MATHEMATICS OF COMPUTATION
Volume 66, Number 220, October 1997, Pages 1729–1741
S 0025-5718(97)00874-0

APPROXIMATING THE NUMBER OF INTEGERS FREE OF LARGE PRIME FACTORS

SIMON HUNTER AND JONATHAN SORENSON

ABSTRACT. Define $\Psi(x,y)$ to be the number of positive integers $n \leq x$ such that n has no prime divisor larger than y. We present a simple algorithm that approximates $\Psi(x,y)$ in $O(y\{\frac{\log\log x}{\log y} + \frac{1}{\log\log y}\})$ floating point operations. This algorithm is based directly on a theorem of Hildebrand and Tenenbaum. We also present data which indicate that this algorithm is more accurate in practice than other known approximations, including the well-known approximation $\Psi(x,y) \approx x\rho(\log x/\log y)$, where $\rho(u)$ is Dickman's function.

1. INTRODUCTION

Let $\Psi(x,y)$ denote the number of positive integers $n \leq x$ such that n has no prime divisors larger than y. In order to optimize the running times of many integer factoring and discrete logarithm algorithms, including the number field sieve methods (see [9, 13]), good estimates for $\Psi(x,y)$ are required.

Unfortunately, computing the exact value of $\Psi(x,y)$ is quite prohibitive in practice, as x is typically 20–200 decimal digits long. For example, the obvious way to compute the exact value of $\Psi(x,y)$ is to factor all the integers $\leq x$ over the primes up to y. If sieving is used, this method requires time proportional to $x \log\log y$. There is a simple recursive algorithm to compute $\Psi(x,y)$ using the Buchstab identity $\Psi(x,y) = \Psi(x,2) + \sum_{2<p\leq y}\Psi(x/p,p)$, which holds for $x > 1$, $y \geq 2$. This algorithm requires $O(\Psi(x,y))$ operations, but is easy to implement. Bernstein [1] gives several algorithms for computing the exact value of $\Psi(x,y)$, one of which requires only $O(\Psi(x,y)^{\beta(x,y)})$ operations, where $1/2 < \beta(x,y) \leq 1$. See also §5.1 for an $O(x/\log y)$ method. None of these methods appear to be feasible for applications in factoring.

The natural alternative to computing the exact value of $\Psi(x,y)$ is to use an approximation, and in factoring, a good approximation is normally sufficient. The estimate $\Psi(x,y) \approx x\rho(\log x/\log y)$, where ρ is Dickman's function, is very popular. Currently, the best rigorous version of this estimate is the following theorem. Let $u := u(x,y) = \log x/\log y$.

Received by the editor December 1, 1994 and, in revised form, October 16, 1995 and August 21, 1996.

1991 *Mathematics Subject Classification.* Primary 11N25, 11Y05; Secondary 11Y16.

Key words and phrases. Psixyology, integer factoring, analytic number theory, algorithmic number theory, computational number theory.

The first author was supported in part by a Butler Scholarship.

The second author was supported in part by NSF grant CCR-9204414.

Computing equipment was provided through a grant from the Holcomb Research Institute.

Theorem 1 (Hildebrand [4]). *Let $\epsilon > 0$. Then*

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

for $y \geq 2$ and $1 \leq u \leq \exp((\log y)^{3/5-\epsilon})$.

Here $\rho(u)$ is defined as the (unique) continuous solution to

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

The error term $1+O(\log(u+1)/\log y)$ here is the best possible, although the range for y can be extended when assuming the Riemann Hypothesis.

See also §5.2 for a simple probabilistic approximation algorithm.

In this paper, we present Algorithm HT, an alternative method for approximating $\Psi(x,y)$ based on a theorem of Hildebrand and Tenenbaum. After introducing the necessary notation, we will state this theorem.

Let $\bar{u} := \bar{u}(x,y) = \min\{\log x, y\}/\log y = \min\{u, y/\log y\}$. Define

$$\begin{aligned} \zeta(s,y) &:= \prod_{p \leq y}(1-p^{-s})^{-1}; \\ \phi(s,y) &:= \log \zeta(s,y); \\ \phi_k(s,y) &:= \frac{d^k}{ds^k}\phi(s,y) \quad (k \geq 1); \\ HT(x,y,s) &:= \frac{x^s\zeta(s,y)}{s\sqrt{2\pi\phi_2(s,y)}}. \end{aligned}$$

Let $\alpha = \alpha(x,y)$ be the unique solution to the equation

$$\phi_1(\alpha,y) + \log x = 0.$$

Theorem 2 (Hildebrand and Tenenbaum [5]).

$$\Psi(x,y) = HT(x,y,\alpha(x,y))(1 + O(1/\bar{u}))$$

uniformly for $2 \leq y \leq x$.

Observe that the error terms in Theorems 1 and 2 are roughly equivalent when $\log y$ is proportional to $\sqrt{\log x \log\log x}$; this is precisely the situation in current factoring algorithms. The error term in Theorem 2 is better than that of Theorem 1 for smaller y (larger u), but worse for larger y (smaller u).

Algorithm HT approximates $\Psi(x,y)$ by computing $HT(x,y,\alpha')$, where α' is an approximation of α. It has a running time of

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

floating point operations, which is roughly proportional to y and essentially independent of x. Note that y is normally at most 10^8 in practice, so this algorithm is quite practical. We present and analyze this algorithm in §2.

We also implemented four algorithms based on Theorem 1. The most accurate of these has a running time proportional to u. The other three have constant running times. We discuss these algorithms briefly in §3.

In §4, we compare the accuracy of the five algorithms mentioned above. We conclude in §5 with a brief discussion of two additional algorithms for estimating $\Psi(x, y)$ and suggestions for future work.

For a more thorough introduction to the literature on $\Psi(x, y)$, see Canfield, Erdős, and Pomerance [2], Hildebrand and Tenenbaum [6], Moree [10], and Norton [11]. We now briefly discuss our model of computation and notation.

1.1. **Computation model.** We measure the complexity of our algorithms by counting the number of *floating point* operations. Such operations include $\pm$, $\times$, $\div$, comparisons, exponentiation, and taking logarithms of real numbers. We also include array indexing and branching as basic operations. In practice, we used 80-bit floating point numbers.

1.2. **Notation.** p always denotes a prime number, and sums over p are always sums over primes. For positive functions f and g, we write $f(n) = O(g(n))$ if there exists an absolute constant $c > 0$ such that $f(n) < c \cdot g(n)$ for all n sufficiently large. If the O is subscripted, then c may depend on the subscript. $f(n) \ll g(n)$ means $f(n) = O(g(n))$. When we write $f(n) \approx g(n)$, we mean f is approximately g (this approximation may be very crude).

2. An algorithm based on Theorem 2

The main steps of the algorithm are as follows.

Algorithm HT.

1. Find all primes $p \leq y$.
2. Compute an approximation α' to the solution $s = \alpha$ of $f(s) = 0$, where
$$f(s) := \phi_1(s, y) + \log x.$$
We require that $|\alpha' - \alpha| < \min\{0.0001, 1/(\bar{u} \log x)\}$.
3. Output $HT(x, y, \alpha')$.

In this section, we first discuss the complexity of this algorithm. We then prove its correctness by showing that $\Psi(x, y) = HT(x, y, \alpha')(1 + O(1/\bar{u}))$. Finally, we show how the algorithm might be improved by using Newton's method in Step (2).

2.1. **Complexity.** We now derive the running time for Algorithm HT as claimed in the Introduction.

Step (1) can be performed using a sublinear sieve such as Pritchard's dynamic wheel sieve (see [14, 15]), which uses $O(y/\log\log y)$ operations.

Because the number of primes $\leq y$ is $\pi(y) = O(y/\log y)$, for a given value of s it is possible to compute each of $\zeta(s, y)$, $\phi_1(s, y)$, and $\phi_2(s, y)$ using only $O(y/\log y)$ operations. Thus Step (3) requires $O(y/\log y)$ operations.

For Step (2), observe that when y is fixed, the function
$$\phi_1(s, y) = -\sum_{p \leq y} \frac{\log p}{p^s - 1}$$
is increasing in s when $s > 0$ ($\phi_2(s, y)$ is positive on this interval). Thus α may be approximated using bisection. We know that
$$\frac{\log(1 + y/(5 \log x))}{\log y} \leq \alpha \ll 1$$

(see [5, Lemma 2]) which provides the interval to search. As mentioned above, we must perform enough iterations to reduce the size of our interval to no more than $(\bar{u}\log x)^{-1}$. At $O(y/\log y)$ operations per iteration, this gives a total of $O(y\log(\bar{u}\log x)/\log y) = O(y(\log\log x)/\log y)$ operations.

2.2. Correctness.

We now prove that if we use the approximation α' in place of α, then Theorem 2 still holds. In other words, we will prove that

$$HT(x,y,\alpha') = HT(x,y,\alpha)(1+O(1/\bar{u})).$$

We begin with some results on the ϕ_k functions.

Lemma 3. *Let $\delta = \delta(x,y) > 0$, with $\delta \ll 1$. If $|s-\alpha| < \delta/\log x$, then $\phi_2(s,y) = \phi_2(\alpha,y)(1+O(\delta/\bar{u}))$.*

Proof. Taking a Taylor Series expansion of ϕ_2 about α, we have

$$\phi_2(s,y) = \phi_2(\alpha,y) + \sum_{i=1}^{\infty} \frac{(s-\alpha)^i}{i!}\phi_{i+2}(\alpha,y).$$

From Lemma 4 of [5] we easily obtain

$$\phi_k(\alpha,y) \ll k!(\log x)^k \bar{u}^{1-k}.$$

WLOG, we may assume $\bar{u} > 1$. Since the series converges geometrically, we have

$$\phi_2(s,y) = \phi_2(\alpha,y) + O(\delta(\log x)^2/\bar{u}^2).$$

Again from Lemma 4 of [5] we obtain that $\phi_2(\alpha,y) \gg (\log x)^2/\bar{u}$, which completes the proof. □

Corollary 4. *Under the same hypotheses as the previous lemma, we have*

$$\begin{aligned} \phi(s,y) &= \phi(\alpha,y) + O(\delta); \\ -\phi_1(s,y) &= \log x(1+O(\delta/\bar{u})). \end{aligned}$$

Proof. By the mean value theorem, there exists an s' between s and α such that

$$-\phi_1(s,y) = -\phi_1(\alpha,y) + (s-\alpha)\phi_2(s',y).$$

Applying Lemma 3 and bounding $\phi_2(\alpha,y)$ as before, we obtain

$$-\phi_1(s,y) = -\phi_1(\alpha,y) + O(\delta(\log x)/\bar{u}).$$

Noting that $-\phi_1(\alpha,y) = \log x$ completes the proof for ϕ_1. The proof for ϕ is easier and follows the same lines. □

Lemma 5. *If $|s-\alpha| \le (\bar{u}\log x)^{-1}$, then $HT(x,y,s) = HT(x,y,\alpha)(1+O(1/\bar{u}))$.*

Proof. Assume $|s-\alpha| \le \delta/\log x$ with $\delta = o(1)$. Applying Corollary 4 we have $x^s = x^\alpha e^{O(\delta)} = x^\alpha(1+O(\delta))$. We also have $\log\zeta(s,y) = \phi(s,y) = \phi(\alpha,y)+O(\delta)$ so that $\zeta(s,y) = \zeta(\alpha,y)e^{O(\delta)} = \zeta(\alpha,y)(1+O(\delta))$. Applying Lemma 3 and setting $\delta = 1/\bar{u}$ completes the proof. □

We have proven the following.

Theorem 6. *Algorithm HT outputs an approximation for $\Psi(x,y)$ accurate to within a factor of $1+O(1/\bar{u})$ using at most*

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

floating point operations.

2.3. **Using Newton's method.** Newton's method may be substituted for bisection in Step (2) of Algorithm HT. It is not difficult to show that, using $\alpha_0 := \log(1+y/(5\log x))/\log y$ for a starting point, Newton's method will converge. (The function $f(s) = \phi_1(s,y) + \log x$ has a negative second derivative with respect to s on $(0,1]$, so a starting point smaller than α, like α_0, is preferable.) In practice, we observed quadratic convergence after only 2 or 3 iterations of Newton's method, so that 5 or 6 iterations were sufficient. However, we are only able to prove quadratic convergence when using a starting point within a distance of $O(\bar{u}/\log x)$ from α.

Let x, y be fixed, and let $g(s) := s - f'(s)/f(s)$ so that $g(s)$ is the iteration function for Newton's method. Observe that $f'(s) = \phi_2(s,y)$ and $f''(s) = \phi_3(s,y)$. Thus $g'(s) = f(s)\phi_3(s,y)/\phi_2(s,y)^2$. We will also need formulas for ϕ_2 and ϕ_3:

$$\begin{aligned}\phi_2(s,y) &= \sum_{p\le y} \frac{p^s(\log p)^2}{(p^s-1)^2};\\ \phi_3(s,y) &= -\sum_{p\le y} \frac{p^s(p^s+1)(\log p)^3}{(p^s-1)^3}.\end{aligned}$$

Lemma 7. *For* $0 < s \le 1$, $-\phi_3(s,y)/\phi_2(s,y) \le \log y + 2/s$.

Proof. Observing that $s\log p \le p^s - 1$, we have

$$\begin{aligned}-\phi_3(s,y) &= \sum_{p\le y} \frac{p^s(p^s+1)(\log p)^3}{(p^s-1)^3}\\ &= \sum_{p\le y}\left(\log p + \frac{2\log p}{p^s-1}\right)\frac{p^s(\log p)^2}{(p^s-1)^2}\\ &\le \sum_{p\le y}\left(\log p + \frac{2}{s}\right)\frac{p^s(\log p)^2}{(p^s-1)^2}\\ &\le \left(\log y + \frac{2}{s}\right)\sum_{p\le y}\frac{p^s(\log p)^2}{(p^s-1)^2}\\ &= \left(\log y + \frac{2}{s}\right)\phi_2(s,y). \quad \square\end{aligned}$$

Theorem 8. *There exists a constant* $c > 0$ *such that if* $h \le c\bar{u}/\log x$, *then Newton's method converges quadratically to the solution* $s = \alpha$ *of* $f(s) = 0$ *on the interval* $[\alpha - h, \alpha]$.

Proof. We need to show that, for every $s \in [\alpha - h, \alpha]$, $|g'(s)| < 1$. We have

$$\begin{aligned}|g'(s)| &= \frac{|(\phi_1(s,y) + \log x)\phi_3(s,y)|}{\phi_2(s,y)^2}\\ &\le \frac{|\phi_1(s,y) + \log x|(\log y + 2/s)}{\phi_2(s,y)}\end{aligned}$$

using Lemma 7. By the mean value theorem, there exists a t, $s \le t \le \alpha$, such that

$\phi_1(s,y) = \phi_1(\alpha,y) + (s-\alpha)\phi_2(t,y)$. Observing that $\phi_1(\alpha,y) = -\log x$, we have

$$\begin{aligned} |g'(s)| &\leq \frac{(\alpha - s)\phi_2(t,y)(\log y + 2/s)}{\phi_2(s,y)} \\ &\leq h(\log y + 2/s), \end{aligned}$$

since $\phi_2(s,y)$ is a decreasing function of s. So if $h(\log y + 2/s) < 1$, we are done.

For c sufficiently small, we have

$$h \log y \leq c\bar{u} \log y/\log x \leq cu \log y/\log x = c < 1/2.$$

Since $\alpha \gg \bar{u}/\log x$ (see [5, Lemma 2]), we can choose c sufficiently small so that $\alpha > 5h$, giving us

$$2h/s \leq 2h/(\alpha - h) < 2h/(5h-h) = 1/2,$$

which completes the proof. □

This result implies that, to find a good approximation to α, one must first use bisection to reduce the interval containing α to size $O(\bar{u}/\log x)$. One may then take the left endpoint of this interval as the starting point for Newton's method to obtain an additional factor of $O(1/\bar{u}^2)$ of precision. If $\bar{u} = u$ (that is, $y \geq \log x$), then this method will slightly improve the operation count of Algorithm HT to $O(y/(\log\log y))$. In other words, finding the primes in Step (1) dominates the running time; the contribution of Step (2) drops to $O(y(\log\log y)/\log y)$. If $\bar{u} = y/\log y$ (that is, $y < \log x$), the improvement is (theoretically) negligible.

3. Algorithms based on Theorem 1

We implemented four different algorithms based on the approximation $\Psi(x,y) \approx x\rho(u)$.

Algorithm A. $\rho(u)$ can be computed by numeric integration using

$$\rho(u) = \frac{1}{u}\int_{u-1}^{u} \rho(t)\,dt \quad (u \geq 1) \tag{1}$$

with, say, Simpson's rule (see [17]). The idea is as follows. For $1 \leq u \leq 2$, we have $\rho(u) = 1 - \log u$, which is easy to compute. Then compute $\rho(i + j/n)$ starting with $i = 2$ and $j = 0 \ldots n$ using (1) and Simpson's rule. Repeat this process with $i = 2, 3, \ldots$, and observe that those points where values of ρ are needed for Simpson's rule are precisely where ρ was previously computed.

This requires $O(n^2 u)$ operations. We used $n = 1000$, and our values of $\rho(u)$ match those given by van de Lune and Wattel [17, Table 1, p. 420] for integral values of u satisfying $2 \leq u \leq 20$.

Algorithm B. We have $\rho(u) = u^{-u+o(u)}$, giving the approximation $\Psi(x,y) \approx x \cdot u^{-u}$. This crude estimate can be computed using only $O(1)$ floating point operations.

Algorithm C. Pomerance [13] observed that for $5 < u < 11$,

$$\rho(u) \approx C(u) := \exp(-u(\log u + 0.56 - 1/\log(u+1))),$$

giving the approximation $\Psi(x,y) \approx x \cdot C(u)$. This can also be computed in $O(1)$ operations.

Algorithm D. Let

$$D(u) := \exp\left\{-u\left(\log u + \log\log u - 1 + \frac{\log\log u - 1}{\log(u+1)}\right)\right\}.$$

De Bruijn [3] showed that

$$\rho(u) = D(u)\exp\left\{O\left(u\left(\frac{\log\log u}{\log u}\right)^2\right)\right\}$$

for $u \geq 3$. This gives the approximation $\Psi(x,y) \approx x \cdot D(u)$, which can be computed in $O(1)$ operations. (This approximation breaks down for small u.)

4. A comparison of approximations

We began by computing $\Psi(x,y)$ for $x = 2^i$, $i = 10, 11, \ldots, 33$ and $y = 2^j$, $j = 1, 2, \ldots, 15$ (see Table 3). We then ran our five algorithms on the same x, y values to obtain estimates to compare to the actual values. On the following pages we present tables giving the ratios between the estimates and the actual data, so that a perfect estimate will yield 1.00 entries. In the interest of space, we present the data only for Algorithm HT and Algorithm A, as the data for the other algorithms can be quickly computing from Table 3. Each table is indexed by x across the top and y down the left.

Let us summarize our results.

- *Algorithm HT* (Table 1). $HT(x, y, \alpha')$ provides a very good approximation to $\Psi(x,y)$, with the ratio always between 0.9 and 1.1. It took about 5 minutes of CPU time on a workstation to produce the approximations used in the table, with the list of primes up to 2^{15} computed only once.
- *Algorithm A* (Table 2). This approximation is fairly good for small values of u, but $HT(x,y)$ appears to be uniformly better. The ratio $x\rho(u)/\Psi(x,y)$ approaches 1 as u decreases, and for fixed u, as x increases. It took less than 2 minutes of CPU time to compute the approximations in the table. $\rho(u)$ was computed once for all u up to 20, with $n = 1000$.
- *Algorithms B, C, and D.* These approximations were only close for small u, with Algorithm B best near $u = 1.5$ and Algorithm C best near $u = 2.5$. None were as good as Algorithm A, and it is fair to say that Algorithm D is essentially useless. As Algorithm A does only a mediocre job of estimating $\Psi(x,y)$, it is not surprising that Algorithms B, C, and D, which are based on poorer estimates of $\rho(u)$, do not do very well.

For a fast, crude approximation, Algorithms B or C may be useful. Otherwise, it is clearly a choice between Algorithms HT and A.

Our program to compute the exact values of $\Psi(x,y)$ uses sieving, and it factors roughly a billion numbers over the primes up to 2^{15} in one CPU day on an HP 9000 series 715/75 workstation. The values in Table 3 for smaller y were checked using the recursive algorithm based on Buchstab's identity mentioned in §1. All programs were implemented in ANSI C.

TABLE 1. The approximation $HT(x, y, \alpha')/\Psi(x, y)$

	x									
y	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}
2^1	1.0000	1.0000	1.0769	1.0714	1.0667	1.0625	1.0588	1.0556	1.0526	1.0500
2^2	1.0244	1.0417	1.0357	1.0308	1.0405	1.0357	1.0316	1.0377	1.0339	1.0385
2^3	1.0140	1.0158	1.0161	1.0189	1.0200	1.0201	1.0212	1.0200	1.0199	1.0212
2^4	1.0122	1.0114	1.0122	1.0119	1.0121	1.0141	1.0140	1.0142	1.0138	1.0142
2^5	1.0023	1.0058	1.0085	1.0076	1.0072	1.0076	1.0084	1.0075	1.0074	1.0076
2^6	1.0085	1.0031	1.0025	1.0043	1.0053	1.0055	1.0044	1.0040	1.0047	1.0051
2^7	1.0082	1.0094	1.0046	1.0003	1.0005	1.0036	1.0050	1.0047	1.0036	1.0026
2^8	1.0000	1.0086	1.0101	1.0087	1.0038	0.9995	1.0002	1.0029	1.0047	1.0047
2^9	0.9863	0.9983	1.0070	1.0121	1.0121	1.0086	1.0035	0.9995	0.9999	1.0027
2^{10}	–	0.9843	0.9969	1.0070	1.0128	1.0145	1.0125	1.0082	1.0032	0.9996
2^{11}	–	–	0.9849	0.9965	1.0063	1.0131	1.0162	1.0158	1.0126	1.0079
2^{12}	–	–	–	0.9843	0.9957	1.0057	1.0133	1.0175	1.0184	1.0165
2^{13}	–	–	–	–	0.9849	0.9954	1.0054	1.0133	1.0183	1.0203
2^{14}	–	–	–	–	–	0.9850	0.9951	1.0051	1.0133	1.0189
2^{15}	–	–	–	–	–	–	0.9854	0.9951	1.0048	1.0132

y	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}	2^{28}	2^{29}
2^1	1.0476	1.0455	1.0435	1.0417	1.0800	1.0769	1.0741	1.0714	1.0690	1.0667
2^2	1.0420	1.0382	1.0409	1.0430	1.0396	1.0413	1.0426	1.0395	1.0406	1.0414
2^3	1.0218	1.0224	1.0213	1.0207	1.0212	1.0217	1.0211	1.0207	1.0206	1.0208
2^4	1.0144	1.0143	1.0142	1.0140	1.0141	1.0140	1.0139	1.0140	1.0138	1.0139
2^5	1.0077	1.0077	1.0076	1.0076	1.0076	1.0076	1.0076	1.0076	1.0076	1.0076
2^6	1.0049	1.0047	1.0046	1.0047	1.0048	1.0048	1.0048	1.0047	1.0047	1.0047
2^7	1.0025	1.0032	1.0036	1.0035	1.0032	1.0029	1.0029	1.0030	1.0032	1.0031
2^8	1.0038	1.0026	1.0018	1.0019	1.0026	1.0030	1.0030	1.0026	1.0023	1.0021
2^9	1.0047	1.0053	1.0048	1.0036	1.0023	1.0016	1.0018	1.0026	1.0030	1.0031
2^{10}	1.0003	1.0031	1.0052	1.0061	1.0058	1.0048	1.0034	1.0023	1.0019	1.0022
2^{11}	1.0031	1.0000	1.0010	1.0038	1.0060	1.0071	1.0071	1.0062	1.0049	1.0036
2^{12}	1.0126	1.0078	1.0033	1.0007	1.0017	1.0045	1.0067	1.0081	1.0083	1.0077
2^{13}	1.0196	1.0167	1.0125	1.0078	1.0037	1.0014	1.0026	1.0053	1.0076	1.0090
2^{14}	1.0218	1.0221	1.0202	1.0168	1.0124	1.0079	1.0041	1.0022	1.0035	1.0061
2^{15}	1.0193	1.0229	1.0241	1.0231	1.0204	1.0166	1.0122	1.0080	1.0046	1.0031

y	2^{30}	2^{31}	2^{32}	2^{33}
2^1	1.0645	1.0625	1.0606	1.0588
2^2	1.0421	1.0426	1.0400	1.0431
2^3	1.0211	1.0211	1.0210	1.0208
2^4	1.0139	1.0139	1.0140	1.0140
2^5	1.0076	1.0076	1.0076	1.0076
2^6	1.0048	1.0047	1.0047	1.0047
2^7	1.0030	1.0029	1.0029	1.0029
2^8	1.0021	1.0023	1.0024	1.0023
2^9	1.0028	1.0023	1.0020	1.0018
2^{10}	1.0030	1.0034	1.0035	1.0032
2^{11}	1.0027	1.0024	1.0028	1.0036
2^{12}	1.0066	1.0052	1.0040	1.0032
2^{13}	1.0095	1.0091	1.0081	1.0068
2^{14}	1.0084	1.0100	1.0106	1.0104
2^{15}	1.0044	1.0070	1.0093	1.0109

TABLE 2. The approximation $x\rho(u)/\Psi(x,y)$

	x									
y	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}
2^1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2^2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2^3	0.1678	0.1211	0.0806	0.0536	0.0350	0.0241	0.0147	0.0093	0.0055	0.0037
2^4	0.5406	0.4729	0.4061	0.3467	0.2936	0.2463	0.2043	0.1681	0.1372	0.1112
2^5	0.7104	0.6546	0.6068	0.5593	0.5124	0.4666	0.4243	0.3843	0.3468	0.3115
2^6	0.8506	0.8226	0.7881	0.7506	0.7204	0.6909	0.6609	0.6309	0.5993	0.5695
2^7	0.9015	0.8835	0.8665	0.8483	0.8249	0.8000	0.7791	0.7597	0.7420	0.7221
2^8	0.9376	0.9208	0.9072	0.8970	0.8857	0.8728	0.8564	0.8386	0.8246	0.8119
2^9	0.9652	0.9435	0.9287	0.9191	0.9108	0.9028	0.8940	0.8835	0.8702	0.8565
2^{10}	–	0.9696	0.9512	0.9401	0.9321	0.9261	0.9205	0.9148	0.9083	0.9003
2^{11}	–	–	0.9737	0.9571	0.9471	0.9406	0.9359	0.9319	0.9279	0.9234
2^{12}	–	–	–	0.9752	0.9606	0.9517	0.9461	0.9423	0.9390	0.9359
2^{13}	–	–	–	–	0.9780	0.9645	0.9566	0.9518	0.9486	0.9461
2^{14}	–	–	–	–	–	0.9792	0.9668	0.9598	0.9555	0.9527
2^{15}	–	–	–	–	–	–	0.9808	0.9696	0.9630	0.9592

y	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}	2^{28}	2^{29}
2^1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2^2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2^3	0.0023	0.0013	0.0006	0.0005	0.0004	0.0000	0.0000	0.0000	0.0000	0.0000
2^4	0.0893	0.0711	0.0562	0.0441	0.0344	0.0266	0.0204	0.0155	0.0118	0.0089
2^5	0.2788	0.2489	0.2213	0.1959	0.1729	0.1521	0.1333	0.1164	0.1014	0.0880
2^6	0.5421	0.5139	0.4860	0.4597	0.4333	0.4080	0.3845	0.3607	0.3378	0.3167
2^7	0.7013	0.6801	0.6601	0.6409	0.6218	0.6040	0.5847	0.5655	0.5468	0.5286
2^8	0.7998	0.7873	0.7740	0.7601	0.7461	0.7329	0.7204	0.7080	0.6954	0.6826
2^9	0.8459	0.8368	0.8281	0.8178	0.8085	0.7987	0.7884	0.7782	0.7688	0.7600
2^{10}	0.8902	0.8797	0.8719	0.8654	0.8594	0.8534	0.8471	0.8404	0.8333	0.8259
2^{11}	0.9181	0.9118	0.9038	0.8954	0.8893	0.8843	0.8798	0.8754	0.8724	0.8675
2^{12}	0.9326	0.9288	0.9243	0.9188	0.9120	0.9057	0.8998	0.8965	0.8936	0.8892
2^{13}	0.9437	0.9413	0.9386	0.9354	0.9317	0.9271	0.9215	0.9158	0.9118	0.9087
2^{14}	0.9505	0.9486	0.9467	0.9445	0.9421	0.9393	0.9359	0.9318	0.9269	0.9227
2^{15}	0.9568	0.9550	0.9534	0.9519	0.9503	0.9484	0.9463	0.9438	0.9408	0.9373

y	2^{30}	2^{31}	2^{32}	2^{33}
2^1	0.0000	0.0000	0.0000	0.0000
2^2	0.0000	0.0000	0.0000	0.0000
2^3	0.0000	0.0000	0.0000	0.0000
2^4	0.0066	0.0049	0.0036	0.0027
2^5	0.0761	0.0656	0.0563	0.0483
2^6	0.2958	0.2758	0.2574	0.2392
2^7	0.5106	0.4928	0.4765	0.4591
2^8	0.6697	0.6568	0.6442	0.6320
2^9	0.7512	0.7424	0.7317	0.7225
2^{10}	0.8187	0.8121	0.8060	0.8000
2^{11}	0.8624	0.8569	0.8514	0.8460
2^{12}	0.8863	0.8833	0.8783	0.8748
2^{13}	0.9059	0.9033	0.9007	0.8995
2^{14}	0.9185	0.9167	0.9137	0.9123
2^{15}	0.9331	0.9285	0.9267	0.9241

TABLE 3. Values of $\Psi(x, y)$

	x				
y	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
2^1	11	12	13	14	15
2^2	41	48	56	65	74
2^3	143	190	248	317	400
2^4	246	351	490	672	906
2^5	442	689	1053	1586	2346
2^6	589	981	1595	2542	3991
2^7	731	1270	2179	3679	6094
2^8	849	1516	2684	4699	8146
2^9	949	1735	3142	5636	10041
2^{10}	1024	1911	3521	6428	11663
2^{11}	1024	2048	3841	7129	13128
2^{12}	1024	2048	4096	7728	14427
2^{13}	1024	2048	4096	8192	15512
2^{14}	1024	2048	4096	8192	16384
2^{15}	1024	2048	4096	8192	16384
y	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}
2^1	16	17	18	19	20
2^2	84	95	106	118	130
2^3	498	614	749	905	1083
2^4	1202	1576	2041	2616	3317
2^5	3414	4898	6948	9731	13475
2^6	6180	9451	14266	21260	31347
2^7	9931	16011	25559	40390	63173
2^8	13944	23481	39048	64361	105278
2^9	17755	31128	54004	92441	156524
2^{10}	21037	37733	67249	118972	208572
2^{11}	24032	43786	79421	143389	257471
2^{12}	26748	49339	90654	165976	302766
2^{13}	29113	54284	100763	186423	343868
2^{14}	31156	58732	110048	205414	382279
2^{15}	32768	62506	118264	222583	417362
y	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}
2^1	21	22	23	24	25
2^2	143	157	171	186	202
2^3	1286	1517	1780	2074	2402
2^4	4167	5193	6419	7877	9598
2^5	18469	25075	33741	45014	59579
2^6	45781	66227	94933	134903	190163
2^7	97785	149884	227857	343760	514788
2^8	170865	275111	439306	695673	1093094
2^9	263183	439766	730433	1205820	1977771
2^{10}	361431	620771	1060020	1800976	3045268
2^{11}	459301	812792	1424059	2476303	4285281
2^{12}	550007	994335	1787283	3190043	5644630
2^{13}	632444	1159443	2117721	3851153	6967074
2^{14}	709690	1314383	2428041	4472213	8209619
2^{15}	780651	1457130	2714319	5045563	9357112

Values of $\Psi(x, y)$ (continued)

	x								
y	2^{25}	2^{26}	2^{27}	2^{28}	2^{29}	2^{30}	2^{31}	2^{32}	2^{33}
2^1	26	27	28	29	30	31	32	33	34
2^2	218	235	253	271	290	309	329	350	371
2^3	2767	3174	3625	4121	4665	5260	5912	6624	7398
2^4	11621	13988	16738	19928	23604	27828	32659	38169	44433
2^5	78266	102093	132291	170321	217974	277365	351007	441920	553606
2^6	266049	369553	509807	698608	951343	1287797	1733329	2320213	3089396
2^7	765239	1129330	1655298	2410848	3490422	5024347	7191651	10237430	14496552
2^8	1706546	2648250	4085325	6264989	9551140	14477597	21826231	32739765	48876982
2^9	3222076	5213665	8383431	13410847	21350049	33828954	53347812	83727582	130784134
2^{10}	5124129	8577810	14281205	23642843	38922285	63750581	103979865	168933457	273408517
2^{11}	7385245	12678351	21679370	36917071	62587985	105617305	177380226	296511968	493575295
2^{12}	9922778	17371995	30309172	52711748	91378192	157873692	271778274	466079862	796104188
2^{13}	12525194	22345935	39643653	70084947	123541536	217188821	380801428	665799447	1160628960
2^{14}	15010728	27317440	49436058	88862657	158941679	283424267	504144912	894722064	1584321399
2^{15}	17306519	31909829	58621645	107234657	195171858	353089794	636009446	1142579675	2048212809

5. More algorithms and some open problems

We conclude with a brief discussion of two more algorithms, followed by some suggestions for future work.

5.1. **Another algorithm to compute $\Psi(x, y)$.**

The referee pointed out that $\Psi(x, y)$ may be computed using the following:

$$\Psi(x, y) = \sum_{P(n)>y} \mu(n) \left\lfloor \frac{x}{n} \right\rfloor,$$

where $P(n)$ is the smallest prime divisor of n, and μ is the Möbius function. Using $\lfloor x/(ab) \rfloor = \lfloor \lfloor x/a \rfloor / b \rfloor$, this may be evaluated via a tree-traversal based on the primes p where $y < p \leq x$. With a list of such primes p in hand, this takes $O(x/\log y)$ operations when $2 \leq y \leq x/2$.

5.2. **A probabilistic approximation algorithm.** The referee also pointed out that $\Psi(x, y)$ may be estimated probabilistically. This is, in effect, how factoring and discrete logarithm practitioners estimate $\Psi(x, y)$.

The idea is to choose integers $\leq x$ at random and factor them over the primes up to y. The proportion of numbers that completely factor will give an estimate for the ratio $\Psi(x, y)/x$, from which an estimate for $\Psi(x, y)$ can be made. Because this is a binomial distribution, if $kn^2x/\Psi(x, y)$ random numbers are chosen, then with probability at least $1 - 1/k$, the estimate for $\Psi(x, y)$ will be accurate within a factor of $1 \pm 1/n$. For example, we could set $k = 10$ and $n = 10$, which requires $1000x/\Psi(x, y)$ random numbers, giving an estimate for $\Psi(x, y)$ that, with probability at least 90%, is accurate within $\pm 10\%$. We do not know $\Psi(x, y)$ beforehand, but this is not a problem; simply continue sampling until 1000 integers are found that completely factor over the primes $\leq y$.

5.3. **Open problems.** We now discuss two open problems.

Considering how accurate Theorem 2 is in practice, upper and lower bounds for $\Psi(x, y)$ with explicit constants patterned after this theorem would be very useful. It would also be nice to know whether the error term $1 + O(1/\bar{u})$ is best possible. Our data implies it is not, especially for smaller u.

Let $f(t)$ denote a monic, irreducible polynomial with integer coefficients. Let $\Psi_f(x, y)$ denote the number of integers $n \leq x$ such that $f(n)$ has no prime divisors larger than y. Is it possible to construct an algorithm to approximate $\Psi_f(x, y)$ to within a constant factor? For approximations to $\Psi_f(x, y)$, see [7, 8, 16].

Acknowledgments

We wish to thank Pieter Moree for several helpful comments on §2. We also wish to thank the anonymous referee for many very helpful comments that greatly improved this paper.

References

1. D. J. Bernstein, *Enumerating and counting smooth integers*, Ch. 2, Ph.D. Thesis, University of California at Berkeley, May 1995.
2. E. R. Canfield, P. Erdős, and C. Pomerance, *On a problem of Oppenheim concerning "Factorisatio Numerorum"*, Journal of Number Theory **17** (1983), 1–28. MR **85j:**11012
3. 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. MR **13:**326f
4. A. Hildebrand, *On the number of positive integers $\leq x$ and free of prime factors $> y$*, Journal of Number Theory **22** (1986), 289–307. MR **87d:**11066

→ A. Hildebrand and G. Tenenbaum, *On integers free of large prime factors*, Trans. AMS **296** (1986), no. 1, 265–290. MR **87f:**11066

6. ———, *Integers without large prime factors*, Journal de Théorie des Nombres de Bordeaux **5** (1993), 411–484. MR **95d:**11116
7. N. A. Hmyrova, *On polynomials with small prime divisors*, Doklady Akad. Nauk SSSR **155** (1964), 1268–1271, Translation in *Soviet Mathematics*, AMS. MR **28:**3975
8. ———, *On polynomials with small prime divisors II*, Izv. Akad. Nauk SSSR Ser. Mat. **30** (1966), 1367–1372. MR **34:**7475
9. A. K. Lenstra and H. W. Lenstra Jr. (eds.), *The development of the number field sieve*, Lecture Notes in Mathematics, vol. 1554, Springer-Verlag, Berlin and Heidelberg, Germany, 1993. MR **96m:**11116
10. Pieter Moree, *Psixyology and diophantine equations*, Ph.D. thesis, Rijksuniversiteit Leiden, 1993.

11. Karl K. Norton, *Numbers with small prime factors, and the least kth power non-residue*, Memoirs of the American Mathematical Society, vol. 106, American Mathematical Society, Providence, Rhode Island, 1971. MR **44:**3948
12. C. Pomerance (ed.), *Cryptology and computational number theory*, Proceedings of Symposia in Applied Mathematics, vol. 42, American Mathematical Society, Providence, Rhode Island, 1990. MR **92e:**94023
13. ———, *Factoring*, Cryptology and Computational Number Theory [12], American Mathematical Society, 1990, pp. 27–48. MR **92b:**11089
14. P. Pritchard, *A sublinear additive sieve for finding prime numbers*, Communications of the ACM **24** (1981), no. 1, 18–23,772. MR **82c:**10011
15. J. Sorenson, *An introduction to prime number sieves*, Tech. Report #909, Computer Sciences Department, University of Wisconsin-Madison, January 1990.
16. N. M. Timofeev, *On polynomials with small prime divisors*, Taškent. Gos. Univ. Naučn. Trudy; Voprosy Mat. **548** (1977), 87–91,145. MR **58:**27845
17. J. van de Lune and E. Wattel, *On the numerical solution of a differential-difference equation arising in analytic number theory*, Mathematics of Computation **23** (1969), 417–421. MR **40:**1050

Department of Mathematics and Computer Science, Butler University, 4600 Sunset Ave., Indianapolis, Indiana 46208

Department of Mathematics and Computer Science, Butler University, 4600 Sunset Ave., Indianapolis, Indiana 46208
E-mail address: sorenson@butler.edu