How to Not Count Primes

An Analytic Prime Counting Function

D.J. Platt

Department of Mathematics
University of Bristol

28 February 2011 / TCC Number Theory Day

Outline

Sieve of Eratostenes

Enumerate and Count

- for(i=2;i<x;i++) prime[i]=true;

Sieve of Eratostenes

Enumerate and Count

- for(i=2;i<x;i++) prime[i]=true;
- for(i=2;i<sqrt(x);i++) if(prime[i]) for(j=i*i;j<x;j+=i) prime[i]=false;

Sieve of Eratostenes

Enumerate and Count

- for(i=2;i<x;i++) prime[i]=true;
- for(i=2;i<sqrt(x);i++) if(prime[i]) for(j=i*i;j<x;j+=i) prime[i]=false;
- $O(x \log(x) \log(\log(x)))$ in time

Sieve of Eratostenes

Enumerate and Count

- for(i=2;i<x;i++) prime[i]=true;
- for(i=2;i<sqrt(x);i++) if(prime[i]) for(j=i*i;j<x;j+=i) prime[i]=false;
- $O(x \log(x) \log(\log(x)))$ in time
- $O(x)$ in space

Other Sieves?

- "Better" sieves are known

Other Sieves?

- "Better" sieves are known
- BUT

Other Sieves?

- "Better" sieves are known
- BUT
- By the Prime Number Theorem....

Other Sieves?

- "Better" sieves are known
- BUT
- By the Prime Number Theorem....
- All sieves must be at least $O\left(\frac{x}{\log(x)}\right)$ in time

Combinatoric Methods

- $\pi(x) = x - 1 + \pi(\sqrt{x}) \ldots$

Combinatoric Methods

- $\pi(x) = x - 1 + \pi(\sqrt{x}) \ldots$
- $- \left\lfloor \frac{x}{2} \right\rfloor \ldots$

Combinatoric Methods

- $\pi(x) = x - 1 + \pi(\sqrt{x}) \ldots$
- $- \left\lfloor \frac{x}{2} \right\rfloor \ldots$
- $- \left\lfloor \frac{x}{3} \right\rfloor + \left\lfloor \frac{x}{6} \right\rfloor \ldots$

Combinatoric Methods

- $\pi(x) = x - 1 + \pi(\sqrt{x}) \ldots$
- $- \left\lfloor \frac{x}{2} \right\rfloor \ldots$
- $- \left\lfloor \frac{x}{3} \right\rfloor + \left\lfloor \frac{x}{6} \right\rfloor \ldots$
- $- \left\lfloor \frac{x}{5} \right\rfloor + \left\lfloor \frac{x}{10} \right\rfloor + \left\lfloor \frac{x}{15} \right\rfloor - \left\lfloor \frac{x}{30} \right\rfloor \ldots$

Combinatoric/Sieve Methods

Meissel/Lehmer/Lagarias/Miller/Odlyzko/Deléglise/Rivat

- Use various sieve “tricks” to reduce search tree

Combinatoric/Sieve Methods

Meissel/Lehmer/Lagarias/Miller/Odlyzko/Deléglise/Rivat

- Use various sieve "tricks" to reduce search tree
- Is about $O\left(\frac{x^{\frac{2}{3}}}{\log^2(x)}\right)$ in time

Combinatoric/Sieve Methods

Meissel/Lehmer/Lagarias/Miller/Odlyzko/Deléglise/Rivat

- Use various sieve "tricks" to reduce search tree
- Is about $O\left(\frac{x^{\frac{2}{3}}}{\log^2(x)}\right)$ in time
- Has been used to compute $\pi\left(10^{23}\right)$

Combinatoric/Sieve Methods

Meissel/Lehmer/Lagarias/Miller/Odlyzko/Deléglise/Rivat

- Use various sieve "tricks" to reduce search tree
- Is about $O\left(\frac{x^{\frac{2}{3}}}{\log^2(x)}\right)$ in time
- Has been used to compute $\pi\left(10^{23}\right)$
- BUT...

The End Of The Road?

- Parallelised version of M/L/L/M/O/D/R method by Gourdon ran for 360 days

The End Of The Road?

- Parallelised version of M/L/L/M/O/D/R method by Gourdon ran for 360 days
- Was ± 1 out in global checks

The End Of The Road?

- Parallelised version of M/L/L/M/O/D/R method by Gourdon ran for 360 days
- Was ± 1 out in global checks
- Project was abandoned

The End Of The Road?

- Parallelised version of M/L/L/M/O/D/R method by Gourdon ran for 360 days
- Was ± 1 out in global checks
- Project was abandoned
- 6 years later in 2007 Oliveira e Silva recomputed it correctly.

The End Of The Road?

- Parallelised version of M/L/L/M/O/D/R method by Gourdon ran for 360 days
- Was ± 1 out in global checks
- Project was abandoned
- 6 years later in 2007 Oliveira e Silva recomputed it correctly.
- Have such methods reached their limit?

$\pi^*(x)$

- $\pi^*(x) := \frac{1}{2}\left[\sum_{p^n<x} \frac{1}{n} + \sum_{p^n \leq x} \frac{1}{n}\right]$

$\pi^*(x)$

- $\pi^*(x) := \frac{1}{2}\left[\sum_{p^n<x} \frac{1}{n} + \sum_{p^n \le x} \frac{1}{n}\right]$
- $\pi(x)$ is cheap to recover from $\pi^*(x)$ (Möbius inversion)

$\pi^*(x)$

- $\pi^*(x) := \frac{1}{2}\left[\sum_{p^n<x} \frac{1}{n} + \sum_{p^n \le x} \frac{1}{n}\right]$
- $\pi(x)$ is cheap to recover from $\pi^*(x)$ (Möbius inversion)
- Riemann showed given $\Re s > 1$, $\frac{\log \zeta(s)}{s} = \int_0^\infty \pi^*(x) x^{-s} \frac{dx}{x}$

$\pi^*(x)$

- $\pi^*(x) := \frac{1}{2}\left[\sum_{p^n<x} \frac{1}{n} + \sum_{p^n \le x} \frac{1}{n}\right]$
- $\pi(x)$ is cheap to recover from $\pi^*(x)$ (Möbius inversion)
- Riemann showed given $\Re s > 1$, $\frac{\log \zeta(s)}{s} = \int_0^\infty \pi^*(x) x^{-s} \frac{dx}{x}$
- By Mellin inversion $\pi^*(x) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} \log \zeta(s) x^s \frac{ds}{s}$ $(\sigma > 1)$

$\pi^*(x)$

- $\pi^*(x) := \frac{1}{2}\left[\sum_{p^n<x} \frac{1}{n} + \sum_{p^n \le x} \frac{1}{n}\right]$
- $\pi(x)$ is cheap to recover from $\pi^*(x)$ (Möbius inversion)
- Riemann showed given $\Re s > 1$, $\frac{\log \zeta(s)}{s} = \int_0^\infty \pi^*(x) x^{-s} \frac{dx}{x}$
- By Mellin inversion $\pi^*(x) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} \log \zeta(s) x^s \frac{ds}{s}$ $(\sigma > 1)$
- (after some analysis) $= \mathrm{Li}(x) - \sum_\rho \mathrm{Li}\left(x^\rho\right) - \log 2 + O\left(x^{-1}\right)$

Forcing Convergence

- L.-O. suggested introducing a suitable Mellin transform pair $\phi(t)$ and $\hat{\phi}(s)$ so that

Forcing Convergence

- L.-O. suggested introducing a suitable Mellin transform pair $\phi(t)$ and $\hat{\phi}(s)$ so that
- $$\pi^*(x) = \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty} \hat{\phi}(s)\log\zeta(s)\mathrm{d}s + \sum_{p^m}\frac{1}{m}\left[\chi_x\left(p^m\right) - \phi\left(p^m\right)\right]$$

Forcing Convergence

- L.-O. suggested introducing a suitable Mellin transform pair $\phi(t)$ and $\hat{\phi}(s)$ so that
- $\pi^*(x) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \hat{\phi}(s) \log \zeta(s) \mathrm{d}s + \sum_{p^m} \frac{1}{m} \left[\chi_x\left(p^m\right) - \phi\left(p^m\right)\right]$
- where $\chi_x(t) := \begin{cases} 1 & t < x \\ 1/2 & t = x \\ 0 & t > x \end{cases}$

Forcing Convergence

- L.-O. suggested introducing a suitable Mellin transform pair $\phi(t)$ and $\hat{\phi}(s)$ so that
- $\pi^*(x) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \hat{\phi}(s) \log \zeta(s) \mathrm{d}s + \sum_{p^m} \frac{1}{m} \left[\chi_x\left(p^m\right) - \phi\left(p^m\right)\right]$
- where $\chi_x(t) := \begin{cases} 1 & t < x \\ 1/2 & t = x \\ 0 & t > x \end{cases}$
- This algorithm is potentially $O\left(x^{\frac{1}{2}}\right)$ in time.

Forcing Convergence

- L.-O. suggested introducing a suitable Mellin transform pair $\phi(t)$ and $\hat{\phi}(s)$ so that
- $\pi^*(x) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \hat{\phi}(s) \log \zeta(s) \mathrm{d}s + \sum_{p^m} \frac{1}{m} \left[\chi_x\left(p^m\right) - \phi\left(p^m\right)\right]$
- where $\chi_x(t) := \begin{cases} 1 & t < x \\ 1/2 & t = x \\ 0 & t > x \end{cases}$
- This algorithm is potentially $O\left(x^{\frac{1}{2}}\right)$ in time.
- Note $\hat{\phi}(s) = \frac{x^s}{s}$ gives $\phi(t) = \chi_x(t)$ and we are back to Riemann.

J. Buethe, J. Franke, A. Jost, T. Kleinjung

- They have developed a version of this algorithm, but based on Weil's explict formula.

J. Buethe, J. Franke, A. Jost, T. Kleinjung

- They have developed a version of this algorithm, but based on Weil's explict formula.
- It is contingent on R.H. (and no I don't know why).

J. Buethe, J. Franke, A. Jost, T. Kleinjung

- They have developed a version of this algorithm, but based on Weil's explict formula.
- It is contingent on R.H. (and no I don't know why).
- They have computed a value for $\pi\left(10^{24}\right)$.

Galway

- G. suggested $\hat{\phi}(s) = \exp\left(\frac{\lambda^2 s^2}{2}\right)\frac{x^s}{s}$, $\phi(t) = \frac{1}{2}\mathrm{erfc}\left(\frac{\log\left(\frac{t}{x}\right)}{\sqrt{(2)}\lambda}\right)$

Galway

- G. suggested $\hat{\phi}(s) = \exp\left(\frac{\lambda^2 s^2}{2}\right)\frac{x^s}{s}$, $\phi(t) = \frac{1}{2}\mathrm{erfc}\left(\frac{\log\left(\frac{t}{x}\right)}{\sqrt{(2)}\lambda}\right)$
- Self-duality of Gaussian implies these are “optimal”

Galway

- G. suggested $\hat{\phi}(s) = \exp\left(\frac{\lambda^2 s^2}{2}\right)\frac{x^s}{s}$, $\phi(t) = \frac{1}{2}\mathrm{erfc}\left(\frac{\log\left(\frac{t}{x}\right)}{\sqrt{(2)}\lambda}\right)$
- Self-duality of Gaussian implies these are “optimal”
- Now everything converges absolutely so just compute

Galway

- G. suggested $\hat{\phi}(s) = \exp\left(\frac{\lambda^2 s^2}{2}\right)\frac{x^s}{s}$, $\phi(t) = \frac{1}{2}\mathrm{erfc}\left(\frac{\log\left(\frac{t}{x}\right)}{\sqrt{(2)}\lambda}\right)$
- Self-duality of Gaussian implies these are "optimal"
- Now everything converges absolutely so just compute
- $\pi^*(x) = \frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty}\hat{\phi}(s)\log\zeta(s)\mathrm{d}s + \sum_{p^m}\frac{1}{m}\left[\chi_x\left(p^m\right) - \phi\left(p^m\right)\right]$

Computing The Sum

- The sum is a prime sieve, centred on x.

Computing The Sum

- The sum is a prime sieve, centred on x.
- At (e.g.) $x = 10^{24}$ it needs to be 10^{16} or so wide.

Computing The Sum

- The sum is a prime sieve, centred on x.
- At (e.g.) $x = 10^{24}$ it needs to be 10^{16} or so wide.
- This is (just) achievable.

Computing The Integral

- Numerical integration is problematic.

Computing The Integral

- Numerical integration is problematic.
- So let $G(s)$ be a primitive of $\hat{\phi}$ such that $G(2 + i\infty) = -G(2 - i\infty)$

Computing The Integral

- Numerical integration is problematic.
- So let $G(s)$ be a primitive of $\hat{\phi}$ such that $G(2 + i\infty) = -G(2 - i\infty)$
- Then the integral becomes $G(1) - \sum_{\rho} G(\rho) - log2 + O\left(x^{-1}\right)$

Computing The Integral

- Numerical integration is problematic.
- So let $G(s)$ be a primitive of $\hat{\phi}$ such that $G(2 + i\infty) = -G(2 - i\infty)$
- Then the integral becomes $G(1) - \sum_{\rho} G(\rho) - log2 + O\left(x^{-1}\right)$
- (Note similarity to Riemann's explicit formula.)

Computing G

- Start with

$$\hat{\phi}(s_0 + ih) = \hat{\phi}(s_0) \exp(ih(s_0\lambda^2 + \log(x))) \frac{\exp\left(\frac{-\lambda^2 h^2}{2}\right)}{1 + \frac{ih}{s_0}}$$

Computing G

- Start with

$$\hat{\phi}(s_0 + ih) = \hat{\phi}(s_0) \exp(ih(s_0\lambda^2 + \log(x))) \frac{\exp\left(\frac{-\lambda^2 h^2}{2}\right)}{1 + \frac{ih}{s_0}}$$

- Write everything as a Taylor series

Computing G

- Start with

$$\hat{\phi}(s_0 + ih) = \hat{\phi}(s_0) \exp(ih(s_0\lambda^2 + \log(x))) \frac{\exp\left(\frac{-\lambda^2 h^2}{2}\right)}{1 + \frac{ih}{s_0}}$$

- Write everything as a Taylor series
- Integrate term by term

We Need...

- A way of computing approximations to real numbers rigorously.

We Need...

- A way of computing approximations to real numbers rigorously.
- An efficient prime sieve

We Need...

- A way of computing approximations to real numbers rigorously.
- An efficient prime sieve
- Accurate and precise zeros of ζ (maybe 10^{11} of them)

Rigorous Computations

- Computers are finite, the real number system isn't.

Rigorous Computations

- Computers are finite, the real number system isn't.
- Rounding and truncation must be managed.

Rigorous Computations

- Computers are finite, the real number system isn't.
- Rounding and truncation must be managed.
- We use multiple precision interval arithmetic (MPFI by Revol and Rouillier)

Rigorous Computations

- Computers are finite, the real number system isn't.
- Rounding and truncation must be managed.
- We use multiple precision interval arithmetic (MPFI by Revol and Rouillier)
- This is one or two orders of magnitude slower than hardware.

An Efficient Prime Sieve

- Our sieve contains too many primes to compute $\phi(p)$ rigorously each time, so

An Efficient Prime Sieve

- Our sieve contains too many primes to compute $\phi(p)$ rigorously each time, so
- We compute a quadratic approximation to ϕ around some t_0.

An Efficient Prime Sieve

- Our sieve contains too many primes to compute $\phi(p)$ rigorously each time, so
- We compute a quadratic approximation to ϕ around some t_0.
- 1, t and t^2 form a basis for this 3 term approximation

An Efficient Prime Sieve

- Our sieve contains too many primes to compute $\phi(p)$ rigorously each time, so
- We compute a quadratic approximation to ϕ around some t_0.
- 1, t and t^2 form a basis for this 3 term approximation
- We sieve intervals I about 2^{32} wide centred on t_0 and compute the integers

An Efficient Prime Sieve

- Our sieve contains too many primes to compute $\phi(p)$ rigorously each time, so
- We compute a quadratic approximation to ϕ around some t_0.
- 1, t and t^2 form a basis for this 3 term approximation
- We sieve intervals I about 2^{32} wide centred on t_0 and compute the integers
  - $\sum_{p \in I} 1$

An Efficient Prime Sieve

- Our sieve contains too many primes to compute $\phi(p)$ rigorously each time, so
- We compute a quadratic approximation to ϕ around some t_0.
- 1, t and t^2 form a basis for this 3 term approximation
- We sieve intervals I about 2^{32} wide centred on t_0 and compute the integers
  - $\sum_{p \in I} 1$
  - $\sum_{p \in I} (p - t_0)$

An Efficient Prime Sieve

- Our sieve contains too many primes to compute $\phi(p)$ rigorously each time, so
- We compute a quadratic approximation to ϕ around some t_0.
- 1, t and t^2 form a basis for this 3 term approximation
- We sieve intervals I about 2^{32} wide centred on t_0 and compute the integers
  - $\sum_{p \in I} 1$
  - $\sum_{p \in I} (p - t_0)$
  - $\sum_{p \in I} (p - t_0)^2$

Zeros of ζ

- We have developed a rigorous, FFT based algorithm for computing ζ on the $\frac{1}{2}$ line.

Zeros of ζ

- We have developed a rigorous, FFT based algorithm for computing ζ on the $\frac{1}{2}$ line.
- It computes many evenly spaced values of ζ in average time $O(t^{\epsilon})$.

Zeros of ζ

- We have developed a rigorous, FFT based algorithm for computing ζ on the $\frac{1}{2}$ line.
- It computes many evenly spaced values of ζ in average time $O(t^{\epsilon})$.
- We can interpolate using these values to locate the zeros of ζ.

Zeros of ζ

- We have developed a rigorous, FFT based algorithm for computing ζ on the $\frac{1}{2}$ line.
- It computes many evenly spaced values of ζ in average time $O(t^{\epsilon})$.
- We can interpolate using these values to locate the zeros of ζ.
- We use Turing's method to confirm that no zeros have been missed.

Results

- We have tested the basic algorithm using Odlyzko's first $100,000$ zeros to compute $\pi\left(10^{11}\right)$.

Results

- We have tested the basic algorithm using Odlyzko's first $100,000$ zeros to compute $\pi\left(10^{11}\right)$.
- Using UoB's Bluecrystal cluster, we are computing enough zeros to reach $\pi\left(10^{22}\right)$.

Summary

- We have a working version of the Lagarias and Odlyzko analytic $\pi(x)$ algorithm.

Summary

- We have a working version of the Lagarias and Odlyzko analytic $\pi(x)$ algorithm.
- It might be able to reach $\pi\left(10^{24}\right)$.

Summary

- We have a working version of the Lagarias and Odlyzko analytic $\pi(x)$ algorithm.
- It might be able to reach $\pi\left(10^{24}\right)$.
- As a spin off, we will have lots of accurate and precise zeros of ζ to give away.