

## AN ESTIMATE FOR THE NUMBER OF INTEGERS WITHOUT LARGE PRIME FACTORS

KOJI SUZUKI

ABSTRACT.  $\Psi(x, y)$  denotes the number of positive integers  $\leq x$  and free of prime factors  $> y$ . Hildebrand and Tenenbaum provided a good approximation of  $\Psi(x, y)$ . However, their method requires the solution  $\alpha = \alpha(x, y)$  to the equation  $\sum_{p \leq y} \log p / (p^\alpha - 1) = \log x$ , and therefore it needs a large amount of time for the numerical solution of the above equation for large  $y$ . Hildebrand also showed  $1 - \xi_u / \log y$  approximates  $\alpha$  for  $1 \leq u \leq y / (2 \log y)$ , where  $u = (\log x) / \log y$  and  $\xi_u$  is the unique solution to  $e^{\xi_u} = 1 + u \xi_u$ . Let  $E(i)$  be defined by  $E(0) = \log u$ ;  $E(i) = \log u + \log(E(i-1) + 1/u)$  for  $i > 0$ . We show  $E(m)$  approximates  $\xi_u$ , and  $1 - E(m) / \log y$  also approximates  $\alpha$ , where  $m = \lceil (\log u + \log \log y) / \log \log u \rceil + 1$ . Using these approximations, we give a simple method which approximates  $\Psi(x, y)$  within a factor  $1 + O(1/u + 1/\log y)$  in the range  $(\log \log x)^{5/3+\epsilon} < \log y < (\log x)/e$ , where  $\epsilon$  is any positive constant.

### 1. INTRODUCTION

Let  $\Psi(x, y)$  be the number of positive integers  $\leq x$  and free of prime factors  $> y$ . Estimates for  $\Psi(x, y)$  are very useful for many number-theoretic algorithms and modern cryptography. The behavior of  $\Psi(x, y)$  has been investigated by many authors ([2], [3], [4], [5], [6], [7], [8], [10], [13]). We see good summaries for the investigations of  $\Psi(x, y)$  in [9] and [11].

Dickman [5] showed that the probability that a random integer between 1 and  $x$  has no prime factors exceeding  $x^{1/u}$  ( $0 < u$ ) approaches the value  $\rho(u)$  as  $x \rightarrow \infty$ , where  $u = (\log x) / \log y$  and  $\rho(u)$  is the unique solution to the following equations:

$$\begin{aligned} u\rho'(u) + \rho(u-1) &= 0 & (1 < u); \\ \rho(u) &= 1 & (0 \leq u \leq 1). \end{aligned}$$

Hildebrand [7] showed that for any fixed positive number  $\epsilon$ ,  $3 \leq x$ , and  $1 \leq u \leq \log x / (\log \log x)^{5/3+\epsilon}$ ,

$$\Psi(x, y) = x\rho(u) \left( 1 + O_\epsilon \left( \frac{\log(u+1)}{\log y} \right) \right)$$

holds. The estimate  $\Psi(x, y) \approx x\rho(u)$  is, in practice, accurate only for small  $u$ . Hildebrand's result above shows this fact in theory, and Hunter and Sorenson [10] gave some experimental data to show this fact.

---

Received by the editor April 10, 2001 and, in revised form, September 12, 2002.

2000 *Mathematics Subject Classification*. Primary 11N25; Secondary 11Y05.

*Key words and phrases*. Computational number theory, analytic number theory, asymptotic estimates, factoring problem.

©2003 American Mathematical Society

Hildebrand and Tenenbaum [8] gave an estimate of  $\Psi(x, y)$  which is accurate for large  $u$ . They showed that uniformly for  $2 \leq y \leq x$ ,

$$(1.1) \quad \Psi(x, y) = h(x, y, \alpha_u) \left( 1 + O\left(\frac{\log y}{\log x}\right) + O\left(\frac{\log y}{y}\right) \right),$$

where

$$h(x, y, s) = \frac{x^s \prod_{p \leq y} (1 - p^{-s})^{-1}}{s \sqrt{2\pi \phi_2(s, y)}},$$

$$\phi_2(s, y) = \sum_{p \leq y} \frac{p^s (\log p)^2}{(p^s - 1)^2},$$

and  $\alpha_u$  is the unique solution to the equation

$$(1.2) \quad - \sum_{p \leq y} \frac{\log p}{p^{\alpha_u} - 1} + \log x = 0.$$

Hunter and Sorenson [10] provided the following algorithm to evaluate Hildebrand and Tenenbaum's approximation.

**Algorithm HS** [10].

- (1) Find all primes  $\leq y$ .
- (2) Find  $\alpha'_u$  satisfying  $|\alpha_u - \alpha'_u| < \min\{0.0001, 1/(\bar{u} \log x)\}$ , where  $\alpha_u$  is the unique solution to (1.2) and  $\bar{u} = \min\{u, y/\log y\}$ .
- (3) Output  $h(x, y, \alpha'_u)$ .

Step (1) can be done by some sieve algorithms (for example, see [1] and [12]) using  $O(y/\log \log y)$  operations. Step (2) requires bisection, and it can be performed using  $O(y(\log \log x)/\log y)$  operations [10]. Hence, the complexity of Algorithm HS is given by

$$(1.3) \quad O\left(y \left(\frac{\log \log x}{\log y} + \frac{1}{\log \log y}\right)\right).$$

Hunter and Sorenson also showed that Newton's method can improve the complexity of step (2) in Algorithm HS. To prove quadratic convergence for Newton's method, one needs a preliminary search by bisection for obtaining a suitable starting point, and it costs  $O(y(\log((\log x)/\bar{u})/\log y))$  operations. Then, if  $\log x \leq y$ , the total running time of this algorithm is dropped to  $O(y/\log \log y)$ , and if  $\log x > y$ , it corresponds to (1.3). Although Newton's method can reduce the running time of step (2), one can only prove quadratic convergence.

In this paper, we give an estimate of  $\alpha_u$ ,

$$\hat{\alpha}_u = 1 - \frac{E(m)}{\log y}; \quad m = \left\lceil \frac{\log u + \log \log y}{\log \log u} \right\rceil + 1,$$

where

$$(1.4) \quad E(i) = \begin{cases} \log u & \text{for } i = 0; \\ \log u + \log(E(i-1) + 1/u) & \text{for } i > 0. \end{cases}$$

The above  $E(m)$  provides an approximation of  $\xi_u$ , where  $\xi_u$  is the unique solution to the equation

$$e^{\xi_u} = 1 + u\xi_u,$$

and  $E(m)$  can be used in some other estimate formulas for  $\Psi(x, y)$  (for example, see Section 2 of [9]). We prove that the error term of our estimate for  $\alpha_u$  is  $O(1/(u(\log y)^2))$  for  $x \rightarrow \infty$ , in the range

$$(1.5) \quad (\log \log x)^{5/3+\epsilon} < \log y < (\log x)/e,$$

where  $\epsilon$  is any fixed positive number. Furthermore, using this estimate, we show that in the above range

$$(1.6) \quad \Psi(x, y) = h(x, y, \hat{\alpha}_u) \left( 1 + O\left(\frac{1}{u} + \frac{1}{\log y}\right) \right),$$

when  $x$  tends to infinity. Using (1.6), we can compute the estimates of  $\Psi(x, y)$  without the numerical solution of the equation (1.2). Our algorithm for computing the estimate for  $\Psi(x, y)$  is described as follows.

**Algorithm Modified HT.**

- (1) Find all primes  $\leq y$ .
- (2) Set  $E(0) = \log u$  and  $m = \lceil (\log u + \log \log y) / \log \log u \rceil + 1$ .
- (3) Compute  $E(m)$  using (1.4).
- (4) Set  $\hat{\alpha}_u = 1 - E(m) / \log y$ .
- (5) Output  $h(x, y, \hat{\alpha}_u)$ .

Calculating  $\hat{\alpha}_u$  takes time proportional to  $m = O(\log y)$  in the range (1.5). Therefore the total running time is dominated by step (1), and the complexity of Algorithm Modified HT is given by

$$O\left(\frac{y}{\log \log y}\right).$$

In the range (1.5), the above complexity is the same as (1.3). However, since Algorithm Modified HT does not require Newton’s method or bisection, it provides a practical improvement on Algorithm HS. Moreover, if all primes up to  $y$  are precomputed, the complexity of our new algorithm is

$$O\left(\frac{y}{\log y}\right).$$

Then, compared with the conventional algorithms, our new algorithm provides some advantages in rapidly computing multiple values of  $\Psi(x, y')$  for different values of  $x$  and  $y' < y$  after precomputing all primes  $\leq y$ .

The structure of this paper is as follows. In Section 2 we give our estimates for  $\xi_u$  and  $\alpha_u$ , and using these estimates we provide our approximation to  $\Psi(x, y)$ . In Section 3 we give numerical results for showing that Algorithm Modified HT provides an accurate approximation to  $\Psi(x, y)$  and is faster than Algorithm HS.

2. ESTIMATE FOR  $\Psi(x, y)$

Hildebrand and Tenenbaum [8, formula (7.8)] showed that for sufficiently large  $x$ ,

$$(2.1) \quad \alpha_u = 1 - \frac{\xi_u}{\log y} + O\left(\frac{1}{u(\log y)^2}\right) + O(\exp(-(\log y)^{3/5-\epsilon_0})),$$

in the range  $(\log x)^{1+\epsilon_0} < y \leq x$ , where  $\epsilon_0$  is any positive constant and  $\xi_u$  is the unique solution of the equation

$$e^{\xi_u} = 1 + u\xi_u.$$

Then, we have

$$(2.2) \quad \xi_u = \log u + \log \left( \xi_u + \frac{1}{u} \right).$$

Using (2.1), in our range, we can obtain the following lemma, which provides a simpler formula to estimate  $\alpha_u$ .

**Lemma 2.1.** *For sufficiently large  $x$ , we have*

$$(2.3) \quad \alpha_u = 1 - \frac{\xi_u}{\log y} + O \left( \frac{1}{u(\log y)^2} \right),$$

in the range  $\log y > (\log \log x)^{5/3+\epsilon}$ , where  $\epsilon$  is any fixed positive number.

*Proof.* We first see that since  $y > (\log x)^{(\log \log x)^{2/3+\epsilon}}$ , we have  $y \rightarrow \infty$  for  $x \rightarrow \infty$ . Let  $\epsilon'$  be a number such that

$$0 < \epsilon' < \frac{3}{5}\epsilon,$$

and  $\epsilon_0$  be a number such that

$$\epsilon_0 = \frac{3}{5} \left( 1 - \frac{1 + \epsilon'}{1 + 3\epsilon/5} \right) > 0.$$

Then, in the range  $\log y > (\log \log x)^{5/3+\epsilon}$ ,

$$(\log y)^{3/5-\epsilon_0} > (\log \log x)^{1+\epsilon'},$$

for sufficiently large  $y$ . Since  $\log \log y = o((\log y)^{3/5-\epsilon_0})$ , we have

$$(\log y)^{3/5-\epsilon_0} - \log \log y > \left( (\log y)^{3/5-\epsilon_0} \right)^{\frac{1}{1+\epsilon'}} > \log \log x,$$

for sufficiently large  $x$ . Hence,

$$(\log y)^{3/5-\epsilon_0} > \log \log x + \log \log y = \log u + 2 \log \log y.$$

This implies that

$$\exp \left\{ -(\log y)^{3/5-\epsilon_0} \right\} < \frac{1}{u(\log y)^2}.$$

From the above equation and (2.1), we obtain the proof of this lemma.  $\square$

For  $u > e$ ,  $i \geq 0$ , we define  $E(i)$  by (1.4). We also define  $v_u$  for  $u > e$  as follows:

$$v_u = \xi_u + \frac{1}{u}.$$

The following lemma shows that  $E(i)$  approximates  $\xi_u$  (see Remark 2.4 for a geometric interpretation for explaining  $E(i)$  converges to  $\xi_u$ ). This estimate for  $\xi_u$  is utilized in the proof of Theorem 2.3, which is for providing our algorithm to approximate  $\Psi(x, y)$ .

**Lemma 2.2.** *For  $u > e$ ,  $n \geq 1$ ,*

$$(2.4) \quad 0 < \xi_u - E(n) < \frac{1}{(\log u)^{n-1}}.$$

*Proof.* We define  $z_u(n)$  for  $u > e, n \geq 0$  as

$$z_u(n) = E(n) + \frac{1}{u}.$$

It is clear that for  $u > e, n \geq 0,$

$$(2.5) \quad z_u(n) > \log u > 1.$$

We also define  $w_{s,u}(n)$  for  $u > e, s \geq 1, n \geq 0$  as follows:

$$\begin{aligned} w_{s,u}(0) &= s; \\ w_{s,u}(n) &= 1 + \frac{1}{z_u(n-1)} \log w_{s,u}(n-1) \quad \text{for } n \geq 1. \end{aligned}$$

Note that  $z_u(n) \neq 0$  from (2.5), and  $w_{s,u}(n) \geq 1$  for  $s \geq 1.$  We define  $r_{s,u}(n)$  for  $u > e, s \geq 1, n \geq 0$  as follows:

$$r_{s,u}(n) = z_u(n) + \log w_{s,u}(n).$$

Then, for  $n \geq 1,$  we have

$$(2.6) \quad r_{s,u}(n) = \log u + \frac{1}{u} + \log r_{s,u}(n-1).$$

Hence, from (2.5), we have for  $n \geq 1,$

$$\begin{aligned} r_{v_u,u}(n) - r_{1,u}(n) &= \log w_{v_u,u}(n) = \log \left( 1 + \frac{1}{z_u(n-1)} \log w_{v_u,u}(n-1) \right) \\ &\leq \frac{1}{z_u(n-1)} \log w_{v_u,u}(n-1) \\ &< \frac{1}{\log u} \log w_{v_u,u}(n-1) \\ &\vdots \\ &< \frac{1}{(\log u)^n} \log w_{v_u,u}(0) = \frac{\log \left( \xi_u + \frac{1}{u} \right)}{(\log u)^n}. \end{aligned}$$

Since for  $u > e$

$$e^{2 \log u} > 2u \log u + 1,$$

we have  $2 \log u > \xi_u$  for  $u > e.$  Therefore, for  $u > e$  and  $n \geq 1,$

$$(2.7) \quad r_{v_u,u}(n) - r_{1,u}(n) < \frac{\log \left( 2 \log u + \frac{1}{u} \right)}{(\log u)^n} < \frac{1}{(\log u)^{n-1}}.$$

Next, we show that, for  $n \geq 0,$

$$(2.8) \quad r_{v_u,u}(n) = v_u.$$

From (2.2), we have

$$r_{v_u,u}(0) = \log u + \frac{1}{u} + \log \left( \xi_u + \frac{1}{u} \right) = v_u.$$

Assuming  $r_{v_u,u}(n) = v_u,$  then, from (2.2) and (2.6), we have

$$r_{v_u,u}(n+1) = \log u + \frac{1}{u} + \log r_{v_u,u}(n) = \log u + \frac{1}{u} + \log \left( \xi_u + \frac{1}{u} \right) = v_u.$$

Therefore, the equation (2.8) holds for  $n \geq 0.$

From the definitions of  $r_{s,u}(n)$  and  $z_u(n)$ , it is clear that

$$r_{1,u}(n) = E(n) + \frac{1}{u}.$$

Hence, from (2.7), (2.8), and the above equation, we have for  $n \geq 1$

$$\xi_u - E(n) = r_{v_u,u}(n) - r_{1,u}(n) < \frac{1}{(\log u)^{n-1}}.$$

On the other hand, since  $w_{v_u,u} > 1$  for  $u > e$ ,

$$\xi_u - E(n) = r_{v_u,u}(n) - r_{1,u}(n) = \log w_{v_u,u}(n) > 0$$

holds. This completes the proof of Lemma 2.2.  $\square$

Finally, we give our main result. By the following theorem and (1.1), we can obtain a method which approximates  $\Psi(x, y)$  within a factor  $1 + O(1/u + 1/\log y)$ .

**Theorem 2.3.** *For sufficiently large  $x$ ,*

$$h(x, y, \hat{\alpha}_u) = h(x, y, \alpha_u) \left( 1 + O\left(\frac{1}{\log y}\right) \right),$$

*in the range  $(\log \log x)^{5/3+\epsilon} < \log y < (\log x)/e$ , where  $\epsilon$  is any fixed positive number and*

$$\hat{\alpha}_u = 1 - \frac{E(m)}{\log y}; \quad m = \left\lceil \frac{\log u + \log \log y}{\log \log u} \right\rceil + 1.$$

*Proof.* From Lemma 2.2, for  $u = (\log x)/\log y > e$ ,

$$0 < \xi_u - E(m) < \frac{1}{u \log y}.$$

Hence, from Lemma 2.1, we have in the range  $(\log \log x)^{5/3+\epsilon} < \log y < (\log x)/e$ ,

$$\hat{\alpha}_u = \alpha_u + O\left(\frac{1}{u(\log y)^2}\right),$$

for sufficiently large  $x$ . In the same manner as Lemma 5 of [10], we can obtain the proof of Theorem 2.3.  $\square$

*Remark 2.4.* The referee pointed out that the following geometric interpretation allows us to easily understand why  $E(i)$  approaches  $\xi_u$ . The curves for  $e^X$  and  $1 + uX$  intersect at 0 and at  $\xi_u$ , and these two curves make a closed area. As  $i$  increases,  $E(i) = \log(1 + uE(i-1))$  progressively gets closer from  $E(0) = \log u$  to  $\xi_u$  in this closed area. This illustrates that for sufficiently large  $i$ ,  $E(i)$  approximates  $\xi_u$ .

### 3. NUMERICAL RESULTS

In this section, we compare Algorithm Modified HT with Algorithm HS. We implemented both algorithms in C++ programs. To find all primes  $\leq y$  in step (1) of both algorithms, we used Atkin and Bernstein's sieve method [1], which uses  $O(y/\log \log y)$  operations and  $y^{1/2+o(1)}$  bits of memory. In step (2) of Algorithm HS, we used Newton's method for finding an estimate of  $\alpha_u$ . Instead of a value given by a preliminary search by bisection, we used  $\alpha_0 := \log(1 + y/(5 \log x))/\log y$  as a starting point. Although by the use of this starting point we could not prove quadratic convergence in theory, Newton's method converged very well after only a few iterations in practice.

TABLE 1. Estimates of  $\alpha_u$  and  $\Psi(x, y)$  by  $h$  function for  $x = 2^{320}$

$y$	Algorithm	$\alpha_u$	$\Psi(x, y)$	TIME(milliseconds)
$2^{20}$	HS with NM	0.694720	$2.87E75$	90
	Modified HT	0.694926	$2.87E75$	30
$2^{21}$	HS with NM	0.713551	$7.10E76$	170
	Modified HT	0.713768	$7.11E76$	40
$2^{22}$	HS with NM	0.730496	$1.25E78$	290
	Modified HT	0.730717	$1.26E78$	70
$2^{23}$	HS with NM	0.745816	$1.67E79$	590
	Modified HT	0.746038	$1.68E79$	140
$2^{24}$	HS with NM	0.759726	$1.74E80$	1101
	Modified HT	0.759944	$1.75E80$	270
$2^{25}$	HS with NM	0.772407	$1.46E81$	2133
	Modified HT	0.772618	$1.47E81$	520
$2^{26}$	HS with NM	0.784003	$1.02E82$	4075
	Modified HT	0.784209	$1.02E82$	991
$2^{27}$	HS with NM	0.794646	$6.03E82$	7801
	Modified HT	0.794846	$6.04E82$	1932
$2^{28}$	HS with NM	0.804443	$3.07E83$	15001
	Modified HT	0.804632	$3.08E83$	3725
$2^{29}$	HS with NM	0.813488	$1.37E84$	28901
	Modified HT	0.813671	$1.37E84$	7220
$2^{30}$	HS with NM	0.821861	$5.48E84$	62800
	Modified HT	0.822038	$5.49E84$	14070

Table 1 lists the estimates of  $\alpha_u$  and  $\Psi(x, y)$  with  $x = 2^{320}$  and  $y$  ranging from  $2^{20}$  up to  $2^{30}$  and the running times of both algorithms. This table shows that the estimate calculated by Algorithm Modified HT is accurate and this algorithm is faster than Algorithm HS with Newton’s method. In the table, “TIME” denotes the total amount of CPU time (milliseconds), and “HS with NM” denotes Algorithm HS with Newton’s method. All calculations were performed using a PC with Pentium IV 1.8GHz and 523Mbyte memory.

*Remark 3.1.* The reader may have the question of whether a larger value for  $m$  than in Theorem 2.3 can improve the accuracy of our estimate for  $\Psi(x, y)$ . The use of a larger value for  $m$  improves the accuracy slightly, not significantly. For the values of  $x$  and  $y$  used in calculating the data in Table 1,  $m = \lceil (\log u + \log \log y) / \log \log u \rceil + 1$  is 7 or 8. Our calculation with  $m = 1000$  showed that both the error of our estimate for  $\alpha_u$  and that for  $\Psi(x, y)$  decreased by only about 2%. This implies that if  $m$  is set as defined in Theorem 2.3,  $E(m)$  accurately approximates  $\xi_u$  and that the major contribution to the error of our estimate comes from the error term of (2.3).

*Remark 3.2.* Recently, J. P. Sorenson [14] proposed a new method to speed up Hildebrand and Tenenbaum’s algorithm. His algorithm is based on the validity of

the Riemann Hypothesis and has a running time of

$$O\left(\sqrt{y}\left(\frac{\log \log x}{\log y} + \frac{1}{\log \log y}\right)\right)$$

operations. Sorenson’s result is as follows: under the validity of the Riemann Hypothesis, for  $\delta > 0$  such that  $1 > s \geq 1/2 + \delta$ , and sufficiently large  $z$  such that  $2 \leq z \leq y$ ,

$$h_s(x, y, z, \alpha_{u,z}) = h(x, y, \alpha_u) \left(1 + O\left(\frac{1}{u} + \frac{1}{\log y}\right)\right),$$

where

$$\begin{aligned} h_s(x, y, z, s) &= \frac{x^s a(s, y, z)}{s \sqrt{2\pi c(s, y, z)}}; \\ a(s, y, z) &= \left(\prod_{p \leq z} (1 - p^{-s})^{-1}\right) \exp\left(\sum_{k=1}^{\lfloor \log y/s \rfloor} \frac{\text{li}(y^{1-ks}) - \text{li}(z^{1-ks})}{k}\right); \\ c(s, y, z) &= \sum_{p \leq z} \frac{p^s (\log p)^2}{(p^s - 1)^2} + \frac{z \log z}{s(z^s - 1)} - \frac{y \log y}{s(y^s - 1)} \\ &\quad + \sum_{k=1}^{\lfloor \log y/s \rfloor} \frac{1}{s(1 - ks)^2} \left(\frac{(1 - ks) \log y - ks}{y^{ks-1}} - \frac{(1 - ks) \log z - ks}{z^{ks-1}}\right), \end{aligned}$$

and  $\alpha_{u,z}$  is the solution to the equation

$$(3.1) \quad \sum_{p \leq z} \frac{\log p}{p^s - 1} + \sum_{k=1}^{\lfloor \log y/s \rfloor} \frac{y^{1-ks} - z^{1-ks}}{1 - ks} = \log x.$$

Substituting  $\hat{\alpha}_u$  for  $\alpha_{u,z}$ , we can speed up Sorenson’s algorithm. Let  $\epsilon$  be any fixed positive number. Using  $\hat{\alpha}_u$  of Theorem 2.3, we can obtain

$$\Psi(x, y) = h_s(x, y, z, \hat{\alpha}_u) \left(1 + O\left(\frac{1}{u} + \frac{1}{\log y}\right)\right)$$

in the range where both Sorenson’s assumption and the assumption  $(\log \log x)^{5/3+\epsilon} < \log y < (\log x)/e$  hold. Then, although the complexity of this new algorithm is  $O(\sqrt{y}/\log \log y)$ , in that range this complexity is the same as that of Sorenson’s algorithm.

We implemented both algorithms in C++ programs. To obtain the numerical solution to the equation (3.1), required in Sorenson’s original algorithm, we used Newton’s method and a starting point  $\alpha_0 := \log(1 + y/(5 \log x))/\log y$ . Following the setting of parameters in [14], we set  $z$  to  $5\sqrt{y}$ .

In Table 2, we give the estimates for  $\alpha_u$  and  $\Psi(x, y)$  with  $x = 2^{320}$  and  $y$  ranging from  $2^{45}$  up to  $2^{55}$  and the running times of both algorithms. In the table, “S with NW” denotes Sorenson’s original algorithm with Newton’s method, and “Modified S” denotes the modified Sorenson algorithm with  $\hat{\alpha}_u$ . This table shows that the estimates by our new algorithm agree well with those by Sorenson’s original algorithm, and our algorithm is faster than Sorenson’s.

TABLE 2. Estimates of  $\alpha_u$  and  $\Psi(x, y)$  by  $h_s$  function for  $x = 2^{320}$

y	Algorithm	$\alpha_u$	$\Psi(x, y)$	TIME(milliseconds)
$2^{45}$	S with NM	0.898767	$1.38E90$	2203
	Modified S	0.898886	$1.38E90$	741
$2^{46}$	S with NM	0.901916	$2.26E90$	2944
	Modified S	0.902032	$2.27E90$	971
$2^{47}$	S with NW	0.904913	$3.62E90$	4186
	Modified S	0.905027	$3.63E90$	1371
$2^{48}$	S with NW	0.907769	$5.67E90$	5788
	Modified S	0.907881	$5.68E90$	1862
$2^{49}$	S with NW	0.910492	$8.67E90$	8001
	Modified S	0.910602	$8.69E90$	2623
$2^{50}$	S with NW	0.913091	$1.30E91$	10865
	Modified S	0.913199	$1.30E91$	3585
$2^{51}$	S with NW	0.915575	$1.91E91$	15482
	Modified S	0.915568	$1.91E91$	4987
$2^{52}$	S with NW	0.917949	$2.76E91$	21190
	Modified S	0.918053	$2.76E91$	6960
$2^{53}$	S with NW	0.920222	$3.92E91$	29672
	Modified S	0.920323	$3.92E91$	9603
$2^{54}$	S with NW	0.922398	$5.48E91$	40658
	Modified S	0.922498	$5.49E91$	13559
$2^{55}$	S with NW	0.924484	$7.55E91$	57332
	Modified S	0.924583	$7.56E91$	18546

ACKNOWLEDGMENT

The author would like to thank the anonymous referee for his or her insightful remarks that improved this paper.

REFERENCES

- [1] A. Atkin, and D. Bernstein, *Prime sieves using binary quadratic forms*, to appear in Mathematics of Computation.
- [2] D. Bernstein, *Bounding smooth integers*, ANTS-III proceedings, Lecture Notes in Comput. Sci. 1423, Springer, New York, 128-130, 1998.
- [3] E. R. Canfield, P. Erdős, and C. Pomerance, *On a problem of Oppenheim concerning "factorisatio numerorum,"* J. Number Theory 17, 1-28, 1983. MR **85j**:11012
- [4] N. G. de Bruijn, *On the number of positive integers  $\leq x$  and free of prime factors  $> y$ ,* Nederl. Akad. Wetensch. Proc. Ser. A54, 50-60, 1951. MR **13**:724e
- [5] K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude,* Arkiv för Matematik, Astroni Fysik. Band 22 A, n:o 10, 1-14, 1930.
- [6] A. Hildebrand, *On the number of positive integers  $\leq x$  and free of prime factors  $> y$ ,* J. Number Theory 22, 289-307, 1986. MR **87f**:11066
- [7] A. Hildebrand, *On the local behavior of  $\Psi(x, y)$ ,* Trans. Amer. Math. Soc. 297, 729-751, 1986. MR **87k**:11099
- [8] A. Hildebrand and G. Tenenbaum, *On integers free of large prime factors,* Trans. Amer. Math. Soc. 296, 265-290, 1986. MR **87f**:11066

- [9] A. Hildebrand and G. Tenenbaum, *On integers without large prime factors*, Journal de Théorie des Nombres de Bordeaux 5, 411-484, 1993. MR **95d**:11116
- [10] S. Hunter and J. Sorenson, *Approximating the number of integers free of large prime factors*, Math. Comp. 66, 1729-1741, 1997. MR **98c**:11093
- [11] K. K. Norton, *Numbers with small prime factors, and the least  $k$ th power nonresidue*, Mem. Amer. Math. Soc. 106, 9-27, 1971. MR **44**:3948
- [12] P. Pritchard, *A sublinear additive sieve for finding prime numbers*, Communications of the ACM 24(1), 18-23, 772, 1981. MR **82c**:10011
- [13] R. A. Rankin, *The difference between consecutive prime numbers*, J. London Math. Soc. 13, 242-247, 1938. MR **28**:3978
- [14] J. P. Sorenson, *A fast algorithm for approximately counting smooth numbers*, ANTS-IV proceedings, Lecture Notes in Comput. Sci. 1838, Springer, New York, 539-549, 2000. MR **2002e**:11123

INFORMATION MEDIA LABORATORY, FUJI XEROX, 430, SAKAI, NAKAI-MACHI, ASHIGARAKAMI-GUN, KANAGAWA 259-0157, JAPAN

*E-mail address:* `kohji.suzuki@fujixerox.co.jp`