SIAM J. COMPUT.
Vol. 4, No. 3, September 1975

EVERY PRIME HAS A SUCCINCT CERTIFICATE*

VAUGHAN R. PRATT†

Abstract. To prove that a number n is composite, it suffices to exhibit the working for the multiplication of a pair of factors. This working, represented as a string, is of length bounded by a polynomial in $\log_2 n$. We show that the same property holds for the primes. It is noteworthy that almost no other set is known to have the property that short proofs for membership or nonmembership exist for all candidates without being known to have the property that such proofs are easy to come by. It remains an open problem whether a prime n can be recognized in only $\log_2^\alpha n$ operations of a Turing machine for any fixed α.

The proof system used for certifying primes is as follows.

AXIOM. $(x, y, 1)$.

INFERENCE RULES.

$$\mathrm{R}_1: \quad (p, x, a), q \vdash (p, x, qa) \quad \text{provided } x^{(p-1)/q} \not\equiv 1 \pmod{p} \text{ and } q|(p-1).$$

$$\mathrm{R}_2: \quad (p, x, p-1) \vdash p \quad \text{provided } x^{p-1} \equiv 1 \pmod{p}.$$

THEOREM 1. *p is a theorem $\equiv$ p is a prime.*

THEOREM 2. *p is a theorem $\supset$ p has a proof of $\lceil 4 \log_2 p \rceil$ lines.*

Key words. primes, membership, nondeterministic, proof, NP-complete, computational complexity

1. Proofs. We know of no efficient method that will reliably tell whether a given number is prime or composite. By "efficient", we mean a method for which the time is at most a polynomial in the length of the number written in positional notation. Thus the cost of *testing* primes and composites is very high. In contrast, the cost of *selling* composites (persuading a potential customer that you have one) is very low—in every case, one multiplication suffices. The only catch is that the salesman may need to work overtime to prepare his short sales pitch; the effort is nevertheless rewarded when there are many customers.[1]

At a meeting of the American Mathematical Society in 1903, Frank Cole used this property of composites to add dramatic impact to the presentation of his paper. His result was that $2^{67} - 1$ was composite, contradicting a two-centuries-old conjecture of Mersenne. Although it had taken Cole "three years of Sundays" to find the factors, once he had done so he could, in a few minutes and without uttering a word, convince a large audience of his result simply by writing down the arithmetic for evaluating $2^{67} - 1$ and $193707721 \times 761838257287$.

We now show that the primes are to a lesser extent similarly blessed; one may certify p with a proof of at most $\lceil 4 \log_2 p \rceil$ lines, in a system each of whose inference rules are readily applied in time $O(\log^3 p)$. The method is based on the Lucas–Lehmer heuristic (Lehmer (1927)) for testing primeness.

In the system to be described, theorems take one of two forms:

(i) "p", asserting that p is prime, or

(ii) "(p, x, a)", asserting that we are making progress towards establishing that p is a prime and that x is a primitive root (mod p); a is a progress indicator

* Received by the editors May 24, 1974.

† Project MAC Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. This research was supported by the National Science Foundation under Grant GJ-34671.

[1] Edmonds (1965) discusses a similar situation with a "supervisor and his hard-working assistant".

such that when it reaches $p - 1$, we may establish these properties for p and x in one more step.

The system is as follows.

AXIOM. $(x, y, 1)$.

INFERENCE RULES.

R_1: $(p, x, a), q \vdash (p, x, qa)$ provided $x^{(p-1)/q} \not\equiv 1 \pmod p$ and $q|(p - 1)$;

R_2: $(p, x, p - 1) \vdash p$ provided $x^{p-1} \equiv 1 \pmod p$.

A certificate of p is then a proof in this system with last line p.

Some familiar primes are given by the following proofs.

(1)	$(2, 1, 1)$	Axiom;
(2)	2	$(1), R_2, 1^1 \equiv 1 \pmod 2$;
(3)	$(3, 2, 1)$	Axiom;
(4)	$(3, 2, 2)$	$(3), (2), R_1, 2^1 \equiv 2 \pmod 3$;
(5)	3	$(4), R_2, 2^2 \equiv 1 \pmod 3$.

No proof for 4 is possible because we would need to prove $(4, x, 3)$ for some $x \equiv 1 \pmod 4$ (by the condition in R_2), which would contradict the condition in R_1.

(6)	$(5, 2, 1)$	Axiom;
(7)	$(5, 2, 2)$	$(6), (2), R_1, 2^2 \equiv 4 \pmod 5$;
(8)	$(5, 2, 4)$	$(7), (2), R_1, 2^2 \equiv 4 \pmod 5$;
(9)	5	$(8), R_2, 2^4 \equiv 1 \pmod 5$.

No proof for 6 is possible because $x^5 \not\equiv 1 \pmod 6$ for all $x \not\equiv 1 \pmod 6$.

(10)	$(11, 2, 1)$	Axiom;
(11)	$(11, 2, 2)$	$(10), (2), R_1, 2^5 \equiv 10 \pmod{11}$;
(12)	$(11, 2, 10)$	$(11), (9), R_1, 2^2 \equiv 4 \pmod{11}$;
(13)	11	$(12), R_2, 2^{10} \equiv 1 \pmod{11}$;
(14)	$(23, 5, 1)$	Axiom;
(15)	$(23, 5, 2)$	$(14), (2), R_1, 5^{11} \equiv 22 \pmod{23}$;
(16)	$(23, 5, 22)$	$(15), (13), R_1, 5^2 \equiv 2 \pmod{23}$;
(17)	23	$(16), R_2, 23^{22} \equiv 1 \pmod{23}$;
(18)	$(47, 5, 1)$	Axiom;
(19)	$(47, 5, 2)$	$(18), (2), R_1, 5^{23} \equiv 46 \pmod{47}$;
(20)	$(47, 5, 46)$	$(19), (17), R_1, 5^2 \equiv 25 \pmod{47}$;
(21)	47	$(20), R_2, 5^{46} \equiv 1 \pmod{47}$.

Not counting the proof for 3, this (shortest) proof of 47 took 18 steps, not too far from the promised bound of $\lceil 4\log_2 47\rceil = 22$. The gap is mostly due to the proof of 47 not using the proof of 3 that is counted in the bound $\lceil 4\log_2 p\rceil$. A much larger gap is exhibited by the proof of 474397531, which is 23 lines long; here, $\lceil 4\log_2 p\rceil = 116$. This prime was constructed to show that our bound on proof length is not always tight. Steps (1) to (9) are as above.

(10)	$(251, 6, 1)$	Axiom;
(11)	$(251, 6, 2)$	(10), (2), R_1;
(12)	$(251, 6, 10)$	(11), (9), R_1;
(13)	$(251, 6, 50)$	(12), (9), R_1;
(14)	$(251, 6, 250)$	(13), (9), R_1;
(15)	251	(14), R_2;
(16)	$(474397531, 2, 1)$	Axiom;
(17)	$(474397531, 2, 2)$	(16), (2), R_1;
(18)	$(474397531, 2, 6)$	(17), (5), R_1;
(19)	$(474397531, 2, 30)$	(18), (9), R_1;
(20)	$(474397531, 2, 7530)$	(19), (15), R_1;
(21)	$(474397531, 2, 1890030)$	(20), (15), R_1;
(22)	$(474397531, 2, 474397530)$	(21), (15), R_1;
(23)	474397531	(22), R_2.

2. Metaproofs. We now prove soundness and completeness of our system.

THEOREM 1. *p is a prime if and only if p is a theorem.*

Proof. If. No number has multiplicative order $p - 1 \pmod p$ when p is not a prime. If such a p is proved, it must be by application of R_2 to $(p, x, p - 1)$ where $x^{p-1} \equiv 1 \pmod p$. Hence $x^j \equiv 1 \pmod p$ for some $j < p - 1$. Now $j | p - 1$, so $x^{(p-1)/q} \equiv 1 \pmod p$ for some prime q. But to prove $(p, x, p - 1)$, we had to build up $p - 1$ as the product of primes q which satisfied $x^{(p-1)/q} \not\equiv 1 \pmod p$. Applying the fundamental theorem of arithmetic then leads to a contradiction.

Only if. This part proceeds by induction on p. If p is prime, then p has a primitive root $\pmod p$, that is, a number whose multiplicative order $\pmod p$ is $p - 1$. A proof of p may start with the axiom $(p, x, 1)$ for such a primitive root x. By the induction hypothesis, each of the prime factors of $p - 1$ is a theorem. Moreover, for each such prime factor q, $x^{(p-1)/q} \not\equiv 1 \pmod p$; otherwise the order of x would be less than $p - 1$. Hence the proof system permits the inference of any theorem (p, x, a), where a is a product of prime factors of $p - 1$. In particular, $(p, x, p - 1)$ may be inferred, and since $x^{p-1} \equiv 1 \pmod p$, we may infer p. □

We now establish the efficiency of our method.

THEOREM 2. *If p is a theorem, then p has a proof of at most $\lceil 4\log_2 p\rceil$ lines.*

Proof. The construction given in the proof of Theorem 1 yields such a proof.

First prove 2 and 3 in five lines. (These primes p are special because $p - 1$ is not composite.) We now assume as our induction hypothesis that by not counting the proofs of 2 and 3, each prime p can be proved in at most $\lfloor 4 \log_2 p \rfloor - 4$ lines.

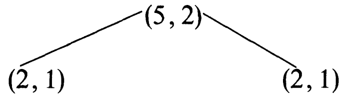
For $p = 2$ or 3, this follows directly from the identities $\lfloor 4 \log_2 2 \rfloor - 4 = 0$ and $\lfloor 4 \log_2 3 \rfloor - 4 = 2$. For $p > 3$, let $p - 1 = p_1 p_2 \cdots p_k$, $k \geqq 2$. Then the cost of proving p is bounded above by

$$\begin{aligned} & 2 + k + \sum_{1 \leqq i \leqq k} (\lfloor 4 \log_2 p_i \rfloor - 4) && \text{(by the induction hypothesis)} \\ & \leqq \lfloor 4 \log_2 p_1 p_2 \cdots p_k) \rfloor - 4 && \text{(since } k \geqq 2) \\ & \leqq \lfloor 4 \log_2 p \rfloor - 4, && \text{(the desired answer).} \end{aligned}$$

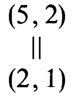
If we now count the 5 lines required to prove 2 and 3, the cost rises to $\lfloor 4 \log_2 p \rfloor + 1$ lines. For $p > 2$, $\log_2 p$ will not be an integer, and so the cost is bounded by $\lceil 4 \log_2 p \rceil$, a bound that is 4 when $p = 2$ and is therefore applicable to all p. □

Almost identical proofs may be used to show that no more than $\lfloor 3 \log_2 p \rfloor$ lines involve an exponentiation and $\lfloor 2 \log_2 p \rfloor$ a multiplication, facts which we will use in the next section.

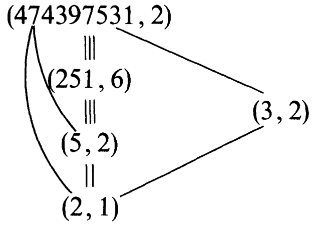
3. Picturesque proofs. The reader should have little difficulty in seeing that all the information in the proof that 5 is prime is contained in the following tree, whose vertices are primes together with their primitive roots.



or, collapsing repeated vertices for brevity:



The proof tree for 474397531, when collapsed, becomes



It is straightforward to check a proof tree without reconstructing the proof. The "VELP" test (vertices, edges, leaves, products) is

(i) For each vertex (p, x), $x^{p-1} \equiv 1 \pmod p$.

(ii) For each edge (p, x) down to (q, y), $x^{(p-1)/q} \not\equiv 1 \pmod p$ and $q | p - 1$.

(iii) Each leaf is $(2, 1)$.

(iv) For each vertex (p, x) with immediate descendants $(p_1, x_1), \cdots, (p_k, x_k)$, $p = p_1 p_2 \cdots p_k + 1$.

The proof tree approach is more picturesque than the proof system, whose raison d'être is that it is more formally and compactly presented.

4. Computations. Returning to our customer, we find him dissatisfied with the exponentiation he must carry out to check a line. He protests that the evaluation of x^b requires $b - 1$ multiplications, and also that the numbers produced along the way have $O(b)$ digits, which he has neither time nor paper to write down for large b.

The first protest is dealt with by the well-known trick of exponentiating by repeated squaring, which yields x^b with at most $2\lfloor\log_2 b\rfloor$ multiplications. This method is an essential feature of the Lucas–Lehmer heuristic. The method may be described recursively as:

$$\begin{aligned} x^b\colon\quad & b = 0 \to 1, \\ & b \text{ odd} \to xx^{b-1}, \\ & b \text{ even} \to (x^2)^{b/2}. \end{aligned}$$

To eliminate the recursion and attendant waste of space, we translate this algorithm into a "deterministic" system whose rules are

$$\begin{aligned} (u, v, w) &\vdash (u^2, v/2, w) && \text{if } v \text{ is even}, \\ (u, v, w) &\vdash (u, v - 1, uw) && \text{if } v \text{ is odd}. \end{aligned}$$

Now wu^v is an invariant of these rules, each of which reduces either the number of significant bits (provided $v \neq 0$) or the number of 1's in v (expressed in binary notation), but not both. Hence $(x, b, 1) \overset{*}{\vdash} (y, 0, x^b)$ in a number of steps exactly one less than the number of bits plus the number of 1's in b, which is at most $2\lceil\log_2 (b + 1)\rceil - 1$. By skipping the multiplication the first time w is multiplied by u, and beginning with (x, b, x), only $2\lfloor\log_2 b\rfloor$ multiplications are required.

The second protest is disposed of by performing each multiplication modulo p in the above algorithms when testing $x^b \equiv 1 \pmod p$.

In any proof of p, each multiplication is performed modulo q for some prime $q \leqq p$. Moreover, in testing $x^b, b < p$. Hence each exponentiation requires at most $2\lfloor\log_2 p\rfloor$ multiplications of numbers smaller than p. At most $\lfloor 3 \log_2 p\rfloor$ exponentiations are required, whence no more than $6 \log_2^2 p$ multiplications plus the $\lfloor 2 \log_2 p\rfloor$ multiplications from R_1 are needed. Each multiplication may be carried out in $O(\log p \log\log p)$ steps on a random access machine (RAM) (Schönhage and Strassen, (1971)), and so $O(\log^3 p \log\log p)$ steps suffice to check a proof of p on a RAM. (A factor of $\log\log\log p$ creeps in for those who do the arithmetic on paper (or on a Turing machine) due to time spent scanning and shuffling the sheets!)

An item that might find a market among consumers of prime numbers would be a pocket calculator with a predicate (x, b, p) that evaluates $x^{(p-1)/b} \equiv 1 \pmod p$. Only one bit of output is required, only integer arithmetic (multiple-precision) is used, and so the unit should cost about \$100 in quantity at today's prices, assuming that it handles integers of up to several hundred bits. Users of the Hewlett Packard HP-65 pocket computer with the appropriate program may find it suitable but expensive. A proof using our method of, say, the smallest Mersenne prime yet undiscovered would require a considerably more expensive unit, with perhaps 30,000-bit integers and sufficient parallelism to make the computation time acceptably low.

5. Complexity. The families NP (P) of sets of strings accepted (recognized) in time some polynomial function of their length by some nondeterministic

(deterministic) Turing machine[2] have recently engaged the attention of computational complexity theorists. The family P is of interest in that it includes all sets that can be recognized reasonably quickly, a property that has become identified to some extent with membership in P. The family NP is of interest (Cook (1971), Karp (1972)) because it includes thirty or more operations-research-related sets each with the astonishing property that if it belonged to P, then NP $=$ P, implying that all of its fellow O.R. sets would be in P, along with other sets in NP (such as {primes} as we showed above) not known at present to belong to P. In view of the effort that has been expended in the past twenty years or so on trying to show that any one of these sets is in P, it is widely conjectured that none is, that is, NP $\neq$ P. These peculiar sets are called NP-complete.

A family of sets that has only very recently attracted any attention is coNP $= \{S|S \in \mathrm{NP}\}$. Of course, coP $=$ P, whence if NP $=$ P, then coNP $=$ NP. However, it is conceivable that NP $\neq$ P but NP $=$ coNP. It is straightforward to show that NP $=$ coNP if and only if some NP-complete set is in coNP, just as NP $=$ P if and only if some NP-complete set is in P, and it is conjectured that NP $\neq$ coNP.

If true, this implies that NP $\cap$ coNP contains no NP-complete problems. One is tempted to speculate that NP $\cap$ coNP $=$ P. After all, the families RE and R of recursively enumerable and recursive sets, whose relationship resembles the NP $-$ P relationship, satisfy RE $\cap$ coRE $=$ R; and until recently, every known member of NP $\cap$ coNP was known to be in P. Thus one could be forgiven for wanting to conjecture that NP $\cap$ coNP $=$ P.

An immediate corollary of § 4 above is that the primes are in NP $\cap$ coNP. Provided NP $\neq$ coNP, this settles in the negative a question raised by Cook as to whether the composites are NP-complete. Conjecture aside, it gives us the first known member of NP $\cap$ coNP not known to be in P. Chvatal has recently exhibited another set with this property, namely the set of pairs (linear programming problem, optimal solution to it). No other such sets are known, although a plausible candidate is the set of irreducible univariate polynomials over the integers. Berlekamp (1967) has shown that over any finite field such a set is in P. A somewhat less plausible candidate is the set of pairs of isomorphic graphs.

If the primes or the optimal-lp-solutions are not in P, it will not be because they are NP-complete (still supposing NP $\neq$ coNP) which is the *usual* reason. One might therefore say that these problems were *anomalously* hard, although any term for this phenomenon lacks the all-or-nothing significance of "NP-completeness". The whole question of proving lower bounds on the complexity of sets in NP is completely open, and any information about the structure of hard problems would be welcome. In particular, the criterion that membership in NP $\cap$ coNP precludes NP-completeness, though based only on a conjecture, is nonetheless a useful guide considering how few tools we have in the area.

6. Conclusion.

We exhibited a simple system whose theorems are exactly the set of all primes and whose proofs are very short. We inferred from this that the primes are in NP $\cap$ coNP, giving us our first example of a member of NP $\cap$ coNP not known to be in P. We advocated membership in NP $\cap$ coNP as a strong

[2] That is, for each such set there is a polynomial and a Turing machine.

reason for presuming non-NP-completeness, based on the plausible and moderately popular conjecture that NP $\neq$ coNP. We observed the striking paucity of sets that are candidates for lying between P and NP-complete sets. It is interesting to find the number theorists' most famous set occupying a special position in complexity theory.

REFERENCES

E. R. BERLEKAMP (1967), *Factoring polynomials over finite fields*, Bell System Tech. J., 46, pp. 1853–1860.

S. A. COOK (1971), *The complexity of theorem-proving procedures*, Conf. Rec. of 3rd Ann. ACM Symp. on Theory of Computing, pp. 151–158.

J. EDMONDS (1965), *Minimum partition of a matroid into independent subsets*, J. Res. Nat. Bur. Standards Sect. B, 69B, pp. 67–72.

R. M. KARP (1972), *Reducibilities among combinatorial problems*, Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, pp. 85–103.

D. H. LEHMER (1927), Bull. Amer. Math. Soc., 33, pp. 327–340.

A. SCHÖNHAGE AND V. STRASSEN (1971), *Fast multiplication of large numbers*, Computing, 7, pp. 281–292. (In German.) English description: D. E. Knuth, The Art of Computer Programming, vol. 2, 2nd printing, pp. 270–275.