Primality testing

Recall that it is extremely difficult to factor large integers (this is the starting point for many cryptosystems). Surprisingly, it is much simpler to tell if a number is prime.

Example 85. The following is the number from Example 83, for which RSA Laboratories, until 2007, offered $100,000 to the first one to factorize it. Nobody has been able to do so to this day.

Has the thought crossed your mind that the challengers might be tricking everybody by chosing M to be a huge prime that cannot be factored further? Well, we'll talk more about primality testing soon. But we can actually quickly convince ourselves that M cannot be a prime. If M was prime then, by Fermat's little theorem, $2^{M-1} \equiv 1 \pmod{M}$. Below, we compute $2^{M-1} \pmod{M}$ and find that $2^{M-1} \not\equiv 1 \pmod{M}$. This proves that M is not a prime. It doesn't bring us any closer to factoring it though.

Comment. Ponder this for a while. We can tell that a number is composite without finding its factors. Both sides to this story (first, being able to efficiently tell whether a number is prime, and second, not being able to factor large numbers) are of vital importance to modern cryptography.

```
Sage] rsa = Integer("135066410865995223349603216278805969938881475605667027524485143851\
                    525510604859533833940287150571909441798207282164471551373680419703\
                    964191743046496589274256239341020864383202110372958725762358509643\
                    110564073501508187510676594629205563685529475213500852879416377328\
                    533906109750544334999811150056977236890927563")
```

```
Sage] power_mod(2, rsa-1, rsa)
```

```
120939094432033615867650595352956996867540098463588951238902808367556733932202059338 53\
348534147116662841968124107288512373904071077139405352848835710498409193003137847878 95\
226029615123284879513798127406300472693925500331497519103479951096634123177725212482 97\
950196643140069546889855131459759160570963857373851
```

Comment. Just for giggles, let us emphasize once more the need to compute $2^{N-1} \pmod{N}$ without actually computing 2^{N-1}. Take, for instance, the 1024 bit RSA challenge number $N = 135...563$ from Example 83. In Example 85, we did compute $2^{N-1} \pmod{N}$, observed that it was $\not\equiv 1$ and concluded that N is not prime. The number 2^{N-1} itself has $N - 1 \approx 2^{1024} \approx 10^{308.3}$ binary digits. It is often quoted that the number of particles in the visible universe is estimated to be between 10^{80} and 10^{100}. Whatever these estimates are worth, our number has WAY more digits (!) than that. Good luck writing it out! [Of course, the binary digits are a single 1 followed by all zeros. However, we need to further compute with that!]

Example 86. (bonus challenge) Find the factors of the following number $M = pq$:

```
89320280057437363393608386387469360495079915773073599087435569428108 27\
07615146116506918133536640188765047775335776026093439165454319252186 33\
75114106509563452970373049082933244013107347141654282924032714311
```

As indicated in Example 83, this is difficult. Through some sort of espionage, however, you have learned that $\phi(M)$ is:

```
89320280057437363393608386387469360495079915773073599087435569428108 27\
07615146116506918133536640188675726495278338662699830779066849891691 25\
75956375773572578614678768000225628866990840223520746283867797512
```

In general, if $M = pq$ is a product of two large primes p, q, given $\phi(M)$, how can we factor M?

Send me the factorization, and an explanation how you found it, by Feb 24 for a bonus point!

Comment. Even if we don't know the number of prime factors of M (in the above case we know that M is a product of two primes), we can "efficiently" factor M if we know the value of $\phi(M)$.

Example 87. Fermat's little theorem can be stated in the slightly stronger form:

n is a prime $\iff$ $a^{n-1} \equiv 1 \pmod{n}$ for all $a \in \{1, 2, \ldots, n-1\}$

Why? Fermat's little theorem covers the "$\Longrightarrow$" part. The "$\Longleftarrow$" part is a direct consequence of the fact that, if n is composite with divisor d, then $d^{n-1} \not\equiv 1 \pmod{n}$. (Why?!)

Fermat primality test

Input: number n and parameter k indicating the number of tests to run

Output: "not prime" or "likely prime"

Algorithm:

Repeat k times:
  Pick a random number a from $\{2, 3, \ldots, n-2\}$.
  If $a^{n-1} \not\equiv 1 \pmod{n}$, then stop and output "not prime".
Output "likely prime".

If $a^{n-1} \equiv 1 \pmod{n}$ although n is composite, then a is often called a **Fermat liar**.

On the other hand, if $a^{n-1} \not\equiv 1 \pmod{n}$, then n is composite and a is called a **Fermat witness**.

Flaw. There exist certain composite numbers n (see Example 89) for which every a is a Fermat liar (or reveals a factor of n). For this reason, the Fermat primality test should not be used as a general test for primality. That being said, for very large random numbers, it is exceedingly unlikely to meet one of these troublesome numbers, and so the Fermat test is indeed used for the purpose of randomly generating huge primes (for instance in PGP). In fact, in that case, we can even always choose $a = 2$ and $k = 1$ with virtual certainty of not messing up.

Next class, we will discuss an extension of the Fermat primality test which solves these issues (and is just mildly slower).

Advanced comment. If n is composite but not an absolute pseudoprime (see Example 89), then at least half of the values for a satisfy $a^{n-1} \not\equiv 1 \pmod{n}$ and so reveal that n is not a prime. This is more of a theoretical result: for most large composite n, almost every a (not just half) will be a Fermat witness.

Example 88. Suppose we want to determine whether $n = 221$ is a prime. Simulate the Fermat primality test for the choices $a = 38$ and $a = 24$.

Solution.

- First, maybe we pick $a = 38$ randomly from $\{2, 3, \ldots, 219\}$.
  We then calculate that $38^{220} \equiv 1 \pmod{221}$. So far, 221 is behaving like a prime.
- Next, we might pick $a = 24$ randomly from $\{2, 3, \ldots, 219\}$.
  We then calculate that $24^{220} \equiv 81 \not\equiv 1 \pmod{221}$. We stop and conclude that 221 is not a prime.

Important comment. We have done so without finding a factor of n. (To wit, $221 = 13 \cdot 17$.)

Comment. Since 38 was giving us a false impression regarding the primality of n, it is called a **Fermat liar** modulo 221. Similarly, we say that 221 is a **pseudoprime** to the base 38.

On the other hand, we say that 24 was a **Fermat witness** modulo 221.

Comment. In this example, we were actually unlucky that our first "random" pick was a Fermat liar: only 14 of the 218 numbers (about 6.4%) are liars. As indicated above, for most large composite numbers, the proportion of liars will be exceedingly small.