Computing $\pi(x)$: An Analytic Method

J. C. Lagarias
A. M. Odlyzko

AT&T Labs - Research
Murray Hill, NJ 07974

1. Introduction

The problem of computing $\pi(x)$, the number of primes $p \le x$, is a very old one. The sieve of Eratosthenes, in its usual form, finds all primes $p \le x$ in

$$O(x \log \log x)$$

steps, where we consider for our model of computation a random access machine with about $x \lceil \log_2 x \rceil$-bit storage locations, and where a ''step'' is defined as any elementary arithmetic operation on integers $\le x$. Recently an improved version of this sieve was found [18], which requires only

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

steps on such a machine. On the other hand, by the Prime Number Theorem

$$\pi(x) \sim \frac{x}{\log x} \quad \text{as} \quad x \to \infty , \tag{1.1}$$

so one cannot hope to find an algorithm that finds all primes $p \le x$ in fewer than about $x(\log x)^{-1}$ operations.

For a long time no method for computing $\pi(x)$ faster than finding all primes $p \le x$ was known. For example, Gauss was led to conjecture the Prime Number Theorem (1.1) by inspection of counts of primes in various intervals that had been compiled by various authors, but those counts were apparently obtained by the sieve of Eratosthenes. Legendre [5] used the principle of inclusion-exclusion to obtain the formula

$$\pi(x) - \pi(\sqrt{x}) + 1 = [x] - \sum_{p \le x^{1/2}} [\frac{x}{p}] + \sum_{p < q \le x^{1/2}} [\frac{x}{pq}] - \ldots,$$

where p and q run over primes. The sum on the right side above contains approximately $6\pi^{-2}(1 - \log\ 2)x$ nonzero terms, and so is not suitable for computation. The first substantial progress towards an efficient algorithm was made in the second half of the 19-th century by the astronomer Meissel, who found a combinatorial method [15] that enabled him to compute several large values of $\pi(x)$. Meissel's work culminated in his publication of the value of $\pi(10^9)$ [16] (which turned out to be too small by 56). Many other mathematicians came up with variants of Meissel's method (see [5]), but in the era of hand computation none were willing to try their methods on $\pi(x)$ for $x > 10^9$. (It was pointed out by B. Berndt that there is a curious table of values of $\pi(x)$ for a few $x \le 10^8$ in a notebook of Ramanujan [19]. They are all either correct or else very close to the true value, but there is no indication how they were derived. They are more accurate than the values given by Ramanujan's approximate formulas for $\pi(x)$ [9].)

Further progress on computing $\pi(x)$ was made by D. H. Lehmer [12], who significantly simplified and generalized Meissel's method and used it to compute $\pi(10^{10})$ on an electronic computer. (His value of $\pi(10^{10})$ turned out to be too high by 1 [2].) Further computations were carried out by Mapes [14] and Bohman [2], who computed several values up to and including $\pi(10^{13})$. (Bohman's value of $\pi(10^{13})$ turned out to be too low by 941 [10].)

The asymptotic running times of Meissel's and Lehmer's algorithms for computing $\pi(x)$ are rather difficult to estimate, but as is pointed out in [10,11], all the published versions seem to require time $\ge c(\varepsilon)x^{1-\varepsilon}$ for any $\varepsilon > 0$. However, V. Miller and the authors of this paper obtained a new version of that method which requires only $O(x^{2/3+\varepsilon})$ time and $O(x^{1/3+\varepsilon})$ space to compute $\pi(x)$, see [10]. This version of the Meissel-Lehmer method has been used to compute values of $\pi(x)$ up to $x = 4\times10^{16}$.

In this paper we present new algorithms for computing $\pi(x)$, based on entirely different ideas, which require $O(x^{3/5+\varepsilon})$ time and $O(x^{\varepsilon})$ space for any $\varepsilon > 0$ in one version and $O(x^{1/2+\varepsilon})$ time and $O(x^{1/4+\varepsilon})$ space in another version. We call this class of algorithms *analytic* $\pi(x)$-*algorithms*. An informal description of the algorithms is presented in Section 2. That section also discusses generalizations of these algorithms to compute other arithmetical functions. Section 3 provides a more formal definition of the algorithms together with a rigorous estimate of their running time, while Section 4 contains the proof of the lemmas needed in Section 3.

The announcement [11] of analytic algorithms for computing $\pi(x)$ presented only the $O(x^{3/5+\varepsilon})$ time and $O(x^{\varepsilon})$ space algorithm. A new method of computing multiple values of the Riemann zeta function [17] enables us to describe here, for each b with $0 \leq b \leq 1/4$, an algorithm for computing $\pi(x)$ that runs in time $O(x^{(3-2b)/5+\varepsilon})$ and space $O(x^{b+\varepsilon})$ for every $\varepsilon > 0$.

Our work on this subject was stimulated by H. S. Wilf's note [23], which discusses the question of what is a satisfactory answer to an enumerative problem.

2. Informal Description and Discussion

We shall carry out the complexity analysis of analytic $\pi(x)$-algorithms using for a model of computation a random access machine (RAM) with $O(\log x)$ bit addresses. In fact the analysis can be carried over to apply to a multi-tape Turing machine, since the x^{ε} factors are more than sufficient to take care of the extra complexity introduced by counting bit operations, and for the specific application that is made in this paper, the algorithm of [17] can be implemented on a multi-tape Turing machine. This result contrasts with the Meissel-Lehmer-type algorithm of [10] which relies on sieving in an essential way, and sieving cannot be implemented quickly on a Turing machine. For simplicity of presentation, we will only count the number of elementary operations on $O(\log x)$ bit numbers that our algorithms perform.

The analytic $\pi(x)$-algorithms are motivated by the exact formulas of prime number theory, which express many arithmetical functions in terms of zeros of the Riemann zeta and related functions. For example, Riemann's formula [6] states that if

$$\pi^*(x) = \sum_p \sum_{\substack{m=1 \\ p^m \le x}}^{\infty}{}' \frac{1}{m} ,$$

where p runs over the primes, and the prime in the summation indicates that if $x = p^m$ for some prime p, then one should take $(2m)^{-1}$ instead of m^{-1} in the sum, then

$$\pi^*(x) = Li(x) - \sum_\rho Li(x^\rho) - \log 2 + \int_x^\infty \frac{dt}{t(t^2-1)\log t} , \tag{2.1}$$

where $Li(u)$ is given by

$$Li(u) = \int_0^u \frac{dt}{\log t} ,$$

the value of the integral being its Cauchy principal value, and ρ runs over zeros of $\zeta(s)$ with $\mathrm{Re}(\rho) > 0$.

The class of analytic $\pi(x)$-algorithms is actually developed from integral transform formulas from which ‘‘explicit formulas’’ like (2.1) are derived. For example, (2.1) comes from the identity

$$\pi^*(x) = \sum_p \sum_{\substack{m=1 \\ p^m \le x}}^{\infty}{}' \frac{1}{m} = \frac{1}{2\pi i} \int_{(a)} \frac{x^s}{s} \log \zeta(s) \, ds , \tag{2.2}$$

where (a) for $a > 1$ denotes the straight line $s = a + it$, $-\infty < t < \infty$. Eq. (2.2) follows immediately from the Euler product for $\zeta(s)$, which yields for $\mathrm{Re}(s) > 1$

$$\log \zeta(s) = \log \prod_p (1-p^{-s})^{-1} = \sum_p \log(1-p^{-s})^{-1} = \sum_p \sum_{m=1}^{\infty} \frac{1}{m}(p^m)^{-s} , \tag{2.3}$$

and the classical formula [3]

$$\frac{1}{2\pi i} \int_{(a)} \frac{(x/n)^s}{s}\, ds = \begin{cases} 1 & n < x, \\ 1/2 & n = x, \\ 0 & n > x\ . \end{cases} \tag{2.4}$$

The sum on the left side of (2.2) is almost exactly what we want to compute; it differs from $\pi(x)$ (for $x \notin Z$, say) only by the contribution of primes $p \leq \sqrt{x}$, and those terms can be computed in $O(x^{1/2+\varepsilon})$ steps to within $\pm 1/10$, say. (These terms could be computed much faster by the recursive use of an analytic $\pi(x)$-algorithm, but that is not important.) Since $\pi(x)$ is an integer, to know it exactly it would therefore suffice to compute the integral on the right side of (2.2) to within $\pm 1/5$. Unfortunately that is not easy to do, since on any line $s = a + it$, $|x^s| = x^a$, and $\log \zeta(s)$ oscillates boundedly, so the integral is not absolutely convergent. The rate of convergence can be estimated [3], but it is not sufficient for our purposes. This seems to be the reason that Riesel and Göhl [20], who investigated formulas like (2.1) numerically, did not obtain very good results.

The main idea of the analytic $\pi(x)$-algorithms is to use, in place of (2.3), another of the family of Mellin transform identities to which (2.2) belongs, namely

$$\sum_p \sum_{m=1}^{\infty} \frac{c(p^m)}{m} = \frac{1}{2\pi i} \int_{(a)} F(s) \log \zeta(s)\, ds\ , \tag{2.5}$$

where $c(u)$ and $F(s)$ are a Mellin transform pair:

$$c(u) = \frac{1}{2\pi i} \int_{(a)} F(s) u^{-s} ds\ , \tag{2.6}$$

$$F(s) = \int_0^{\infty} c(u) u^{s-1} du\ . \tag{2.7}$$

(Here $c(u)$ and $F(s)$ have to be sufficiently ‘‘nice’’ for the integrals in (2.6) and (2.7) to converge, but this will easily hold for the functions we will use.) Ideally we would like to utilize the previously used step function $c_0(u)$ with $c_0(u) = 1$ for $u < x$ and $c_0(u) = 0$ for $u > x$, but then, by (2.7), we would again have $F(s) = s^{-1} x^s$, which makes the integral difficult to compute. Instead, we use a function $c(u)$ that is very close to $c_0(u)$; for an appropriately chosen parameter $y \in (1,x)$, we will have $c(u) = 1$ for $0 \le u \le x-y$, $c(u) = 0$ for $u \ge x$, and $0 \le c(u) \le 1$ for $x-y \le u \le x$. In the interval $(x-y,x)$, $c(u)$ will be chosen so as to make $F(s)$ decrease very rapidly as $|\mathrm{Im}(s)| \to \infty$, which will make it possible to evaluate the integral in (2.4) numerically. The integral, of course, will no longer equal $\pi^*(x)$. However, the difference will be

$$\pi^*(x) - \sum_p \sum_{m=1}^{\infty} \frac{c(p^m)}{m} = \sum_{\substack{p \\ x-y<p^m<x}} \sum_{m=1}^{\infty} \frac{1-c(p^m)}{m} , \tag{2.8}$$

which depends only on the $p^m \in (x-y,x)$. All these prime powers can be determined (using primality testing, or sieving) in about y steps ($O(yx^{\varepsilon})$, to be precise). Therefore, if $c(u)$ is easy to compute accurately, we can obtain $\pi^*(x)$, and hence $\pi(x)$, in about y steps (for $y > x^{1/2}$, say) once we compute the integral in (2.5). It turns out that one can choose $c(u)$ so that in the integral in (2.5) the contribution of the region with $|\mathrm{Im}(s)| \ge T$ is negligible for $T \gtrsim xy^{-1}$. Now to compute the integral over the range $|\mathrm{Im}(s)| \le T$ numerically requires $O(Tx^{\varepsilon})$ evaluations of the integrand, which, provided $F(s)$ is easy to compute, reduces to evaluating $\zeta(s)$. Given any β, $0 \le \beta \le 1/2$, the algorithm of [17] computes all the necessary values of $\zeta(a+it)$ for $u \le t \le u + u^{\beta}$ in about $u^{1/2}$ operations using u^{β} storage, so the computation of the integral takes about $T^{3/2-\beta}$ steps and space T^{β}. (For technical reasons, in some ranges we use the Euler-Maclaurin formula to compute $\zeta(s)$, but this does not affect the running time analysis.) Thus the total running time of the resulting algorithm is on the order of

$$y + T^{3/2-\beta} ,$$

where $yT \gtrsim x$, and the space requirement is on the order of T^{β}. We select $\beta = 5b/(2(1+b))$, $T = x^{b/\beta}$, and y on the order of x/T, and obtain the desired result.

In the next two sections we show that the heuristic analysis presented above can be made rigorous. Three main ingredients are used which make the new algorithm fast: 1) the ability to choose $c(u)$ so that both $c(u)$ and $F(s)$ are easy to compute, and $F(s)$ decreases rapidly, while $c(u)$ is nearly a step function, 2) the ability to find all the primes and prime powers in the interval $(x-y,x)$ in time not much larger than y, and 3) the ability to compute the integral numerically quickly, using the algorithm of [17] to compute values of $\zeta(s)$. It is not possible to improve on the first two of these ingredients. Since a function and its Mellin transform cannot both be rapidly decreasing (this is related to the uncertainty principle of quantum mechanics), we cannot choose $c(u)$ with $c(u) = 1$ for $0 \le u \le x-y$ and $c(u) = 0$ for $u \ge x$ for which $F(s)$ will decrease significantly faster than the one we utilize (cf. [13]). As far as the second ingredient is concerned, there are at least $y^{1-\varepsilon}$ primes and prime powers in the interval $(x-y,x)$, while we can find them in time $O(y^{1+\varepsilon})$, for any $\varepsilon > 0$, so little improvement is possible in this area. Improvements in the algorithm could come from improving the third ingredient. The new algorithm of [17] enables one to compute individual values of $\zeta(s)$ to reasonable accuracy (within $|s|^{-c}$ for any $c>0$) in average time $O(|s|^{\varepsilon})$ (although at the cost of requiring $O(|s|^{1/2})$ storage), which is essentially optimal. However, the new algorithm only requires the computation of the integral in (2.5), and it is conceivable that there are ways to compute the whole integral directly using various functional equations or integral representations that would be faster than using quadrature formulae but we have not found any way to do this.

In terms of space requirements, numerical integration requires only $O(x^{\varepsilon})$ locations. Finding all the primes $p \in (x-y,x)$ requires more than $O(x^{\varepsilon})$ locations if one uses sieving, but the recent primality testing method of Adleman et al. [1] makes it possible to bypass that problem and implement the entire algorithm in $O(x^{\varepsilon})$ storage locations. Thus the only place where nontrivial storage is required is in the implementation of the algorithm of [17] for computing the zeta function. It should also be noted that analytic $\pi(x)$-algorithms lend themselves to parallel implementation, as both numerical integration and primality testing can be carried out by breaking the ranges of operation into blocks and performing the operation independently on each block. Thus with *m* processors one can achieve essentially *m*-fold speedup.

It is possible to replace the $O(x^{\varepsilon})$ terms in the bounds on both the running times and the space requirements in these algorithms by smaller and more explicit expressions. To simplify the exposition, we have not done so. In order to implement these algorithms, it would be necessary to optimize many parameters and to explicitly determine various constants, and this would require substantial additional work. For example, one would need to determine explicit estimates for the remainder terms in the Riemann-Siegel formula in the region $\mathrm{Re}(s) > 1/2$, similar to those that have been obtained for $\mathrm{Re}(s) = 1/2$ [7]. It would also be necessary to investigate the various possible choices of the function $c(u)$, balancing the need for ease of computing $c(u)$ and $F(s)$ against the requirement that $F(s)$ decrease rapidly as $|\mathrm{Im}(s)| \to \infty$. (See [13] for some possible functions to use.) In practice, the integral in (2.5) would probably be estimated by deforming the contour of integration to go partially through the critical strip, which would create additional computational difficulties. Instead of first estimating $\pi^*(x)$ and then computing $\pi(x)$, it might be advantageous to estimate $\pi(x)$ itself, which involves using a formula somewhat different from (2.2). It would also be necessary to investigate various quadrature rules for numerical integration to determine which is best. Finally, it would be necessary to implement the algorithm of [17]. In summary, implementing an analytic $\pi(x)$ algorithm that is competitive with the Meissel-Lehmer method *in practice* would require substantial additional analysis.

The basic ideas underlying analytic $\pi(x)$-algorithms can be used for computing other arithmetical functions, such as

$$M(x) = \sum_{n \le x} \mu(n) ,$$

where $\mu(n)$ is the Möbius μ-function, or $\pi(x;q,a)$, the number of primes $p \le x$, $p \equiv a \pmod q$.

Generally speaking, the analytic $\pi(x)$-algorithm can be adapted to compute

$$G(x) = \sum_{n \le x} g(n) \ ,$$

where $g(n)$ is an integer-valued function (so that computing $G(x)$ exactly makes sense) such that the Dirichlet series

$$\sum_{n=1}^{\infty} \frac{g(n)}{n^s}$$

is easy to compute for large $\mathrm{Re}(s)$ (in the case of $M(x)$, this series equals $\zeta(s)^{-1}$, whereas for $\pi(x;q,a)$ it equals a combination of logarithms of Dirichlet L-functions $L(s,\chi)$, which can be computed by a generalized Riemann-Siegel type formula [4] and the method of [17]), and where the values of $g(n)$ for $x-y < n < x$ can be computed fast. The exact time and storage requirements depend on the method of computing $G(x)$ and the $g(n)$ for $x-y < n < x$.

3. The Algorithm

Given any $b \in [0,1/4]$ and say positive δ, we describe an algorithm for computing $\pi(x)$, which we call Algorithm $A(b,\delta)$, and show that it computes $\pi(x)$ in $O(x^{(3-2b)/5+\delta})$ elementary operations on numbers of $O(\log x)$ bits using $O(x^{b+\delta})$ space as $x \to \infty$. (The constants implied by the O-notation depend on δ). The verification of the correctness of the algorithm and of the bounds for the storage and running time depend on a number of lemmas, which are stated in this section and proved in Section 4. At the end of the section we indicate how the algorithms $A(b,\delta)$ may be combined to give a single algorithm, Algorithm $A(b)$, which computes $\pi(x)$ in time $O(x^{(3-2b)/5+\varepsilon})$ and space $O(x^{b+\varepsilon})$ as $x \to \infty$ for any $\varepsilon > 0$.

Algorithm $A(b,\delta)$ is given run-time exponents b and δ and a test value x as input, and produces $\pi(x)$ as output. The run-time exponent δ is used to compute a parameter $k = [200\ \delta^{-1}] + 1$, which is held fixed in the rest of the algorithm. In particular the kernel functions $F_{x,y}(s)$ and $g(w)$ and the auxiliary function $h_{a,x,y}(t)$ defined below depend on the parameter k.

We first describe the kernel function $F(s)$ that we use. We start with the polynomial

$$f(v) = f_k(v) = c_k\, v^k (1-v)^k \ , \tag{3.1}$$

where c_k is a normalization constant chosen to make

$$\int_0^1 f_k(v)\ dv = 1 \ . \tag{3.2}$$

(Note that c_k is rational.) We define the parametrized family $F_{x,y}(s)$ of kernel functions by

$$F_{x,y}(s) = s^{-1} \int_0^1 (x-vy)^s f_k(v)\, dv \ . \tag{3.3}$$

The integral defines $F_{x,y}(s)$ as an analytic function of s for $\sigma = \mathrm{Re}(s) > 0$, provided $0 \le y \le x$. We will use $F_{x,y}(s)$ and its inverse Mellin transform after making an appropriate choice of the parameter y, which turns out to be $y = x^{(3-2b)/(5-2b)}$. Our first lemma computes this Mellin transform pair explicitly and bounds the growth of $F_{x,y}(s)$.

Lemma 1. (1) We have

$$F_{x,y}(s) = [y^k s(s+1)\ldots(s+k)]^{-1} \sum_{r=1}^{k+1} \frac{a_r x^{s+k+r} - b_r (x-y)^{s+k+r}}{y^r (s+k+1)\ldots(s+k+r)} \ , \tag{3.4}$$

where $\quad a_r = f_k^{(k+r-1)}(0) \ , \quad b_r = f_k^{(k+r-1)}(1) = (-1)^{k+r-1} a_r \ .$

(2) *For* $\sigma = \mathrm{Re}(s) > 0$ and $|s| \ge 1$,

$$|F_{x,y}(s)| \le (2k)^k x^{\sigma+k} y^{-k} |s|^{-k-1} \ . \tag{3.5}$$

(3) For $u > 0$ *and* $a > 0$

$$\frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty} F_{x,y}(s) u^{-s} ds = g_k\left[\frac{x-u}{y}\right] \ , \tag{3.6}$$

where

$$g_k(w) \;=\; \begin{cases} 0 & w \le 0 \;, \\ \int_0^w f_k(v)\,dv & 0 < w < 1 \;, \\ 1 & w \ge 1 \;\; . \end{cases} \tag{3.7}$$

Our object is to compute $\pi(x)$ by evaluating the contour integral

$$P(x,y) \;=\; \frac{1}{2\pi i} \int_{a-i\infty}^{a+i\infty} F_{x,y}(s) \;\log \zeta(s) \quad ds \tag{3.8}$$

in two different ways, to within an error of $\pm \frac{1}{10}$. We choose the vertical line of integration to be $a \;=\; \frac{3}{2}$.

First we evaluate $P(x,y)$ in terms of $\pi(x)$. Using the inverse Mellin transform, we have by (2.3) and (3.6) that

$$P(x,y) \;=\; \pi(x) \;+\; \sum_{m=2}^{\infty} \frac{1}{m} \sum_{\substack{p \\ p^m \le x}} 1 \;-\; \sum_{m=1}^{\infty} \;\frac{1}{m} \sum_{\substack{p \\ x-y<p^m<x}} \left[1 \;-\; g\left[\frac{x-p^m}{y} \right] \right] \; . \tag{3.9}$$

The difference between $P(x,y)$ and $\pi(x)$ can be computed quickly, as is shown in the next lemma.

Lemma 2. A machine with $O(x^{\varepsilon})$ bits of storage can compute

$$\sum_{m=2}^{\infty} \frac{1}{m} \sum_{\substack{p \\ p^m \le x}} 1 \tag{3.10}$$

to within $\pm 10^{-3}$ in $O(x^{1/2+\varepsilon/10})$ operations for $x \ge x_0(\varepsilon)$. Such a machine can compute

$$\sum_{m=1}^{\infty} \;\frac{1}{m} \sum_{\substack{p \\ x-y<p^m<x}} \left[1 \;-\; g\left[\frac{x-p^m}{y} \right] \right] \tag{3.11}$$

to within $\pm 10^{-3}$ in $O(yx^{\varepsilon/10})$ operations for $x \ge x_0(\varepsilon)$.

Second we evaluate $P(x,y)$ in (3.8) directly to within $\pm 10^{-1}$. By the bound (3.5) of Lemma 1, for $T \geq 10$, say,

$$\left| \frac{1}{2\pi i} \int_{3/2+iT}^{3/2+i\infty} F_{x,y}(s) \log \zeta(s)\, ds \right| = O(x^{3/2+k} y^{-k} T^{-k}) \ , \tag{3.12}$$

so if we choose

$$T = y^{-1} x^{1+\delta/10} \ , \tag{3.13}$$

then the quantity on the right side of (3.12) will be $O(x^{-8})$, since $k \geq 200\, \delta^{-1}$. Therefore, for the above choice of T, we will have

$$\left| P(x,y) - \frac{1}{2\pi i} \int_{3/2-iT}^{3/2+iT} F_{x,y}(s) \log \zeta(s)\, ds \right| = O(x^{-8}) \ . \tag{3.14}$$

We have thus reduced the problem of computing $P(x,y)$ to that of estimating a finite integral. In our case we have $\overline{F_{x,y}} = F_{x,y}(\bar{s})$ and $\overline{\zeta(s)} = \zeta(\bar{s})$, so the finite integral is

$$\frac{1}{2\pi i} \int_{3/2-iT}^{3/2+iT} F_{x,y}(s) \log \zeta(s)\, ds = \frac{1}{\pi} \int_0^T h_{3/2}(t)\, dt \ , \tag{3.15}$$

where

$$h_a(t) = h_{a,x,y}(t) = \mathrm{Re}\, \{ F_{x,y}(a+it) \log \zeta(a+it) \} \ . \tag{3.16}$$

We evaluate this last integral numerically using the Euler-Maclaurin summation formula ([8], Eqn. 25.4.7).

Lemma 3. Let $g(t)$ be complex-valued, with its first $2m$ derivatives continuous on [0,T]. Then for any positive integer n we have

$$\int_0^T g(t)\,dt \;=\; \frac{T}{n}\left[\sum_{j=0}^{n} g(\frac{jT}{n}) \;-\; \frac{g(0)\;+\;g(T)}{2}\right]$$

$$-\sum_{i=1}^{m-1} \frac{B_{2i}}{(2i)!}\;(\frac{T}{n})^{2i-1}[g^{(2i-1)}(T)\;-\;g^{(2i-1)}(0)]\;+\;R_{2m}\;, \qquad (3.17)$$

where the B_{2i} are the Bernoulli numbers, and the remainder R_{2m} is bounded by

$$|R_{2m}| \le n\;\frac{|B_{2m}|}{(2m)!}\;(\frac{T}{n})^{2m+1}\max_{0\le t\le T}|g^{(2m)}(t)|\;. \qquad (3.18)$$

We will apply Lemma 3 for $g(t) \;=\; h_a(t)$ with $a \;=\; 3/2$. To estimate the remainder term, we observe that $h_{a,x,y}(t)\;\varepsilon\;C^\infty(-\infty,\infty)$ and its derivatives are easily bounded, as follows.

Lemma 4. For integers $m \ge 1$,

$$\left|\frac{d^m}{dt^m}\;h_{a,x,y}(t)\right| \;=\; O\left[m!(\frac{a-1}{2})^{-m}\;x^{(a+1)/2\;+\;k}\right]\;, \qquad (3.19)$$

and the constant implied by the O-symbol depends on k and a.

Now we apply Lemma 3, choosing $a \;=\; 3/2$ with number of steps

$$n \;=\; \left[T\,x^{\delta/10}\right] \;=\; \left[y^{-1}\;x^{1\;+\;\delta/5}\right] \qquad (3.20)$$

and number of derivatives $2m \;=\; 2k^2$. Using Lemma 4 to estimate the remainder term gives

$$|R_{2k^2}| \;=\; O(x^{-5})\;, \qquad (3.21)$$

where the implied constant depends on $k \;=\; [200\;\delta^{-1}]\;+\;1$.

It remains to evaluate the main term in the Euler-Maclaurin summation formula (3.17). Since $F(3/2+it)$ and all its derivatives up to some fixed order can be computed to within $O(x^{-10})$ in $O(x^{\delta/10})$ operations, and they are all $O(x^{3/2})$ in size, it suffices to show that we can compute $\log\,\zeta(s)$ and its derivatives at $s \;=\; 3/2+it$ to within $O(x^{-10})$ efficiently. Since

$b^{-1} \leq |\zeta(3/2+it)| \leq b$ for some fixed $b > 0$, it suffices to show that we can compute $\zeta(3/2+it)$ and its derivatives up to some fixed order efficiently. The Euler-Maclaurin summation formula gives the following result for evaluating $\zeta(s)$ and its derivatives.

Lemma 5. For any positive integer m and any $\varepsilon > 0$ *and for* $1 < a < 2$, *there is an algorithm to compute*

$$\frac{\partial^j}{\partial t^j} \zeta(a+it)$$

for any t with $0 \leq t \leq x$ *and any j with* $0 \leq j \leq 2m$ *with an error of* $O(x^{-10})$ *in* $O(tx^{\varepsilon/10})$ *operations using* $O(x^{\varepsilon})$ *space, for* $x \geq x_0(\varepsilon)$. *The constant in the O-symbol depends on m,* ε, *and a.*

The Riemann-Siegel formula can be used to compute values of $\zeta(s)$ even faster. It is difficult to obtain good bounds for the error in that formula for small *t*, but one easily obtains a result valid for $t \geq x^{1/10}$.

Lemma 6. For $x^{1/10} \leq t \leq x$ *and* $1 \leq a \leq 2$, *the Riemann-Siegel formula can be used to compute* $\zeta(a+it)$ *with an error of* $O(x^{-10})$ *in* $O(t^{1/2} x^{\varepsilon/10})$ *operations using* $O(x^{\varepsilon})$ *space, for* $x \geq x_0\ (\varepsilon)$.

This bound already suffices to give the desired Algorithm $A(b,\delta)$ in the case $b = 0$. For the case of general b in $0 < b \leq \frac{1}{4}$ we need a yet faster method to compute many values of $\zeta(s)$, and to this end we use the method of [17].

Lemma 7. If $0 \leq \beta \leq 1/2$, $1 \leq a \leq 2$, *and* $\varepsilon > 0$ *are given, then there is a constant c and an algorithm such that for every* $x \geq 1$ *and* $T \geq x^{1/10}$ *it will perform* $\leq cT^{1/2+\varepsilon}$ *operations using* $\leq cT^{\beta+\varepsilon}$ *bits of storage and will then be capable of computing any value* $\zeta(a+it)$, $T \leq t \leq T + T^{\beta}$ *to within* $\pm\ x^{-10}$ *in* $\leq cT^{\varepsilon}$ *operations using the precomputed values.*

We use the algorithm of Lemma 5 to compute $h_{3/2}(\frac{jT}{n})$ for $\frac{jT}{n} \le x^{1/10}$ and also to compute $h^{(j)}(0)$ and $h^{(j)}(T)$ for $1 \le j \le 2k^2$. We then use either the algorithm of Lemma 7 to or that of Lemma 6 in the case $b = 0$ to compute $h(\frac{jT}{n})$ for $x^{1/10} \le jT/n \le T$. By Lemma 3 and (3.21), this gives us $\int_0^T h_{3/2}(t)\,dt$ to within an error of $O(x^{-8})$ using $O(T^{\beta+\varepsilon})$ space in the computation and using

$$O(x^{1/5+\delta/10} + T^{3/2 - \beta + \delta/10})$$

operations. Combining this with the previous estimates shows that we can compute $\pi(x)$ to within $\pm 10^{-1}$ in

$$O(\{x^{1/2} + y + x^{1/5} + T^{3/2 - \beta}\}x^{\delta/10})$$

operations and $O(T^{\beta+\delta})$ space. By (3.13), this yields a running time for Algorithm $A(b,\delta)$ of $O(x^{(3 - 2\beta)/(5 - 2\beta) + 2\delta/5)}$ and space requirement of $O(x^{2\beta/(5 - 2\beta) + \delta})$ if we take $y = x^{(3 - 2\beta)/(5 - 2\beta)}$, where the constant implied by the O-symbol depends on the given δ. If we now choose $\beta = 5b/(2(1+b))$, we obtain the desired estimate.

Finally, we observe that these algorithms may be combined to obtain a single Algorithm $A(b)$ which computes $\pi(x)$ in $O(x^{(3-2b)/5+\varepsilon})$ steps using $O(x^{b+\varepsilon})$ space for any $\varepsilon > 0$. When given x, Algorithm $A(b)$ computes a value $\delta = \delta(x)$ depending on x and then runs Algorithm $A(b,\delta)$. If the value $\delta(x)$ goes to zero sufficiently slowly as $x \to \infty$, Algorithm $A(b)$ will have the required running time property. Note that Algorithm $A(b)$ must compute various quantities such as $f_k(r)$ which are treated as fixed overhead in Algorithm $A(b,\delta)$. An analysis which we omit of the dependence on δ of the fixed overhead and the running times of various steps in Algorithm $A(b,\delta)$ indicates that choosing $\delta(x) = (\log\log x)^{-1}$ is adequate to obtain the stated running time bound for Algorithm $A(b)$.

4. Proofs of Lemmas

Proof of Lemma 1. Integration by parts shows that

$$F_{x,y}(s) = \frac{1}{y^k s(s+1)\ldots(s+k)} \int_0^1 (x - vy)^{s+k} f_k^{(k)}(v)\, dv \ , \tag{4.1}$$

since $f_k^{(r)}(1) = f_k^{(r)}(0) = 0$ for $0 \leq r \leq k-1$. Since

$$| f_k^{(k)}(r) | \leq (2k)^k \ , \quad 0 \leq r \leq 1 \ .$$

this implies (3.5). Another k integrations by parts of (4.1) yield (3.4).

To prove (3), we use (3.3) and an easily justified exchange of the order of integration to obtain for $a > 1$ that

$$\frac{1}{2\pi i} \int_{a-i\infty}^{a+i\infty} F_{x,y}(s)\ u^{-s}\ ds = \int_0^1 \left[\frac{1}{2\ \pi\ i} \int_{a-i\infty}^{a+i\infty} \frac{1}{s}\ \left(\frac{x-vy}{u}\right)^s ds \right] f_k(v)\, dv \ .$$

We obtain the desired result by using (2.4) on the inner integrand. □

Proof of Lemma 2. We make use of the recent algorithm of Adleman, Pomerance, and Rumely [1], which tests whether an integer n is prime in

$$O((\log\ n)^{c \log \log \log n})$$

operations (and also storage locations) on a random access machine for some $c > 0$. (This algorithm can also be implemented on a Turing machine.)

Any prime power p^m that contributes to the sum (3.10) must have $p \leq \sqrt{x}$. Therefore we initially set $S = 0$, and proceed to test each $n \leq \sqrt{x}$ for primality in $O(x^{\varepsilon/100})$ bit operations. If n is composite, we go on to test $n + 1$. If n is prime, we compute successively $n, n^2, \ldots, n^M$, where M is the smallest integer such that $n^M > x$ (note that $M \leq 1 + \log_2 x$), and then increment S by the floating point approximations to $1/2, 1/3, \ldots, 1/(M-1)$, each accurate to $\pm x^{-1}$. Thus all arithmetic will be performed on $O(\log\ x)$-bit binary numbers. At the end, S will differ from the sum (3.10) by at most $O(x^{-1/2})$, since S is obtained by adding up $O(x^{1/2})$ terms, each of which differs from the desired amount by a most x^{-1}. This proves the first part of the lemma.

To prove the second part of the lemma, we determine, for each $n \in [x-y,x]$, the largest m such that $n = p^m$ for some $p \in Z^+$. There are few values of m to test, since $m \le \log_2 x$, and for each m we can check whether n is an m-th power or not by computing $t = n^{1/m}$ using $O(\log\ x)$-bit floating point approximations, and then testing whether $[t-1]^m$, $[t]^m$, or $[t+1]^m$ equals n. Thus this part takes $O(x^{\varepsilon/100})$ operations. Once p is determined, we test it for primality. If it is not prime, we go on to $n+1$. If p is prime, we compute

$$\frac{1}{m}\left[1-g\left[\frac{x-p^m}{y}\right]\right]$$

to within $\pm x^{-2}$ using $O(\log\ x)$-bit floating point approximations, where we use the fact that $g(w)$ is a polynomial of degree $2k+1$, and thus its values are easy to compute, so that each computation takes $O(x^{\varepsilon/100})$ steps. Since there are $\le y+1$ terms, this completes the proof of the lemma. □

Proof of Lemma 4. Let $a > 1$, and recall that

$$h_{a,x,y}(t) = \mathrm{Re}\ \{H(t)\}\ ,$$

where

$$H(t) = H_{a,x,y}(t) = F_{x,y}(a+it)\ \log\ \zeta(a+it)$$

is an analytic function of t for $|\mathrm{Im}(t)| < a-1$. Now by Cauchy’s formula, for real t we have

$$\frac{d^m}{dt^m}\ H(t) = \frac{m!}{2\pi i} \int_{|z-t|=\frac{a-1}{2}} \frac{H(z)\,dz}{(z-t)^{m+1}}\ .$$

Hence

$$\left| \frac{d^m}{dt^m} H(t) \right| \le m! \left[\frac{a-1}{2} \right]^{-m} \left[\max_{|s-(a+it)| = \frac{a-1}{2}} |F_{x,y}(s)| \right] \left[\max_{|s-(a+it)| = \frac{a-1}{2}} |\log \zeta(s)| \right] \tag{4.2}$$

Now

$$|\log \zeta(s)| = O(1)$$

for $\mathrm{Re}(s) \ge a-(a-1)/2 = (a+1)/2$ and by Lemma 1 we have

$$\max_{|s-(a+it)| \le \frac{a-1}{2}} |F_{x,y}(s)| \le (2k)^k x^{(a+1)/2 + k} y^{-k} \left| \frac{a+1}{2} \right|^{-k-1} .$$

Combining these inequalities with (4.2) proves the lemma. □

Proof of Lemma 5. This follows from the Euler-Maclaurin formula as applied to $\zeta(a+it)$ and its derivatives. We present in detail only the analysis for $\zeta(s)$. It is well known [6, p. 114], [21] that for any positive integers q and n and for $s = \sigma + it$ with $\sigma > 1$ we have

$$\zeta(s) = \sum_{n=1}^{q-1} n^{-s} + \frac{1}{2} q^{-s} + \frac{q^{1-s}}{s-1} + \sum_{r=1}^{n} \frac{B_{2r}}{(2r)!} q^{1-s-2r} \prod_{j=0}^{2r-2} (s+j) + R_{2n}^*(q,s) \tag{4.3}$$

where

$$R_{2n}^*(q,s) = - \frac{s(s+1)...(s+2n)}{(2n+1)!} \int_q^\infty \overline{B}_{2n+1}(t) t^{-s-2n} dt , \tag{4.4}$$

and $\overline{B}_{2n}(t)$ is a periodic function with period 1, which equals the Bernoulli polynomial of order $2n$ for $0 \le t \le 1$. We have the estimate [6, p. 115]

$$|R_{2n}^*(q,s)| = O\left[\frac{|B_{2n+2}|}{(2n+2)!} q^{-1-\sigma-2n} \frac{|s+2n+1|}{\sigma+2n+1} \prod_{j=0}^{2n} |s+j| \right] ,$$

where the constant implied by the O-notation is independent of n. Since

$$|B_{2r}| = \frac{2(2r)!}{(2\pi)^{2r}} \zeta(2r) ,$$

the O-term above is, for $1 < \sigma \le 2$,

$$O\left[\frac{|s+2n+2|^{2n+2}}{q^{2+2n}} \right] .$$

We choose $n = [200\,\varepsilon^{-1}] + 1$ and $q \sim tx^{\varepsilon/20}$, and this term becomes $O(2^{2n+2}(2n+4)^{2n+2}x^{-10})$. The time needed to compute the remaining terms is proportional to q multiplied by the time needed to compute each term to an accuracy of $O(x^{-11})$, which is $O(x^{\varepsilon/20})$. The only term whose computation time depends on a is $\dfrac{q^{1-s}}{s-1}$, and this only for s very near 1. A similar analysis can be given for computing $\dfrac{\partial^j}{\partial s^j}\ \zeta(s)$ from the formulae (4.3) and (4.4) differentiated j times, for $1 \le j \le m$. $\square$

Proof of Lemma 6. We use the Riemann-Siegel formula [21]. (This formula is also derived in [6,22], but for a restricted range of values.) If $s = \sigma + it$, $1 < \sigma < 2$, $m = \left\lfloor (2\pi)^{-1/2} t^{1/2} \right\rfloor$, $N \in Z^+$, and $t \ge c\ N$ for a certain fixed $c > 0$, then

$$\zeta(s) = \sum_{n=1}^{m} n^{-s} + \chi(x) \sum_{n=1}^{m} n^{s-1} \tag{4.5}$$

$$+ (-1)^{m-1} (2\pi t)^{(s-1)/2} \exp(-it/2 - i\pi/8 - i\pi(s-1)/2)\,\Gamma(1-s)\ \{S_N + O(e^{-\delta' t}) + O((c'Nt^{-1})^{N/6})\} ,$$

where δ' , $c' > 0$ are fixed constants,

$$\chi(s) = 2^s\ \pi^{s-1} \sin(\frac{\pi s}{2})\,\Gamma(1-s) ,$$

and

$$S_N = S_N(s) = \sum_{n=0}^{N-1} \sum_{r \le n/2} \frac{n!\, i^{r-n}}{r!(n-2r)!2^n}\ (\frac{2}{\pi})^{n/2-r}\ a_n(s)\Psi^{(n-2r)}\ (2\{t^{1/2}(2\pi)^{-1/2}\}) ,$$

with

$$\Psi(u) \;=\; \frac{\cos\pi(\frac{1}{2}u^2 - u - \frac{1}{8})}{\cos\,\pi u} \;\;, \tag{4.6}$$

and the coefficients $a_n\,(s)$ are given by the recurrence

$$a_n\,(s) \;=\; 1,\; a_1\,(s) \;=\; \frac{\sigma - 1}{\sqrt{t}}\;,\;\; a_2\,(s) \;=\; \frac{(\sigma-1)\,(\sigma-2)}{2t}\;\;,$$

$$(n+1)\,\sqrt{t}\;a_{n+1}\,(s) \;=\; (\sigma - n - 1)\,a_n\,(s) \;+\; i\;a_{n-2}\,(s) \quad \text{for } n \geq 2\;, \tag{4.7}$$

and satisfy the bound

$$|a_n\,(s)| \;=\; O(t^{-n/6})\;.$$

(In the definition of S_N, $\{u\}$ denotes the fractional part of u.)

We take $N \;=\; 600$, say. Since [8, Eq. (6.1.45)]

$$\lim_{|y| \to \infty} \;(2\pi)^{-1/2}\,|\Gamma(x+iy)|\,e^{1/2\pi|y|}\;\;|y|^{1/2-x} \;=\; 1$$

we have

$$|(2\pi t)^{(s-1)/2}\;\exp(-it/2 - i\pi/8 - i\pi(s-1)/2)\,\Gamma(1-s)| \;=\; (2\pi t)^{(\sigma-1)/2}\,e^{\pi t/2}\,|\Gamma(1-s)|$$

$$\sim (2\pi)^{\sigma/2}\,t^{-\sigma/2} \quad \text{as} \quad t \to \infty$$

Hence when $|t| \geq x^{1/10}$ and $1 \leq \sigma < 2$, the formula (4.5) gives

$$\zeta(s) \;=\; \sum_{n=1}^{m} n^{-s} \;+\; \chi(x)\;\sum_{n=1}^{m} n^{s-1} \tag{4.8}$$

$$+\;(-1)^{m-1}\;(2\pi t)^{(s-1)/2}\;\exp\left[-\;\frac{it}{2}\;-\;\frac{i\pi}{8}\;-\;\frac{i\pi(s-1)}{2}\right]\Gamma(1-s)\,S_N\,(s) \;+\; O(x^{-10})\;.$$

We first observe that for any fixed N and $x^{1/10} \leq t \leq x$, the term containing $S_N\,(s)$ can be evaluated to accuracy $O(x^{-10})$ in $O(x^{\varepsilon})$ operations. To see this, we first note that by Stirling's formula [8; Eq. 6.1.42],

$$\log\,\Gamma(1-s) \;=\; (1/2-s)\,\log\,(1-s) \;-\; (1-s) \;+\; \frac{1}{2}\log(2\pi)$$

$$+\;\sum_{h=1}^{M}\;\frac{B_{2h}}{2h\,(2h-1)}\;(1-s)^{1-2h} \;+\; O(t^{-1-2M})\;, \qquad (4.9)$$

where we choose $M\;=\;20$ and the constant implied by the O-notation is (as will be the case for all other such constants) dependent only on σ and δ. Hence we can compute $\log\,\Gamma(1-s)$ using (4.9) to an accuracy of $O(x^{-20})$ in $O(x^{\varepsilon})$ operations. Similarly, we can compute

$$w \;=\; -i\pi(s-1)/2 \;-\; it/2 \;-\; i\pi/8$$

to within $O(x^{-20})$ in $O(x^{\varepsilon})$ operations. Since [8; Eq. 6.1.45]

$$|\Gamma(1-s)| \;=\; O(t^{1/2-\sigma}\;e^{-\pi t/2})$$

(a result that follows from (4.8) also), we have

$$|(2\pi t)^{(s-1)/2}\;e^{w}\;\Gamma(1-s)| \;=\; O(t^{-\sigma/2})\;,$$

and we can compute it to an accuracy of $O(x^{-20})$ in $O(x^{\varepsilon})$ operations.

Next we consider $S_N\,(s)$, where we selected $N\;=\;600$. Note that $\Psi(u)$ is actually an entire function, and its Taylor series expansion around $u\;=\;1/2$ can be used to evaluate $\Psi(u)$ and its first N derivatives in a neighborhood of $u\;=\;1/2$ (and similarly in a neighborhood of $u\;=\;-1/2$), while away from $u\;=\;\pm\,1/2$, we can differentiate the expansion (4.7) and evaluate individual terms. Each of the first N derivatives takes $O(x^{\varepsilon})$ operations to compute to within $O(x^{-20})$. Finally the $a_n\,(s)$ can be computed using the recurrence (4.7) to within $O(x^{-20})$ in $O(x^{\varepsilon})$ operations. Combining all these estimates, we conclude that the entire term in (4.8) containing $S_N\,(s)$ can be computed to within $O(x^{-20})$ in $O(x^{\varepsilon})$ operations, using $O(x^{\varepsilon})$ storage.

Stirling’s formula (4.9) allows us to compute $\chi(s)$ to accuracy $O(x^{-12})$ in a similar way. For $x^{1/10}\;\le\;t\;\le\;x$ the two sums in (4.8) are evaluated to accuracy $O(x^{-10})$ by evaluation term-by-term in time $O(mx^{\varepsilon})\;=\;O(t^{1/2+\varepsilon})$. Lemma 6 follows. □

REFERENCES

[1] L. M. Adleman, C. Pomerance, and R. S. Rumely, On distinguishing prime numbers from composite numbers, *Annals Math.* 117 (1983), 173-206.

[2] J. Bohman, On the number of primes less than a given limit, *BIT 12* (1972), 576-577.

[3] H. Davenport, *Multiplicative Number Theory*, 2nd ed. (revised by H. L. Montgomery), Springer-Verlag, 1980.

[4] M. Deuring, Asymptotische Entwicklungen der Dirichletschen L-Reihen, *Math. Annalen 168* (1967), 1-30.

[5] L. E. Dickson, *History of the Theory of Numbers*, Chelsea reprint. Vol. 1, Chapter XVIII.

[6] H. M. Edwards, *Riemann's Zeta Function*, Academic Press, 1974.

[7] W. Gabcke, Neue Herleitung und explizite Restabschätzung der Riemann-Siegel-Formel, Ph.D. Dissertation, Göttingen 1979.

[8] *Handbook of Mathematical Functions*, M. Abramowitz and I. A. Stegun, eds., National Bureau of Standards, 9th printing, 1970.

[9] G. H. Hardy, *Ramanujan*, Cambridge Univ. Press, 1940. (Reprinted by Chelsea.)

[10] J. C. Lagarias, V. S. Miller, and A. M. Odlyzko, Computing $\pi(x)$: the Meissel-Lehmer method, *Math. Comp.*, *44* (1985), 537-560.

[11] J. C. Lagarias and A. M. Odlyzko, New algorithms for computing $\pi(x)$, pp. 176-193 in *Number Theory: New York 1982*, D. V. Chudnovsky, G. V. Chudnovsky, H. Cohn, and M. B. Nathanson, eds., Lecture Notes in Mathematics #1052, Springer-Verlag, 1984.

[12] D. H. Lehmer, On the exact number of primes less than a given limit, *Illinois J. Math. 3* (1959), 381-388.

[13] B. F. Logan, Note on ideal sharp-cutoff filters, preprint.

[14] D. C. Mapes, Fast method of computing the number of primes less than a given limit, *Math. Comp. 17* (1963), 179-185.

[15] E. D. F. Meissel, Über die Bestimmung der Primzahlmenge innerhalb gegebener Grenzen, *Math. Annalen, 2* (1870), 636-642.

[16] E. D. F. Meissel, Berechrung der Menge der Primzahlen, welche innerhalb der ersten Milliarde näturlichen Zahlen vorkommen, *Math. Annalen, 25* (1885), 251-257.

[17] A. M. Odlyzko and A. Schönhage, Fast algorithms for multiple evaluations of the Riemann zeta function, to be published.

[18] P. Pritchard, A sublinear additive sieve for finding prime numbers, *Comm. ACM 24* (1981), 18-23.

[19] S. Ramanujan, *Notebooks*, Tata Institute, 1957, 3rd notebook, p. 371.

[20] H. Riesel and G. Göhl, Some calculations related to Riemann's prime number formula, *Math. Comp. 24* (1970), 969-983.

[21] C. L. Siegel, Über Riemanns Nachlass zur analytischen Zahlentheorie, *Quellen und Studien zur Geschichte der Math. Astr. Phys.* 2 (1932), 45-80. Reprinted in C. L. Siegel, *Gesammelte Abhandlungen*, Spring-Verlag, 1966, Vol. 1, pp. 275-310.

[22] E. C. Titchmarsh, *The Theory of the Riemann Zeta-Function*, Oxford Univ. Press, 1951.

[23] H. S. Wilf, What is an answer? *Am. Math. Monthly, 89* (1982), 289-292.

Computing π(x): An Analytic Method

J. C. Lagarias
A. M. Odlyzko

AT&T Labs - Research
Murray Hill, NJ 07974

ABSTRACT

This paper presents a class of algorithms for computing $\pi(x)$, the number of primes $\leq x$. For each b with $0 \leq b \leq 1/4$ and each $\varepsilon > 0$ there is an algorithm $A(b, \varepsilon)$ which computes $\pi(x)$ using $O(x^{3/5 - 2b/5 + \varepsilon})$ bit operations and uses $O(x^{b + \varepsilon})$ bits of storage. In particular one can compute $\pi(x)$ using $O(x^{3/5 + \varepsilon})$ bit operations and $O(x^{\varepsilon})$ space, or using $O(x^{1/2 + \varepsilon})$ bit operations and $O(x^{1/4 + \varepsilon})$ space. These algorithms are based on numerical integration of certain integral transforms of the Riemann ζ-function. This technique can be generalized to evaluate many other arithmetic functions, including the functions $\pi(x; k, \ell)$ counting the number of primes $p \equiv \ell \pmod{k}$ with $p \leq x$ and the function $M(x)$ which is the partial sum of the Möbius function $\mu(n)$ for all $n \leq x$.