period A random-number generator with period *P* generates the same random sequence of random numbers after *P* iterations. (263)

permutation A permutation of 1, 2, …, *N* is a sequence of *N* integers that includes each of 1, 2, …, *N* exactly once. (259)

Poisson distribution A distribution that models the number of occurrences of a rare event. (265)

pseudorandom numbers Numbers that have many properties of random numbers. Good generators of pseudorandom numbers are hard to find. (260)

random permutation A random arrangement of *N* items. Can be generated in linear time using one random number per item. (268)

randomized algorithm An algorithm that uses random numbers rather than deterministic decisions to control branching. (269)

seed The initial state of the random-number generator. (262)

trial division A simple algorithm for primality testing. It is fast for small (32-bit) numbers but cannot be used for larger numbers. (272)

uniform distribution A distribution in which all numbers in the specified range are equally likely to occur. (260)

witness to compositeness A value of *A* that proves that a number is not prime using Fermat's Little Theorem. (273)

Common Errors

1. Using an initial seed of zero will give bad random numbers.
2. Inexperienced users occasionally reinitialize the seed prior to generating a random permutation. This guarantees that the same permutation will be repeatedly produced, which is probably not what is intended.
3. Many random number generators are notoriously bad. For serious applications in which long sequences of random numbers are required, the linear congruential generator is also unsatisfactory.
4. The low-order bits of linear congruential generators are known to be somewhat nonrandom, so avoid using them. As an example, `randomInt()%2` is a bad way to flip a coin.
5. When random numbers are being generated in some interval, a common error is to be slightly off at the boundaries and either allow some number outside of the interval to be generated or not allow the smallest number to be generated with fair probability.
6. Many random permutation generators do not generate all permutations with equal likelihood. As discussed in the text, our algorithm is limited by the random-number generator.
7. Tinkering with a random number generator is likely to weaken its statistical properties.