EXACT PRIME TEST WITH ELLIPTIC CURVES ASSYMPTOTIC RUNNING TIME ANALYSIS TALK 2014, MARCH 5

ANTAL JÁRAI

The algorithm

The ECPP program can be divided three parts. The first will be called "Downto" the second will be called "Findcurves" and the third is "Proof".

The steps of Downto are the following:

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

D1: Square roots of small primes. We start with the probable prime n. Up to a limit s, for any prime number p for which $(n|p) = 1$, the square roots of p or $-p$ is calculated. The primes come from a precalculated prime list, containing the half of the differences between odd primes. Roughly for half of the primes have to calculate a square root. Calculating one square root with algorithm of Tonelly and Shank (see Cohen [6], p. 32) roughly the same time as the probabilistic primality test for n. Indeed, the algorithm has a probabilistic part which find a generator of $(\mathbb{Z}/n\mathbb{Z})^*$ and a deterministic part using $2\lceil \lg(n) \rceil + e^2$ multiplication modulo n, where e is the largest exponent for which $2^e|n-1$. Because the value of e is $1, 2, 3, \ldots$ with probability $1/2, 1/4, 1/8, \ldots$, the expected number of multiplications is $O(\ln n)$. Heuristic suggest that the running time of the probabilistic part can be neglected compared to the running time of the deterministic part. If $m(k)$ denote the multiplication time of k bit numbers, then the total running time of one square root operation is $O\bigl(m(\ln n) \ln n\bigr)$.

D2: Square roots of discriminants. A precalculated, prefactored and preordered list of fundamental discriminants $D \leq -7$ is read;

$$\gcd(n, 2D) = 1$$

necessary, i.e. n have to be free from small factors. For all odd prime factor of $-D$ a square root have to be given. The Jacobi symbol $(D|n)$ have to be 1. The modular square root of $-D$ is calculated using the roots of some factors of D.

D3: Reduction of quadratic forms. For a D passing the previous step, the reduction algorithm is applied for the quadratic form $ax^2 + bxy + cy^2$ where $a = n$, $b^2 \equiv -D \pmod{4n}$ and $c = (b^2 + D)/(4n)$. Here b can be found as the square root of D modulo n and adding n if necessary to have the same parity. The result of the reduction checked whether it results an algebraic integer ν from $\mathbb{Z}(\sqrt{D})$ for which $\nu\overline{\nu} = n$. If this is the case, then $s = \nu + \overline{\nu}$ is calculated. The chance for this is $1/(2h(D))$ where $h(D)$ is the ideal class number. The reduction time using binary reduction algorithm is proportional to $(\ln n)^2$, but this can be decreased to $O\bigl(m(\ln n)\ln n\bigr)$ with the Cornacchia algorithm (see Cohen [6], p. 34) combined with controlled euclidean descent of Schönhage (see Schönhage – Grotefeld – Vetter [30]).

D4: Factorization of curve orders. Elliptic curves with cardinality $n+1\pm s$ can be found. These cardinal numbers trial divided with primes up to a limit t. Only $s \bmod p$ is calculated, because $n+1 \bmod p$ is precalculated. Another halfing the work is possible, see [2], p. 55. The division with the found divisor is done and the quotient is stored. The necessary time is proportional to $t \ln n$.

D5: Factorization of curve orders. A "batch trial division" using fast multiplication and fast gcd can be applied to remove divisors from $10^4 \dots 10^6$ up to an upper limit b, say up to 10^8. This trick was suggested by one of the authors several years ago and seem to have appear independently by several authors; Pollard, Strassen, D. J. Bernstein, for example; see [31], p. 353. The product of curve orders is calculated: first product of pairs, then product of quads, etc. Primes products are read from file. It is divided by the product of curve orders, then the gcd of the remainder and the product of curve orders is calculated. This is small. It is distributed to partial products of curve orders, etc., product of quads, product of pairs, and finally to curve orders. The suprise with in this, that the dependence on $\lg n$, the magnitude of curve orders, is very week. The algorithm seems to be superior to other factorisation procedures useful here, as Pollard ρ and $p-1$, elliptic curves, etc., for factors below 10^8, but further investigation is necessary to find the best limit.

D6: Miller–Rabin test. The remaining unfactored parts are tested with the Rabin-Miller test (see Cohen [6], p. 415). A limited number of basis (say, 10, for example 2,3,5,7,11,13,19,23,29,31) is used. For any probable prime n' the corresponding D and s, for which n' is the factor of $n+1+s$, is outputted as a line containing D, s and n' (or better, n/n' instead of n') each in hexadecimal form. The running time of the test is $O\big(m(\ln n)\ln n\big)$.

D7: Iterate. The procedure is repeated for n' or each n', if there are more. If no n' is found, then after taking a larger limit s, d or b further D's and prime divisors may be include to obtain at least one n'.

D8: Backtrack. If “enough” work is done on n, and it is possible, backtrack.

The steps of “Findcurves” and “Proof” are the following:

F1: Calculate polynomial. By finding the proof, the first step is the calculation of the Hilbert polynomial (or another polynomial) using floating point calculations. The resulting coefficients are rounded to integers.

F2: Modular root of polynomial. The second step is the calculation of a root of the Hilbert polynomial having degree $h(D_n)$. This use the well-known splitting procedure, until a degree 1 part is obtained. To do this, we have to calculate the nth power of a random first-degree polynomial modulo the Hilbert polynomial modulo n. If the more refined version of Atkin and Morain is used, then the degree is $g(D_n) = h(D_n)/2^t$, where t is the number of factors of D_n.

F3: Find curve and a point on it. The third step to find the appropriate elliptic curve from the two possibility, and an appropriate point on it.

P: Partial proof. Calculate $n'P$ and nP on the given elliptic curve for the given point P.

Assymptotic running time analysis.

A heuristic running time analysis of the elliptic curve primality proof method is given by Lenstra and Lenstra [16]. They found the running time $O(\ln^{6+\varepsilon} n)$ for n and they found probable that using fast multiplication, division, polynom calculation, etc. methods it is possible to reduce the running time down to $O(\ln^{5+\varepsilon} n)$ for any positive ε. They mention without explicit treatement that using an observation of Shallit this can be reduced to $O(\ln^{4+\varepsilon} n)$ if we only use discriminants which are the product of some small primes. Here we will show that if some plausible assumptions are true then the running time can be reduced down to $o(\ln^4 n)$ bit operation. Our analysis is very close to our implementation.

The key to the running time analysis is the following observation: If we can find all prime factors of elliptic curve cardinalities $m = n + 1 \pm s$ below a bound $b(n)$, then the chance that one such number m will result a downto step, is equal to the probability that the second largest prime factor of m is less than b supposed to be approximately $\bigl($see Knuth [13], 4.5.4.(18)$\bigr)$.

$$e^\gamma \frac{\ln b(n)}{\ln n} \approx 1.7811 \frac{\ln b(n)}{\ln n}.$$

Hence reasonable to suppose that if we have alltogether $e(n)$ such numbers m, then the number of downto possibilities has a probability distribution with a mean approximately

$$\lambda = 1.7811 \frac{\ln b(n)}{\ln n} e(n).$$

Because for each negative discriminant $D \leq -7$ the probability of the succes is

$$\frac{1}{2h(D)},$$

where $h(D)$ denotes the ideal class number, but each succes results 2 different m's, we expect

$$e(n) \approx \sum_D \frac{1}{h(D)}.$$

Let us take a more detailed analysis of the running time separately for each step. Of course, this strongly depends on the time of multiplication, which has the same order for division and squaring, and for any known practical methods larger then the time for addition, substraction, and multiplication by two-powers. To keep our calculations as far as possible independent from the choose of fast multiplication methods, we shall denote my $m(k)$ the time which we need to multiple an arbitrary k-bit number with itself on a given computer using a given multiplication method. FFT technics, as the well-known method of Schönhage and Strassen make possible to keep this time as low as $O(k \ln k \ln \ln k)$ and our implementation make very realistic to use this estimate for the practical range of application with some ten thousend decimal digits of this test.

D1: modular square roots. We need the square root of D modulo n for several discriminants. To keep the amount of square root calculation reasonable, we may consider only discriminants D having absolute value below a bound $d(n)$ and are $s(n)$-smooth for some smooth bound function s. The following question is very important from the viewpoint of running time analysis.

Q1. Let $h(D)$ denote for a given negative discriminant D the ideal class number. What can be said about the assymptotic behavior of the function

$$\bar{e}(n) = \sum_{D} \frac{1}{h(D)},$$

where the summation is taken for all $s(n)$-smooth negative discriminants with absolute value between 7 and $d(n)$. The bound function $d(n)$ and the smooth function $s(n)$ are prescribed function of n.

For example, would be good to known that for $d(n) = (s(n))^2$ we have $\bar{e}(n) \asymp s(n)$.

As a standard example, we may choose $d(n) \asymp (\ln n)^2$. To obtain a lower estimate to the number of fundamental discriminants below $d(n)$ we consider from the remainder classes mod 16 only the classes of -3, -4, -7, -8, -11 and -15, because any number from these classes which if free from a square of odd primes is necessarily a fundamental discriminant. We estimate the density of numbers in these residue classes free from the square of any odd prime. The density of square free numbers is $\prod_{p\in\mathbb{P}}(1-1/p^2)$ having reciprocal

$$\begin{aligned}
&\prod_{p\in\mathbb{P}}\Big(1-\frac{1}{p^2}\Big)^{-1}\\
&=\prod_{p\in\mathbb{P}}\Big(1+\frac{1}{p^2}+\frac{1}{p^4}+\cdots\Big)\\
&=\sum_{n\in\mathbb{N}+}\frac{1}{n^2}=\frac{\pi^2}{6}.
\end{aligned}$$

Hence the density of odd square free numbers in general and in each in the residue classes modulo 16 is $(6/\pi^2)\cdot(4/3)=8/\pi^2$, and assymptotically we obtain that there are at least

$$3d(n)/\pi^2$$

fundamental discriminants below $d(n)$. To take the modular square root of all these discriminants is too much time. Suppose, that we only keep the fundamental discriminants which are $s(n) \asymp d(n)^c$ smooth with some $0 < c < 1$. Then we only have to calculate one square root modulo n for each prime below $s(n)$ for which $(n|p) = 1$. This is the half of the primes below $d(n)^c$, i.e. $O\big(d(n)^c/\ln d(n)\big)$ prime. So if $d(n) = O(\ln^2 n)$ and $m(k) = O(k \ln k \ln\ln k)$ then to take the roots we need $O(\ln^3 n \ln\ln\ln n)$ bit operations if $c = 1/2$ and $o(\ln^3 n)$ bit operations if $c < 1/2$. Still we may (see Knuth [13], 4.5.4) suppose that the number of discriminants which can be factored is $\asymp \ln^2(n)$.

To check all the fundamental discriminants up to $d(n)$ whether they are $d(n)^c$ smooth and to find their factorizations we may use simple trial division by all primes up to this limit. This takes at most $O\big(d(n)^{1+c}\ln^3 d(n)\big)$ bit operations using the classical division procedure.

So, as we will see, with an appropriate choose of $c < 1/2$ the total amount of time for this step can be neglected compared with the amount of time to the other steps.

D2: Square root of discriminants. For a given discriminant D first we check whether the conditions $(D|n) = 1$ and $(n|p) = 1$ for all odd prime factor of D are satisfied. The probability that these conditions are satisfied is 2^{-t} where t is the number of "prime" factors of D; here -4 or -8 are considered as a "prime" factor but -1 not. Because of the Erdős–Kac theorem the mean of the number of the factors for naturals up to x has in limit normal distribution with expected value $\ln\ln x$ and variance $\ln\ln x$, we obtain that this probability is

$$2^{-\ln\ln d} = (e^{-\ln\ln d})^{\ln 2} = 1/\ln^{\ln 2} d(n);$$

(¡¡¡¡ This was a serious error, I apologize for it; fortunately the results seems to be remain true ????) until $\ln d(n) \asymp \ln\ln n$, this probability is $\asymp 1/\ln\ln^{\ln 2} n$. For example, this is the case for any choose of $d(n) \asymp \ln^{\alpha}(n)\ln\ln^{\beta}(n)$ with $\alpha > 0$. The time to calculate the Jacobi symbols can be neglected, because $(n|p) = 1$ are already checked, $(4|n)$ and $(\pm 8|n)$ precalculated. Moreover, we have to calculate the square roots of the discriminants. This needs

$$O\big(m(\ln n)d(n)/\ln^{\ln 2} d(n)\big)$$

time, with the following simple trick: we precalculate the square root modulo n of all $\pm k$ for each k up to, say, $\sqrt{d(n)}$ which is the product of primes having $(n|p) = 1$. After this, at most 3 modular multiplications gives the square root of D modulo n; indeed it D is written as the product of 4 factors, then the 2 least has a product which is not greater then $\sqrt{d(n)}$.

D3: Reduction of quadratic forms. The remaining D's are investigated by reduction algorithms of quadratic forms or the Cornachia algorithm is applied. The running time in both cases can be decreased down to

$$O\big(m(\ln n)\ln\ln n\big),$$

hence the total running time in our case is

$$O\big(m(\ln n)d(n)\ln\ln n/\ln^{\ln 2} d(n)\big).$$

D4: Trial division. Twice the number of the remaining good discriminants is the number of the elliptic curves $e(n)$. This is approximated by $\bar{e}(n)$. The small factors of their cardinalities are removed with a simple trial division up to a limit $t(n)$. This needs

$$O\big(e(n)t(n)\ln n\big)$$

time altogether. If $t(n) = O(1)$, then this time can be neglected.

D5: Factorization. This step is to remove small factors and find large factors of elliptic curve cardinalities.

(a) Fourther trial division. If we use only simple trial division method to remove small factors up to the bound $b(n)$, then this needs
$$O\bigl(e(n)b(n)\ln n\bigr)$$
time.

(b) Batch trial division. One of the most practical method is our "batch trial division" up to a limit $b(n)$. If $b(n) \geq e(n)\ln n$, then dividing the product of primes up the $b(n)$ (which is in magnitude $e^{b(n)}$) to parts having the same size as the products of curve orders, we obtain
$$\frac{b(n)}{e(n)\ln n}$$
parts. Then we take the gcd of each part with the product of curve orders. This needs
$$O\left(\frac{b(n)}{e(n)\ln n} m\bigl(e(n)\ln n\bigr)\ln\bigl(e(n)\ln n\bigr)\right)$$

time and

$$O\bigl(e(n)\ln n\bigr)$$

core space. A somewhat better solution may be to divide the product of primes (as it read from disc, in binary form, starting with MSB) with the product of curve orders, and then taking the gcd of the remainder and the product. This also needs

$$O\bigl(e(n)\ln n\bigr)$$

core space, but only

$$O\left(m\bigl(e(n)\ln n\bigr)\Bigl(\frac{b(n)}{e(n)\ln n}+\ln\bigl(e(n)\ln n\bigr)\Bigr)\right)$$

time.

(c) Pollard's ϱ. The so-called Pollard ϱ methods can be applied. To find factors up to $b(n)$ roughly, we need $O\big(\sqrt{b(n)}\big)$ iterations, and each iterations needs $O\big(m(\ln n)\big)$ bit operations. We also needs gcd operations which takes $O\big(m(\ln n)\ln\ln n\big)$ bit operations, but seems to be enough to use a gcd operation only after $\ln\ln n$ iteration. In this way the total running time is

$$O\big(e(n)\sqrt{b(n)}m(\ln n)\big).$$

(d) Pollard's $p-1$. The $p-1$ method of Pollard also can be applied. Here we can choose parameters so that all factors $p < b(n)$ are found for which $p-1$ is $\sqrt{b(n)}$-smooth. Hence, a prime $p \ll b(n)$ will be found almost surely, but by $p \approx b(n)$ with probability ≈ 0.34 (see Knuth [13], 4.5.4 about the limit distribution of size of largest prime factor). The running time is similar as by the previous method of Pollard.

(e) We also may use the elliptic curve factorization method to find the small factors of the elliptic curve cardinalities. By Lenstra and Lenstra [17], 4.3, p. 698 the optimal choose to find factors below $b(n)$ to take the "smoothness bound" s to $L_{b(n)}(1/\sqrt{2})$ and to use $L_{b(n)}(1/\sqrt{2})$ elliptic curves, where $L_k(\beta)$ defined by

$$L_k(\beta) = e^{\beta\sqrt{\ln(k)\ln\ln(k)}}.$$

In this case we shall find prime factors below $b(n)$ with large probabilities. The total amount of time we need for $e(n)$ cardinalities is

$$O\left(e(n)\ln b(n) m(\ln n) L_{b(n)}(\sqrt{2})\right).$$

Moreover, two important theoretical question appears.

Q2. For different factorization methods such as trial division, Pollard ϱ, Pollard $p-1$, elliptic curves, etc., applying them with various parameters to find prime factors of cardinality of $e(n)$ elliptic curves, what is the expected number of curve cardinalities completely factored. For example, using trial division up to a bound $b(n)$ this expected number is hoped to be assymptotically

$$e^{\gamma}\frac{\ln b(n)}{\ln n}e(n).$$

Q3. For each different factorization methods, applying them with various parameters to find prime factors, if a given elliptic curve cardinality completely factors, what is the expected value and distribution of the "gain" defined as ln of the "smooth part" of the cardinality. For example, using trial division or batch trial division up to a bound $b(n)$ the expected value supposed to be $\asymp \ln b(n)$. The heuristic behind this is the following: for each prime p the probability that p divides a given curve order m is $1/p$ and this results a "gain" $\ln p$. Hence the expected value of the total gain is

$$\sum_{p\in\mathbb{P},p\leq b(n)} \frac{\ln p}{p} \sim \int_2^{b(n)} \ln x \frac{1}{\ln x}\frac{1}{x}\,dx \sim \ln b(n).$$

By outher methods still we may expect that the "gain" only differs by a bounded factor, so still remains $\asymp \ln b(n)$.

D6: Miller-Rabin test. We have to test $e(n)$ remaining unfactored parts with a fast probabilistic test, the Miller-Rabin test. This needs

$$O\big(e(n)m(\ln n)\ln n\big)$$

time.

D7: Iterate. We may suppose that if the expected value of the downone possibilities is large enough, then with a fixed positive probability we will succed. This gives the condition

$$e^{\gamma}\frac{\ln b(n)}{\ln n}e(n) > 1.$$

D8: Backtrack. We have the possibility always to use more effort in a given step or backtrack. We also want to find the optimal solution for this problem, too.

Q4. Not clear, what is the best tactic by the downto process, and what probability of succes can be obtained. This problem is similar to some problems in percolation theory. We plan to investigate what is the optimal choose of parameters. From one side, we should like to use minimal time, and on the other side we should like to obtain so large "gain" in the length of primes as possible. Occasionally we need a backtrack step. The probability of this possibility have to keep small. Some of the calculated data obtained earlier can be reused by bactracking, which makes even harder to find the optimal tactic.

F1: Calculate polynomial. By finding the proof, the first step is the calculation of the Hilbert polynomial. The time can be neglected?

F2: Find a root of the polynomial. The second step is the calculation of a root of the Hilbert polynomial having degree $h(D_n)$. The degree certainly $\leq 2\sqrt{d(n)}\ln d(n)$, and we may hope $\asymp \sqrt{d(n)}$. If the more refined version of Atkin and Morain is used, then the degree is $g(D_n)$, so we may hope degree $\asymp \sqrt{d(n)}/\ln^{\ln 2} d(n)$. So we have two cases:

(A) In a simple implementation the running time for one step is

$$O\Big(m\big(h(D_n)\ln n\big)\ln n\Big).$$

(B) In a more refined implementation the running time for one step is

$$O\Big(m\big(g(D_n)\ln n\big)\ln n\Big).$$

F3: Find elliptic curve. The third step to find an appropriate elliptic curve. The number of elliptic curves to try only two, and the expected number of points to try also bounded. Hence this step needs

$$O\big(m(\ln n)\ln n\big)$$

time. This can be neglected.

P: Proof step. One proof step needs

$$O\big(m(\ln n)\ln n\big)$$

time. This can be neglected.

Heuristic

From the running time analysis we obtain the following estimates for the running time of the important parts:

D3: Reduction of quadratic forms:

$$O\big(d(n)m(\ln n)\ln\ln n/\ln^{\ln 2} d(n)\big).$$

D5: Factorization:

(a) $O\big(e(n)b(n)\ln n\big)$.

(b) $O\left(m\big(e(n)\ln n\big)\Big(\frac{b(n)}{e(n)\ln n}+\ln\big(e(n)\ln n\big)\Big)\right)$

(c) $O\big(e(n)\sqrt{b(n)}m(\ln n)\big)$.

(d) $O\big(e(n)\sqrt{b(n)}m(\ln n)\big)$.

(e) $O\big(e(n)\ln b(n)m(\ln n)L_{b(n)}(\sqrt{2})\big)$.

D6: Miller-Rabin test: $O\big(e(n)m(\ln n)\ln n\big)$.

F2: Find a root of a polynomial:

(A) $O\Big(m\big(h(D_n)\ln n\big)\ln n\Big)$.

(B) $O\Big(m\big(g(D_n)\ln n\big)\ln n\Big)$.

"Strong factorization " heuristic. Let us take simple $d(n) \asymp \ln^2(n)$ and suppose that the "gain" is $\asymp \ln b(n)$, moreover $e(n) \asymp \sqrt{d(n)}$,

$$h(D_n) = O\bigl(\sqrt{d(n)}\bigr)$$

and

$$g(D_n) = O\bigl(\sqrt{d(n)}/\ln^{\ln 2} d(n)\bigr).$$

We suppose that $m(k) = O(k \ln k \ln\ln k)$. The time of one step for the most critical steps D3, D6 and F2 is independent from the method of factorization and the following:

D3: $O\bigl(\ln^3 n \ln\ln^{2-\ln 2} n \ln\ln\ln n\bigr)$;
D6: $O\bigl(\ln^3 n \ln\ln n \ln\ln\ln n\bigr)$;
F2A: $O\bigl(\ln^3 n \ln\ln n \ln\ln\ln n\bigr)$;
F2B: $O\bigl(\ln^3 n \ln\ln^{1-\ln 2} n \ln\ln\ln n\bigr)$.

In the different cases of the factorization the following one-step running time for factorization and and total running time for the whole test can be obtained using only the Hilbert polynomials:

(a): trial division. If $b(n)$ between $\asymp \ln(n)$ and $\asymp \ln(n)\ln\ln^{2-\ln 2} n \ln\ln\ln n$ then $O\big(b(n)\ln^2 n\big)$ and $O(\ln^4 n \ln\ln^{1-\ln 2} n \ln\ln\ln n)$.

(b): batch trial division. If $b(n)$ is between $\asymp \ln n$ and $\asymp \ln^3(n)\ln\ln^{1-\ln 2} n$, then the factorization time is between $O\big(\ln^2 n \ln\ln^2 n \ln\ln\ln n\big)$ and $O\big(\ln^3 n \ln\ln^{2-\ln 2} n \ln\ln\ln n\big)$ and the total running time is

$$O\big(\ln^4 \ln\ln^{1-\ln 2} n \ln\ln\ln n\big).$$

(c/d): Pollard. If $b(n)$ is between $\asymp \ln n$ and $\asymp \ln^2 n \ln\ln^{2-2\ln 2} n$, then

$$O\big(\sqrt{b(n)}\ln^2 n \ln\ln n \ln\ln\ln n\big)$$

and

$$O\big(\ln^4 n \ln\ln^{1-\ln 2} n \ln\ln\ln n\big).$$

(e): Elliptic curves. If

$$b(n) \asymp (\ln n)^{\ln\ln n/\ln\ln\ln^2 n}$$

then the "gain" is $\asymp \ln\ln^2 n/\ln\ln\ln^2 n$, the factorizaton time is $o(\ln^3 n)$, and the total running time is

$$O(\ln^4 n \ln\ln\ln^3 n/\ln\ln^{\ln 2} n).$$

“Strong factorization plus small discriminans” heuristic. Let us take

$$d(n) \asymp \ln^2 n / \ln\ln^2 n$$

and suppose that the “gain” is $\asymp \ln b(n)$, moreover $e(n) \asymp \sqrt{d(n)}$, $h(D_n) = O(\sqrt{d(n)})$ and $g(D_n) = O\big(\sqrt{d(n)}/\ln^{\ln 2} d(n)\big)$. We suppose that $m(k) = O(k \ln k \ln\ln k)$. Note that still the expected value of $e(n)$ can be greater then 1. Again, the time of one step for the most critical steps D3, D6 and F2 is independent from the method of factorization and the following:

D3: $O\big(\ln^3 n \ln\ln\ln n / \ln\ln^{\ln 2} n\big)$;
D6: $O\big(\ln^3 n \ln\ln\ln n\big)$;
F2A: $O\big(\ln^3 n \ln\ln\ln n\big)$;
F2B: $O\big(\ln^3 n \ln\ln\ln n / \ln\ln^{\ln 2} n\big)$;

In the different cases of the factorization the following one-step running time for factorization and total running time for the whole test can be obtained using only the Hilbert polynomials:

(a): trial division. If $b(n)$ is between $\asymp \ln n$ and $\asymp \ln n \ln\ln\ln n$, then $O\big(b(n)\ln^2 n\big)$ and

$$O(\ln^4 n \ln\ln\ln n/\ln\ln n).$$

(b): batch trial division. If $b(n)$ is between $\asymp \ln n$ and $\asymp \ln^3 n/\ln\ln^3 n$, then the factorization time is between $O\big(\ln^2 n \ln\ln n \ln\ln\ln n\big)$ and $O\big(\ln^3 n \ln\ln\ln n\big)$ and the total running time is

$$O\big(\ln^4 n \ln\ln\ln n/\ln\ln n\big).$$

(c/d): Pollard. If $b(n)$ is between $\asymp \ln n$ and $\asymp \ln^2 n$, then the factorization time is

$$O\big(\sqrt{b(n)}\ln^2 n \ln\ln\ln n\big)$$

and the total running time is

$$O\big(\ln^4 n \ln\ln\ln n/\ln\ln n\big).$$

(e): Elliptic curves. If

$$b(n) \asymp (\ln n)^{\ln\ln n/\ln\ln\ln^2 n}$$

then the "gain" is $\asymp \ln\ln^2 n/\ln\ln\ln^2 n$, the factorizaton time is $o(\ln^3 n)$, and the total running time is $O(\ln^4 n \ln\ln\ln^3 n/\ln\ln^2 n)$.

References

1. L. M. Adleman, *On Constructing A Molecular Computer. Draft* (1995), 1–20.
2. A. O. L. Atkin and F. Morain, *Elliptic curves and primality proving*, Math. of Computation **61** (1993), 29–68.
3. D. J. Bernstein, A. K.Lenstra, *A general number field sieve implementation*, A. K. Lenstra, H. W. Lenstra, Jr. (Eds.): The development of the number field sieve, Springer-Verlag, LNM 1554, 1993, pp. 103–126.
4. D. Boneh, R. J. Lipton, *Quantum Cryptanalysis of Hidden Linear Functions*, D. Coppersmith (Ed.): Advances in Cryptology — CRYPTO'95, Springer-Verlag, LNCS 963, 1995, pp. —.
5. J. Buhler, H. W. Lenstra, Jr., C. Pomerance, *Factoring integers with the number field sieve*, A. K. Lenstra, H. W. Lenstra, Jr. (Eds.): The development of the number field sieve, Springer-Verlag, LNM 1554, 1993, pp. 50–94.
6. H. Cohen, *A course in computational algebraic number theory*, Springer Verlag.
7. R. Crandall, J. Doenias, C. Norrie, J. Young, *The twenty-second Fermat number is composite*, Math. Comp. **64**, 863–868.
8. F. Damm, F.-P. Heider, G. Wambach, *MIMD-Factorisation on Hypercubes*, A. De Santis (Ed.): Advances in Cryptology — EUROCRYPT'94, Springer-Verlag, LNCS 950, 1994, pp. 400–409.
9. B. Dodson, A. K.Lenstra, *NFS with Four Large Primes: An explosive Experiment*, D. Coppersmith (Ed.): Advances in Cryptology — CRYPTO'95, Springer-Verlag,

LNCS 963, 1995, pp. 372–385.

10. K.-H. Indlekofer, A. Járai, *Largest known twin primes*, Math. Comp. (1996) (to appear).
11. K.-H. Indlekofer, A. Járai, *Largest known twin primes and Sophie Germain primes*, Math. Comp. (1996) (to appear).
12. D. E. Knuth, *The Art of Computer Programming, Vol. 1–3.*, Reading, MA: Addison-Wesley.
13. D. E. Knuth, *The Art of Computer Programming, Vol. 1–3. Second Edition*, Reading, MA: Addison-Wesley.
14. P. Kirrinnis, *Zur Berechnung von Partialbruchzerlegungen und Kurvenintegralen rationaler Funktionen, PhD. Thesis*, Rheinischen Friedrich-Wilhelms Universität zu Bonn.
15. P. Kirrinnis, *Partial Fraction Decomposition in* $\mathbb{C}(z)$ *and Simultaneous Newton Iteration for Factorization in* $\mathbb{C}[z]$, J. of Complexity (to appear).
16. A. K. Lenstra, H. W. Lenstra, Jr., *Algorithms in Number Theory*, J. v. Leeuwen (Ed.): Algorithms in Complexity, Vol. A, Elsevier, pp. 673–716.
17. A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard, *The number field sieve*, A. K. Lenstra, H. W. Lenstra, Jr. (Eds.): The development of the number field sieve, Springer-Verlag, LNM 1554, 1993, pp. 11–42.
18. R. Lipton, *Speeding up computations via molecular biology Draft* (1994), 1–7.
19. F. Morain, *Distributed primality proving and the primality of* $(2^{3539}+1)/3$, I. B.Damgąrd (Ed.) Advances in Cryptology – EUROCRYPT'90, 1991, pp. 110–123.

20. P. L. Montgomery, *A Block Lanczos Algorithm for Finding Dependencies over $GF(2)$*, L. C. Guillou, J.-J. Quisquater (Eds.): Advances in Cryptology — EUROCRYPT'95, Springer-Verlag, LNCS 921, 1995, pp. 95–105.
21. B. K. Parady, J. F. Smith, S. E. Zarantonello, *Largest known twin primes*, Math. Comp. **55**, 381–382.
22. J. M. Pollard, *Factoring with cubic integers*, A. K. Lenstra, H. W. Lenstra, Jr. (Eds.): The development of the number field sieve, Springer-Verlag, LNM 1554, 1993, pp. 4–10.
23. J. M. Pollard, *The lattice sieve*, A. K. Lenstra, H. W. Lenstra, Jr. (Eds.) The development of the number field sieve, Springer-Verlag, LNM 1554, 1993, pp. 43–49.
24. M. E. Pohst, *Computational Algebraic Number Theory*, Birkhäser.
25. P. Ribenboim, *The Book of Prime Number Records*, Springer-Verlag.
26. H. Riesel, *Prime Numbers and Computer Methods for Factorisation*, Birkhäuser.
27. A. Schönhage, *The fundamental theorem of algebra in terms of computational complexity – Preliminary report*, Univ. Tübingen, 1–74.
28. A. Schönhage, *Equation Solving in Terms of Computational Complexity*, Proceedings of the International Congress of Mathematicians, Berkeley, California, USA, 1986, pp. 131–154.
29. A. Schönhage, *Numerik analytischer Funktionen und Komplexität*, Jber. d. Dt. Math.-Verein **92** (1990), 1–20.
30. A. Schönhage, A. F. W. Grotefeld, E. Vetter, *Fast Algorithms: A Multitape Turing Machine Implemen-*

tation, B. I. Wissenschaftverlag, Mannheim.
31. T. Gowers (Ed.), *The Princeton Companion to Mathematics*, Princeton University Press, Princeton and Oxford.
32. D. Weber, *An Implementation of the General Number Field Sieve to Compute Discrete Logarithms mod p*, L. C. Guillou, J.-J. Quisquater (Eds.): Advances in Cryptology — EUROCRYPT'95, Springer-Verlag, LNCS 921, 1995, pp. 95–105.