An Algorithm to Generate Random Factored Smooth Integers

Eric Bach, University of Wisconsin-Madison (bach@cs.wisc.edu)
Jonathan Sorenson, Butler University (jsorenso@butler.edu)

ANTS XIV poster session
Arxiv.org link: https://arxiv.org/abs/2006.07445

Results

We say an integer n is *y-smooth* if every prime divisor of n is $\leq y$. Let $\Psi(x, y)$ count the number of y-smooth integers $\leq x$.

We present a new algorithm that does the following:

Inputs:

- Integers x,y, with $x \geq y > 0$, and
- A real number $r \in [0, 1)$ (supposedly chosen uniformly at random).

Outputs:

- Integer n, with $0 < n \leq x$,
- A list of primes $p_1, p_2, \ldots$ such that $n = p_1 p_2 \cdots$, with $p_i \leq y$ for all i, and
- n is at position $r\Psi(x, y)(1 + o(1))$ in the enumeration of y-smooth integers $\leq x$, lexicographically ordered by prime divisors. In other words, n is chosen asymptotically uniformly.

Our algorithm takes

$$O\left(\frac{(\log x)^3}{\log\log x}\right)$$

arithmetic operations on average. This running time analysis uses the following:

- We assume the ERH to extend the range of applicability to the estimate $x\rho(u)$ for $\Psi(x, y)$, where ρ is the Dickman-DeBruijn function [12], and
- We use the Miller-Rabin probabilistic primality test, so that our list of primes all are in fact prime with probability $1 - o(1)$ [9, 10].

In the draft of our full paper, we show how to derive running times with differing sets of assumptions and conditions. Other special cases we discuss there include

- Setting $y = x$ to get an alternative to Bach's algorithm [2],
- Looking and what gets easier if all primes $\leq y$ are available, and
- Generating random *semismooth* integers with known prime factorization.

Buchstab's Identity

$$\Psi(x, y) = 1 + \sum_{p \leq y} \Psi(x/p, p), \tag{1}$$

which decomposes $\Psi(x, y)$ by its largest prime divisor [12, §5.3].

We can use this equation to see that, for a prime $p \leq y$, the number of y-smooth integers with p as their largest prime divisor is $\Psi(x/p, p)$. In other words, when generating n, we choose p as n's largest prime divisor with probability $\Psi(x/p, p)/\Psi(x, y)$. Then $n = 1$ with probability $1/\Psi(x, y)$.

Algorithm

This leads us to the following algorithm.
Inputs: x, y, r

1. If $r < 1/\Psi(x,y)$, output 1 and halt.

2. Find real t such that $\Psi(x,t) - r\Psi(x,y)$ is near zero.

3. Find consecutive primes $p_1 < p_2$, near t, such that $\Psi(x,p_1) < r\Psi(x,y) \le \Psi(x,p_2)$.

   Note that, from Buchstab's identity above, this gives us
$$1 + \sum_{p \le p_1} \Psi(x/p, p) < r\Psi(x,y) \le 1 + \sum_{p \le p_2} \Psi(x/p, p),$$
   which tells us that p_2 is our largest prime divisor.

4. Output p_2.

5. Set $r' = \frac{r\Psi(x,y) - \Psi(x,p_1)}{\Psi(x/p_2, p_2)}$.

6. Recurse on x/p_2, p_2, and r'.

Algorithm Details

- We estimate Ψ with either the $x\rho(u)$ estimate mentioned above [13], if y is not too small, or with a saddle-point based method [8, 11] for smaller y. See Hildebrand [6] for the cutoff point.

- If we use the $x\rho(u)$ method, we can find t with Newton's method. Otherwise, the Illinois algorithm with some bisection steps [4], or Brent's algorithm [3] gives quick convergence.

- We can find p_1, p_2 by sieving a short interval and using strong pseudoprime tests [9, 10].

- Due to the work of Alladi [1], Hensley [5], and Hildebrand [7], we know the average number of prime divisors of n is
$$O\left(\log\log x + \frac{\log x}{\log y}\right),$$
  which is also the recursion depth of the algorithm.

- If we were to use an exact method to compute Ψ, then we would generate an n at exactly position $\lfloor r\Psi(x,y) \rfloor$ in the enumeration of y-smooth numbers $\le x$.

Example Run

We implemented our algorithm in C++ on a linux desktop workstation and ran it with $x = 10^{100}$, $y = 10000$, and $r = 0.5$. It generated the following list of prime divisors for n:

2 3 5 7 29 31 97 113 113 113 157 223 241 503 509 569 691 727 1033 1367 1571 2141 2339
2617 2741 3041 3221 3547 3989 4021 4513 4999 5573 6577 7573 9463

The resulting n is roughly $4.29 \cdot 10^{97}$, which occupies a position near $2.05 \cdot 10^{61}$ in the enumeration. The run took less than 0.35 seconds of wall time.

References

[1] Krishnaswami Alladi. An Erdős-Kac theorem for integers without large prime factors. *Acta Arith.*, 49(1):81–105, 1987.

[2] E. Bach. How to generate factored random numbers. *SIAM Journal on Computing*, 2:179–193, 1988.

[3] R. P. Brent. *Algorithms for Minimization Without Derivatives.* Dover, 2013. Originally published by Prentice-Hall 1973.

[4] M. Dowell and P. Jarratt. A modified regula falsi method for computing the root of an equation. *BIT*, 11:168–174, 1971.

[5] Douglas Hensley. The distribution of $\Omega(n)$ among numbers with no large prime factors. In *Analytic number theory and Diophantine problems (Stillwater, OK, 1984)*, volume 70 of *Progr. Math.*, pages 247–281. Birkhäuser Boston, Boston, MA, 1987.

[6] A. Hildebrand. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Journal of Number Theory*, 22:289–307, 1986.

[7] Adolf Hildebrand. On the number of prime factors of integers without large prime divisors. *J. Number Theory*, 25(1):81–106, 1987.

[8] Simon Hunter and Jonathan P. Sorenson. Approximating the number of integers free of large prime factors. *Mathematics of Computation*, 66(220):1729–1741, 1997.

[9] G. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13:300–317, 1976.

[10] M. O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128–138, 1980.

[11] Jonathan P. Sorenson. A fast algorithm for approximately counting smooth numbers. In W. Bosma, editor, *Proceedings of the Fourth International Algorithmic Number Theory Symposium (ANTS IV)*, pages 539–549, Leiden, The Netherlands, 2000. LNCS 1838.

[12] Gérald. Tenenbaum. *Introduction to Analytic and Probabilistic Number Theory*, volume 46 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, english edition, 1995.

[13] J. van de Lune and E. Wattel. On the numerical solution of a differential-difference equation arising in analytic number theory. *Mathematics of Computation*, 23:417–421, 1969.