

**31.7-2**

Prove that if Alice's public exponent  $e$  is 3 and an adversary obtains Alice's secret exponent  $d$ , where  $0 < d < \phi(n)$ , then the adversary can factor Alice's modulus  $n$  in time polynomial in the number of bits in  $n$ . (Although you are not asked to prove it, you may be interested to know that this result remains true even if the condition  $e = 3$  is removed. See Miller [255].)

**31.7-3 ★**

Prove that RSA is multiplicative in the sense that

$$P_A(M_1)P_A(M_2) \equiv P_A(M_1M_2) \pmod{n}.$$

Use this fact to prove that if an adversary had a procedure that could efficiently decrypt 1 percent of messages from  $\mathbb{Z}_n$  encrypted with  $P_A$ , then he could employ a probabilistic algorithm to decrypt every message encrypted with  $P_A$  with high probability.

---

**★ 31.8 Primality testing**

In this section, we consider the problem of finding large primes. We begin with a discussion of the density of primes, proceed to examine a plausible, but incomplete, approach to primality testing, and then present an effective randomized primality test due to Miller and Rabin.

**The density of prime numbers**

For many applications, such as cryptography, we need to find large “random” primes. Fortunately, large primes are not too rare, so that it is feasible to test random integers of the appropriate size until we find a prime. The **prime distribution function**  $\pi(n)$  specifies the number of primes that are less than or equal to  $n$ . For example,  $\pi(10) = 4$ , since there are 4 prime numbers less than or equal to 10, namely, 2, 3, 5, and 7. The prime number theorem gives a useful approximation to  $\pi(n)$ .

**Theorem 31.37 (Prime number theorem)**

$$\lim_{n \rightarrow \infty} \frac{\pi(n)}{n / \ln n} = 1.$$

■

The approximation  $n / \ln n$  gives reasonably accurate estimates of  $\pi(n)$  even for small  $n$ . For example, it is off by less than 6% at  $n = 10^9$ , where  $\pi(n) =$

50,847,534 and  $n / \ln n \approx 48,254,942$ . (To a number theorist,  $10^9$  is a small number.)

We can view the process of randomly selecting an integer  $n$  and determining whether it is prime as a Bernoulli trial (see Section C.4). By the prime number theorem, the probability of a success—that is, the probability that  $n$  is prime—is approximately  $1 / \ln n$ . The geometric distribution tells us how many trials we need to obtain a success, and by equation (C.32), the expected number of trials is approximately  $\ln n$ . Thus, we would expect to examine approximately  $\ln n$  integers chosen randomly near  $n$  in order to find a prime that is of the same length as  $n$ . For example, we expect that finding a 1024-bit prime would require testing approximately  $\ln 2^{1024} \approx 710$  randomly chosen 1024-bit numbers for primality. (Of course, we can cut this figure in half by choosing only odd integers.)

In the remainder of this section, we consider the problem of determining whether or not a large odd integer  $n$  is prime. For notational convenience, we assume that  $n$  has the prime factorization

$$n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}, \quad (31.39)$$

where  $r \geq 1$ ,  $p_1, p_2, \dots, p_r$  are the prime factors of  $n$ , and  $e_1, e_2, \dots, e_r$  are positive integers. The integer  $n$  is prime if and only if  $r = 1$  and  $e_1 = 1$ .

One simple approach to the problem of testing for primality is **trial division**. We try dividing  $n$  by each integer  $2, 3, \dots, \lfloor \sqrt{n} \rfloor$ . (Again, we may skip even integers greater than 2.) It is easy to see that  $n$  is prime if and only if none of the trial divisors divides  $n$ . Assuming that each trial division takes constant time, the worst-case running time is  $\Theta(\sqrt{n})$ , which is exponential in the length of  $n$ . (Recall that if  $n$  is encoded in binary using  $\beta$  bits, then  $\beta = \lceil \lg(n+1) \rceil$ , and so  $\sqrt{n} = \Theta(2^{\beta/2})$ .) Thus, trial division works well only if  $n$  is very small or happens to have a small prime factor. When it works, trial division has the advantage that it not only determines whether  $n$  is prime or composite, but also determines one of  $n$ 's prime factors if  $n$  is composite.

In this section, we are interested only in finding out whether a given number  $n$  is prime; if  $n$  is composite, we are not concerned with finding its prime factorization. As we shall see in Section 31.9, computing the prime factorization of a number is computationally expensive. It is perhaps surprising that it is much easier to tell whether or not a given number is prime than it is to determine the prime factorization of the number if it is not prime.

### Pseudoprimality testing

We now consider a method for primality testing that “almost works” and in fact is good enough for many practical applications. Later on, we shall present a re-

finement of this method that removes the small defect. Let  $\mathbb{Z}_n^+$  denote the nonzero elements of  $\mathbb{Z}_n$ :

$$\mathbb{Z}_n^+ = \{1, 2, \dots, n-1\}.$$

If  $n$  is prime, then  $\mathbb{Z}_n^+ = \mathbb{Z}_n^*$ .

We say that  $n$  is a **base- $a$  pseudoprime** if  $n$  is composite and

$$a^{n-1} \equiv 1 \pmod{n}. \quad (31.40)$$

Fermat's theorem (Theorem 31.31) implies that if  $n$  is prime, then  $n$  satisfies equation (31.40) for every  $a$  in  $\mathbb{Z}_n^+$ . Thus, if we can find any  $a \in \mathbb{Z}_n^+$  such that  $n$  does *not* satisfy equation (31.40), then  $n$  is certainly composite. Surprisingly, the converse *almost* holds, so that this criterion forms an almost perfect test for primality. We test to see whether  $n$  satisfies equation (31.40) for  $a = 2$ . If not, we declare  $n$  to be composite by returning COMPOSITE. Otherwise, we return PRIME, guessing that  $n$  is prime (when, in fact, all we know is that  $n$  is either prime or a base-2 pseudoprime).

The following procedure pretends in this manner to be checking the primality of  $n$ . It uses the procedure MODULAR-EXPONENTIATION from Section 31.6. We assume that the input  $n$  is an odd integer greater than 2.

PSEUDOPRIME( $n$ )

```

1  if MODULAR-EXPONENTIATION(2, n - 1, n)  $\not\equiv$  1 (mod n)
2      return COMPOSITE           // definitely
3  else return PRIME              // we hope!
```

This procedure can make errors, but only of one type. That is, if it says that  $n$  is composite, then it is always correct. If it says that  $n$  is prime, however, then it makes an error only if  $n$  is a base-2 pseudoprime.

How often does this procedure err? Surprisingly rarely. There are only 22 values of  $n$  less than 10,000 for which it errs; the first four such values are 341, 561, 645, and 1105. We won't prove it, but the probability that this program makes an error on a randomly chosen  $\beta$ -bit number goes to zero as  $\beta \rightarrow \infty$ . Using more precise estimates due to Pomerance [279] of the number of base-2 pseudoprimes of a given size, we may estimate that a randomly chosen 512-bit number that is called prime by the above procedure has less than one chance in  $10^{20}$  of being a base-2 pseudoprime, and a randomly chosen 1024-bit number that is called prime has less than one chance in  $10^{41}$  of being a base-2 pseudoprime. So if you are merely trying to find a large prime for some application, for all practical purposes you almost never go wrong by choosing large numbers at random until one of them causes PSEUDOPRIME to return PRIME. But when the numbers being tested for primality are not randomly chosen, we need a better approach for testing primality.

As we shall see, a little more cleverness, and some randomization, will yield a primality-testing routine that works well on all inputs.

Unfortunately, we cannot entirely eliminate all the errors by simply checking equation (31.40) for a second base number, say  $a = 3$ , because there exist composite integers  $n$ , known as **Carmichael numbers**, that satisfy equation (31.40) for all  $a \in \mathbb{Z}_n^*$ . (We note that equation (31.40) does fail when  $\gcd(a, n) > 1$ —that is, when  $a \notin \mathbb{Z}_n^*$ —but hoping to demonstrate that  $n$  is composite by finding such an  $a$  can be difficult if  $n$  has only large prime factors.) The first three Carmichael numbers are 561, 1105, and 1729. Carmichael numbers are extremely rare; there are, for example, only 255 of them less than 100,000,000. Exercise 31.8-2 helps explain why they are so rare.

We next show how to improve our primality test so that it won't be fooled by Carmichael numbers.

### The Miller-Rabin randomized primality test

The Miller-Rabin primality test overcomes the problems of the simple test PSEUDOPRIME with two modifications:

- It tries several randomly chosen base values  $a$  instead of just one base value.
- While computing each modular exponentiation, it looks for a nontrivial square root of 1, modulo  $n$ , during the final set of squarings. If it finds one, it stops and returns COMPOSITE. Corollary 31.35 from Section 31.6 justifies detecting composites in this manner.

The pseudocode for the Miller-Rabin primality test follows. The input  $n > 2$  is the odd number to be tested for primality, and  $s$  is the number of randomly chosen base values from  $\mathbb{Z}_n^+$  to be tried. The code uses the random-number generator RANDOM described on page 117: RANDOM(1,  $n - 1$ ) returns a randomly chosen integer  $a$  satisfying  $1 \leq a \leq n - 1$ . The code uses an auxiliary procedure WITNESS such that WITNESS( $a, n$ ) is TRUE if and only if  $a$  is a “witness” to the compositeness of  $n$ —that is, if it is possible using  $a$  to prove (in a manner that we shall see) that  $n$  is composite. The test WITNESS( $a, n$ ) is an extension of, but more effective than, the test

$$a^{n-1} \not\equiv 1 \pmod{n}$$

that formed the basis (using  $a = 2$ ) for PSEUDOPRIME. We first present and justify the construction of WITNESS, and then we shall show how we use it in the Miller-Rabin primality test. Let  $n - 1 = 2^t u$  where  $t \geq 1$  and  $u$  is odd; i.e., the binary representation of  $n - 1$  is the binary representation of the odd integer  $u$  followed by exactly  $t$  zeros. Therefore,  $a^{n-1} \equiv (a^u)^{2^t} \pmod{n}$ , so that we can

compute  $a^{n-1} \bmod n$  by first computing  $a^u \bmod n$  and then squaring the result  $t$  times successively.

WITNESS( $a, n$ )

```

1  let  $t$  and  $u$  be such that  $t \geq 1$ ,  $u$  is odd, and  $n - 1 = 2^t u$ 
2   $x_0 = \text{MODULAR-EXPONENTIATION}(a, u, n)$ 
3  for  $i = 1$  to  $t$ 
4       $x_i = x_{i-1}^2 \bmod n$ 
5      if  $x_i == 1$  and  $x_{i-1} \neq 1$  and  $x_{i-1} \neq n - 1$ 
6          return TRUE
7  if  $x_t \neq 1$ 
8      return TRUE
9  return FALSE

```

This pseudocode for WITNESS computes  $a^{n-1} \bmod n$  by first computing the value  $x_0 = a^u \bmod n$  in line 2 and then squaring the result  $t$  times in a row in the **for** loop of lines 3–6. By induction on  $i$ , the sequence  $x_0, x_1, \dots, x_t$  of values computed satisfies the equation  $x_i \equiv a^{2^i u} \pmod{n}$  for  $i = 0, 1, \dots, t$ , so that in particular  $x_t \equiv a^{n-1} \pmod{n}$ . After line 4 performs a squaring step, however, the loop may terminate early if lines 5–6 detect that a nontrivial square root of 1 has just been discovered. (We shall explain these tests shortly.) If so, the algorithm stops and returns TRUE. Lines 7–8 return TRUE if the value computed for  $x_t \equiv a^{n-1} \pmod{n}$  is not equal to 1, just as the PSEUDOPRIME procedure returns COMPOSITE in this case. Line 9 returns FALSE if we haven't returned TRUE in lines 6 or 8.

We now argue that if WITNESS( $a, n$ ) returns TRUE, then we can construct a proof that  $n$  is composite using  $a$  as a witness.

If WITNESS returns TRUE from line 8, then it has discovered that  $x_t = a^{n-1} \bmod n \neq 1$ . If  $n$  is prime, however, we have by Fermat's theorem (Theorem 31.31) that  $a^{n-1} \equiv 1 \pmod{n}$  for all  $a \in \mathbb{Z}_n^+$ . Therefore,  $n$  cannot be prime, and the equation  $a^{n-1} \bmod n \neq 1$  proves this fact.

If WITNESS returns TRUE from line 6, then it has discovered that  $x_{i-1}$  is a nontrivial square root of 1, modulo  $n$ , since we have that  $x_{i-1} \not\equiv \pm 1 \pmod{n}$  yet  $x_i \equiv x_{i-1}^2 \equiv 1 \pmod{n}$ . Corollary 31.35 states that only if  $n$  is composite can there exist a nontrivial square root of 1 modulo  $n$ , so that demonstrating that  $x_{i-1}$  is a nontrivial square root of 1 modulo  $n$  proves that  $n$  is composite.

This completes our proof of the correctness of WITNESS. If we find that the call WITNESS( $a, n$ ) returns TRUE, then  $n$  is surely composite, and the witness  $a$ , along with the reason that the procedure returns TRUE (did it return from line 6 or from line 8?), provides a proof that  $n$  is composite.

At this point, we briefly present an alternative description of the behavior of WITNESS as a function of the sequence  $X = \langle x_0, x_1, \dots, x_t \rangle$ , which we shall find useful later on, when we analyze the efficiency of the Miller-Rabin primality test. Note that if  $x_i = 1$  for some  $0 \leq i < t$ , WITNESS might not compute the rest of the sequence. If it were to do so, however, each value  $x_{i+1}, x_{i+2}, \dots, x_t$  would be 1, and we consider these positions in the sequence  $X$  as being all 1s. We have four cases:

1.  $X = \langle \dots, d \rangle$ , where  $d \neq 1$ : the sequence  $X$  does not end in 1. Return TRUE in line 8;  $a$  is a witness to the compositeness of  $n$  (by Fermat's Theorem).
2.  $X = \langle 1, 1, \dots, 1 \rangle$ : the sequence  $X$  is all 1s. Return FALSE;  $a$  is not a witness to the compositeness of  $n$ .
3.  $X = \langle \dots, -1, 1, \dots, 1 \rangle$ : the sequence  $X$  ends in 1, and the last non-1 is equal to  $-1$ . Return FALSE;  $a$  is not a witness to the compositeness of  $n$ .
4.  $X = \langle \dots, d, 1, \dots, 1 \rangle$ , where  $d \neq \pm 1$ : the sequence  $X$  ends in 1, but the last non-1 is not  $-1$ . Return TRUE in line 6;  $a$  is a witness to the compositeness of  $n$ , since  $d$  is a nontrivial square root of 1.

We now examine the Miller-Rabin primality test based on the use of WITNESS. Again, we assume that  $n$  is an odd integer greater than 2.

MILLER-RABIN( $n, s$ )

```

1  for  $j = 1$  to  $s$ 
2       $a = \text{RANDOM}(1, n - 1)$ 
3      if WITNESS( $a, n$ )
4          return COMPOSITE           // definitely
5  return PRIME                       // almost surely
```

The procedure MILLER-RABIN is a probabilistic search for a proof that  $n$  is composite. The main loop (beginning on line 1) picks up to  $s$  random values of  $a$  from  $\mathbb{Z}_n^+$  (line 2). If one of the  $a$ 's picked is a witness to the compositeness of  $n$ , then MILLER-RABIN returns COMPOSITE on line 4. Such a result is always correct, by the correctness of WITNESS. If MILLER-RABIN finds no witness in  $s$  trials, then the procedure assumes that this is because no witnesses exist, and therefore it assumes that  $n$  is prime. We shall see that this result is likely to be correct if  $s$  is large enough, but that there is still a tiny chance that the procedure may be unlucky in its choice of  $a$ 's and that witnesses do exist even though none has been found.

To illustrate the operation of MILLER-RABIN, let  $n$  be the Carmichael number 561, so that  $n - 1 = 560 = 2^4 \cdot 35$ ,  $t = 4$ , and  $u = 35$ . If the procedure chooses  $a = 7$  as a base, Figure 31.4 in Section 31.6 shows that WITNESS computes  $x_0 \equiv a^{35} \equiv 241 \pmod{561}$  and thus computes the sequence

$X = \langle 241, 298, 166, 67, 1 \rangle$ . Thus, WITNESS discovers a nontrivial square root of 1 in the last squaring step, since  $a^{280} \equiv 67 \pmod{n}$  and  $a^{560} \equiv 1 \pmod{n}$ . Therefore,  $a = 7$  is a witness to the compositeness of  $n$ , WITNESS(7,  $n$ ) returns TRUE, and MILLER-RABIN returns COMPOSITE.

If  $n$  is a  $\beta$ -bit number, MILLER-RABIN requires  $O(s\beta)$  arithmetic operations and  $O(s\beta^3)$  bit operations, since it requires asymptotically no more work than  $s$  modular exponentiations.

### Error rate of the Miller-Rabin primality test

If MILLER-RABIN returns PRIME, then there is a very slim chance that it has made an error. Unlike PSEUDOPRIME, however, the chance of error does not depend on  $n$ ; there are no bad inputs for this procedure. Rather, it depends on the size of  $s$  and the “luck of the draw” in choosing base values  $a$ . Moreover, since each test is more stringent than a simple check of equation (31.40), we can expect on general principles that the error rate should be small for randomly chosen integers  $n$ . The following theorem presents a more precise argument.

#### Theorem 31.38

If  $n$  is an odd composite number, then the number of witnesses to the compositeness of  $n$  is at least  $(n - 1)/2$ .

**Proof** The proof shows that the number of nonwitnesses is at most  $(n - 1)/2$ , which implies the theorem.

We start by claiming that any nonwitness must be a member of  $\mathbb{Z}_n^*$ . Why? Consider any nonwitness  $a$ . It must satisfy  $a^{n-1} \equiv 1 \pmod{n}$  or, equivalently,  $a \cdot a^{n-2} \equiv 1 \pmod{n}$ . Thus, the equation  $ax \equiv 1 \pmod{n}$  has a solution, namely  $a^{n-2}$ . By Corollary 31.21,  $\gcd(a, n) \mid 1$ , which in turn implies that  $\gcd(a, n) = 1$ . Therefore,  $a$  is a member of  $\mathbb{Z}_n^*$ ; all nonwitnesses belong to  $\mathbb{Z}_n^*$ .

To complete the proof, we show that not only are all nonwitnesses contained in  $\mathbb{Z}_n^*$ , they are all contained in a proper subgroup  $B$  of  $\mathbb{Z}_n^*$  (recall that we say  $B$  is a *proper* subgroup of  $\mathbb{Z}_n^*$  when  $B$  is subgroup of  $\mathbb{Z}_n^*$  but  $B$  is not equal to  $\mathbb{Z}_n^*$ ). By Corollary 31.16, we then have  $|B| \leq |\mathbb{Z}_n^*|/2$ . Since  $|\mathbb{Z}_n^*| \leq n - 1$ , we obtain  $|B| \leq (n - 1)/2$ . Therefore, the number of nonwitnesses is at most  $(n - 1)/2$ , so that the number of witnesses must be at least  $(n - 1)/2$ .

We now show how to find a proper subgroup  $B$  of  $\mathbb{Z}_n^*$  containing all of the nonwitnesses. We break the proof into two cases.

*Case 1:* There exists an  $x \in \mathbb{Z}_n^*$  such that

$$x^{n-1} \not\equiv 1 \pmod{n}.$$

In other words,  $n$  is not a Carmichael number. Because, as we noted earlier, Carmichael numbers are extremely rare, case 1 is the main case that arises “in practice” (e.g., when  $n$  has been chosen randomly and is being tested for primality).

Let  $B = \{b \in \mathbb{Z}_n^* : b^{n-1} \equiv 1 \pmod{n}\}$ . Clearly,  $B$  is nonempty, since  $1 \in B$ . Since  $B$  is closed under multiplication modulo  $n$ , we have that  $B$  is a subgroup of  $\mathbb{Z}_n^*$  by Theorem 31.14. Note that every nonwitness belongs to  $B$ , since a nonwitness  $a$  satisfies  $a^{n-1} \equiv 1 \pmod{n}$ . Since  $x \in \mathbb{Z}_n^* - B$ , we have that  $B$  is a proper subgroup of  $\mathbb{Z}_n^*$ .

Case 2: For all  $x \in \mathbb{Z}_n^*$ ,

$$x^{n-1} \equiv 1 \pmod{n}. \quad (31.41)$$

In other words,  $n$  is a Carmichael number. This case is extremely rare in practice. However, the Miller-Rabin test (unlike a pseudo-primality test) can efficiently determine that Carmichael numbers are composite, as we now show.

In this case,  $n$  cannot be a prime power. To see why, let us suppose to the contrary that  $n = p^e$ , where  $p$  is a prime and  $e > 1$ . We derive a contradiction as follows. Since we assume that  $n$  is odd,  $p$  must also be odd. Theorem 31.32 implies that  $\mathbb{Z}_n^*$  is a cyclic group: it contains a generator  $g$  such that  $\text{ord}_n(g) = |\mathbb{Z}_n^*| = \phi(n) = p^e(1 - 1/p) = (p-1)p^{e-1}$ . (The formula for  $\phi(n)$  comes from equation (31.20).) By equation (31.41), we have  $g^{n-1} \equiv 1 \pmod{n}$ . Then the discrete logarithm theorem (Theorem 31.33, taking  $y = 0$ ) implies that  $n-1 \equiv 0 \pmod{\phi(n)}$ , or

$$(p-1)p^{e-1} \mid p^e - 1.$$

This is a contradiction for  $e > 1$ , since  $(p-1)p^{e-1}$  is divisible by the prime  $p$  but  $p^e - 1$  is not. Thus,  $n$  is not a prime power.

Since the odd composite number  $n$  is not a prime power, we decompose it into a product  $n_1 n_2$ , where  $n_1$  and  $n_2$  are odd numbers greater than 1 that are relatively prime to each other. (There may be several ways to decompose  $n$ , and it does not matter which one we choose. For example, if  $n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$ , then we can choose  $n_1 = p_1^{e_1}$  and  $n_2 = p_2^{e_2} p_3^{e_3} \cdots p_r^{e_r}$ .)

Recall that we define  $t$  and  $u$  so that  $n-1 = 2^t u$ , where  $t \geq 1$  and  $u$  is odd, and that for an input  $a$ , the procedure WITNESS computes the sequence

$$X = \langle a^u, a^{2u}, a^{2^2 u}, \dots, a^{2^t u} \rangle$$

(all computations are performed modulo  $n$ ).

Let us call a pair  $(v, j)$  of integers *acceptable* if  $v \in \mathbb{Z}_n^*$ ,  $j \in \{0, 1, \dots, t\}$ , and  $v^{2^j u} \equiv -1 \pmod{n}$ .



Acceptable pairs certainly exist since  $u$  is odd; we can choose  $v = n - 1$  and  $j = 0$ , so that  $(n - 1, 0)$  is an acceptable pair. Now pick the largest possible  $j$  such that there exists an acceptable pair  $(v, j)$ , and fix  $v$  so that  $(v, j)$  is an acceptable pair. Let

$$B = \{x \in \mathbb{Z}_n^* : x^{2^j u} \equiv \pm 1 \pmod{n}\}.$$

Since  $B$  is closed under multiplication modulo  $n$ , it is a subgroup of  $\mathbb{Z}_n^*$ . By Theorem 31.15, therefore,  $|B|$  divides  $|\mathbb{Z}_n^*|$ . Every nonwitness must be a member of  $B$ , since the sequence  $X$  produced by a nonwitness must either be all 1s or else contain a  $-1$  no later than the  $j$ th position, by the maximality of  $j$ . (If  $(a, j')$  is acceptable, where  $a$  is a nonwitness, we must have  $j' \leq j$  by how we chose  $j$ .)

We now use the existence of  $v$  to demonstrate that there exists a  $w \in \mathbb{Z}_n^* - B$ , and hence that  $B$  is a proper subgroup of  $\mathbb{Z}_n^*$ . Since  $v^{2^j u} \equiv -1 \pmod{n}$ , we have  $v^{2^j u} \equiv -1 \pmod{n_1}$  by Corollary 31.29 to the Chinese remainder theorem. By Corollary 31.28, there exists a  $w$  simultaneously satisfying the equations

$$w \equiv v \pmod{n_1},$$

$$w \equiv 1 \pmod{n_2}.$$

Therefore,

$$w^{2^j u} \equiv -1 \pmod{n_1},$$

$$w^{2^j u} \equiv 1 \pmod{n_2}.$$

By Corollary 31.29,  $w^{2^j u} \not\equiv 1 \pmod{n_1}$  implies  $w^{2^j u} \not\equiv 1 \pmod{n}$ , and  $w^{2^j u} \not\equiv -1 \pmod{n_2}$  implies  $w^{2^j u} \not\equiv -1 \pmod{n}$ . Hence, we conclude that  $w^{2^j u} \not\equiv \pm 1 \pmod{n}$ , and so  $w \notin B$ .

It remains to show that  $w \in \mathbb{Z}_n^*$ , which we do by first working separately modulo  $n_1$  and modulo  $n_2$ . Working modulo  $n_1$ , we observe that since  $v \in \mathbb{Z}_n^*$ , we have that  $\gcd(v, n) = 1$ , and so also  $\gcd(v, n_1) = 1$ ; if  $v$  does not have any common divisors with  $n$ , then it certainly does not have any common divisors with  $n_1$ . Since  $w \equiv v \pmod{n_1}$ , we see that  $\gcd(w, n_1) = 1$ . Working modulo  $n_2$ , we observe that  $w \equiv 1 \pmod{n_2}$  implies  $\gcd(w, n_2) = 1$ . To combine these results, we use Theorem 31.6, which implies that  $\gcd(w, n_1 n_2) = \gcd(w, n) = 1$ . That is,  $w \in \mathbb{Z}_n^*$ .

Therefore  $w \in \mathbb{Z}_n^* - B$ , and we finish case 2 with the conclusion that  $B$  is a proper subgroup of  $\mathbb{Z}_n^*$ .

In either case, we see that the number of witnesses to the compositeness of  $n$  is at least  $(n - 1)/2$ . ■

### Theorem 31.39

For any odd integer  $n > 2$  and positive integer  $s$ , the probability that MILLER-RABIN( $n, s$ ) errs is at most  $2^{-s}$ .

**Proof** Using Theorem 31.38, we see that if  $n$  is composite, then each execution of the **for** loop of lines 1–4 has a probability of at least  $1/2$  of discovering a witness  $x$  to the compositeness of  $n$ . MILLER-RABIN makes an error only if it is so unlucky as to miss discovering a witness to the compositeness of  $n$  on each of the  $s$  iterations of the main loop. The probability of such a sequence of misses is at most  $2^{-s}$ . ■

If  $n$  is prime, MILLER-RABIN always reports PRIME, and if  $n$  is composite, the chance that MILLER-RABIN reports PRIME is at most  $2^{-s}$ .

When applying MILLER-RABIN to a large randomly chosen integer  $n$ , however, we need to consider as well the prior probability that  $n$  is prime, in order to correctly interpret MILLER-RABIN's result. Suppose that we fix a bit length  $\beta$  and choose at random an integer  $n$  of length  $\beta$  bits to be tested for primality. Let  $A$  denote the event that  $n$  is prime. By the prime number theorem (Theorem 31.37), the probability that  $n$  is prime is approximately

$$\begin{aligned}\Pr\{A\} &\approx 1/\ln n \\ &\approx 1.443/\beta.\end{aligned}$$

Now let  $B$  denote the event that MILLER-RABIN returns PRIME. We have that  $\Pr\{\bar{B} \mid A\} = 0$  (or equivalently, that  $\Pr\{B \mid A\} = 1$ ) and  $\Pr\{B \mid \bar{A}\} \leq 2^{-s}$  (or equivalently, that  $\Pr\{\bar{B} \mid \bar{A}\} > 1 - 2^{-s}$ ).

But what is  $\Pr\{A \mid B\}$ , the probability that  $n$  is prime, given that MILLER-RABIN has returned PRIME? By the alternate form of Bayes's theorem (equation (C.18)) we have

$$\begin{aligned}\Pr\{A \mid B\} &= \frac{\Pr\{A\} \Pr\{B \mid A\}}{\Pr\{A\} \Pr\{B \mid A\} + \Pr\{\bar{A}\} \Pr\{B \mid \bar{A}\}} \\ &\approx \frac{1}{1 + 2^{-s}(\ln n - 1)}.\end{aligned}$$

This probability does not exceed  $1/2$  until  $s$  exceeds  $\lg(\ln n - 1)$ . Intuitively, that many initial trials are needed just for the confidence derived from failing to find a witness to the compositeness of  $n$  to overcome the prior bias in favor of  $n$  being composite. For a number with  $\beta = 1024$  bits, this initial testing requires about

$$\begin{aligned}\lg(\ln n - 1) &\approx \lg(\beta/1.443) \\ &\approx 9\end{aligned}$$

trials. In any case, choosing  $s = 50$  should suffice for almost any imaginable application.

In fact, the situation is much better. If we are trying to find large primes by applying MILLER-RABIN to large randomly chosen odd integers, then choosing a small value of  $s$  (say 3) is very unlikely to lead to erroneous results, though

we won't prove it here. The reason is that for a randomly chosen odd composite integer  $n$ , the expected number of nonwitnesses to the compositeness of  $n$  is likely to be very much smaller than  $(n - 1)/2$ .

If the integer  $n$  is not chosen randomly, however, the best that can be proven is that the number of nonwitnesses is at most  $(n - 1)/4$ , using an improved version of Theorem 31.38. Furthermore, there do exist integers  $n$  for which the number of nonwitnesses is  $(n - 1)/4$ .

### Exercises

#### 31.8-1

Prove that if an odd integer  $n > 1$  is not a prime or a prime power, then there exists a nontrivial square root of 1 modulo  $n$ .

#### 31.8-2 ★

It is possible to strengthen Euler's theorem slightly to the form

$$a^{\lambda(n)} \equiv 1 \pmod{n} \text{ for all } a \in \mathbb{Z}_n^*,$$

where  $n = p_1^{e_1} \cdots p_r^{e_r}$  and  $\lambda(n)$  is defined by

$$\lambda(n) = \text{lcm}(\phi(p_1^{e_1}), \dots, \phi(p_r^{e_r})). \quad (31.42)$$

Prove that  $\lambda(n) \mid \phi(n)$ . A composite number  $n$  is a Carmichael number if  $\lambda(n) \mid n - 1$ . The smallest Carmichael number is  $561 = 3 \cdot 11 \cdot 17$ ; here,  $\lambda(n) = \text{lcm}(2, 10, 16) = 80$ , which divides 560. Prove that Carmichael numbers must be both "square-free" (not divisible by the square of any prime) and the product of at least three primes. (For this reason, they are not very common.)

#### 31.8-3

Prove that if  $x$  is a nontrivial square root of 1, modulo  $n$ , then  $\gcd(x - 1, n)$  and  $\gcd(x + 1, n)$  are both nontrivial divisors of  $n$ .

---

## ★ 31.9 Integer factorization

Suppose we have an integer  $n$  that we wish to **factor**, that is, to decompose into a product of primes. The primality test of the preceding section may tell us that  $n$  is composite, but it does not tell us the prime factors of  $n$ . Factoring a large integer  $n$  seems to be much more difficult than simply determining whether  $n$  is prime or composite. Even with today's supercomputers and the best algorithms to date, we cannot feasibly factor an arbitrary 1024-bit number.