.034697210                          |
| 1-84000                  | 8190                        | 5.241493939                          | 2.632841670                          | 0.035254327                          |
| 1-85000                  | 8277                        | 5.359394352                          | 2.692156129                          | 0.035924893                          |
| 1-86000                  | 8362                        | 5.476140825                          | 2.750696703                          | 0.036505509                          |
| 1-87000                  | 8450                        | 5.599071087                          | 2.812422767                          | 0.037086125                          |
| 1-88000                  | 8543                        | 5.729572054                          | 2.877901567                          | 0.037681328                          |
| 1-89000                  | 8619                        | 5.837839418                          | 2.931894451                          | 0.038227504                          |
| 1-90000                  | 8713                        | 5.973040016                          | 2.999929099                          | 0.038830000                          |
| 1-91000                  | 8802                        | 6.103092049                          | 3.064765696                          | 0.039413453                          |
| 1-92000                  | 8887                        | 6.228005645                          | 3.127284284                          | 0.039995285                          |
| 1-93000                  | 8984                        | 6.372415479                          | 3.198993873                          | 0.040602238                          |
| 1-94000                  | 9070                        | 6.501452142                          | 3.263947971                          | 0.041186501                          |
| 1-95000                  | 9157                        | 6.634097703                          | 3.330335985                          | 0.041833567                          |

| <u>Numbers Range</u> | <u>Number of Primes</u> | <u>Time Taken by Sieve 1.2.1</u> | <u>Time Taken by Sieve 1.2.2</u> | <u>Time Taken by Sieve 1.2.3</u> |
|----------------------|-------------------------|----------------------------------|----------------------------------|----------------------------------|
| 1-96000              | 9252                    | 6.780153764                      | 3.402797985                      | 0.042466857                      |
| 1-97000              | 9336                    | 6.910726447                      | 3.467893489                      | 0.043071379                      |
| 1-98000              | 9418                    | 7.039057293                      | 3.532274264                      | 0.043663341                      |
| 1-99000              | 9505                    | 7.176810497                      | 3.601510662                      | 0.044275966                      |
| 1-100000             | 9592                    | 7.315765452                      | 3.671328082                      | 0.044884540                      |

Comparison of these time requirements with those for Sieves 1, 2, 3 given in [4] cannot be one-to-one. The values here are found to be very small. The execution in [4] was on a 32 bit electronic computer while this on a 64 bit machine. Also the version of the operating system is advanced for present work.

And the graph makes the trend clear in a glimpse.

**FIGURE 1. RUNTIME COMPARISON OF THREE IMPROVED SIEVES**



It is necessary to note the fact that these runtimes correspond to the data during a specific execution of the algorithms. As implementation of no programming language presently guarantees precision/accuracy of the time measurement of order  $10^{-9}$  seconds, a minor variations of the order of  $10^{-6}$  seconds may be observed during repeated executions.

From the data and its plotted version – the graph – it is obvious beyond doubt that like Sieves 1, 2 and 3 were successive refinements, so is the case of Sieves 1.2.1, 1.2.2. & 1.2.3, each also being respectively better than Sieve 1, 2 & 3 and the last one being the best amongst all the six.

### References

- [1] Eric Temple Bell, “Men of Mathematics”, Simon and Schuster, New York, p. 477, 1986.
- [2] Donald E. Knuth, “The Art of Computer Programming”, Volume 1: Fundamental Algorithms, Addison-Wesley, Reading, MA, 1968.
- [3] Evan Niven, Herbert S. Zuckerman, Hugu L. Montgomery, “An Introduction to the Theory of Numbers”, John Wiley & Sons Inc., U.K., 2008.
- [4] Neeraj Anant Pande, “Evolution of Algorithms: A Case Study of Three Prime Generating Sieves”, Journal of Science and Arts, 13, 3(24), 267-276, 2013.
- [5] Herbert Schildt, “Java : The Complete Reference, 7<sup>th</sup> Edition”, Tata McGraw - Hill Education, 2006

### Acknowledgments

The author is thankful to the Java (7 Update 25) Programming Language Development Team and the NetBeans IDE 7.3.1 Development Team whose software were useful in implementing the algorithms on their platforms.

Thanks will also be due to University Grants Commission (U.G.C.), New Delhi of the Government of India for funding awaited for this work under a proposed Research Project.