COUNTING THE INTEGERS CYCLOTOMIC METHODS CAN FACTOR

by

Jonathan Sorenson

Computer Sciences Technical Report #919

March 1990

Counting the Integers Cyclotomic Methods Can Factor

Jonathan Sorenson

Computer Sciences Department
University of Wisconsin-Madison
1210 West Dayton Street
Madison, WI 53706
sorenson@cs.wisc.edu

March 16, 1990

Sponsored by NSF grant CCR-8552596

Abstract

Let $F(x,t,A)$ count those integers $\leq x$ that algorithm A can completely factor with probability at least $1/2$ using at most t arithmetic operations. We give estimates of this function for integer factoring algorithms based on cyclotomic polynomials. Setting t to a polynomial in $\log x$ tells us how many integers these methods can completely factor in random polynomial time.

We give fully rigorous bounds on this function for the $p \pm 1$ methods, which are the simplest cyclotomic algorithms. Our results show that $F(x,t,A)$ for these methods falls properly between that of trial division and Lenstra's elliptic curve method, both of which were analyzed previously by Hafner and McCurley. Using reasonable heuristics, we improve our upper and lower bounds so that they fall within a small constant multiple of one another.

We also give a heuristic upper bound on $F(x,t,A)$ for Bach and Shallit's $\Phi_k(p)$ integer factoring algorithm. This algorithm is the most general method known based on cyclotomic polynomials, and choosing $k = 1,2$ gives the $p \pm 1$ methods as special cases.

Contents

1 Introduction.

Given a positive integer n, the *complete factorization problem* is to write n as a product of primes. In this paper, we estimate how many integers can be completely factored in random polynomial time using Pollard's $p-1$ factoring algorithm, the $p+1$ factoring algorithm, and Bach and Shallit's cyclotomic polynomial factoring algorithm. This continues the work of Hafner and McCurley, who analyzed the trial division algorithm and Lenstra's elliptic curve method.

The Problem

To date, there are no known factoring algorithms, even probabilistic ones, which run in time bounded by a polynomial in the length of the input. As a result, integer factoring is believed to be intractable, and this belief is often used as the basis for cryptographic protocols.

The fastest known algorithms to factor a positive integer n in the general case have running times of the form $L(n)^c$, where $L(n) = \exp[\sqrt{\log n \log\log n}(1 + o(1))]$ and c is a small constant. The list of such algorithms has grown quite long, and includes Morrison and Brillhart's continued fraction algorithm [MB75], Pomerance's quadratic sieve [Pom85], Seysen's algorithm based on quadratic forms [Sey87], and Lenstra's elliptic curve method [Len87]. The new number field sieve [LLMP89] factors integers of the form $n = r^e \pm s$ in expected time $\exp[(\log n)^{1/3+o(1)}]$.

Since we are a long way from a random polynomial time factoring algorithm, the question we ask is:

How many integers can we completely factor in random polynomial time?

Another version of this question asks if the set of integers which are completely factorable in random polynomial time has positive relative density [AM87].

To address this question, Hafner and McCurley [HM89] defined the function $F(x,t,A)$, the number of integers $\leq x$ that algorithm A can factor completely with probability at least $1/2$ using at most t arithmetic operations. If we have a good approximation for $F(x,t,A)$, then $F(x,(\log x)^{O(1)},A)$ tells us how many integers algorithm A can factor in random polynomial time. Our goal, then, is to approximate this function for several different factoring algorithms.

For some algorithms, however, estimating $F(x,t,A)$ tells us nothing new. Consider a factoring algorithm A that takes the same amount of time for all inputs of a given length. Unless A runs in random polynomial time, it cannot factor any integers whatsoever in random polynomial time. Clearly $F(x,t,A)$ for such an algorithm is not very interesting for our purposes. Aside from Lenstra's elliptic curve method, all of the algorithms mentioned above have this property.

The factoring algorithms for which $F(x,t,A)$ is worth considering are ones which adapt to their inputs in some fashion. Such algorithms have running times which vary for different inputs of the same length, and as we just argued, only ones of this type have a chance to factor integers in random polynomial time. Some examples of adaptive integer factoring algorithms are:

Trial division,
The Pollard-Strassen FFT method [Str76],
The Pollard-ρ method (whose analysis is only heuristic) [Pol75],
The $p \pm 1$ methods [Pol74, Wil82],
Bach and Shallit's $\Phi_k(p)$ methods [BS89a],
Lenstra's elliptic curve method [Len87].

Previous Work

Knuth and Trabb Pardo [KTP76] were the first to analyze the statistical behavior of the running time of trial division. They gave estimates for $F(x, x^{1/u}, TD)$ where $u > 1$ is fixed and where TD denotes the standard trial division algorithm.

Let $TDSS$ denote a trial division algorithm enhanced with the Solovay-Strassen probabilistic primality test [SS77]. Hafner and McCurley [HM89] proved the following asymptotic estimate on the number of integers factorable by $TDSS$. If $\log t = o(\log x)$ and $t/\log^2 x \log\log x \to \infty$, then

$$F(x,t,TDSS) \quad \sim \quad e^{\gamma}\frac{x}{\log x}\log t. \tag{1}$$

So, in random polynomial time, $TDSS$ can completely factor $\Theta(x \log\log x/\log x)$ integers $\leq x$. Using the Miller-Rabin prime test instead of Solovay-Strassen leads to the same result [Mil76, Rab80].

The analysis for the Pollard-Strassen method and the (heuristic) Pollard-ρ method would resemble that of trial division.

Let ECM denote Lenstra's elliptic curve method with an initial trial division phase and a probabilistic primality test. Hafner and McCurley also proved the following lower bound on the number of integers factorable by ECM. For any $C < 6/5$, if $t \geq \log^4 x$ and $\log t = o(\log x)^{1/c}$, then

$$F(x,t,ECM) \quad \geq \quad e^{\gamma}\frac{x}{\log x}\left(\log\frac{t}{\log x}\right)^{C}(1+o(1)). \tag{2}$$

This leads to the lower bound of $\Omega(x(\log\log x)^C/\log x)$ on the number of integers factorable in random polynomial time, which is the best one known. Heuristic arguments imply that (2) holds for any $C < 2$.

New Results

We continue this work by analyzing integer factoring algorithms based on cyclotomic polynomials. Bach and Shallit's cyclotomic polynomial factoring algorithm, the $\Phi_k(p)$ method, generalizes a number of algorithms, including the $p \pm 1$ methods [Pol74, Wil82], Williams and Judd's $p^2 + 1$ and $p^2 \pm p + 1$ methods [WJ76a, WJ76b], and Bach, Miller, and Shallit's sums of divisors method [BMS86]. We focus our attention on the $p \pm 1$ methods since they are the simplest ones and probably the only ones used regularly in practice. If we make use of some heuristics, our techniques generalize to the $\Phi_k(p)$ method.

Let PM denote Pollard's $p-1$ factoring algorithm and PP the $p+1$ factoring algorithm, both modified by adding an initial trial division phase and a probabilistic primality test. We prove the following upper and lower bounds on the number of integers factorable by the PM and PP algorithms. For every $\epsilon > 0$,

$$F(x,t,PM \text{ or } PP) \leq (3+\epsilon)e^{\gamma}\frac{x}{\log x}\left(\log\frac{t}{\log x}\right)\frac{\log\log t}{\log\log\log t}(1+o(1)) \quad \text{and} \tag{3}$$

$$F(x,t,PM \text{ or } PP) \geq 1.07446 \cdot e^{\gamma}\frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)) \tag{4}$$

if $t \geq (\log x)^{2+\delta}$ and $\log t \leq (\log x)^{1-\delta}$ for $\delta > 0$. We also require that t, viewed as a function of x, be "nice" in the sense that $\sqrt{\log x}/\log t$ either tend to infinity or be bounded.

By making a reasonable heuristic assumption, we can improve both of these bounds. Let $\Pi(x,y)$ count those primes $p \leq x$ such that $p+a$ has no prime divisors larger than y, where $a \neq 0$ is a fixed integer. Assuming that $\Pi(x,y) \sim \pi(x)\rho(u)$ for all u such that $1 \leq u \leq (3+\epsilon)\log\log y/\log\log\log y$, where $u = \log x/\log y$, $\epsilon > 0$, and ρ is Dickman's function, we show that

$$F(x,t,PM \text{ or } PP) \leq 1.78108 \cdot e^{\gamma}\frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)) \quad \text{and} \tag{5}$$

$$F(x,t,PM \text{ or } PP) \geq 1.30685 \cdot e^{\gamma}\frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)). \tag{6}$$

Our rigorous results prove the widely-held belief that the $p \pm 1$ methods factor more integers than trial division but fewer than the elliptic curve method. To be more specific, (1), (2), (3), and (4) imply that

$$\lim_{x\to\infty}\frac{F(x,t,PM \text{ or } PP)}{F(x,t,TDSS)} > 1 \quad \text{and} \quad \lim_{x\to\infty}\frac{F(x,t,ECM)}{F(x,t,PM \text{ or } PP)} = \infty$$

hold when $t \geq (\log x)^{14.5}$, $\log t = o(\log x)^{5/6}$, and $\sqrt{\log x}/\log t$ is either bounded or tends to infinity with x. Using (6), we can drop the lower bound on t to $(\log x)^{4.26}$.

Let C_k denote the $\Phi_k(p)$ method, where $\Phi_k(x)$ is the kth cyclotomic polynomial. Assuming that an integer of the form $\Phi_k(p)$ for p a prime has all small prime divisors with probability no greater than a randomly chosen number near $\Phi_k(p)$, we show that

$$F(x,t,C_k) \leq \left(1+\frac{\rho(\phi(k))}{\phi(k)}\right)\cdot e^{\gamma}\frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)) \tag{7}$$

where ρ is Dickman's function and ϕ is Euler's function.

Suppose we construct an algorithm which tries the $\Phi_k(p)$ methods for all values of k up to some bound l. Does such an algorithm factor a significantly larger number of integers? Our answer is no. Let $MC(l)$ denote this algorithm. Making the same assumptions as in (7), we prove that

$$F(x,t,MC(l)) = \Theta\left(\frac{x\log t}{\log x}\right). \tag{8}$$

This bound is independent of l and holds even if $l \to \infty$.

Let us summarize our results. Using reasonable heuristic assumptions, any known cyclotomic polynomial based algorithm will factor $\Theta(x \log\log x / \log x)$ integers $\leq x$ in random polynomial time. If we restrict ourselves to fully rigorous arguments, we have shown that the $p \pm 1$ methods factor at most $O(x(\log\log x)^{1+o(1)}/\log x)$ and at least $\Omega(x \log\log x/\log x)$ integers $\leq x$ in random polynomial time.

Our heuristic assumptions ignore the many known algebraic properties of cyclotomic polynomials. To partially justify this, we give some computational evidence that supports our heuristics.

Organization

After giving some definitions and notation in section 2, we cover the $p-1$, $p+1$, and $\Phi_k(p)$ methods in sections 3, 4, and 5. We conclude in section 6 with computational results to support our heuristic assumptions.

2 Notation.

In this section we go over our use of notation and some definitions. This is meant to be used as a reference by the reader. Most of these definitions also occur in other parts of this paper.

Asymptotics

If f and g are two real-valued functions, we write $f = O(g)$ to mean that there exists $c > 0$ such that $|f(x)| \leq c|g(x)|$ for sufficiently large x. Similarly we write $f = \Omega(g)$ to mean $|f(x)| \geq c|g(x)|$ for sufficiently large x. $f = o(g)$ means $\lim_{x\to\infty} f(x)/g(x) = 0$. $f \sim g$ means $\lim_{x\to\infty} f(x)/g(x) = 1$, or equivalently, $f(x) = g(x)(1 + o(1))$. $f \ll g$ means $f = O(g)$. $f = \Theta(g)$ means $f = O(g)$ and $g = O(f)$. All these implied constants are absolute.

Let f be a real-valued function. We say $f(x)$ is *well-behaved* if either $f(x) \to \infty$ as $x \to \infty$ or $f(x)$ is bounded.

Number Theoretic Functions

If S is a finite set, we write $\#S$ to denote its cardinality.

$\Phi_k(x)$ is the kth cyclotomic polynomial, whose roots are precisely the kth primitive roots of unity and has degree $\phi(k)$, where ϕ is Euler's function. $\psi_k(x) = (x^k - 1)/\Phi_k(x)$.

$P_k(n)$ denotes the kth largest prime divisor of n. If n has fewer than k prime divisors, then $P_k(n) = 1$, and if n is prime, $P_1(n) = n$. An integer n is y-smooth if $P_1(n) \leq y$.

$\mathbf{Z}/(n)$ is the ring of integers modulo n, $(\mathbf{Z}/(n))^*$ is its multiplicative group of units, and we have $\phi(n) = \#(\mathbf{Z}/(n))^*$.

$GF(q)$ is the finite field of q elements, where q is a prime power. $GF(q)^*$ is its multiplicative group, and we have $\#GF(q)^* = q - 1$.

$\rho_k(u)$ is Dickman's function; see Knuth and Trabb Pardo [KTP76].

$$\begin{aligned}
\pi(x) &= \#\{p \leq x \ : \ p \text{ is prime }\} \\
\Psi_k(x,y) &= \#\{n \leq x \ : \ P_k(n) \leq y\} \\
\Pi_k(x,y) &= \#\{p \leq x \ : \ p \text{ is prime and } P_1(\Phi_k(p)) \leq y\} \\
S(x) &= \#\{n \leq x \ : \ P_1(n) = P_2(n)\} \\
G(x,p) &= \#\{n \leq x \ : \ P_2(n) = p\} \\
H(x,y,p) &= \#\{n \leq x \ : \ P_2(n) = p \text{ and } P_3(n) \leq y\} \\
\Lambda_k(x,y,z) &= \#\{n \leq x \ : \ P_1(\Phi_k(P_2(n))) \leq y \text{ and } P_3(n) \leq z\} \\
\Lambda_k(x,y) &= \Lambda_k(x,y,x) \\
\Lambda^*(x,y,l) &= \#\{n \leq x \ : \ \text{for some } k,\ 1 \leq k \leq l,\ P_1(\Phi_k(P_2(n))) \leq y\} \\
L(x) &= e^{\sqrt{\log x \log\log x}} \\
Li(x) &= \int_2^x \frac{dt}{\log t} \\
V_k(x,y) &= \#\{n \leq x \ : \ P_1(\Phi_k(P_1(n))) \leq y\} \\
V^*(x,y,l) &= \#\{n \leq x \ : \ \text{for some } k,\ 1 \leq k \leq l,\ P_1(\Phi_k(P_1(n))) \leq y\} \\
D_k(z) &= \int_1^z \rho\left(\frac{z}{u} - 1\right) \rho\left(\phi(k)u\right) du
\end{aligned}$$

For all of the above functions, if the subscript k is omitted, assume it is 1. For example, $\Psi(x, y)$ is $\Psi_1(x, y)$.

$\Delta f(w) = f(w) - f(w-1)$, and for functions of two (or more) variables, we subscript Δ according to the variable which differs: $\Delta_w f(w, y) = f(w, y) - f(w-1, y)$.

Algorithms

C_k is the $\Phi_k(p)$ method. PM is the $p-1$ method, which is also C_1. PP is the $p+1$ method, which is also C_2. $MC(l)$ is the combined C_k method, for all $1 \leq k \leq l$ to the bound l.

All of these algorithms are assumed to be enhanced with probabilistic primality tests, such as Solovay-Strassen [SS77] or Miller-Rabin [Mil76, Rab80], which allows the algorithms to determine with high probability whether or not they have completely factored an integer. They also perform preliminary trial division, as would be done in practice.

All of these algorithms are assumed to have access to random bits. Define $F(x, t, A)$ to be the number of integers $\leq x$ that, with probability at least $1/2$, algorithm A can completely factor in t arithmetic operations. In other words, we define

$$F(x, t, A) \quad = \quad \#\{n \leq x \ : \ \Pr(\ A \text{ can completely factor } n \text{ in } t \text{ steps }) \geq 1/2\}\,.$$

A more precise definition appears in Hafner and McCurley [HM89].

Our definition of *arithmetic operation* or *step* is one addition, subtraction, multiplication, division, greatest common divisor, or assignment operation on $O(\log x)$-bit integers. Note that using classical algorithms as found in Knuth [Knu81], these all have bitwise complexity of $O(\log^2 x)$, and with the methods of Schönhage and Strassen [SS71], they have complexity $O((\log x)^{1+o(1)})$.

3 Pollard's $p-1$ Integer Factoring Algorithm.

In this section we give our results on the Pollard $p-1$ integer factoring algorithm, which we will denote PM. The proof techniques we use here apply directly to the $p+1$ method, and what we prove on general cyclotomic algorithms uses the same approach.

An integer is smooth if all of its prime divisors are small. The central idea of the $p-1$ algorithm is that it can find any prime divisor p of an integer if $p-1$ is smooth. Since a number is more likely to be smooth if it is small, the $p-1$ method typically removes the smallest prime divisors first, leaving the largest prime divisor for last. So we define a function Λ that counts the integers whose behavior under the $p-1$ method is "typical." Our results center around approximating this function asymptotically. We also must take care of the "non-typical" cases: when the largest prime is not removed last, or when the largest and second largest prime divisors are equal. We define the functions V and S to count these cases, and show that they occur far less frequently than the "typical" case.

We start by reviewing Pollard's $p-1$ factoring algorithm and giving its arithmetic complexity. We then define $\Lambda(x, y, z)$. We then cover previous work on smooth numbers, and use them to give upper bounds on S and V. We prove some upper and lower bounds on Λ and apply them to $F(x, t, PM)$ giving our main results. We finish this section with some much tighter bounds using reasonable heuristic assumptions on smooth numbers of the form $p-1$ for p a prime.

3.1 The Algorithm.

Let $n = p \cdot q$ be the integer we wish to factor. Further suppose $p \leq \sqrt{n}$, p is prime, $p-1$ factors completely over the primes below a bound B, and that q is relatively prime to p. Let $p_1, \ldots, p_r$ be the primes below B, and let e_i be the largest integer such that $p_i^{e_i} \leq \sqrt{n}$.

Choose $a \in (\mathbf{Z}/(n))^*$ uniformly at random and compute $b \equiv a^E \pmod{n}$ where $E = \prod_{i=1}^{r} p_i^{e_i}$. Since $p-1 \mid E$, we have $b \equiv 1 \pmod{p}$. If $b \not\equiv 1 \pmod{q}$, then $d := \gcd(b-1, n)$ is a proper divisor of n by the Chinese Remainder Theorem. If d is not prime, we know that E is a multiple of the order of $(\mathbf{Z}/(d))^*$, so we can use Miller's algorithm to factor d in at most $O(\log E)$ expected arithmetic operations for each prime divisor of d [Mil76].

In addition, we add trial division, a probabilistic prime test such as Solovay-Strassen [SS77] or Miller-Rabin [Mil76, Rab80], and one of the perfect power testing algorithms of Bach and Sorenson [BS89b] to give us the following algorithm.

Algorithm PM

1. Find all the primes $p_i \leq B$, and compute the e_i.

2. Remove all primes factors below B from n with trial division. If n is now (probably) a prime power, halt.

3. Repeat the following *main loop* for several choices of a. Use Miller's factoring algorithm to completely factor any composite divisors that are found.

```
Choose $a \in (\mathbf{Z}/(n))^*$;
$b := a$;
For $i := 1$ to $r$ do:
    $v := p_i^{e_i}$;
    $b := b^v \bmod n$;
    If $d := \gcd(b-1, n) > 1$ Then:
        $n := n/d$;
        If $n$ is probably prime or a prime power Then halt;
    End if;
End for;
```

Complexity

Step 1.

All the primes below B can be found in $O(B \log\log B)$ arithmetic operations using the Sieve of Eratosthenes. Since $B \leq \sqrt{n}$, this is only $O(B \log\log n)$ operations.

Step 2.

Trial Division will use at most $O(B)$ arithmetic operations.

Step 3.

The main loop will use $O(\log E)$ operations. Since $\log E \leq \pi(B) \log \sqrt{n}$, this is $O(B \log n)$ operations.

Since n has at most $\log n$ divisors, we need execute only $O(\log n)$ prime tests, each of which uses $O(\log n)$ operations. We will repeat these $O(\log\log n)$ times to insure a total error probability of at most $1/2$, giving a total of $O(\log^2 n \log\log n)$ operations for prime testing. A perfect power can be recognized in $O(\log n \log\log n)$ arithmetic operations, and the same time bound will produce the root as well. (See algorithm A using the modified Newton's method in [BS89b].)

Thus the total complexity of the algorithm is $O(\log^2 n \log\log n + B \log n)$ arithmetic operations. If we are allowed at most t arithmetic operations, then we choose $B = \Theta(t/\log n)$.

The $p-1$ factoring algorithm also has two extensions, the standard extension and the FFT extension. The FFT extension gives the algorithm a worst-case running time of $O(n^{1/4+\epsilon})$ (see [Pol74]). We do not consider either of these extensions here for the following reasons. The FFT extension complicates the analysis, and its inclusion would only increase the upper bounds on $F(x,t,PM)$ by at most a factor of 2. The inclusion of the standard extension would only effect lower order terms in our estimate of $F(x,t,PM)$. For more details on these extensions and on other improvements to the $p-1$ factoring algorithm, see Montgomery [Mon87], Montgomery and Silverman [MS88], and Pollard [Pol74].

3.2 The Function $\Lambda(x,y,z)$.

As we mentioned earlier, we are going to estimate $F(x,t,PM)$ by first bounding it in terms of the function Λ, which counts the "typical" numbers algorithm PM can factor, and the functions S and V, which count the rest. We will now define these functions and prove a simple but important lemma that correlates them with $F(x,t,PM)$.

Let $P_i(n)$ be the ith largest prime divisor of the integer n. If n has fewer than i divisors, we define $P_i(n)=1$, and also $P_i(0)=0$. When $i=1$, we will drop the subscript altogether so that $P(n)$ is the largest prime divisor of n. Thus, $n=\prod_{i=1}^{\infty} P_i(n)$.

We say an integer n is *y-smooth* if $P(n)\leq y$.

We define

$$\Lambda(x,y,z) \quad = \quad \#\{n\leq x \ : \ P(P_2(n)-1)\leq y \ \text{ and } \ P_3(n)\leq z\}.$$

If we leave off the third argument z, assume it is equal to x, that is, $\Lambda(x,y):=\Lambda(x,y,x)$. For example, $\Lambda(x,x)=x$ and $\Lambda(x,1,1)=\pi(x)+\pi(x/2)+1$ since $P_2(n)=1$ or 2.

We also define the functions $S(x)$ and $V(x,y)$ as follows. $S(x)$ is the number of integers below x whose largest prime divisor divides at least twice. In other words (or symbols),

$$S(x) \quad = \quad \#\{n\leq x \ : \ P_1(n)=P_2(n)\}.$$

$V(x,y)$ is the number of integers whose largest prime divisor p has the property that $p-1$ is y-smooth, or

$$V(x,y) \quad = \quad \#\{n\leq x \ : \ P_1(P_1(n)-1)\leq y\}.$$

Lemma 3.1 *Let $B=B(x,t)=\Theta(t/\log x)$ be the bound in algorithm PM as described above. Then*

$$\begin{aligned} F(x,t,PM) &\leq \Lambda(x,B)+S(x)+V(x,B) \quad \textit{and} \\ F(x,t,PM) &\geq \Lambda(x,B,B). \end{aligned}$$

Proof: To prove the upper bound, we must show that $\Lambda(x,B)+S(x)+V(x,B)$ counts every $n\leq x$ where n is completely factorable by PM with probability $\geq 1/2$. If the algorithm removes $P_1(n)$ before $P_2(n)$, then n is counted by $V(x,B)$. If $P_1(n)=P_2(n)$, then n is counted by $S(x)$. That leaves the case where $P_1(n)>P_2(n)$ and the algorithm removes $P_2(n)$ before $P_1(n)$. Clearly it must have removed $p=P_2(n)$ either by trial division or in the main loop of the algorithm. In either case we have $P(p-1)\leq B$.

To prove the lower bound, we must show that any integer n counted by $\Lambda(x,B,B)$ can be completely factored by PM with probability at least 1/2. Since $P_3(n)\leq B$, trial division will remove all prime divisors from n except for possibly the two largest. (The main loop could also remove them.) Since $P(P_2(n)-1)\leq B$, the main loop of the algorithm will remove $P_2(n)$, unless $P_2(n)=P_1(n)$ or $P_2(n)=P_3(n)$. In the first case, n is a prime power after $P_3(n)$ was removed, and so n is completely factored anyway. In the second case, $P_2(n)$ is removed during trial division, leaving n a prime after trial division. □

3.3 Smooth Numbers.

Generally speaking, a smooth number is one with small prime divisors. To give estimates for the functions Λ, S, and V defined above, we make use of several functions that count various kinds of smooth numbers. The results we quote below characterize the asymptotic growth of these functions. We will use these results frequently throughout the rest of this paper.

First, we define the function

$$\Psi_k(x,y) \;=\; \#\{n \le x \;:\; P_k(n) \le y\}.$$

We are primarily interested in this function for $k=1$ or 2, and we often drop the subscript when $k=1$.

In 1930, Dickman [Dic30] proved that $\Psi(x,y) \sim x\rho(u)$ where $u = \log x/\log y$ is fixed and ρ is Dickman's function. This function is defined as the continuous solution to the equations

$$\rho(u) = 1 \text{ for } 0 \le u \le 1 \quad\text{and}\quad -u\rho'(u) = \rho(u-1) \text{ for } u > 1. \tag{9}$$

Note that $\rho(u)$ is differentiable for $0 < u < 1$ and $u > 1$. De Bruijn [dB51a] gave the asymptotic estimate

$$\begin{aligned}\rho(u) &= \exp\left[-u\left(\log u + \log\log u - 1 - \frac{1}{\log u} + \frac{\log\log u}{\log u} + O\left(\left(\frac{\log\log u}{\log u}\right)^2\right)\right)\right] \\ &= u^{-u\cdot(1+o(1))}.\end{aligned} \tag{10}$$

Knuth and Trabb Pardo [KTP76] generalized the ρ function to all values of k, showing that $\Psi_k(x, x^{1/u}) \sim x\rho_k(u)$ for u fixed. For $k=2$ they also showed that

$$\rho_2(u) \;=\; \frac{e^\gamma}{u}\left(1 + O\left(\frac{1}{u}\right)\right) \tag{11}$$

for $u \to \infty$, where γ is Euler's constant, and $e^\gamma = 1.78107241\cdots$.

Many improvements to Dickman's result have appeared over the years. De Bruijn [dB66] showed that

$$\Psi(x,y) \;\le\; x\rho(u)(1+o(1)) \tag{12}$$

where $u = \log x/\log y$, $(\log x)^{1+\epsilon} \le y \le x^\epsilon$, and $\epsilon > 0$. Hildebrand [Hil86] showed that

$$\Psi(x,y) \;=\; x\rho(u)\left(1 + O_\epsilon\left(\frac{\log(1+u)}{\log y}\right)\right) \tag{13}$$

uniformly for $x \ge 3$ and $y \ge \exp((\log\log x)^{5/3+\epsilon})$ where $\epsilon > 0$ is fixed. If the Extended Riemann Hypothesis holds, then the 5/3 above can be replaced by 1. For our purposes, any exponent smaller than 2 suffices. See also de Bruijn [dB51b, dB66] and Canfield, Erdös, and Pomerance [CEP83].

For Ψ_2, Hafner and McCurley [HM89] proved that

$$\Psi_2(x,y) \;=\; x\rho_2(u)\left(1 + O\left(\frac{1}{\log y}\right)\right) \tag{14}$$

for $2 \le y \le x$.

We also need the following identities involving $\rho(u)$. For proofs, see Knuth and Trabb Pardo [KTP76].

$$
\begin{aligned}
\rho(u) &= 1-\log u \quad \text{for} \quad 1 \leq u \leq 2 && (15)\\
\rho(u) &< \rho(u-1)/u \quad \text{for all } u > 1 && (16)\\
\int_{a-1}^{a} \rho(u)du &= a\rho(a) \quad \text{for all } a > 1 && (17)\\
\int_{a}^{b} \rho(u)du &< \rho(a)-\rho(b) \quad \text{for } a \geq 1 && (18)\\
\int_{0}^{\infty} \rho(u)du &= e^{\gamma} && (19)
\end{aligned}
$$

We conclude our review of smooth numbers by defining a function which counts smooth numbers of the form $p-1$ for p a prime:

$$\Pi(x,y) \;=\; \#\{p \leq x \;:\; P(p-1) \leq y \text{ and } p \text{ is prime } \}.$$

This function has a major role to play in our estimates of Λ, as we shall see.

3.4 Estimates for $S(x)$ and $V(x,y)$.

We now have the tools necessary to prove a crude upper bound for $S(x)$. Then, after introducing some more notation and discussing Stieltjes integrals, we give an upper bound for $V(x,y)$. The important point is that both of these are of much lower order than Λ, allowing us to ignore them later. Chances are that tighter bounds can be found for both functions, probably at the expense of more effort.

Let $L(x) = \exp[\sqrt{\log x \log\log x}]$. Notice that we can write $(\log x)^c \cdot L(x) = L(x)^{1+o(1)}$ for c any fixed constant.

Theorem 3.2 $S(x) \leq x/L(x)^{1+o(1)}$.

Proof: Using definitions only, we get the identity

$$S(x) \;=\; 1 + \sum_{p \leq \sqrt{x}} \Psi\left(\frac{x}{p^2}, p\right).$$

By allowing the sum to run over all integers and splitting the sum at $y = \lfloor L(x) \rfloor$ gives

$$
\begin{aligned}
S(x) &\leq \sum_{y=1}^{\sqrt{x}} \Psi\left(\frac{x}{y^2}, y\right) \;=\; \sum_{y=1}^{\lfloor L(x)\rfloor} \Psi\left(\frac{x}{y^2}, y\right) + \sum_{y=\lfloor L(x)\rfloor}^{\sqrt{x}} \Psi\left(\frac{x}{y^2}, y\right)\\
&= T_1 + T_2.
\end{aligned}
$$

First we estimate T_1 using (13) and (10):

$$
\begin{aligned}
T_1 &\leq \sum_{y=1}^{\lfloor L(x)\rfloor} \Psi\left(\frac{x}{y^2}, L(x)\right) \;\leq\; \sum_{y=1}^{\lfloor L(x)\rfloor} \frac{x}{y^2}\rho\left(\frac{\log(x/L(x)^2)}{\log L(x)}\right)(1+o(1))\\
&\ll x\rho\left(\sqrt{\frac{\log x}{\log\log x}} - 2\right) \;=\; \frac{x}{L(x)^{1+o(1)}}.
\end{aligned}
$$

And second we estimate T_2:

$$T_2 \quad \leq \quad \sum_{y=\lfloor L(x)\rfloor}^{\sqrt{x}} \frac{x}{y \cdot \lfloor L(x)\rfloor} \quad \ll \quad \frac{x \log x}{L(x)} \quad = \quad \frac{x}{L(x)^{1+o(1)}}$$

since $\Psi(w,y) \leq w$. □

Since PM includes a prime test, we know that $F(x,t,PM) \geq \pi(x) \sim x/\log x$ if $t = \Omega(\log x)$. Thus Theorem 3.2 implies that the $S(x)$ term in Lemma 3.1 is insignificant.

Frequently, we wish to evaluate or approximate a sum by rewriting it as a Stieltjes integral. In other words, we use

$$\sum_{i=a+1}^{b} f(i)\Delta g(i) \quad = \quad \int_a^b f(w)\, dg(\lfloor w \rfloor)$$

where $\Delta g(i) = g(i) - g(i-1)$. Once in this form, we often want to either approximate g asymptotically or bound it from above or below. The following lemma tells us how to do this.

Lemma 3.3 *Assume the integrals $\int_a^b f(w)\, dg(w)$ and $\int_a^b f(w)\, dh(w)$ exist. Then the following are true:*

1. *If f is decreasing on $[a,b]$ and $g \geq h$ on $[a,b]$, then*

$$\int_a^b f(w)\, dg(w) \quad \geq \quad \int_a^b f(w)\, dh(w) + O\left(f(a)g(a) + f(b)g(b)\right).$$

2. *If $g \sim h$ and a, $b \to \infty$, then*

$$\int_a^b f(w)\, dg(w) \quad \sim \quad \int_a^b f(w)\, dh(w) + o\left(f(a)h(a) + f(b)h(b)\right).$$

Proof: Both are easy to prove by integrating by parts twice. (See Knuth and Greene [GK81, chapter 4] and Apostol [Apo57, chapter 9] for more on Stieltjes integration.) □

Define $D_k(z) = \int_1^z \rho(z/u-1)\rho(\phi(k)u)du$. For our upper bound on $V(x,y)$, we need an upper bound for $D_1(z)$. We generalized the definition because we need it later for the $p+1$ and $\Phi_k(p)$ algorithms, and the proof of following lemma generalized easily.

Lemma 3.4 *If $z \to \infty$, then $D_k(z) = O\left(\rho\left(\sqrt{\phi(k)z} - 1\right)\right)$.*

Proof:

$$\begin{aligned} D_k(z) &= \int_1^z \rho(z/u-1) \cdot \rho(\phi(k)u)\, du \\ &= \left\{ \int_1^{\sqrt{z/\phi(k)}} + \int_{\sqrt{z/\phi(k)}}^z \right\} \rho(z/u-1) \cdot \rho(\phi(k)u)\, du \\ &\leq \rho\left(\sqrt{\phi(k)z} - 1\right) \int_1^{\sqrt{z/\phi(k)}} \rho(u)\, du \quad + \quad \int_{\sqrt{z/\phi(k)}}^z \rho(\phi(k)u)\, du \\ &= O\left(\rho\left(\sqrt{\phi(k)z} - 1\right)\right), \end{aligned}$$

using (18) and (16). □

And now for our upper bound on $V(x,y)$.

Lemma 3.5 *Let $\delta > 0$. If $\log x/\log y \to \infty$ and $y \geq (\log x)^{1+\delta}$, then*

$$V(x,y) \;=\; O\left(x(\log y)\rho\left(\sqrt{\frac{\log x}{\log y}}-1\right)\right).$$

Further, if $\log y \leq (\log x)^{1-\delta}$, then

$$V(x,y) \;\leq\; x\cdot\exp\left[-\Omega\left((\log x)^{\delta/2}\log\log x\right)\right].$$

Proof: From the definition of $V(x,y)$, we have

$$\begin{aligned} V(x,y) &= \Psi(x,y) + \sum_{y<p\leq x,\ P(p-1)\leq y} \Psi(x/p,p) \\ &\leq \Psi(x,y) + \sum_{w=y+1}^{x} \Psi(x/w,w)\Delta_w\Psi(w-1,y) \end{aligned}$$

where the subscript on Δ indicates that y is fixed: $\Delta_w\Psi(w-1,y) = \Psi(w-1,y) - \Psi(w-2,y)$. Notice this is 1 when $w-1$ has no prime divisors larger than y, and 0 otherwise. Now writing this as a Stieltjes integral gives

$$\Psi(x,y) + \int_y^x \Psi(x/w,w)\,d\Psi(w-1,y).$$

Approximating this asymptotically using Lemma 3.3 and (12) gives

$$\begin{aligned} V(x,y) &\sim \Psi(x,y) + \int_y^x \frac{x}{w}\rho\left(\frac{\log x}{\log w}-1\right) d\left(w\rho\left(\frac{\log w}{\log y}\right)\right) \\ &\quad + \; o\left(\Psi(x/y,y)\Psi(y-1,y) + \Psi(1,w)\Psi(x-1,y)\right) \\ &= \int_y^x \frac{x}{w}\rho\left(\frac{\log x}{\log w}-1\right)\rho\left(\frac{\log w}{\log y}\right)(1+o(1))\,dw \\ &\quad + \; O\left(\Psi(x,y) + y\cdot\Psi(x/y,y)\right). \end{aligned}$$

The $O(\cdot)$ term is bounded by $O(x\rho(\log x/\log y - 1))$ using (12). By substituting $u = \log w/\log y$, the integral can be rewritten as

$$x(\log y)(1+o(1))\int_1^{\log x/\log y} \rho\left(\frac{\log x}{u\log y}-1\right)\rho(u)du \;=\; x(\log y)(1+o(1))\cdot D_1\left(\frac{\log x}{\log y}\right)$$

where $D_1(z) = \int_1^z \rho(z/u-1)\rho(u)du$. By Lemma 3.4 we have $D_1(z) = O(\rho(\sqrt{z}-1))$, and since ρ is a decreasing function, we arrive at $V(x,y) = O\left(x(\log y)\rho(\sqrt{\log x/\log y}-1)\right)$, as desired. Substituting $\log y = (\log x)^{1-\delta}$ and using (10) completes the proof for when $\log y \leq (\log x)^{1-\delta}$. □

That completes the crude bounds for the "unusual" numbers factorable by the $p-1$ method. The rest of this section is devoted to bounds on $\Lambda(x,y,z)$.

3.5 An Upper Bound.

Next, we give a rigorous upper bound for $F(x,t,PM)$ by using the results proved above and by giving an upper bound for $\Lambda(x,y)$.

Let $\epsilon > 0$ be fixed. We define

$$\alpha_\epsilon(y) = (3+\epsilon)\frac{\log\log y}{\log\log\log y}. \tag{20}$$

Our choice for $\alpha_\epsilon(y)$ is motivated by the following lemma.

Lemma 3.6 *Let $\epsilon > 0$. If $y \to \infty$, then $\rho(\alpha_\epsilon(y)) = o(1/\log^3 y)$.*

Proof: Follows from (10) and (20). □

Our upper bound for Λ is given by the following lemma.

Lemma 3.7 *Let ϵ, $\delta > 0$. If $\log x/(\alpha_\epsilon(y)\log y) \to \infty$, $y \geq (\log x)^{1+\delta}$, and $\sqrt{\log x}/\log y$ is well-behaved, then*

$$\Lambda(x,y) \leq e^\gamma \frac{x}{\log x}\alpha_\epsilon(y)\log y(1+o(1)).$$

We postpone the proof until after the following theorem, which gives an upper bound for $F(x,t,PM)$.

Theorem 3.8 *If ϵ, $\delta > 0$, $t/\log^2 x\log\log x \to \infty$, $\sqrt{\log x}/\log t$ is well-behaved, and $\log t \leq (\log x)^{1-\delta}$, then*

$$F(x,t,PM) \leq (3+\epsilon)e^\gamma \frac{x}{\log x}\left(\log\frac{t}{\log x}\right)\frac{\log\log t}{\log\log\log t}(1+o(1)).$$

Proof: Follows from Lemma 3.1, Lemma 3.7, Theorem 3.2, Lemma 3.5, and (20). □

This proves (3) for the $p-1$ method.

Proof of Lemma 3.7: Our first step is to express $\Lambda(x,y)$ in terms of simpler functions. With this in mind, let us define $G(x,p)$ to be the number of positive integers below x with p as their second largest prime divisor, that is

$$G(x,p) = \#\{n \leq x \ : \ p = P_2(n)\}.$$

Using definitions, we have the identity

$$G(x,p) = \Psi_2(x/p,p) - \Psi(x/p,p-1). \tag{21}$$

Of course, $G(x,y) = 0$ when $y > 1$ is not prime, and $G(x,1) = \pi(x)$. We also have

$$\sum_{w=1}^{y} G(x,w) = \pi(x) + \sum_{p \leq y} G(x,p) = \Psi_2(x,y). \tag{22}$$

This now allows us to write

$$\begin{aligned}\Lambda(x,y) &= G(x,1) + \sum_{p\leq\sqrt{x},\ P(p-1)\leq y} G(x,p) \\ &= \pi(x) + \sum_{p\leq\sqrt{x}} G(x,p)\Delta_p\Pi(p,y)\end{aligned}$$

since $\Delta_p\Pi(p,y) = 1$ precisely when $P(p-1) \leq y$. Now we dissect the sum at $y^{\alpha_\epsilon(y)}$, giving

$$\begin{aligned}\Lambda(x,y) &= \pi(x) + \sum_{p\leq y^{\alpha_\epsilon(y)}} G(x,p)\Delta_p\Pi(p,y) + \sum_{y^{\alpha_\epsilon(y)}<p\leq\sqrt{x}} G(x,p)\Delta_p\Pi(p,y) \\ &= T_1 + T_2\end{aligned}$$

with the $\pi(x)$ grouped with T_1.

For T_1, notice that $\Delta_w\Pi(w,y) \leq \Delta\pi(w)$, giving us $T_1 \leq \Psi_2(x, y^{\alpha_\epsilon(y)})$ by (22). Using (14) and (11), this gives

$$T_1 \leq e^\gamma \frac{x}{\log x}\alpha_\epsilon(y)(\log y)(1+o(1)).$$

It remains to show that T_2 is small; we will show that $T_2 = o((x/\log x)\log y)$. Noticing that $\Delta_w\Pi(w,y) \leq \Delta_w\Psi(w-1,y)$, we have

$$\begin{aligned}T_2 &= \sum_{y^{\alpha_\epsilon(y)}<p\leq\sqrt{x}} G(x,p)\Delta_p\Pi(p,y) \quad \leq \sum_{w=\lfloor y^{\alpha_\epsilon(y)}\rfloor}^{\lfloor\sqrt{x}\rfloor} G(x,w)\Delta_w\Psi(w-1,y) \\ &\leq \sum_{w=\lfloor y^{\alpha_\epsilon(y)}\rfloor}^{\lfloor y^{\log y}\rfloor} G(x,w)\Delta_w\Psi(w-1,y) + \sum_{w=\lceil y^{\log y}\rceil}^{\lfloor\sqrt{x}\rfloor} G(x,w)\Delta_w\Psi(w-1,y) \\ &= S_1 + S_2. \end{aligned} \tag{23}$$

We will show that $S_1 = o(x\log y/\log x)$ and $S_2 = o(x/\log x)$.

First for S_2, the easier one. Since $w \geq y^{\log y}$, we have

$$\Delta_w\Psi(w-1,y) \leq \Delta_w\Psi(w-1, w^{1/\log y}),$$

giving us

$$\begin{aligned}S_2 &\leq \sum_{w=\lceil y^{\log y}\rceil}^{\lfloor\sqrt{x}\rfloor} G(x,w)\Delta_w\Psi(w-1,w^{1/\log y}) \quad = \int_{\lceil y^{\log y}\rceil-1}^{\sqrt{x}} G(x,w)d\Psi(w-1,w^{1/\log y}) \\ &\sim \int_{y^{\log y}}^{\sqrt{x}} G(x,w)d(w-1)\rho(\log y) \\ &\quad + \ o\left(G(x,y^{\log y})\Psi(y^{\log y},y) + G(x,\sqrt{x})\Psi(\sqrt{x},x^{1/(2\log y)})\right) \\ &= \int_{y^{\log y}}^{\sqrt{x}} G(x,w)\rho(\log y)dw + o(x\rho(\log y))\end{aligned}$$

using (13), (12), and Lemma 3.3. Since $G(x,w) \le \Psi_2(x/w,w) \le x/w$ by (21), this gives us $S_2 \ll x(\log x)\rho(\log y)$. Using (10), $\rho(\log y) = (\log y)^{-(\log y)(1+o(1))} = y^{-\Theta(\log\log y)}$, and since $y \ge \log x$, we have

$$S_2 \quad \ll \quad x(\log x)^{-\Theta(\log\log\log x)} \quad = \quad o(x/\log x). \tag{24}$$

Now for S_1. We will split this into two cases; either $\log y = \Omega(\sqrt{\log x})$ or $\log y = o(\sqrt{\log x})$. We are allowed to do this because $\sqrt{\log x}/\log y$ is well-behaved.

Case 1: $\log y = \Omega(\sqrt{\log x})$.

Using the same type of argument as we did for S_2 above, we deduce that

$$S_2 \quad \ll \quad x(\log x)\rho(\alpha_\epsilon(y)).$$

By Lemma 3.6, $x(\log x)\rho(\alpha_\epsilon(y)) = o(x(\log x)(\log y)^{-3})$, and by our assumption that $\log y = \Omega(\sqrt{\log x})$, we have

$$S_1 \quad = \quad o\left(\frac{x}{\log x}\log y\right). \tag{25}$$

Case 2: $\log y = o(\sqrt{\log x})$.

This assumption allows us to estimate Ψ_2, so using (21), (14), (11), and the fact that $\log(x/w) \ge (1/2)\log x$, we have

$$G(x,w) \quad \le \quad \Psi_2(x/w,w) \quad \le \quad e^\gamma \frac{2x}{w\log x}(\log w)(1+o(1)).$$

This gives us

$$\begin{aligned} S_1 \quad &= \quad \sum_{w=\lfloor y^{\alpha_\epsilon(y)}\rfloor}^{\lfloor y^{\log y}\rfloor} G(x,w)\Delta_w\Psi(w-1,y) \quad = \quad \int_{\lfloor y^{\alpha_\epsilon(y)}\rfloor - 1}^{y^{\log y}} G(x,w)\, d\Psi(w-1,y) \\ &\ll \quad \int_{y^{\alpha_\epsilon(y)}}^{y^{\log y}} \frac{x}{w\log x}(\log w)\, d\Psi(w-1,y). \end{aligned}$$

Lemma 3.3 and (13) then gives

$$\begin{aligned} S_1 \quad &\ll \quad \int_{y^{\alpha_\epsilon(y)}}^{y^{\log y}} \frac{x}{w\log x}(\log w) d\left[(w-1)\rho\left(\frac{\log w}{\log y}\right)\right] \\ &\quad + \quad o\left(\frac{x}{\log x}\cdot\left(\alpha_\epsilon(y)(\log y)\rho(\alpha_\epsilon(y)) + (\log y)^2\rho(\log y)\right)\right) \\ &\ll \quad \frac{x}{\log x}\int_{y^{\alpha_\epsilon(y)}}^{y^{\log y}} \frac{\log w}{w}\rho\left(\frac{\log w}{\log y}\right) dw + o\left(\frac{x}{\log x\log y}\right) \end{aligned} \tag{26}$$

using Lemma 3.6. Substituting $u = \log w/\log y$ into the integral above gives

$$\begin{aligned}\int_{y^{\alpha_\epsilon(y)}}^{y^{\log y}} \frac{\log w}{w}\rho\left(\frac{\log w}{\log y}\right) dw &= (\log y)^2 \int_{\alpha_\epsilon(y)}^{\log y} u\rho(u)du \quad \leq \quad (\log y)^3 \int_{\alpha_\epsilon(y)}^{\log y} \rho(u)du \\ &\leq (\log y)^3 \rho(\alpha_\epsilon(y)) = o(1)\end{aligned}$$

using Lemma 3.6 and (18). Plugging into (26) we have

$$S_1 = o(x/\log x) = o\left(\frac{x}{\log x}\log y\right). \tag{27}$$

Putting together (23), (24), (25), and (27), we have $T_2 = o((x/\log x)\log y)$, as desired, which completes the proof. □

Later on we show how to tighten this upper bound using heuristics.

3.6 A Lower Bound.

We now prove a lower bound for $F(x, t, PM)$.

Our upper bound was based on two approximations for $\Pi(x, y)$; essentially, we used $\Pi(x, y) \leq \Psi(x, y)$ and $\Pi(x, y) \leq \pi(x)$. To prove a lower bound for F, (which implies proving a lower bound for Λ) we need a lower bound for $\Pi(x, y)$. We will use the following result by Pomerance [Pom80]:

$$\begin{aligned}\Pi(x, \sqrt{x}) &\geq (1 - 4\log(5/4) + o(1))\pi(x) \\ &\approx 0.10742 \cdot \pi(x)\end{aligned} \tag{28}$$

This result generalizes to counting smooth numbers of the form $p + a$ where p is prime and $a \neq 0$ is any fixed integer (see Goldfeld [Gol69] and Hooley [Hoo73]). In other words, if we generalize our definition of Π as follows,

$$\Pi(x, y, a) = \#\{p \leq x \ : \ p \ \text{ is prime and } \ P(p + a) \leq y\},$$

then we have

$$\Pi(x, \sqrt{x}, a) \geq (1 - 4\log(5/4) + o(1))\pi(x).$$

for $a \neq 0$ a fixed integer. We will use this later in our lower bound proof for the $p+1$ method.

As Pomerance points out [Pom88], many researchers have been working on lower bounds of the form $\Pi(x, x^{1/a}) \geq b\pi(x)$, and it seems likely that improved results will appear in the future. With this in mind, we will leave a and b as parameters in our lower bound results.

In using Pomerance's result, notice that we are limiting our count of integers n to those with their second largest prime divisor p with $P(p - 1) \leq B$ and $p \leq B^2$. Even though most of the integers factorable by the $p - 1$ method have a small second largest prime divisor, we would expect to get a much tighter lower bound by counting integers n with p ranging all the way up to $\sqrt{n}$. And yet, our crude estimate will be good enough to place $F(x, t, PM)$ higher than that of trial division.

Before we proceed to our lower bound for Λ, we state the following well-known theorem. Let $Li(x) = \int_2^x (\log t)^{-1} dt$.

Theorem 3.9 (Prime Number Theorem) *$\pi(x) \sim Li(x) \sim x/\log x$.*

Proof: For a proof, see Davenport [Dav80, chapter 18]. □

Lemma 3.10 *Let $\delta > 0$, and $a > 1$, $b > 0$ such that $\Pi(x, x^{1/a}) \geq b\pi(x)$. If $\log x/\log y \to \infty$ and $y \geq (\log x)^{1+\delta}$ then*

$$\Lambda(x,y,y) \quad \geq \quad (1 + b\log a + o(1))e^{\gamma}\frac{x}{\log x}\log y.$$

Proof: Like in the proof of Lemma 3.7, we start by defining a new function $H(x,y,p)$:

$$H(x,y,p) \quad = \quad \#\left\{n \leq x \ : \ P_2(n) = p \ \text{ and } \ P_3(n) \leq y\right\}.$$

H satisfies $H(x,y,1) = \pi(x)$, $H(x,x,p) = G(x,p)$, $H(x,y,z) = 0$ for $z > 1$ not prime, and

$$\sum_{z=1}^{y} H(x,y,z) \quad = \quad \Psi_2(x,y). \tag{29}$$

Then by definition, for $y \leq p$ we see that,

$$H(x,y,p) \ \geq \ \Psi_2\left(\frac{x}{p}, y\right) - \Psi\left(\frac{x}{p}, p\right). \tag{30}$$

Using the definition of Λ, by (29) and (30) we have

$$\begin{aligned}
\Lambda(x,y,y) \ &\geq \ \Psi_2(x,y) + \sum_{y<p\leq y^a,\ P(p-1)\leq y} H(x,y,p) \\
&\geq \ \Psi_2(x,y) + \sum_{y<p\leq y^a,\ P(p-1)\leq y} \Psi_2\left(\frac{x}{p}, y\right) - \sum_{y<p\leq y^a,\ P(p-1)\leq y} \Psi\left(\frac{x}{p}, p\right) \\
&= \ T_1 + T_2 - T_3.
\end{aligned}$$

Using (14) and (11) we have

$$T_1 = \Psi_2(x,y) \quad = \quad e^{\gamma}\frac{x}{\log x}(\log y)(1 + o(1)).$$

For T_2, we use (14) and (11) and noticing that $\log x \geq \log(x/p)$ gives

$$T_2 \quad \geq \quad \sum_{y<p\leq y^a,\ P(p-1)\leq y} e^{\gamma} \cdot \frac{1}{p} \cdot \frac{x}{\log x}(\log y)(1 + o(1)).$$

To evaluate this sum, we focus on the $1/p$ part, since the rest does not depend on p, the index of summation. Using $\pi(w) \geq \Pi(w,y) \geq \Pi(w, w^{1/a}) \geq b\pi(w)$, Lemma 3.3, the prime number theorem, and the formula $\sum_{p\leq x} 1/p = \log\log x + C + o(1)$ where C is constant (see Hardy and Wright [HW79]), we have

$$\begin{aligned}
\sum_{y<p\leq y^a,\ P(p-1)\leq y} \frac{1}{p} &= \int_y^{y^a} \frac{1}{w} d\Pi(w,y) \\
&\geq \int_y^{y^a} \frac{b}{w} d\pi(w) + O\left(\frac{1}{y}\Pi(y,y) + \frac{1}{y^a}\Pi(y^a,y)\right) \\
&= \sum_{y<p\leq y^a} \frac{b}{p} + O\left(\frac{1}{\log y}\right) \\
&= b(\log a) + o(1).
\end{aligned}$$

Thus, we have

$$T_2 \quad \geq \quad b(\log a)e^{\gamma}\frac{x}{\log x}(\log y)(1+o(1)).$$

It remains to show that T_3 is of lower order. Using the identity

$$\Psi(x,y) \quad = \quad \Psi(x,w) + \sum_{w<p\leq y} \Psi\left(\frac{x}{p}, p\right)$$

which holds for $w \geq 2$, we have

$$\begin{aligned}
T_3 \leq \Psi(x,y^a) &\leq x\rho\left(\frac{\log x}{a\log y}\right)(1+o(1)) \\
&\ll x\left(\frac{a\log y}{\log x}\right)^2 \rho\left(\frac{\log x}{a\log y} - 2\right) \\
&= o\left(x\frac{a^2(\log y)^2}{(\log x)^2}\right) \\
&= o((x/\log x)\log y)
\end{aligned}$$

by (12), two applications of (16), and the hypothesis that $\log x/\log y \to \infty$. □

Note that $1 + (1 - 4\log(5/4))(\log 2) = 1.07446\cdots$.

Theorem 3.11 *Let $\delta > 0$ and $a > 1$, $b > 0$ such that $\Pi(x, x^{1/a}) \geq b\pi(x)$. If $t \geq (\log x)^{2+\delta}$ and $\log x/\log t \to \infty$, then*

$$F(x,t,PM) \quad \geq \quad (1 + b\log a + o(1))e^{\gamma}\frac{x}{\log x}\left(\log\frac{t}{\log x}\right).$$

Proof: Follows immediately from Lemma 3.1 and Lemma 3.10. □

Applying Pomerance's result (28) gives (4) for the $p-1$ method.

3.7 Heuristics.

Our results up to this point have relied on somewhat crude approximations of the asymptotic behavior of the function $\Pi(x,y)$. Essentially, we use the inequalities $\Pi(x,y) \leq \Psi(x,y)$, $\Pi(x,y) \leq \pi(x)$, and $\Pi(x,x^{1/a}) \geq b\pi(x)$ for a and b certain fixed constants. Although these bounds are correct, they are not very tight. Suppose we had a much better approximation for Π. How much would this improve our results for $F(x,t,PM)$? We will give a reasonable heuristic estimate of the asymptotic behavior of $\Pi(x,y)$ that leads to upper and lower bounds for $F(x,t,PM)$ that are within a small (< 2) constant multiple of one another.

Since $\Pi(x,y)$ counts primes p such that $p-1$ has all of its prime divisors less than y, and $\Psi(x,y)$ counts all integers n such that all of n's prime divisors are less than y, it makes sense to approximate $\Pi(x,y)/\pi(x)$ as $\Psi(x,y)/x$. In other words, let us assume the following.

Hypothesis 3.12 $\Pi(x,y) \sim \pi(x)\rho(u)$, *where* $u = \log x/\log y$.

This was conjectured by Pomerance [Pom80, Pom88] and no doubt by others as well.

We show below that this heuristic leads to upper and lower bounds on $F(x,t,PM)$ that are within a small constant multiple of one another.

Actually, Hypothesis 3.12 is stronger than we need; we weaken it to the following.

Hypothesis 3.13 *Let* $\epsilon > 0$ *be fixed. Then* $\Pi(x,y) \sim \pi(x)\rho(u)$ *for* $1 \leq u \leq \alpha_\epsilon(y)$, *where* $u = \log x/\log y$.

We use this to prove the following bounds on Λ.

Lemma 3.14 *Let* ϵ, $\delta > 0$. *If* $\log x/(\alpha_\epsilon(y)\log y) \to \infty$, $y \geq (\log x)^{1+\delta}$, *and* $\sqrt{\log x}/\log y$ *is well-behaved, then Hypothesis 3.13 implies*

$$\Lambda(x,y) \quad \leq \quad e^{2\gamma}\frac{x}{\log x}(\log y)(1+o(1)).$$

Proof: Similar to the proof of Lemma 3.7, we write $\Lambda(x,y) = T_1 + T_2$ where

$$\begin{aligned} T_1 &= G(x,1) + \sum_{p \leq y^{\alpha_\epsilon(y)}} G(x,p)\Delta_p\Pi(p,y) \\ T_2 &= \sum_{y^{\alpha_\epsilon(y)} < p \leq \sqrt{x}} G(x,p)\Delta_p\Pi(p,y). \end{aligned}$$

By the proof of Lemma 3.7, we have $T_2 = o((x/\log x)\log y)$.

It remains to compute an upper bound for T_1. By the prime number theorem, $\pi(x) \sim Li(x)$, and using Hypothesis 3.13, Lemma 3.3, and (22), we have

$$\begin{aligned} T_1 &= \int_1^{y^{\alpha_\epsilon(y)}} G(x,w)d\Pi(w,y) \\ &= \Psi_2(x,y) + \int_y^{y^{\alpha_\epsilon(y)}} G(x,w)d\Pi(w,y) \\ &\sim \Psi_2(x,y) + \int_y^{y^{\alpha_\epsilon(y)}} G(x,w)d\left(Li(w)\rho\left(\frac{\log w}{\log y}\right)\right) \\ &+ \quad o\left(\Pi(y,y)G(x,y) + \Pi(y^{\alpha_\epsilon(y)},y)G(x,y^{\alpha_\epsilon(y)})\right). \end{aligned}$$

Since $\pi(w) \geq \Pi(w,y)$ and $G(x,w) \leq e^\gamma (x/w \log(x/w)) \log w(1+o(1))$ using (21), (14), and (11), the $o(\cdot)$ term is $o(x/\log x)$. This then gives us

$$\begin{aligned} T_1 &\leq e^\gamma \frac{x}{\log x}(\log y)(1+o(1)) + \int_y^{y^{\alpha_\epsilon(y)}} e^\gamma \frac{x}{w \log x}(\log w)\cdot(1+o(1))\cdot \frac{1}{\log w}\rho\left(\frac{\log w}{\log y}\right) dw \\ &\sim e^\gamma \frac{x}{\log x}(\log y)\left[1+\int_1^{\alpha_\epsilon(y)} \rho(u)du\right] \\ &\sim (e^\gamma)e^\gamma \frac{x}{\log x}(\log y) \end{aligned}$$

by substituting $u = \log w/\log y$ and by (16), (19), and Lemma 3.6. □

Lemma 3.15 *Let $\epsilon, \delta > 0$. If $y \geq (\log x)^{1+\delta}$, and $\log x/\log y \to \infty$, then Hypothesis 3.13 implies*

$$\Lambda(x,y,y) \quad \geq \quad (2-\log 2)e^\gamma \frac{x}{\log x}(\log y)(1+o(1)).$$

Proof: Let β be a positive, real-valued function such that $\beta(y) \to \infty$ as $y \to \infty$, $\beta \leq \alpha_\epsilon$, and $\beta \leq (\log x/\log y)^{1/3}$. Following the proof of Lemma 3.10, we have

$$\begin{aligned} \Lambda(x,y,y) &\geq \Psi_2(x,y) + \sum_{y<p\leq y^{\beta(y)},\ P(p-1)\leq y} \left(\Psi_2\left(\frac{x}{p},y\right) - \Psi\left(\frac{x}{p},p\right)\right) \\ &= T_1 + T_2 - T_3. \end{aligned}$$

From the proof of Lemma 3.10, plugging in $\beta(y)$ for a gives us both

$$\begin{aligned} T_1 &= e^\gamma \frac{x}{\log x}(\log y)(1+o(1)) \quad \text{and} \\ T_3 &= o((x/\log x)\log y), \end{aligned}$$

allowing us to focus on T_2.

Again, from the proof of Lemma 3.10, we have

$$T_2 \;\geq \sum_{y<p\leq y^{\beta(y)},\ P(p-1)\leq y} e^\gamma \frac{x}{p\log x}(\log y)(1+o(1)). \tag{31}$$

We need to evaluate the sum on $1/p$. Using Hypothesis 3.13 and Lemma 3.3, we have

$$\begin{aligned} \sum_{y<p\leq y^{\beta(y)},\ P(p-1)\leq y} \frac{1}{p} &= \int_y^{y^{\beta(y)}} \frac{1}{w} d\Pi(w,y) \\ &\sim \int_y^{y^{\beta(y)}} \frac{1}{w} d\left(Li(w)\rho\left(\frac{\log w}{\log y}\right)\right) \quad + \quad o\left(\frac{1}{y}\Pi(y,y) + \frac{1}{y^{\beta(y)}}\Pi(y^{\beta(y)},y)\right). \end{aligned}$$

Bounding $\Pi(w,y)$ with $\pi(w)$, differentiating, using (9), and noticing that $Li(w) = O(w/\log w)$ gives

$$\geq \int_y^{y^{\beta(y)}} \frac{1}{w}\left[\frac{\rho(\log w/\log y)}{\log w} - O\left(\frac{\rho(\log w/\log y - 1)}{\log^2 w}\right)\right] dw \quad + \quad o(1/\log y).$$

Substituting u for $\log w / \log y$ and using (18) gives

$$= \int_1^{\beta(y)} \left[\frac{\rho(u)}{u} - O\left(\frac{\rho(u-1)}{u^2 \log y} \right) \right] du \quad + \quad o(1/\log y)$$
$$= \int_1^{\beta(y)} \frac{\rho(u)}{u} du \quad - \quad O(1/\log y).$$

Since $\rho(u)/u \geq \rho(u)/(u+1) = -\rho'(u+1)$ by (9) and (16), this is

$$\geq \ \rho(2) - \rho(\beta(y)+1) - O(1/\log y) \quad \sim \quad \rho(2) = 1 - \log 2.$$

Plugging back into (31) completes the proof. □

Notice that $2 - \log 2 = 1.30685 \cdots$.

The above bounds on Λ give the following improved results for $F(x, t, PM)$.

Theorem 3.16 *Let $\epsilon, \delta > 0$. If $t \geq (\log x)^{2+\delta}$, $\log t \leq (\log x)^{1-\delta}$, and $\sqrt{\log x}/\log t$ is well-behaved, then Hypothesis 3.13 implies*

$$F(x,t,PM) \ \leq \ 1.78108 \cdot e^{\gamma} \frac{x}{\log x} \left(\log \frac{t}{\log x} \right) (1 + o(1)) \quad \text{and}$$
$$F(x,t,PM) \ \geq \ 1.30685 \cdot e^{\gamma} \frac{x}{\log x} \left(\log \frac{t}{\log x} \right) (1 + o(1)).$$

Proof: Follows from Lemmas 3.1, 3.14, 3.15, Lemma 3.5, Theorem 3.2, and the fact that $e^{\gamma} = 1.78107421 \cdots$. □

This proves (5) and (6) for the $p-1$ method, and concludes our results for the $p-1$ integer factoring algorithm.

4 The $p+1$ Integer Factoring Algorithm.

We now will prove results for the $p+1$ factoring algorithm. Let PP denote this algorithm with an initial trial division phase and a probabilistic primality test. We start by briefly describing the algorithm and estimating its running time. Perhaps the most important observation to make about the $p+1$ algorithm, for our purposes, is that it removes prime divisors p such that $p+1$ is smooth.

We will use the same approach here as we did in section 3 for the $p-1$ method; we define functions $\Lambda_2(x,y,z)$ and $V_2(x,y)$, where Λ_2 counts the "typical" integers factorable by the $p+1$ method, and S and V_2 count the rest (S is the same as before). As a result, the work we did in section 3 on the $p-1$ method carries over almost word-for-word, making our proofs in this section quite short.

4.1 The Algorithm.

The following description of the $p+1$ method in terms of finite fields is from Bach and Shallit [BS89a]. Williams also gives a nice description of the $p+1$ method, but in terms of recurrences [Wil82]. Our goal here is to give a general idea of how this algorithm works, and to point out the similarities between this method and the $p-1$ method.

Let n be the integer we wish to factor, and assume that $n = p \cdot q$ where $p \leq \sqrt{n}$ is prime and q is relatively prime to p. Choose a bound B, and let $p_1 \ldots p_r$ be the primes less than B. Let e_i be the largest integer such that $p_i^{e_i} \leq \sqrt{n}$. Let $E = \prod_{i=1}^{r} p_i^{e_i}$.

If $p+1 \mid E$, that is, if $p+1$ factors completely over the primes below B, the following procedure is likely to split n:

```
Choose $a$, and $b$ uniformly at random from $\mathbf{Z}/(n)$;
Choose $d$ uniformly at random from $(\mathbf{Z}/(n))^*$;
$s := a + b\sqrt{d}$;
By rationalizing the denominator, compute
    $w := \bar{s}/s \bmod n$;
Compute $u$ and $v$ such that
    $w^E \equiv u + v\sqrt{d} \pmod{n}$;
Try to split $n$ with $\gcd(u-1, v, n)$;
```

Computation occurs within the ring $\mathbf{Z}[\sqrt{d}]/(n)$. By the Chinese Remainder Theorem for commutative rings, this is isomorphic to $(\mathbf{Z}[\sqrt{d}]/(p)) \oplus (\mathbf{Z}[\sqrt{d}]/(q))$. If d is a quadratic nonresidue modulo p, then we have $\mathbf{Z}[\sqrt{d}]/(p) \simeq GF(p^2)$, where $GF(p^2)$ is the finite field of order r. Notice $GF(p^2)^*$ has order $p^2 - 1 = (p-1)(p+1)$. w is constructed so that w is likely to have norm 1; this gives $w^{p+1} \equiv 1$ modulo p. Since $p+1 \mid E$ and any power of 1 is 1, we have $u \equiv 1$ and $v \equiv 0$ modulo p. If nothing unusual happens modulo q, which is likely, we split n.

As we did with the $p-1$ method, we will precede this procedure with trial division up to B, incorporate probabilistic primality tests and perfect power tests. We also place the last two steps above in a loop as follows:

```
For $i := 1$ to $r$ Do:
    $w := w^{p_i^{e_i}}$ in $\mathbf{Z}[\sqrt{d}]/(n)$;
    Compute $u$ and $v$ such that
        $w \equiv u + v\sqrt{d} \pmod n$;
    If $d := \gcd(u-1, v, n)$ is a proper divisor $n$,
        $n := n/d$, and output $d$;
```

The main loop of the algorithm will take $O(B \log n)$ operations, since $\log E = O(\pi(B) \log n)$ and computing $w^E \bmod n$ dominates the running time. As in the $p-1$ algorithm, the total number of operations spent in trial division is $O(B)$, in finding the primes is $O(B \log\log n)$, and in prime tests and perfect power tests is $O(\log^2 n \log\log n)$. Thus, the total expected number of arithmetic operations used by PP is $O(B \log n + \log^2 n \log\log n)$. If we are limited to t operations, we choose $B = \Theta(t/\log n)$, just as in the $p-1$ method.

Again, the $p+1$ method has two extensions, a standard extension, which is described by Williams [Wil82], and an FFT extension which is described by Montgomery and Silverman [MS88]. We do not consider these extensions for the same reasons as in the case of the $p-1$ algorithm. For more details on the $p+1$ method of factorization, see Guy [Guy75], Montgomery [Mon87], Montgomery and Silverman [MS88], Bach and Shallit [BS89a], and Williams [Wil82]. For more on finite fields, see Lang [Lan71, chapter VII §5].

4.2 Upper and Lower Bounds for $F(x, t, PP)$.

We now define

$$\Lambda_2(x, y, z) \quad = \quad \#\left\{ n \le x \;:\; P(P_2(n) + 1) \le y \;\text{ and }\; P_3(n) \le z \right\}.$$

If we leave off the third argument z, assume it is equal to x, that is, $\Lambda_2(x, y) := \Lambda_2(x, y, x)$. Also, we define

$$V_2(x, y) \quad = \quad \#\left\{ n \le x \;:\; P(P(n) + 1) \le y \right\}.$$

In other words, $V_2(x, y)$ denotes how many integers n below x have the property that the $p+1$ algorithm can remove n's largest prime divisor.

Lemma 4.1 *Let $B = B(x, t) = \Theta(t/\log x)$ be the bound in algorithm PP. Then*

$$\begin{aligned} F(x, t, PP) &\le \Lambda_2(x, B) + S(x) + V_2(x, B) \quad \textit{and} \\ F(x, t, PP) &\ge \Lambda_2(x, B, B) \end{aligned}$$

Proof: Identical to the proof of Lemma 3.1. □

We now define $\Pi_2(x, y)$ to be the number of integers of the form $p+1$ for p a prime, all of whose prime divisors are at most y:

$$\Pi_2(x, y) \quad = \quad \#\left\{ p \le x \;:\; p \text{ is prime, and } P(p + 1) \le y \right\}.$$

As we mentioned in section 3, Pomerance's result (28) applies to $\Pi_2(x, y)$, and we make use of that below. There does not seem to be any special reason for $\Pi_2(x, y)$ to differ from $\Pi(x, y)$ by very much asymptotically. This also seems to be the case in practice, as our computational results show in section 6. So, we make the following heuristic assumption.

Hypothesis 4.2 *Let $\epsilon > 0$ be fixed. Then $\Pi_2(x,y) \sim \pi(x)\rho(u)$, where $u = \log x/\log y$ with $1 \le u \le \alpha_\epsilon(y)$.*

Lemma 4.3 *Let $\epsilon, \delta > 0$ be fixed and let $a > 1$, $b > 0$ such that $\Pi_2(x, x^{1/a}) \ge b\pi(x)$. If $\log x/(\alpha_\epsilon(y)\log y) \to \infty$, $y \ge (\log x)^{1+\delta}$, and $\sqrt{\log x}/\log y$ is well-behaved, then*

$$\begin{aligned} \Lambda_2(x,y) &\le (3+\epsilon)e^\gamma \frac{x}{\log x}\alpha_\epsilon(y)(\log y)(1+o(1)) \quad \text{and} \\ \Lambda_2(x,y,y) &\ge (1+b\log a)e^\gamma \frac{x}{\log x}(\log y)(1+o(1)). \end{aligned}$$

Further, if Hypothesis 4.2 holds, then

$$\begin{aligned} \Lambda_2(x,y) &\le e^{2\gamma} \frac{x}{\log x}(\log y)(1+o(1)) \quad \text{and} \\ \Lambda_2(x,y,y) &\ge (2-\log 2)e^\gamma \frac{x}{\log x}(\log y)(1+o(1)). \end{aligned}$$

Proof: Using definitions, we have the following identities:

$$\begin{aligned} \Lambda_2(x,y) &= G(x,1) + \sum_{p \le \sqrt{x},\ P(p+1) \le y} G(x,p), \\ \Lambda_2(x,y,y) &\ge H(x,y,1) + \sum_{p \le \sqrt{x},\ P(p+1) \le y} H(x,y,p). \end{aligned}$$

We substitute $\Pi_2(x,y)$ in for $\Pi(x,y)$ in the proofs of Lemmas 3.7, 3.10, 3.14, and 3.15, and substitute Hypothesis 4.2 for Hypothesis 3.13. The results follow. □

Lemma 4.4 *Let $\delta > 0$. If $\log x/\log y \to \infty$ and $y \ge (\log x)^{1+\delta}$, then*

$$V_2(x,y) = O\left(x(\log y)\rho\left(\sqrt{\frac{\log x}{\log y}} - 1\right)\right).$$

Further, if $\log y \le (\log x)^{1-\delta}$, then

$$V_2(x,y) \le x \cdot \exp\left[-\Omega\left((\log x)^{\delta/2}\log\log x\right)\right].$$

Proof: The proof is essentially the same as the proof of Lemma 3.5. □

Theorem 4.5 *Let $\epsilon, \delta > 0$ be fixed, and let $a > 1$, $b > 0$ such that $\Pi_2(x, x^{1/a}) \ge b\pi(x)$. If $t \ge (\log x)^{2+\delta}$, $\log t \le (\log x)^{1-\delta}$, and $\sqrt{\log x}/\log t$ is well-behaved, then*

$$\begin{aligned} F(x,t,PP) &\le (3+\epsilon)e^\gamma \frac{x}{\log x}\left(\log\frac{t}{\log x}\right)\frac{\log\log t}{\log\log\log t}(1+o(1)) \quad \text{and} \\ F(x,t,PP) &\ge (1+b\log a)\cdot e^\gamma \frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)). \end{aligned}$$

Further, Hypothesis 4.2 implies

$$\begin{aligned} F(x,t,PP) &\le 1.78108 \cdot e^\gamma \frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)) \quad \text{and} \\ F(x,t,PP) &\ge 1.30685 \cdot e^\gamma \frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)). \end{aligned}$$

Proof: Follows from Lemmas 4.1 and 4.3, Lemma 4.4, and Theorem 3.2. □

Using (28), this proves (3), (4), (5), and (6) for the $p+1$ integer factoring algorithm.

5 Bach and Shallit's $\Phi_k(p)$ Integer Factoring Algorithm.

In this section we will prove an upper bound on $F(x,t,C_k)$, where C_k is the kth cyclotomic polynomial integer factoring algorithm, or the $\Phi_k(p)$ method. The kth cyclotomic polynomial $\Phi_k(x)$ is the unique monic polynomial with integer coefficients whose roots are precisely the kth primitive roots of 1 in the complex numbers. Perhaps the most important thing to note is that the $\Phi_k(p)$ method removes prime divisors p from an integer n with high probability if $\Phi_k(p)$, the kth cyclotomic polynomial evaluated at p, is smooth.

As we mentioned in the introduction, Bach and Shallit's algorithm is a generalization of many other factoring algorithms. For example, since $\Phi_1(x) = x - 1$ and $\Phi_2(x) = x + 1$, C_1 corresponds to the $p-1$ method and C_2 to the $p+1$ method; in previous sections we analyzed both of these methods. Williams and Judd's algorithms [WJ76a, WJ76b] correspond to $k = 3,4,6$. Bach, Miller, and Shallit's sums of divisors algorithm [BMS86] corresponds to k a prime power. Because the $\Phi_k(p)$ method is a generalization of the $p \pm 1$ methods, we can generalize our techniques from previous sections to give a heuristic upper bound for $F(x,t,C_k)$. So we will introduce the functions $\Lambda_k(x,y,z)$ and $V_k(x,y)$ as before, and bound them by the use of a function $\Pi_k(x,y)$ which counts primes below x such that $\Phi_k(p)$ is y-smooth.

Suppose we construct an algorithm that factors integers by trying the C_k method for all values of k up to a bound l. Call this the $MC(l)$ algorithm for the multiple cyclotomic polynomial method. Bach and Shallit [BS89a] called integers susceptible to factorization by $MC(l)$ *vulnerable numbers.* We also give a heuristic upper bound on $F(x,t,MC(l))$, the number of vulnerable numbers. For this bound, we use the functions $\Lambda^*(x,y,l)$, $V^*(x,y,l)$, and $\Pi^*(x,y,l)$.

We start by giving the general idea of how the $\Phi_k(p)$ factoring algorithm works. This method is quite complicated, so we do not attempt to describe it in complete detail. For that, we refer the reader to Bach and Shallit's paper [BS89a]. We then prove our upper bound for $F(x,t,C_k)$. We conclude with our result on the multiple cyclotomic polynomial method.

5.1 The Algorithm.

The following description of the $\Phi_k(p)$ method is not meant to be complete or rigorous. Our goal is to give the reader an idea of how the algorithm works and to show the similarities between this algorithm and those discussed in previous sections. For complete details, see Bach and Shallit [BS89a].

Let n be the integer we wish to factor, and for simplicity assume that $n = p \cdot q$ where p and q are prime with $p < q$. Choose a bound B, and let $p_1 \dots p_r$ be the primes less than B. Let e_i be the largest integer such that $p_i^{e_i} \le \Phi_k(\sqrt{n}) \sim n^{\phi(k)/2}$. Let $E = \prod_{i=1}^{r} p_i^{e_i}$. Notice that $\log E = O(\pi(B)\phi(k)\log n)$.

Before giving a pseudocode description, we must make a few definitions.

For a prime m, $m \equiv 1 \pmod{k}$, we define the ring R_m as follows. Let K_m be the extension field of degree k over $\mathbf{Q}$, the rationals, inside the mth cyclotomic extension field of degree

$m-1$ over $\mathbf{Q}$. Let O_m be the ring of integers in K_m. Then R_m is $O_m/(n)$. We omit the details on how to represent R_m for computation.

Let ψ_k be the polynomial defined by $\psi_k(x) := (x^k - 1)/\Phi_k(x)$.

We use $u^{f(\sigma)}$ to denote a symbolic power; for example $u^{\sigma^2-1} = u^{\sigma^2}/u$.

Also, if $\alpha \in R_m$, define $\gcd^*(\alpha, n)$ to be the greatest common divisor of pairs of coefficients from a basis representation of α in R_m.

If $\Phi_k(p) \mid E$, that is, if $\Phi_k(p)$ factors completely over the primes below B, then the following procedure is likely to split n.

```
Repeat:
    Choose $m$ a prime, with $m \equiv 1 \pmod{k}$;
    Construct the ring $R_m$ with automorphism $\sigma$,
        were $\sigma$ is a generator for the Galois group $\mathrm{Gal}(K_m/\mathbf{Q})$, pulled down to $R_m$
    Choose $s \in R_m$ at random;
    $w := s^E$;
    For $i \in (\mathbf{Z}/(k))^*$ Do:
        $\tau := \sigma^i$;
        $v := w^{\psi_k(\tau)}$;
        $d := \gcd^*(v-1, n)$;
        Check to see if $d$ is a proper divisor of $n$;
    End for;
Until $n$ splits (or time runs out);
```

We hope that p remains prime in O_m and that q splits completely. If this happens, then R_m mod p is the finite field $GF(p^k)$, and some power τ of σ is the Frobenius automorphism on $GF(p^k)$, giving

$$w^{E\psi_k(\tau)} \equiv w^{\Phi_k(p)\psi_k(p)\cdot(E/\Phi_k(p))} \equiv w^{(p^k-1)\cdot(E/\Phi_k(p))} \equiv 1^{E/\Phi_k(p)} \equiv 1$$

allowing $\gcd^*(v-1,n)$ to split n. A form of the generalized Riemann hypothesis (GRH) guarantees the existence of a small prime m with the required properties.

As we did for the $p \pm 1$ methods, we make some modifications to the procedure above. We incorporate a preliminary trial division step, add perfect power and probabilistic primality tests after divisors are found, and we break down the computation of w^E into a loop that, on the ith iteration, raises w to the power $p_i^{e_i}$ and then checks for a divisor.

Bach and Shallit [BS89a] analyzed the complexity of this algorithm. They showed that one pass through the main repeat loop above uses a number of arithmetic operations bounded by a polynomial in m, k, $\log n$, and $\log E$.

We are concerned with an upper bound on $F(x,t,C_k)$. So it suffices to find an upper bound on B in terms of t, the number of arithmetic operations allowed in running the algorithm. It is not difficult to see that $O(t/\log n)$ is an overestimate for B, so this is the bound we will use.

5.2 Results for $F(x,t,C_k)$.

Following the same pattern as in previous sections, we define

$$\Lambda_k(x,y,z) \;=\; \#\{n \le x \;:\; P(\Phi_k(P_2(n))) \le y \text{ and } P_3(n) \le z\}$$

and we if we omit z, we assume it is x: $\Lambda_k(x,y) := \Lambda_k(x,y,x)$. Similarly, we define

$$V_k(x,y,z) \;=\; \#\{n \le x \;:\; P(\Phi_k(P(n))) \le y\}.$$

Note that for $k = 1, 2$, these definitions reduce to our definitions of Λ and Λ_2, and V and V_2 from previous sections. The following lemma bounds $F(x,t,C_k)$ in terms of Λ_k, S, and V_k.

Lemma 5.1 *Let $B = B(x,t,k) = \Theta(t/\log n)$ be the bound in algorithm C_k. Then*

$$F(x,t,C_k) \;\le\; \Lambda_k(x,B) + S(x) + V_k(x,B).$$

Proof: Follows from the discussion above and the proof of Lemma 3.1. □

We also generalize our definition for $\Pi(x,y)$ as follows:

$$\Pi_k(x,y) \;=\; \#\{p \le x \;:\; p \text{ is prime, and } P(\Phi_k(p)) \le y\}.$$

Little is known about the order of magnitude of $\Pi_k(x,y)$. In order to derive anything, we must make some sort of an assumption in this area. It seems reasonable to assume that $\Phi_k(p)$ has all small prime divisors with probability no greater than a randomly chosen integer of the same size. Since the degree of the polynomial $\Phi_k(x)$ is $\phi(k)$, $\Phi_k(p)$ is roughly $p^{\phi(k)}$ in magnitude. This leads us to the following hypothesis.

Hypothesis 5.2 $\Pi_k(x,y) \le \pi(x)\rho(\phi(k)\cdot u)(1+o(1))$, *where* $u = \log x/\log y$.

Now this heuristic for Π_k ignores many of the special properties of cyclotomic polynomials. For example, if a prime $q \mid \Phi_k(p)$ and $q > k$, then $q \equiv 1 \pmod{k}$. However, computations made by the author indicate that Hypothesis 5.2 is probably correct. We discuss the results of these computations in section 6.

We now use this heuristic to bound $V_k(x,y,l)$.

Lemma 5.3 *Let $\delta > 0$. If $\log x/\log y \to \infty$ and $y \ge (\log x)^{1+\delta}$, then Hypothesis 5.2 implies*

$$\begin{aligned} V_k(x,y) &\le O(y\Psi(x/y,y)) + x(\log y)D_k(\log x/\log y)\cdot(1+o(1)) \\ &= O\left(x\rho(\log x/\log y - 1) + x(\log y)\rho\left(\sqrt{\phi(k)\frac{\log x}{\log y}} - 1\right)\right) \end{aligned}$$

Further, if $\log y \le (\log x)^{1-\delta}$ and $\phi(k) \le \log x/\log y$ then Hypothesis 5.2 implies

$$V_k(x,y) \;=\; x\cdot\exp\left[-\Omega\left(\sqrt{\phi(k)}(\log x)^{\delta/2}\log\log x\right)\right].$$

Proof: The proof follows from the proof of Lemma 3.5 by the appropriate substitution of Π_k for Ψ and the application of Hypothesis 5.2, and Lemma 3.4. □

We now give our bound for $\Lambda_k(x,y)$.

Lemma 5.4 *Let $\epsilon, \delta > 0$. If $y \ge (\log x)^{1+\delta}$, $\log x/(\alpha_\epsilon(y)\log y) \to \infty$, and $\sqrt{\log x}/\log y$ is well-behaved, then Hypothesis 5.2 implies*

$$\Lambda_k(x,y) \;\le\; \left(1 + \frac{\rho(\phi(k))}{\phi(k)}\right) e^\gamma \frac{x}{\log x}(\log y)(1+o(1)).$$

Proof: By our hypothesis and (12), since $\phi(k) \geq 1$, we have $\Pi_k(x,y) \leq \Psi(x,y)(1+o(1))$ and $\Pi_k(x,y) \leq \pi(x)$. By definition, (22), and Lemma 3.3 this gives

$$\begin{aligned}
\Lambda_k(x,y) &= G(x,1) + \sum_{p\leq\sqrt{x},\ P(\Phi_k(p))\leq y} G(x,p) \\
&\leq \Psi(x,y) + \int_y^{y^{\alpha_\epsilon(y)}} G(x,w)d\Pi_k(w,y) + \int_{y^{\alpha_\epsilon(y)}}^{\sqrt{x}} G(x,w)d\Psi(w,y) \\
&+ O\left(G(x,\sqrt{x})\Psi(\sqrt{x},y) + G(x,y^{\alpha_\epsilon(y)})\Psi(y^{\alpha_\epsilon(y)},y)\right) \\
&= T_1 + T_2 + T_3 + E.
\end{aligned}$$

From the proof of Lemma 3.7 $E = O(x/\log x)$. Using (14) and (11) we have

$$T_1 \sim e^\gamma \frac{x}{\log x} \log y.$$

By the proof of Lemma 3.7, we also have $T_3 = o((x/\log x)\log y)$. It remains to bound T_2.

By approximating $G(x,w) \leq e^\gamma(x/(w\log x))(\log w)(1+o(1))$ using (21), (14), and (11) gives

$$T_2 \leq e^\gamma \frac{x}{\log x}(\log y)(1+o(1)) \int_y^{y^{\alpha_\epsilon(y)}} \frac{\log w}{w\log y} d\Pi_k(w,y).$$

We then evaluate the integral using Hypothesis 5.2, the prime number theorem, Lemma 3.3, differentiating, and using (18):

$$\begin{aligned}
\int_y^{y^{\alpha_\epsilon(y)}} \frac{\log w}{w\log y} d\Pi_k(w,y) &= \int_y^{y^{\alpha_\epsilon(y)}} \rho\left(\phi(k)\frac{\log w}{\log y}\right)\frac{1+o(1)}{w\log y}dw + o\left(\frac{1}{\log y}\right) \\
&= \frac{1+o(1)}{\phi(k)} \int_{\phi(k)}^{\phi(k)\alpha_\epsilon(y)} \rho(u)du \quad \leq \quad \frac{\rho(\phi(k))}{\phi(k)}(1+o(1)).
\end{aligned}$$

That completes the proof. □

Using the previous two lemmas, we arrive at the following upper bound for $F(x,t,C_k)$.

Theorem 5.5 *Let $\epsilon, \delta > 0$. If $t \geq (\log x)^{2+\delta}$, $\log x/(\alpha_\epsilon(t)\log t) \to \infty$, and $\sqrt{\log x}/\log t$ is well-behave then Hypothesis 5.2 implies*

$$F(x,t,C_k) \leq \left(1 + \frac{\rho(\phi(k))}{\phi(k)}\right) e^\gamma \frac{x}{\log x}\left(\log\frac{t}{\log x}\right)(1+o(1)).$$

Proof: The results follows from Lemmas 5.1,5.4, Lemma 5.3, and Theorem 3.2. □

This proves (7).

5.3 A Multiple Cyclotomic Polynomial Method.

Finally, we consider an algorithm which performs trial division and then each $\Phi_k(p)$ method for $1 \leq k \leq l$ to a bound l. As we mentioned at the beginning of this section, we will call this algorithm $MC(l)$ for the multiple cyclotomic polynomial method. We will show that $MC(l)$

factors at most a constant multiple more integers than a method based on one cyclotomic polynomial.

Following our standard approach, we start by defining $\Lambda^*(x,y,l)$ and $V^*(x,y,l)$ as follows.

$$\begin{aligned}\Lambda^*(x,y,l) &= \#\{n \le x \ : \ \text{for some } k,\ 1 \le k \le l,\ P(\Phi_k(P_2(n))) \le y\}\\ V^*(x,y,l) &= \#\{n \le x \ : \ \text{for some } k,\ 1 \le k \le l,\ P(\Phi_k(P_1(n))) \le y\}\end{aligned}$$

Correlating Λ^* and $F(x,t,MC(l))$ is done with the following lemma.

Lemma 5.6 *Let $B = B(x,t,l) = O(t/\log x)$ be the bound for the multiple cyclotomic polynomial algorithm $MC(l)$. Then*

$$F(x,t,MC(l)) \ \le \ \Lambda^*(x,B,l) + S(x) + V^*(x,B,l).$$

Proof: Follows from the definitions above and the proof of Lemma 3.1. □

We also define $\Pi^*(x,y,l)$ as follows.

$$\Pi^*(x,y,l) \ = \ \#\{p \le x \ : \ p \text{ is prime, and } P(\Phi_k(p)) \le y \text{ for some } k,\ 1 \le k \le l\}.$$

The important thing to note at this point is that $\Delta_w \Pi^*(w,y,l) \ \le \ \Delta_w \pi(w)$ and $\Delta_w \Pi^*(w,y,l) \le \sum_{k=1}^{l} \Delta_w \Pi_k(w,y)$. We will use these two upper bounds, along with Hypothesis 5.2 to give upper bounds for V^* and Λ^*.

Before we proceed to give an upper bound for V^*, we need the following lemma.

Lemma 5.7

$$\sum_{k=1}^{\infty} D_k(z) \ = \ \rho(\Omega(\sqrt{z}))$$

The proof is rather technical, so we postpone it to the end of this section.

And now for V^*.

Lemma 5.8 *Let $\delta > 0$. If $\log x/\log y \to \infty$ and $y > (\log x)^{1+\delta}$ then Hypothesis 5.2 implies*

$$V^*(x,y,l) \ = \ x(\log y) \cdot \rho\left(\Omega(\sqrt{\log x/\log y})\right).$$

Further, if $\log y < (\log x)^{1-\delta}$, then Hypothesis 5.2 implies

$$V^*(x,y,l) \ = \ x \cdot \exp\left[-\Omega\left((\log x)^{\delta/2}\log\log x\right)\right].$$

Both are independent of l, and hold even if $l \to \infty$.

Proof: Again, following the proof of Lemma 3.5 and using Hypothesis 5.2, we have

$$\begin{aligned}V^*(x,y,l) &\le \Psi(x,y) + \sum_{w=y+1}^{x} \Psi(x/w,w)\Delta_w \Pi^*(w,y,l)\\ &\le \Psi(x,y) + \sum_{w=y+1}^{x} \Psi(x/w,w) \sum_{k=1}^{l} \Delta_w \Pi_k(w,y)\\ &= \Psi(x,y) + O(x\log y) \cdot \sum_{k=1}^{l} D_k(\log x/\log y).\end{aligned}$$

Lemma 5.7 and (12) gives the result. □

The following is our upper bound for Λ^*.

Lemma 5.9 *Let $\epsilon, \delta > 0$. If $y \geq (\log x)^{1+\delta}$, $\log x/(\alpha_\epsilon(y) \log y) \to \infty$, and $\sqrt{\log x}/\log y$ is well-behaved, then Hypothesis 5.2 implies*

$$\Lambda^*(x, y, l) \;=\; O\left(\frac{x \log y}{\log x}\right).$$

Further, this bound holds for $l \to \infty$.

Proof: Recall that $\Delta_w \Pi^*(w, y, l)$ is smaller than $\Delta_w \pi(w)$ and $\sum_{k=1}^{l} \Delta_w \Pi_k(w, y)$. By definition, (22), and Lemma 3.3, we have

$$\begin{aligned}\Lambda^*(x, y, l) \;&=\; G(x, 1) + \sum_{p \leq y} G(x, p) + \sum_{w=y}^{\sqrt{x}} G(x, w) \Delta_w \Pi^*(w, y, l) \\ &\leq\; \Psi_2(x, y) + \sum_{k=1}^{l} \int_y^{\sqrt{x}} G(x, w) d\Pi_k(w, y) \\ &=\; T_1 + T_2.\end{aligned}$$

Then by (14) and (11) we have

$$T_1 \;\sim\; e^\gamma \frac{x}{\log x} \log y.$$

It remains to estimate T_2.

By Hypothesis 5.2 and Lemma 3.3, we have

$$\begin{aligned}T_2 \;&\leq\; \sum_{k=1}^{l} \int_y^{\sqrt{x}} \frac{G(x, w)}{\log w} \rho\left(\phi(k) \frac{\log w}{\log y}\right) dw \cdot (1 + o(1)) \\ &+\; O\left(G(x, \sqrt{x}), \pi(\sqrt{x})) + G(x, y)\pi(y)\right).\end{aligned}$$

At this point we must break up the sum into three pieces; $T_1 = S_1 + S_2 + S_3$ where S_1 covers the interval $[y, y^{\alpha_\epsilon(y)}]$, S_2 the interval $[y^{\alpha_\epsilon(y)}, y^{\log y}]$, and S_3 the interval $[y^{\log y}, \sqrt{x}]$. Using the fact that $\sqrt{\log x}/\log y$ is well-behaved, we use the same techniques as in the proof of Lemma 3.7. The only interval which makes a significant contribution is S_1, so we will prove the upper bound for that one, and leave the intervals covered by S_2 and S_3 to the reader.

We approximate $G(x, w) \;\leq\; e^\gamma (x/(w \log x) \log w)(1 + o(1))$, and substituting $u = \phi(k) \log w/\log y$ gives

$$\begin{aligned}S_1 \;&\ll\; \frac{x}{\log x} (\log y) \sum_{k=1}^{l} \frac{1}{\phi(k)} \int_{\phi(k)}^{\phi(k)\alpha_\epsilon(y)} \rho(u) du + O(x/\log x) \\ &\ll\; \frac{x}{\log x} (\log y) \sum_{k=1}^{l} \frac{\rho(\phi(k))}{\phi(k)} \quad \ll \quad \frac{x}{\log x} (\log y) \sum_{k=1}^{l} \frac{\rho(\phi(k) - 1)}{\phi(k)^2} \\ &=\; O\left(\frac{x \log y}{\log x}\right)\end{aligned}$$

which follows from (18), (16), and the sum converges using the fact that $\phi(k) = \Omega(k/\log \log k)$ holds asymptotically (see Hardy and Wright [HW79]). □

Using the previous lemmas, we now have the following upper bound for $F(x, t, MC(l))$.

Theorem 5.10 *Let $\epsilon, \delta > 0$. If $t \geq (\log x)^{2+\delta}$, $\log t \leq (\log x)^{1-\delta}$, and $\sqrt{\log x}/\log t$ is well-behaved, then Hypothesis 5.2 implies*

$$F(x,t,MC(l)) = \Theta\left(\frac{x \log t}{\log x}\right).$$

Proof: The upper bound follows from Lemmas 5.6, 5.9, Lemma 5.8, and Theorem 3.2. The lower bound follows from Theorem 3.11 and (28). □

This proves (8), and completes our results for cyclotomic polynomial based integer factoring algorithms.

5.4 A Proof of Lemma 5.7.

We start by proving a simple lower bound on $\phi(k)$.

Lemma 5.11 *For $k \geq 2$, $\phi(k) \geq \sqrt{k}/2$.*

Proof: We break this down into several cases.

If $k = 2^e$, then $\phi(k) = 2^{e-1} = k/2$.

If $k = p^e$ with p an odd prime, for $e = 1$ we have $\phi(k) = p - 1 = k - 1 \geq \sqrt{k}$, and for $e > 1$, $\phi(k) = p^{e-1}(p-1) \geq p^{e-1} \geq \sqrt{k}$.

If k is not a prime power, factor $k = \prod_i k_i$ such that each k_i is a prime power and $i \neq j \Rightarrow \gcd(k_i, k_j) = 1$, and k_1 is even. Then $\phi(k) = \prod_i \phi(k_i) \geq (k_1/2) \prod_i \sqrt{k_i} \geq \sqrt{k}/2$. □

Proof of Lemma 5.7: Using the definition of $D_k(z)$ and the linearity of integrals, we have

$$\begin{aligned}\sum_{k=1}^{\infty} D_k(z) &= \int_1^z \rho(z/u-1) \sum_{k=1}^{\infty} \rho(\phi(k)u)du \\ &\ll \int_1^z \rho(z/u-1) \left\{\int_1^{\infty} \rho(\phi(t)u)dt\right\} du.\end{aligned}$$

Focusing on the inner integral, using Lemma 5.11, the fact that ρ is a decreasing function, and substituting $s = \sqrt{t}u/2$, we have

$$\begin{aligned}\int_1^{\infty} \rho(\phi(t)u)dt &\leq \int_1^{\infty} \rho(\sqrt{t}u/2)dt \\ &= 8/u^2 \int_{u/2}^{\infty} s\rho(s)ds.\end{aligned}$$

Using (16) and (18) gives the bound $8/u^2\rho(u/2-1)$. Plugging this in above and following the same method as the in proof of Lemma 3.4 gives

$$\begin{aligned}\sum_{k=1}^{\infty} D_k(z) &\ll \int_1^z \rho(z/u-1)8/u^2\rho(u/2-1)du \\ &\ll 8\rho(\sqrt{z/2}-1) \quad = \quad \rho(\Omega(\sqrt{z})).\end{aligned}$$

□

6 Computations.

In previous sections we made heuristic assumptions about the factorization patterns of numbers of the form $\Phi_k(p)$ for p a prime. Our heuristic (Hypothesis 5.2) says that these numbers are smooth with probability no greater than the probability numbers of the same size as $\Phi_k(p)$ are smooth. Although reasonable, this heuristic ignores many of the special properties of cyclotomic polynomials, giving one reason to question the soundness of this heuristic. For example, if p and q are primes with $q > k$, then $q \mid \Phi_k(p)$ if and only if p has order k modulo q and $q \equiv 1 \pmod{k}$. (See Washington [Was82, Lemma 2.9 and Proposition 2.10].)

In an effort to justify the heuristic, the author wrote a program to test it. In the rest of this section, we describe the algorithm used, we give the results of the computations, and we attempt to interpret these results.

6.1 The Algorithm.

The goal of the algorithm is to factor many values of cyclotomic polynomials and count how many are smooth. This can be done efficiently using a sieve for each value of k. So, the algorithm is based loosely on the segmented sieve of Eratosthenes. It performs several sieves on segments of moderate size, and combining the results gives data for a much larger interval.

First, the primes up to 32,768 were found. For each prime p, a generator of its multiplicative group was also found by factoring $p-1$ using trial division, and then each prime, starting with 2, was tested to see if it generated $(\mathbf{Z}/(p))^*$.

Then the program sieved 640 segments, each 16K long, giving an interval of length roughly 10 million. The midpoint of this interval was $2^{30} = 1073741824$. The following algorithm counted factorizations for one segment:

1. The current segment was sieved to give complete factorizations of all the integers in the segment. From this, the primes were identified and counted, and the functions $\Psi(x,y)$, $\Psi_2(x,y)$, $\Pi_1(x,y)$, and $\Pi_2(x,y)$ were tallied.

2. For k between 3 and 10, the following steps were performed:

   (a) The segment was sieved again, but only the primes p with $p \leq k$ or $p \equiv 1 \pmod{k}$ were used. To do this, the roots of $\Phi_k(x)$ modulo p were needed. Since these roots are roots of unity, the generator was used to construct them.

   (b) For each prime q in the segment, the integer $\Phi_k(q)$ was computed using extended precision arithmetic and then factored over the list of primes generated by the sieve. If it factored completely, the prime q was marked.

   (c) Finally, the segment was passed over one more time to tally $\Pi_k(x,y)$ for the marked primes.

This algorithm was implemented in Pascal and run on a VAXstation 3200, which has a word size of 32 bits. During the debugging phase of the program several factorizations of cyclotomic polynomials were checked using MACSYMA. The program took roughly 24 hours to sieve the entire interval.

6.2 Computational Results.

The table on the following page summarizes the information acquired from running the program we just described. This table requires a bit of explanation:

For each function tallied by the program, there are two columns. The first gives the actual count found by the program (Act), and the second gives a prediction based on the heuristics and results quoted in earlier sections (Est). Let I be the interval that was sieved, dx the length of I, and x the midpoint. Let $u = \log x / \log y$. If the interval I is equal to $[a, b]$, we abuse notation and write $f(I)$ for $f(b) - f(a)$. Then our predictions using results and heuristics from previous sections are

$$\begin{aligned}
\Psi(I, y) &= dx \cdot \rho(u) \\
\Psi_2(I, y) &= dx \cdot \rho_2(u) \\
\Pi_k(I, y) &= \pi(I) \cdot \rho(\phi(k)u).
\end{aligned}$$

To compute actual values, the functions ρ and ρ_2 were estimated using the methods of van de Lune and Wattel [vdLW69] and Simpson's rule. For $\pi(I)$, we used the actual number of primes found in the interval.

The functions Ψ_1 and Ψ_2 are included both because their counts came for free in the program and because they give a perspective on the size of the difference to expect between our estimates and the actual counts.

As can be seen, the results are quite interesting.

Notice that our heuristics for Π_1 and Π_2 (Hypotheses 3.13, 4.2) seem accurate. The actual counts were within 15% of the estimates, and further the counts for Π_1 and Π_2 were very close to each other.

However, for the quadratic polynomials (k = 3,4,6), our predictions give overestimates. These numbers support Hypothesis 5.2. It seems our predictions are of the right order of magnitude. Notice, however, that the actual counts differ from our predictions by around 50%, and further the counts for the different quadratics are not very close to one another.

Recall the heuristics ignore many of the special properties of cyclotomic polynomials. It may be possible to develop better heuristics by taking these properties into account. Clearly our results suggest that further study of the asymptotic behavior of $\Pi_k(x, y)$ is needed.

Computational Results

y	$\Psi(I,y)$		$\Psi_2(I,y)$		$\Pi_1(I,y)$		$\Pi_2(I,y)$	
	Act	Est	Act	Est	Act	Est	Act	Est
2	1	0.000	1046231	644389	0	0.000	0	0.000
4	1	0.000	1617772	1337598	0	0.000	0	0.000
8	9	0.000	2464732	2089269	0	0.000	2	0.000
16	63	1.801	3025794	2913126	11	0.087	12	0.087
32	931	206.0	3997649	3829586	137	9.907	109	9.907
64	5453	3720	4855640	4857237	615	178.8	630	178.8
128	25608	24799	5885098	5965496	2527	1192	2493	1192
256	84285	94678	6928144	7092735	7285	4552	7169	4552
512	218241	249835	7939883	8115914	16975	12013	16796	12013
1024	441207	509696	8766741	8942039	31512	24508	31671	24508
2048	775057	884252	9412945	9556402	52077	42518	52080	42518
4096	1212328	1366500	9884372	9997016	77563	65706	77245	65706
8192	1722213	1914867	10201300	10277477	106406	92073	106216	92073
16384	2304206	2555103	10396239	10438790	138203	122858	137817	122858
32768	2932380	3217585	10485760	10485760	171663	154712	171357	154712

y	$\Pi_3(I,y)$		$\Pi_4(I,y)$		$\Pi_5(I,y)$		$\Pi_6(I,y)$	
	Act	Est	Act	Est	Act	Est	Act	Est
64	0	0.000	0	0.000	0	0.000	0	0.000
128	0	0.002	0	0.002	0	0.000	0	0.002
256	0	0.087	0	0.087	0	0.000	0	0.087
512	1	1.259	0	1.259	0	0.000	1	1.259
1024	4	9.907	1	9.907	0	0.000	4	9.907
2048	21	50.11	22	50.11	0	0.000	21	50.11
4096	85	178.8	105	178.8	0	0.000	89	178.8
8192	285	502.6	337	502.6	0	0.000	263	502.6
16384	691	1192	821	1192	0	0.002	682	1192
32768	1492	2476	1781	2476	0	0.016	1532	2476

y	$\Pi_7(I,y)$		$\Pi_8(I,y)$		$\Pi_9(I,y)$		$\Pi_{10}(I,y)$	
	Act	Est	Act	Est	Act	Est	Act	Est
4096	0	0.000	0	0.000	0	0.000	0	0.000
8192	0	0.000	0	0.000	0	0.000	0	0.000
16384	0	0.000	0	0.002	0	0.000	0	0.002
32768	0	0.000	0	0.016	0	0.000	0	0.016

Interval length $= 16384 \cdot 640 = 10485760$
Midpoint of the interval $= 2^{30} = 1073741824$
Number of primes in the interval $\equiv \pi(I) = 504190$

References

[AM87] L. M. Adleman and K. S. McCurley. Open problems in number theoretic complexity. In D. S. Johnson, A. Nishizeki, A. Nozaki, and H. S. Wilf, editors, *Discrete Algorithms and Complexity: Proceedings of the Japan-US Joint Seminar.* Academic Press, Boston, 1987. Perspectives in Computing Series, volume 15.

[Apo57] T. M. Apostol. *Mathematical Analysis: A Modern Approach to Advanced Calculus.* Addison-Wesley, Reading, Mass., 1957.

[BMS86] E. Bach, G. Miller, and J. Shallit. Sums of divisors, perfect numbers, and factoring. *SIAM J. Comput.*, 4:1143–1154, 1986.

[BS89a] E. Bach and J. Shallit. Factoring with cyclotomic polynomials. *Math. Comp.*, 52(185):201–219, 1989.

[BS89b] E. Bach and J. Sorenson. Sieve algorithms for perfect power testing. Technical Report #852, Computer Sciences Department, University of Wisconsin-Madison, 1989.

[CEP83] E. R. Canfield, P. Erdös, and C. Pomerance. On a problem of Oppenheim concerning "Factorisatio Numerorum". *J. Number Theory*, 17:1–28, 1983.

[Dav80] H. Davenport. *Multiplicative Number Theory.* Springer-Verlag, New York, 1980.

[dB51a] N. G. de Bruijn. The asymptotic behavior of a function occurring in the theory of primes. *J. Indian Math. Soc. (N. S.)*, 15:25-32, 1951.

[dB51b] N. G. de Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Indag. Math.*, 13:50–60, 1951.

[dB66] N. G. de Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$, II. *Indag. Math.*, 28:239-247, 1966.

[Dic30] K. Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Arkiv för Matematik, Astronomi och Fysik*, 22A(10):1–14, 1930.

[GK81] D. H. Greene and D. E. Knuth. *Mathematics for the Analysis of Algorithms.* Birkhäuser, Boston, 1981.

[Gol69] M. Goldfeld. On the number of primes p for which $p + a$ has a large prime factor. *Mathematika*, 16:23–27, 1969.

[Guy75] R. K. Guy. How to factor a number. In *Proceedings of the 5th Manitoba Conference on Numerical Mathematics*, pages 49–89, Winnipeg, 1975.

[Hil86] A. Hildebrand. On the number of positive integers $\leq x$ and free of prime factors $> y$. *J. Number Theory*, 22:289–307, 1986.

[HM89] J. L. Hafner and K. S. McCurley. On the distribution of running times of certain integer factoring algorithms. *J. Algorithms*, 10(4):531–556, 1989.

[Hoo73] C. Hooley. On the largest prime factor of $p + a$. *Mathematika*, 20:135–143, 1973.

[HW79] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers.* Oxford University Press, 5th edition, 1979.

[Knu81] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley, Reading, Mass., 2nd edition, 1981.

[KTP76] D. E. Knuth and L. Trabb Pardo. Analysis of a simple factorization algorithm. *Theor. Comp. Sci.*, 3:321–348, 1976.

[Lan71] S. Lang. *Algebra.* Addison-Wesley, 1971.

[Len87] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. Math.*, 126:649–673, 1987.

[LLMP89] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard. The number field sieve. A report on work in progress, 1989.

[MB75] M. A. Morrison and J. D. Brillhart. A method of factoring and the factorization of F_7. *Math. Comp.*, 29:183–205, 1975.

[Mil76] G. Miller. Riemann's hypothesis and tests for primality. *J. Comp. Sys. Sciences*, 13:300–317, 1976.

[Mon87] P. L. Montgomery. Speeding the pollard methods of factorization. *Math. Comp.*, 48(177):243–264, 1987.

[MS88] P. L. Montgomery and R. D. Silverman. An FFT extension to the $p-1$ factoring algorithm. manuscript, 1988.

[Pol74] J. M. Pollard. Theorems on factorization and primality testing. *Proc. Camb. Phil. Soc.*, 76:521–528, 1974.

[Pol75] J. M. Pollard. A Monte Carlo algorithm for factorization. *BIT*, 15:331–334, 1975.

[Pom80] C. Pomerance. Popular values of Euler's function. *Mathematika*, 27:84–89, 1980.

[Pom82] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In H. W. Lenstra Jr. and R. Tijdeman, editors, *Computational Methods in Number Theory*, pages 89–141. Math. Centre, Amsterdam, 1982. Math. Centre Tract 154.

[Pom85] C. Pomerance. The quadratic sieve factoring algorithm. In *EUROCRYPT '84*, pages 169–182, Berlin, 1985. Springer-Verlag.

[Pom88] C. Pomerance. Two methods in elementary analytic number theory. In *Proc. NATO Advanced Study Institute on Number Theory and Applications*, Banff, 1988.

[Rab80] M. O. Rabin. Probabilistic algorithm for testing primality. *J. Number Theory*, 12:128–138, 1980.

[Sey87] M. Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Math. Comp.*, 48(178):757–780, 1987.

[SS71] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[SS77] R. Solovay and V. Strassen. A fast Monte Carlo test for primality. *SIAM J. Comput.*, 6:84–85, 1977. Erratum in vol. 7, p. 118, 1978.

[Str76] V. Strassen. Einige Resulte über Berechnungskomplexität. *Jahresbericht d. Deutschen Mathem.-Vereinigung*, 78:1–8, 1976.

[Val89] B. Vallée. Provably fast integer factoring with quasi-uniform small quadratic residues. In *21st Annual ACM Symposium on Theory of Computing*, pages 98–106, 1989.

[vdLW69] J. van de Lune and E. Wattel. On the numerical solution of a differential-difference equation arising in analytic number theory. *Math. Comp.*, 23:417–421, 1969.

[Was82] L. Washington. *Introduction to Cyclotomic Fields*. Springer-Verlag, New York, 1982.

[Wil82] H. C. Williams. A $p+1$ method of factoring. *Math. Comp.*, 39(159):225–234, 1982.

[WJ76a] H. C. Williams and J. S. Judd. Determination of the primality of N by using factors of $N^2 \pm 1$. *Math. Comp.*, 30(133):157–172, 1976.

[WJ76b] H. C. Williams and J. S. Judd. Some algorithms for prime testing using generalized Lehmer functions. *Math. Comp.*, 30(136):867–886, 1976.