MATHEMATICS OF COMPUTATION
Volume 75, Number 254, Pages 1015–1024
S 0025-5718(05)01798-9
Article electronically published on December 2, 2005

APPROXIMATING 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 gave a smooth approximation formula for $\Psi(x, y)$ in the range $(\log x)^{1+\epsilon} < y \leq x$, where ϵ is a fixed positive number $\leq 1/2$. In this paper, by modifying their approximation formula, we provide a fast algorithm to approximate $\Psi(x, y)$. The computational complexity of this algorithm is $O(\sqrt{(\log x)(\log y)})$. We give numerical results which show that this algorithm provides accurate estimates for $\Psi(x, y)$ and is faster than conventional methods such as algorithms exploiting Dickman's function.

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 useful for many number-theoretic algorithms and modern cryptography. The behavior of $\Psi(x, y)$ has been investigated by many authors ([3],[4],[6],[8],[9],[10],[11],[13],[16]). We see good summaries for the investigations of $\Psi(x, y)$ in [12] and [15].

Dickman [8] 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 $x \longrightarrow \infty$, where $u = (\log x)/\log y$ and $\rho(u)$ is the unique solution to the following equations:

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

The estimate $\Psi(x, y) \approx x\rho(u)$ is, in practice, accurate only for small u. Hunter and Sorenson [13] gave some experimental data to show this fact. Dickman's ρ can be computed by using the following equation:

$$\rho(u) = \frac{1}{u}\int_{u-1}^{u} \rho(t)dt \quad \text{for } u \geq 1. \tag{1.1}$$

Several authors [2, 5, 14] proposed other methods to efficiently calculate Dickman's ρ.

Hildebrand and Tenenbaum [11] gave an estimate of $\Psi(x, y)$ which is accurate for large u. They showed that uniformly for $2 \leq y \leq x$,

$$\Psi(x, y) = h(x, y, \alpha_u)\left(1 + O\left(\frac{\log y}{\log x}\right) + O\left(\frac{\log y}{y}\right)\right), \tag{1.2}$$

Received by the editor September 30, 2004 and, in revised form, December 13, 2004.
2000 *Mathematics Subject Classification.* Primary 11N25; Secondary 11Y05.
Key words and phrases. Computational number theory, analytic number theory, asymptotic estimates, factoring problem.

where

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

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

and α_u is the unique solution to the equation

$$-\sum_{p\le y}\frac{\log p}{p^{\alpha_u}-1}+\log x = 0. \tag{1.3}$$

In [11], they also gave a smooth approximation for $h(x,y,s)$ in the range u is not too large. They showed that uniformly for $x\ge 2, (\log x)^{1+\epsilon} < y \le x$,

$$\begin{aligned} &h(x,y,\alpha_u) \\ &= x\left(\frac{\xi_u'}{2\pi}\right)\exp\left\{\gamma - u\xi_u + \int_0^{\xi_u}\frac{e^t-1}{t}dt \right. \\ &\qquad \left. +O_\epsilon\left(\frac{\log(1+u)}{\log y}\right) + u\exp(-(\log y)^{3/5-\epsilon})\right\}, \end{aligned} \tag{1.4}$$

where ϵ is a fixed positive number $\le 1/2$ and ξ_u is the unique solution to the equation

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

The use of the above equation for approximating $\Psi(x,y)$ requires good estimates for ξ_u, ξ_u', and $\int_0^{\xi_u}(e^t-1)/tdt$.

Hunter and Sorenson [13] provided an algorithm to evaluate Hildebrand and Tenenbaum's approximation (1.2). Their algorithm uses a bisection method for obtaining approximations of α_u, the unique solution to (1.3). The complexity of their algorithm is given by

$$O\left(y\left(\frac{\log\log x}{\log y}+\frac{1}{\log\log y}\right)\right). \tag{1.5}$$

Hunter and Sorenson also showed that Newton's method can improve the complexity of the algorithm. 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 $\log x\le y$, the total running time of this algorithm is dropped to $O(y/\log\log y)$, and if $\log x > y$, it corresponds to (1.5). Although Newton's method can reduce the running time for finding the unique solution of (1.3), one can only prove quadratic convergence. Furthermore, assuming the validity of the Riemann Hypothesis (RH), Sorenson [17] proposed a modification of Hunter and Sorenson's algorithm. The running time of this modified algorithm is roughly proportional to $\sqrt{y}$.

The author [18] gave another algorithm to evaluate (1.2). Let

$$m = \left\lceil\frac{\log u+\log\log y}{\log\log u}\right\rceil + 1$$

and

$$\hat{\alpha}_u(l) = 1-\frac{E(l)}{\log y},$$

where l is a positive integer and

$$E(l) = \begin{cases} \log u & \text{for } l = 0, \\ \log u + \log\left(E(l-1) + 1/u\right) & \text{for } l > 0. \end{cases} \tag{1.6}$$

Then, we have $\alpha_u = \hat{\alpha}_u(m) + O(1/(u(\log y)^2))$ for $x \to \infty$, in the range

$$(\log\log x)^{5/3+\epsilon'} < \log y < (\log x)/e, \tag{1.7}$$

where ϵ' is any fixed positive number. $E(m)$ is an approximation of ξ_u. Using the above $\hat{\alpha}_u(m)$, one can obtain

$$\Psi(x,y) = h(x,y,\hat{\alpha}_u(m))\left(1 + O\left(\frac{1}{u} + \frac{1}{\log y}\right)\right) \text{ for } x \to \infty, \tag{1.8}$$

in the above range. The complexity of the algorithm to approximate $\Psi(x,y)$ with (1.8) is the same as (1.5). By assuming the validity of the RH and applying the same method as in [17] to this algorithm, one can obtain an algorithm with the running time roughly proportional to $\sqrt{y}$ [18].

In this paper, by modifying (1.4), we provide a fast algorithm to approximate $\Psi(x,y)$. To approximate $\int_0^{\xi_u}(e^t-1)/tdt$ in (1.4), we use the midpoint method and (1.6). Let $t_k(l) = (k-1/2)E(l)/l$ for positive integers k and l. Define

$$g(l) = \frac{x^{\hat{\alpha}_u(l)}e^{\gamma+f(l)}}{\hat{\alpha}_u(l)\sqrt{2\pi u(1+(\log x)/y)}},$$

where γ is Euler's constant and

$$f(l) = \frac{E(l)}{l}\sum_{k=1}^{l}\frac{e^{t_k(l)}-1}{t_k(l)}.$$

Let

$$m' = \lceil\sqrt{u}\log y\rceil + 1.$$

The following theorem shows $g(m')$ can approximate $\Psi(x,y)$ in almost the same range as (1.7).

Theorem 1.1. *Let ϵ be any fixed positive number $\leq 1/2$. In the range $(\log\log x)^{5/3+\epsilon}$ $< \log y \leq e^{-1}(1-\epsilon)\log x$, we have*

$$\Psi(x,y) = g(m')\exp\left\{O\left(\frac{\log(1+u)}{\log y} + u\exp(-(\log y)^{3/5-\epsilon}) + \frac{1}{\log(1+u)}\right)\right\},$$

for $x \to \infty$.

Using the above theorem, we can obtain the following algorithm to approximate $\Psi(x,y)$.

Algorithm A. (1) *Set $E(0) = \log u$ and $m' = \lceil\sqrt{u}(\log y)\rceil + 1$.*
(2) *Compute $E(m')$ using (1.6).*
(3) *Compute and output $g(m')$.*

Since the conventional methods based on (1.2) need to find all primes $\leq y$, these methods require at least $O(y/\log y)$ operations. However, our new algorithm does not require these primes and therefore it is very fast. The complexity of Algorithm A is given by

$$O(\sqrt{u}(\log y)) = O(\sqrt{(\log x)(\log y)}).$$

Also, the algorithm to approximate $\Psi(x,y)$ using (1.1) requires $O(u)$ operations. Therefore, our algorithm is faster than the methods with (1.1) in the range $\log x > (\log y)^3$.

The structure of this paper is as follows. In Section 2 we give the proof of Theorem 1.1. In Section 3 we give numerical results for showing that Algorithm A provides an accurate approximation to $\Psi(x,y)$ and is much faster than conventional algorithms.

2. Proof of Theorem 1.1

In this section, we provide the proof of Theorem 1.1. The author [18] showed that for $u > e,\ n \geq 1$,

$$0 < \xi_u - E(n) < \frac{1}{(\log u)^{n-1}}. \tag{2.1}$$

Using the above equation iteratively, we can have good approximations for ξ_u. By this iterative method and the midpoint one, we can obtain an approximation for $\int_0^{\xi_u}(e^t-1)/t dt$.

Lemma 2.1. *Let ϵ be any fixed positive number. For $u \geq e^{1+\epsilon}$ and sufficiently large y, we have*

$$f(m') = \int_0^{\xi_u} \frac{e^t-1}{t} dt + O\left(\frac{\log u}{\log y}\right). \tag{2.2}$$

Proof. Let $s(t) = (e^t-1)/t$ and $h = E(m')/m'$. In the following, we denote $t_k(m')$ by t_k. Since $m' \geq m = \lceil (\log u + \log\log y)/\log\log u \rceil + 1$ for $u \geq e^{1+\epsilon}$ and sufficiently large y, we have

$$0 < \xi_u - E(m') \leq \xi_u - E(m) = O\left(\frac{1}{u \log y}\right). \tag{2.3}$$

From (2.1), we obtain

$$\log u < E(1) < \xi_u < E(1) + 1 < \log u + \log\log u + 2.$$

Hence, since $s(t)$ is increasing for $t \geq 0$, we get

$$\begin{aligned} 0 < \int_0^{\xi_u} s(t)dt - \int_0^{E(m')} s(t)dt &= \int_{E(m')}^{\xi_u} s(t)dt \\ &< (\xi_u - E(m'))\frac{e^{\xi_u}-1}{\xi_u} \\ &= O\left(\frac{1}{\log y}\right). \end{aligned} \tag{2.4}$$

Since $s''(t)$ is increasing for $t \geq 0$, we have for an integer $k \geq 1$,

$$\begin{aligned} \int_{t_k-h/2}^{t_k+h/2} \{s(t) - s(t_k)\}\, dt &\leq \int_{t_k-h/2}^{t_k+h/2} \left\{(t-t_k)s'(t_k) + \frac{(t-t_k)^2}{2} s''(t_k+h/2)\right\} dt \\ &= \frac{h^3}{24} s''(t_k+h/2). \end{aligned}$$

Hence,

$$\begin{aligned}\int_0^{E(m')} s(t)dt - f(m') &< \frac{h^2}{24}\sum_{k=1}^{m'} s''(t_k + h/2)h \\ &< \frac{h^2}{24}\int_h^{E(m')+h} s''(t)dt \\ &< \frac{h^2}{24}(s'(E(m')+h) - s'(h)) \\ &= O\left(\frac{(\log u)^2}{(\log y)^2}\right) = O\left(\frac{\log u}{\log y}\right).\end{aligned}$$

Similarly, we have

$$\int_0^{E(m')} s(t)dt - f(m') > \frac{h^2}{24}(s'(E(m')-h) - s'(0)) = O\left(\frac{\log u}{\log y}\right).$$

Hence,

$$\int_0^{E(m')} s(t)dt - f(m') = O\left(\frac{\log u}{\log y}\right).$$

From the above equation and (2.4), we obtain the proof of this lemma. □

Using the above lemma, we obtain the proof of Theorem 1.1.

Proof of Theorem 1.1. By Theorem 2 of [11], we have

$$\phi_2(\alpha_u, y) = \left(1 + \frac{\log x}{y}\right)(\log x)(\log y)\left(1 + O\left(\frac{1}{\log(1+u)} + \frac{1}{\log y}\right)\right),$$

for $x \geq y \geq 2$. Also, from the proof of Theorem 2 of [11, p. 289], we obtain

$$\begin{aligned}&\prod_{p\leq y}(1-p^{-\alpha_u})^{-1} \\ &= e^{\gamma}(\log y)\exp\left\{\int_0^{\xi_u}\frac{e^t-1}{t}dt + O_{\epsilon}\left(\frac{\log(1+u)}{\log y} + u\exp(-(\log y)^{3/5-\epsilon})\right)\right\}.\end{aligned}$$

By Lemma 2.1 of [18], we have

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

in the range $(\log\log x)^{5/3+\epsilon} < \log y$. From the above equation and (2.3), we obtain $\hat{\alpha}_u(m') = \alpha_u + O(1/(u(\log y)^2))$ in the range of this theorem. Hence, we have

$$\frac{x^{\hat{\alpha}_u(m')}}{\hat{\alpha}_u(m')} = \frac{x^{\alpha_u}}{\alpha_u}\exp\left\{O\left(\frac{1}{\log y}\right)\right\}.$$

In the range of this theorem, $y \to \infty$ for $x \to \infty$. Hence, by the above equations, Lemma 2.1, and (1.2), we can obtain the proof of Theorem 1.1. □

3. Numerical results

In this section, we compare Algorithm A with conventional algorithms to approximate $\Psi(x, y)$. We implemented Algorithm A, Hunter and Sorenson's (HS), Sorenson's based on the RH (SO on RH), Suzuki's (SU), Suzuki's based on the RH (SU on RH), de Bruijn's method (DB), and Patterson and Rumsey's method (PR) to compute the dickman's ρ in C++ programs.

To find primes required for the algorithms HS, SO on RH, SU, and SU on RH, 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 for finding all primes $\le y$. For Algorithm HS and SO on RH, we used Newton's method for finding an estimate of α_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.

Using Patterson and Rumsey's method, one can efficiently compute Dickman's ρ [2]. Let r denote a positive integer, and let $\rho_r(x) = \rho(x)$ for $r-1 \le x \le r$. Let $0 \le \nu \le 1$. Then, $\rho_r(x)$ can be computed by

$$\rho_r(r-\nu) = \sum_{i=0}^{\infty} \phi(i,r)\nu^r,$$

where $\phi(i,j)$'s are given by the following relations:

$$\begin{aligned} \phi(0,1) &= 1, & \phi(i,0) &= 0 \quad \text{for } i \ge 1, \\ \phi(0,2) &= 1-\log 2, & \phi(i,2) &= 1/(i2^i) \quad \text{for } i \ge 1, \\ \phi(0,r) &= \frac{1}{r-1}\sum_{j=1}^{\infty}\frac{\phi(j,r)}{j+1}, & \phi(i,r) &= \sum_{j=0}^{i-1}\frac{\phi(j,r-1)}{ir^{i-j}} \quad \text{for } r \ge 3. \end{aligned}$$

In the calculation of $\rho_r(x)$ with the above equation, we truncated the infinite sums in the above equations at $i = j = 100$. This algorithm requires $O(u)$ operations. We can obtain another method to calculate Dickman's ρ by using de Bruijn's approximation formula. De Bruijn [7] showed

$$\rho(u) = \exp\left\{-u\left(\log u + \log\log u - 1 + \frac{\log\log u - 1}{\log(u+1)} + O\left(u\left(\frac{\log\log u}{\log u}\right)^2\right)\right)\right\},$$

for $u \ge 3$. The above equation can be calculated in $O(1)$ operations.

Tables 1, 2, and 3 provide the estimates of $\Psi(x, y)$ and the running times of the algorithms. Tables 1 and 2 list the estimates of $\Psi(x, y)$ for $x = 2^{350}$ and 2^{1000}, respectively, with y ranging from 2^{20} up to 2^{25}. Table 3 lists the estimates of $\Psi(x, y)$ for $y = 2^{15}$, with x ranging from 2^{50} up to 2^{300}. In the tables, "TIME" denotes the total amount of CPU time (milliseconds), "RATIO" denotes the ratio of the estimates by SO on RH, SU, SU on RH, PR, DB, and A to that by HS. In Tables 1 and 3, "R-L" and "R-U", respectively, denote the ratio of those estimates to the lower bound and the upper bound of $\Psi(x, y)$ given by Bernstein's method [3]. In our calculations for Bernstein's method, we used the scaling factor = 500 and the precision parameter $=2^{18}$. All calculations were performed using a PC with Pentium III 1.066GHz and 130Mbyte memory.

The tables show that Algorithm A provides accurate estimates and is much faster than the conventional methods, except for Algorithm DB. Although Algorithm DB is as fast as Algorithm A, the estimates by Algorithm DB are very crude.

TABLE 1. Estimates of $\Psi(x, y)$ function for $x = 2^{350}$ and y ranging from 2^{20} up to 2^{25}

y	Algorithm	$\Psi(x, y)$	RATIO	R-L	R-U	TIME(millisec.)
2^{20}	Algorithm HS	$4.77E81$	–	1.01	0.988	180
	Alg. SO on RH	$4.99E81$	1.04	1.05	1.03	50
	Algorithm SU	$4.78E81$	1.00	1.01	0.989	60
	Alg. SU on RH	$4.99E81$	1.04	1.05	1.03	40
	Algorithm PR	$3.91E81$	0.820	0.830	0.810	20
	Algorithm DB	$1.20E83$	25.2	25.5	24.9	0
	Algorithm A	$5.05E81$	1.05	1.07	1.04	0
2^{21}	Algorithm HS	$1.75E83$	–	1.01	0.988	311
	Alg. SO on RH	$1.80E83$	1.02	1.04	1.01	50
	Algorithm SU	$1.75E83$	1.00	1.01	0.989	90
	Alg. SU on RH	$1.80E83$	1.02	1.04	1.01	40
	Algorithm PR	$1.46E83$	0.835	0.846	0.826	20
	Algorithm DB	$4.57E84$	26.1	26.4	25.8	0
	Algorithm A	$1.83E83$	1.04	1.06	1.03	0
2^{22}	Algorithm HS	$4.42E84$	–	1.01	0.989	541
	Alg. SO on RH	$4.52E84$	1.02	1.03	1.01	40
	Algorithm SU	$4.43E84$	1.00	1.01	0.990	140
	Alg. SU on RH	$4.53E84$	1.02	1.03	1.01	40
	Algorithm PR	$3.75E84$	0.848	0.859	0.839	10
	Algorithm DB	$1.19E86$	26.9	27.3	26.7	0
	Algorithm A	$4.59E84$	1.03	1.05	1.02	0
2^{23}	Algorithm HS	$8.13E85$	–	1.01	0.989	932
	Alg. SO on RH	$8.24E85$	1.01	1.02	1.00	50
	Algorithm SU	$8.14E85$	1.00	1.01	0.990	281
	Alg. SU on RH	$8.24E85$	1.01	1.02	1.00	40
	Algorithm PR	$6.99E85$	0.860	0.871	0.851	20
	Algorithm DB	$2.26E87$	27.8	28.1	27.5	0
	Algorithm A	$8.36E85$	1.02	1.04	1.01	0
2^{24}	Algorithm HS	$1.13E87$	–	1.01	0.989	1622
	Alg. SO on RH	$1.14E87$	1.01	1.02	1.00	50
	Algorithm SU	$1.13E87$	1.00	1.01	0.991	471
	Alg. SU on RH	$1.14E87$	1.01	1.02	1.00	50
	Algorithm PR	$9.86E86$	0.869	0.880	0.860	20
	Algorithm DB	$3.23E88$	28.5	28.9	28.2	0
	Algorithm A	$1.15E87$	1.01	1.03	1.00	0
2^{25}	Algorithm HS	$1.24E88$	–	1.01	0.990	2904
	Alg. SO on RH	$1.25E88$	1.01	1.02	1.00	50
	Algorithm SU	$1.24E88$	1.00	1.01	0.991	782
	Alg. SU on RH	$1.25E88$	1.01	1.02	1.00	50
	Algorithm PR	$1.09E88$	0.877	0.888	0.868	10
	Algorithm DB	$3.63E89$	29.2	29.6	28.9	0
	Algorithm A	$1.25E88$	1.00	1.02	0.999	0

TABLE 2. Estimates of $\Psi(x, y)$ function for $x = 2^{1000}$ and y ranging from 2^{20} up to 2^{25}

y	Algorithm	$\Psi(x,y)$	RATIO	TIME(millisec.)
2^{20}	Algorithm HS	$9.99E204$	–	170
	Alg. SO on RH	$1.12E205$	1.12	50
	Algorithm SU	$9.99E204$	1.00	50
	Alg. SU on RH	$1.12E205$	1.12	40
	Algorithm PR	$7.19E204$	0.71	51
	Algorithm DB	$1.46E206$	14.6	0
	Algorithm A	$1.09E205$	1.10	0
2^{21}	Algorithm HS	$6.30E210$	–	310
	Alg. SO on RH	$6.81E210$	1.08	50
	Algorithm SU	$6.30E210$	1.00	80
	Alg. SU on RH	$6.81E210$	1.07	50
	Algorithm PR	$4.74E210$	0.75	50
	Algorithm DB	$9.87E211$	15.6	0
	Algorithm A	$6.97E210$	1.10	0
2^{22}	Algorithm HS	$1.05E216$	–	501
	Alg. SO on RH	$1.11E216$	1.05	50
	Algorithm SU	$1.05E216$	1.00	170
	Alg. SU on RH	$1.11E216$	1.05	50
	Algorithm PR	$8.18E215$	0.77	50
	Algorithm DB	$1.73E217$	16.4	0
	Algorithm A	$1.16E216$	1.10	0
2^{23}	Algorithm HS	$5.55E220$	–	851
	Alg. SO on RH	$5.74E220$	1.03	50
	Algorithm SU	$5.55E220$	1.00	270
	Alg. SU on RH	$5.74E220$	1.03	50
	Algorithm PR	$4.40E220$	0.79	50
	Algorithm DB	$9.61E221$	17.3	0
	Algorithm A	$6.11E220$	1.10	0
2^{24}	Algorithm HS	$1.08E225$	–	1522
	Alg. SO on RH	$1.11E225$	1.03	60
	Algorithm SU	$1.08E225$	1.00	471
	Alg. SU on RH	$1.11E225$	1.03	50
	Algorithm PR	$8.82E224$	0.81	40
	Algorithm DB	$1.94E226$	17.9	0
	Algorithm A	$1.18E225$	1.09	0
2^{25}	Algorithm HS	$8.83E228$	–	2855
	Alg. SO on RH	$9.06E228$	1.02	61
	Algorithm SU	$8.84E228$	1.00	821
	Alg. SU on RH	$9.06E225$	1.02	50
	Algorithm PR	$7.31E228$	0.82	40
	Algorithm DB	$1.61E230$	18.2	0
	Algorithm A	$9.57E228$	1.08	0

TABLE 3. Estimates of $\Psi(x,y)$ function for $y = 2^{15}$ and x ranging from 2^{50} up to 2^{300}

x	u	Algorithm	$\Psi(x,y)$	RATIO	R-L	R-U	TIME
2^{50}	3.33	Algorithm HS	$3.00E13$	–	1.00	1.00	30
		Alg. SO on RH	$3.05E13$	1.01	1.02	1.01	30
		Algorithm SU	$3.02E13$	1.00	1.01	1.00	20
		Alg. SU on RH	$3.06E13$	1.02	1.03	1.02	30
		Algorithm PR	$2.66E13$	0.887	0.894	0.888	0
		Algorithm DB	$1.95E15$	65.1	65.7	65.3	0
		Algorithm A	$2.75E13$	0.917	0.925	0.919	0
2^{100}	6.66	Algorithm HS	$3.88E24$	–	1.00	0.996	30
		Alg. SO on RH	$4.04E24$	1.04	1.04	1.03	30
		Algorithm SU	$3.88E24$	1.00	1.00	0.998	20
		Alg. SU on RH	$4.04E24$	1.04	1.04	1.03	30
		Algorithm PR	$3.19E24$	0.824	0.830	0.821	10
		Algorithm DB	$1.45E26$	37.4	37.7	37.3	0
		Algorithm A	$3.88E24$	1.00	1.00	0.996	0
2^{150}	10.0	Algorithm HS	$5.09E34$	–	1.00	0.993	30
		Alg. SO on RH	$4.04E34$	1.06	1.07	1.05	30
		Algorithm SU	$3.88E34$	1.00	1.00	0.994	20
		Alg. SU on RH	$5.42E34$	1.06	1.07	1.05	30
		Algorithm PR	$3.95E34$	0.775	0.781	0.770	10
		Algorithm DB	$1.49E36$	29.4	29.6	29.2	0
		Algorithm A	$5.26E34$	1.03	1.04	1.02	0
2^{200}	13.3	Algorithm HS	$1.56E44$	–	1.00	0.991	30
		Alg. SO on RH	$1.71E44$	1.09	1.10	1.08	40
		Algorithm SU	$1.56E44$	0.999	1.00	0.991	20
		Alg. SU on RH	$1.70E44$	1.08	1.09	1.08	40
		Algorithm PR	$1.14E44$	0.732	0.739	0.726	10
		Algorithm DB	$3.89E45$	24.8	25.0	24.6	0
		Algorithm A	$1.63E44$	1.04	1.05	1.03	0
2^{250}	16.6	Algorithm HS	$1.66E53$	–	1.01	0.990	30
		Alg. SO on RH	$1.86E53$	1.11	1.12	1.10	40
		Algorithm SU	$1.66E53$	0.998	1.00	0.989	20
		Alg. SU on RH	$1.86E53$	1.11	1.12	1.10	40
		Algorithm PR	$1.15E53$	0.691	0.698	0.685	20
		Algorithm DB	$3.61E54$	21.6	21.8	21.4	0
		Algorithm A	$1.72E53$	1.03	1.04	1.02	0
2^{300}	20.0	Algorithm HS	$7.68E61$	–	1.01	0.988	30
		Alg. SO on RH	$8.80E61$	1.14	1.15	1.13	40
		Algorithm SU	$7.67E61$	0.998	1.00	0.987	20
		Alg. SU on RH	$8.78E61$	1.14	1.15	1.13	40
		Algorithm PR	$5.01E61$	0.652	0.659	0.645	20
		Algorithm DB	$1.46E63$	19.1	19.3	18.8	0
		Algorithm A	$7.83E61$	1.01	1.03	1.00	0

Acknowledgments

The author would like to thank the anonymous referees for carefully reading the paper and for helpful suggestions.

References

[1] A. Atkin and D. Bernstein, *Prime sieves using binary quadratic forms*, Math. Comp. 73, 1023-1030, 2004. MR2031423 (2004i:11147)

[2] E. Bach and R. Peralta, *Asymptotic semismoothness probabilities*, Math. Comp. 65, 1701-1715, 1996. MR1370848 (98a:11123)

[3] D. Bernstein, *Bounding smooth integers*, ANTS-III proceedings, Lecture Notes in Compt. Sci. 1423, Springer, New York, 128-130, 1998. MR1726065

[4] E. R. Canfield, P. Erdös, and C. Pomerance, *On a problem of Oppenheim concerning "factorisatio numerorum,"* J. Number Theory 17, 1-28, 1983. MR0712964 (85j:11012)

[5] A. Y. Cheer and D. A. Goldston, *A differential delay equation arising from the sieve of Eratosthenes*, Math. Comp. 55, 129-141, 1990. MR1023043 (90j:11091)

[6] 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. MR0046375 (13:724e)

[7] N. G. de Bruijn, *The asymptotic behavior of a function occurring in the theory of primes*, Journal of the Indian Mathematical Society (New Series) 15 (1951), 25-32. 50-60, 1951. MR0043838 (13:326f)

[8] K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude*, Arkiv För Matematik, Astromi Fysik. Band 22 A, no. 10, 1-14, 1930.

[9] A. Hildebrand, *On the number of positive integers $\leq x$ and free of prime factors $> y$*, J. Number Theory 22, 289-307, 1986. MR0831874 (87d:11066)

[10] A. Hildebrand, *On the local behavior of $\Psi(x, y)$* Trans. Amer. Math. Soc. 297, 729-751, 1986. MR0854096 (87k:11099)

[11] A. Hildebrand and G. Tenenbaum, *On integers free of large prime factors*, Trans. Amer. Math. Soc. 296, 265-290, 1986. MR0837811 (87f:11066)

[12] A. Hildebrand and G. Tenenbaum, *On integers without large prime factors*, Journal de Théorie des Nombres de Bordeaux 5, 411-484, 1993. MR1265913 (95d:11116)

[13] S. Hunter and J. Sorenson, *Approximating the number of integers free of large prime factors*, Math. Comp. 66, 1729-1741, 1997. MR1423076 (98c:11093)

[14] G. Marsaglia, A. Zaman, and J. C. W. Marsaglia, *Numerical solution of some classical differential-difference equations*, Math. Comp. 53, 191-201, 1989. MR0969490 (90h:65124)

[15] K. K. Norton, *Numbers with small prime factors, and the least kth power nonresidue*, Mem. Amer. Math. Soc. 106, 9-27, 1971. MR0286739 (44:3948)

[16] R. A. Rankin, *The difference between consecutive prime numbers*, J. London Math. Soc. 13, 242-247, 1938.

[17] J. P. Sorenson, *A fast algorithm for approximately counting smooth numbers*, ANTS-IV proceedings, Lecture Notes in Compt. Sci. 1838, Springer, New York, 539-549, 2000. MR1850632 (2002e:11123)

[18] K. Suzuki, *An estimate for the number of integers without large prime factors*, Math. Comp., 73, 1013-1022, 2004. MR2031422 (2005a:11142)

Corporate Research Group, Fuji Xerox, 430, Sakai, Nakai-machi, Ashigarakami-gun, Kanagawa 259-0157, Japan

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