In this problem set, you will learn about an efficient method to find large prime numbers. The main application of large prime numbers is to generate keys for the RSA cryptosystem, where one needs to find random prime numbers of at least 500–1000 binary digits.

Let $n$ be an integer. Our goal is to devise a test which tells us whether or not $n$ is prime.

**Problem 1** (Naive Primality Test) The naive way of testing whether $n$ is prime is to check whether $n$ is divisible by each integer $x$ where $1 \leq x \leq \sqrt{n}$. (a) Why is it sufficient to check only integers up to $\sqrt{n}$? (b) If $n$ is a number with 1000 binary digits, and if each division takes 1 microsecond (one millionth of a second) to perform, then how long would it take to check whether $n$ is prime by this method? (Express your answers in years. Round your answer to the closest power of 10). (c) One may try to improve the efficiency of this algorithm by using only those numbers $x$ which are not obviously composite. For instance, one may wish to only use odd numbers for $x$. Does this significantly improve your answer to part (b)?

Here is another idea for testing whether $n$ is prime, using Fermat’s Theorem. Let $a$ be any number in the range $1 \leq a \leq n - 1$. We say that $n$ passes the $a$-pseudoprime test if $a^{n-1} \equiv 1 \pmod{n}$. If $n$ is prime, then it passes the $a$-pseudoprime test for all $a$ with $1 \leq a \leq n - 1$, by Fermat’s Theorem.

**Problem 2** ($a$-Pseudoprime Test) Show that 45 passes the $a$-pseudoprime test for $a = 8$, $a = 17$, $a = 19$, but not for $a = 4$, $a = 7$, or $a = 11$. Thus 45 is not prime. Now, show that 323 is not prime by finding an $a$ such that 323 fails the $a$-pseudoprime test.

A number $n$ is called pseudoprime if for every $a$ relatively prime to $n$, $n$ passes the $a$-pseudoprime test. Thus every prime number is pseudoprime. There are some pseudoprime numbers which are not prime; such numbers are called Carmichael numbers. Carmichael numbers are relatively rare; the smallest one is 561.

**Problem 3** The number $561 = 3 \cdot 11 \cdot 17$ is not prime. Prove that 561 is pseudoprime. (Hint: Prove that for all $a$ relatively prime to 561, one has $a^{560} \equiv 1 \pmod{3}$, modulo 11, and modulo 17. Conclude that 561 passes the $a$-pseudoprime test for any such $a$).

We propose the following test to determine whether a given $n$ is pseudoprime.

**Randomized Pseudoprime Test.** Given: A number $n$ to be tested. Pick $k$ random numbers $a_1, \ldots, a_k \in \{1, \ldots, n - 1\}$. If $n$ fails the $a_i$-pseudoprime test for one or more of these numbers, then we conclude that $n$ is not pseudoprime (and therefore not prime). If $n$ passes the test for all $a_1, \ldots, a_k$, then we conclude that $n$ is probably pseudoprime.

If $n$ is pseudoprime, the test will always say so. However, if $n$ is not pseudoprime, then there is a certain probability that the test will wrongly conclude that $n$ is pseudoprime, if the numbers $a_1, \ldots, a_k$ are chosen unlucky (compare Problem 2). Thus the success rate of the pseudoprime test hinges on the question: if $n$ is not pseudoprime, then what is the probability that the randomized pseudoprime test gives the wrong answer? We will now see that the probability of the algorithm being fooled is quite small.

If $X$ is a finite set, then we say that the cardinality of $X$ is the number of elements in $X$. 

1
**Problem 4** Let $R = \mathbb{Z}/n\mathbb{Z}$ be the ring of integers mod $n$. Let $U$ be the set of units in $R$. Let $V = \{a \in U \mid n \text{ passes the } a\text{-pseudoprime test}\}$.

(a) Prove: $V$ is closed under products, i.e., if $a \in V$ and $b \in V$, then $ab \in V$.

(b) Prove: if $b \in U - V$, then for all $a \in V$, one has $ab \in U - V$.

(c) Prove: if $n$ is not pseudoprime, then the cardinality of $V$ is at most half the cardinality of $U$. (Hint: let $b$ be some unit for which $n$ fails the $b$-pseudoprime test. Define a map $f : V \rightarrow U - V$ by $f(a) = ab$. Prove that the map is one-to-one. Thus, there are at least as many elements in $U - V$ as there are in $V$).

(d) Prove: if $n$ is not pseudoprime, and we pick some random $a \in \{1, \ldots, n-1\}$, then the probability that $n$ passes the $a$-pseudoprime test is at most $1/2$. (Here “random” means uniformly distributed: each $a \in \{1, \ldots, n-1\}$ has the same chance of being picked).

(e) Conclude: if $n$ is not pseudoprime, then the probability that the randomized pseudoprime test gives the wrong answer is less than $1/2^k$.

We can pick the number $k$ of randomized tests so that the failure probability $1/2^k$ becomes as small as we want to tolerate. For instance, if $k = 10$, then $1/2^k$ is less than 0.001.

**Problem 5** Apply the pseudoprime test, with $k = 4$, to determine whether $n = 371$ is pseudoprime. Repeat for $n = 377$. You should choose $a_1, \ldots, a_4$ truly at random, for instance by throwing dice.

**Problem 6** If $n$ is a number of 1000 binary digits, estimate the number of steps it takes to test whether $n$ is pseudoprime, with error probability less than $1/1000$. A rough estimate is fine. Here a “step” means an elementary operation, such as a division or a multiplication, or choosing a random number. Make sure you account for all the operations performed, for instance, to calculate $a^{n-1} \pmod{n}$. Compare your answer to that of Problem 1.

Remark: The randomized pseudoprime test finds out whether a given number is pseudoprime. It is not a good enough test to distinguish Carmichael numbers from “genuine” primes. However, genuine primes can be detected by the so-called “Strong pseudoprime test” (which is also easy to implement). If you are interested, you can read about this stronger test in Chapter 25.B–C.

Finally, how does one go about finding large prime numbers? We have only talked about how to test whether a given number is prime (or pseudoprime). The key to finding a large prime number is simply to pick a random odd number $n$ of the desired size, and test whether it is prime. If it is not, discard $n$ and try a different random number, until you succeed. How many numbers do we expect to have to try until we find a prime number? The density of prime numbers among the numbers with $b$ binary digits is approximately $\frac{1}{\ln 2}$, so among the numbers of 1000 binary digits, about one in every 350 odd numbers is prime (and one can further improve this by only trying numbers that are not divisible by small primes such as 3, 5, and 7). For a computer, it is a manageable task to try this many numbers until a prime is found.